DESDE 2013
https://repository.uaeh.edu.mx/revistas/index.php/icbi/issue/archive
Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI
ISSN: 2007-6363

Publicación Semestral Pädi Vol. 12 No. Especial 4 (2024) 92–108

Programación de una rutina de un robot manipulador industrial mediante el método paso a paso
Programming a routine of an industrial manipulator robot using the step-by-step method

Alejandro Benítez-Morales [a], José R. Alvarado-Hernández [b], Jorge E. Pérez-Perea [b], Esteban M. Ramos-Gálvez [b], Alfredo Martínez-De-la-Cruz [b], José G. Benítez-Morales [c,*]

[a] *Área Académica de Computación y Electrónica, Universidad Autónoma del Estado de Hidalgo, 42184, Pachuca, Hidalgo, México.*
[b] *Escuela de Ingeniería, Universidad La Salle Pachuca, 42160, Pachuca, Hidalgo, México.*
[c] *Programa Educativo de Ingeniería Mecatrónica, Universidad Politécnica de Pachuca, 43830, Zempoala, Hidalgo, México.*

Resumen

En este trabajo se muestra el método paso a paso aplicado a la programación de un robot para tener una estructura homologada que facilite el diseño y análisis del programa; dando como resultado que cualquier programador comprenda rápidamente el algoritmo para realizar las modificaciones que se requieran debido a los cambios que se puedan suscitar en el proceso de producción. La ventaja del método es su simplicidad de implementación, facilitando la capacitación del personal, en comparación con otros métodos que requieren conocimientos complejos. Primeramente, se describe el método paso a paso aplicado a circuitos electroneumáticos; posteriormente, se presenta el procedimiento para programar la rutina de un robot, y se indican los cambios en el método para su implementación. Finalmente, se describen dos ejemplos aplicados al robot IRB 140.

Palabras Clave: Método paso a paso, Robótica, Automatización, Programación de robots.

Abstract

This work presents a step-by-step method applied to robot programming to establish a standardized structure that facilitates the design and analysis of the program. This approach enables any programmer to quickly understand the algorithm to make the necessary modifications in response to changes that may arise in the production process. The advantage of this method lies in its ease of implementation, which simplifies staff training compared to other methods that require complex knowledge. First, the step-by-step method applied to electro-pneumatic circuits is described; subsequently, the procedure for programming a robot's routine and the necessary modifications to the method's implementation are presented. Finally, two examples applied to the IRB 140 robot are described.

Keywords: Step by step method, Robotics, Automations, Robot programming.

1. Introducción

La automatización es fundamental en la autonomía de los procesos industriales con el propósito de impulsar el crecimiento y la rentabilidad de las empresas. Esto se logra a partir de la reducción de los tiempos de fabricación, el aumento de la producción, la mejora de la calidad del producto, la disminución de costos, el descenso de accidentes y la eliminación del trabajo monótono de los trabajadores, (Duran Moyano J. L., 2019).

Por lo tanto, ingenieros, tecnólogos y científicos orientan sus esfuerzos hacia el desarrollo e innovación de tecnologías que mejoren los procesos de producción mediante las nuevas tendencias como el internet de las cosas, la conectividad con la nube, la realidad virtual, la simulación 3D, la inteligencia artificial, entre otras áreas de la industrial 4.0, (Sánches Martínez, 2019).

*Autor para correspondencia: josegerardo@upp.edu.mx
Correo electrónico: abenitez@uaeh.edu.mx (Alejandro Benítez-Morales), jose.alvarado@lasallep.mx (José Raúl Alvarado-Hernández), jorge.perez.perea@lasallep.mx (Jorge Enrique Pérez-Perea), esteban.ramos@lasallep.mx (Esteban Misael Remos-Gálvez), alfredo.martinez@lasallep.mx (Alfredo Martínez-De-la-Cruz), josegerardo@upp.edu.mx (José Gerardo Benítez-Morales).

Historial del manuscrito: recibido el 01/07/2024, última versión-revisada recibida el 21/09/2024, aceptado el 18/09/2024, publicado el 30/11/2024. **DOI:** https://doi.org/10.29057/icbi.v12iEspecial4.13349

En este tenor, los robots manipuladores juegan un papel importante en los sectores productivos, tal como la industria eléctrica y electrónica, la industria automotriz, el sector metalmecánico, la industria del plástico y los productos químicos, la industria alimentaria, entre otros. Esto se debe a la flexibilidad que tienen los robots para realizar diversas tareas como la soldadura, el ensamble, la distribución y paletización, la manipulación de objetos, etc. Aunado a esto, el sector productivo está empleando cada vez más robots. De acuerdo con los datos estadísticos de (IFR International Federation of Robotics, 2021), el número de robots que se encuentran en operación ha aumentado, pasando de 1,059,000 unidades en 2010 a 3,015,000 unidades en 2020.

Esta creciente demanda de robots provoca la necesidad de contar con personal capacitado en la programación de robots. Cada experto cuenta con diferentes habilidades en la programación, ocasionando que los programadores pueden programar la misma tarea de diferente forma; es decir, un programador realiza el programa en diez líneas de código, otro programador en 30 líneas, otro en 50 líneas; además de usar diferentes tipos de instrucciones para alcanzar el mismo objetivo. Esto no sería un problema si el proceso fuera fijo, pero en la mayoría de las ocasiones los procesos son flexibles o programables, ya que las condiciones y tipos de producción cambian en determinados periodos, provocando que los programas de los robots tengan que ser modificados para adecuarse a los nuevas condiciones del proceso. Aunado a esto, la rotación y el cambio de personal ocasiona que el primer programador del robot, ya no sea el mismo que realice las modificaciones, dificultando el análisis del programa para realizar las modificaciones, a pesar que el programa esté documentado; esto conlleva a que el programador dedique un tiempo de estudio en el programa realizado o que opte por realizar un nuevo programa, lo cual es perjudicial para los tiempos de producción.

En este sentido, el presente trabajo propone contribuir en la implementación de un método para la programación de robots industriales, junto con los equipos involucrados en el proceso, logrando así la automatización del proceso. Para dar un contexto sobre las técnicas y los métodos empleados en la automatización de procesos, se mencionan algunas de ellas, tal como el uso del álgebra booleana y los mapas de Karnaugh que permite la reducción de elementos como contactos y bobinas en el diseño de diagramas de escalera (Soria-Tello, 2017), (Floyd, 2006). Las máquinas de estado son empleadas en el control de secuencias a través de eventos y condiciones para pasar de un estado a otro (Nelson V. P., 2006), mientras que las redes de Petri son usadas para modelar, analizar, simular y controlar los sistemas concurrentes y distribuidos, asegurándose que las transiciones entre diferentes estados del sistema ocurran de manera correcta y segura (Flores, 2016), (Reisig, 2013), (Nuñez-Alvarez, 2023). La programación por semáforos es empleada en la automatización para sincronizar procesos o tareas que necesiten acceder a recursos comunes. Respecto al control de secuencias de los cilindros neumáticos, existen dos métodos, el método de cascada y el método paso a paso; para desarrollar circuitos neumáticos y electroneumáticos (Vásquez Cortés J C, 2017) (Millan-Teja, 1996), y (Escaño-González y García-Caballero, 2019). Además, el método paso a paso es ocupado para realizar la programación de los controladores lógicos programables (PLC's) para controlar secuencias de cilindros neumáticos (Dorantes y Vásquez, 2004).

Los métodos anteriormente mencionados se han implementado generalmente en el PLC, cuya programación es mediante diagramas de escalera, diagramas de bloques de funciones, lista de instrucciones, texto estructurado y diagramas de funciones secuenciales. Los autores del presente trabajo no han encontrado información de la aplicación del método paso a paso enfocado a la programación de los robots manipuladores, los cuales se programan en un lenguaje estructurado.

Por otra parte, (Yongqiang, 2020) presenta una clasificación sobre los niveles de programación de robots a nivel prototipo y didáctico. El primer nivel es una programación basada en los movimientos de cada articulación en lazo abierto, el segundo nivel es la programación del robot en lazo cerrado, el tercer nivel trata sobre la programación de los robots empleando información de los múltiples sensores con el propósito de percibir el entorno que rodea al robot. El cuarto nivel consiste en aplicar una combinación de diferentes algoritmos, así como múltiples sensores para realizar una tarea.

El trabajo de (Adbelaal, 2019) describe el diseño de un algoritmo de programación para mejorar la velocidad y precisión de los robots de la marca Mitsubishi. El método consiste en analizar el programa desarrollado por el programador con la intención de aplicar un algoritmo de interpolación para posteriormente pasarlo a un algoritmo de control externo a la configuración del robot, el cual emplea los sensores internos del robot. Dando por consecuencia que el método no se pueda aplicar de forma inmediata y fácil a un robot industrial con la arquitectura cerrada.

En el trabajo de (Zhang, 2016) emplea la inteligencia artificial con el fin de facilitar la programación de un robot industrial. El método consiste en que el robot aprende al observar cómo el ser humano realiza una tarea y luego traducir la mayor cantidad de información posible de la demostración en el movimiento de sí mismo de manera automática. Esto libera al ser humano de codificar al robot a través de cualquier programación a nivel de lenguaje de máquina. Sin embargo, el método no es factible de aplicarlo en procesos que involucren alto riesgo al programador, tal como tareas de fundición y manipulación de cargas pesadas. Además, involucra tiempos de aprendizaje para realizar una tarea; así como la adición de sensores para captar la información para su aprendizaje.

Respecto a las tecnologías empleadas en la programación

de robots se tiene a la programación fuera de línea en ambientes virtuales tal como se describe en (Anfindsen, 2007) y (Garbev, 2020). Este es un método de programación es eficiente, sin embargo la calibración profesional entre el mundo virtual y el real siempre supone un desafío para los usuarios (Zhang, 2016). Esto ha ocasionado que las empresas en robótica mejoren en el diseño de los gemelos digitales que son representaciones virtuales precisas de objetos físicos o sistemas que permiten la simulación, monitoreo y optimización de sus versiones reales. Otra tecnología que se emplea en la programación de robots es la realidad aumentada, la cual permite ampliar los sentidos y mejoran la conciencia de la situación, incorporando objetos virtuales que coexisten perfectamente con objetos reales (Gallo, 2018) y (Jetter, 2018).

El artículo presenta la implementación del método paso a paso para programar a un robot, permitiendo trabajar bajo una estructura específica de programación con el propósito de facilitar tanto el diseño como el estudio del programa, dando la posibilidad de interpretar rápidamente la función del programa para realizar la modificaciones pertinentes ante los repentinos cambios del sistema de producción. Además, comparando este método con lo reportado en la literatura, no se tienen que realizar cambios físicos o de software; ya que se puede implementar a los lenguajes de programación establecidos por las diferentes marcas de robots como FANUC, ABB, KUKA, entre otros. Aunado a esto, este método puede ser usado en la programación de gemelos digitales robóticos en ambientes virtuales.

2. Método paso a paso aplicado a la automatización de circuitos electroneumáticos

Primeramente, se describe el método paso a paso con el propósito de introducir al lector en este tema. El método consta de cinco pasos para lograr el diseño de circuitos electroneumáticos (Mendoza Vargas J. A., 2011), los cuales se describen a continuación. Además, se proporciona un ejemplo para describir al método.

2.1. Identificación de los elemntos a automatizar mediante cilindros

Aquí se determina la forma en que se llevará a cabo la automatización del proceso, estipulando el tipo y número de cilindros necesarios para realizar las tareas. A cada cilindro se le asigna una letra empezando con la letra A y continuando con el orden de las letras del abecedario, con base a la norma DIN 1219-2.

2.2. Establecer la secuencia de movimientos de los cilindros.

Se determina el orden de los movimientos de los cilindros para desarrollar la tarea de automatización, definiendo qué cilindro se activa primero, cuál, en segundo, el tercero y así sucesivamente hasta obtener la secuencia de movimientos. Si el vástago del cilindro sale, se indicará con signo más (+), mientras que, si el cilindro entra, se indica con el signo menos (-). Por ejemplo, la secuencia de entrada y salida de un cilindro se marca como A+A-, (Berkheim, 1980).

2.3. Formación de grupos

Una vez obtenida la secuencia de movimientos, se procede a dividir a ésta en grupos. El primer grupo se forma al leer la secuencia de izquierda a derecha, si las dos primeras letras son diferentes, se pasa a revisar a la tercera letra con las dos anteriores, si estas son diferentes entre sí, se continúa con la cuarta letra, para verificar que esta letra sea diferente con las tres anteriores; y así sucesivamente. En el momento que se observe que un cilindro se repite (letra del cilindro), se procede a colocar una línea vertical o diagonal para separar a las primeras letras con la letra que se repite; y de esta manera formar al primer grupo. Enseguida, se procede a agrupar el siguiente conjunto de letras, observando que las letras restantes de la secuencia sean diferentes, y en el momento justo donde aparece una letra repetida, se procede nuevamente a colocar una línea divisoria para evitar la repetición de letras en un mismo grupo; es decir, que en un grupo no aparezca el mismo cilindro (letra del cilindro); esto continúa hasta agrupar toda la secuencia de movimientos. Por ejemplo, la secuencia A+B+B-C+A-C- se divide de la forma A+B+/B-C+A-/C- o de forma alternativa se puede separar a la secuencia de movimientos, de tal manera que no aparezca dos veces la misma letra en un mismo grupo, tal como A+B+/B-C+/A-C-, observe que en los grupos no se repite la misma letra. Además, a cada grupo se le asigna un número romano para identificarlos.

Por último, se recomienda que el número mínimo de grupos sean tres y que el máximo número de grupos sea el mínimo posible con la finalidad de ocupar el menor número de dispositivos.

2.4. Asignación de los sensores o interruptores para la transición entre los elementos de la secuencia

En cada cilindro se colocan dos sensores o interruptores de final de carrera con el propósito de detectar la salida y entrada del vástago. A ambos sensores se les etiqueta con la misma letra del cilindro y se les añade un subíndice, el subíndice cero indica que el sensor detecta que el vástago está adentro, mientras que el subíndice uno señala que el sensor detecta que el vástago está afuera. Esto es importante ya que los sensores o interruptores son empleados como las transiciones para pasar de la activación de un cilindro a otro, o de un grupo a otro. A continuación, se describe la forma de pasar de una letra a otra del mismo grupo, así como los cambios de grupo.

2.4.1. *Transición entre letras del mismo grupo*

Cuando se cumple la primera acción de la secuencia (la acción es la salida o entrada el vástago representado por una letra y un signo), esta es detectada por uno de los dos sensores, y en ese momento se envía una señal para iniciar la segunda acción de la secuencia. Cuando el segundo cilindro ha efectuado su acción, este será detectado por uno de sus sensores, dando paso a la activación de la tercera letra de la secuencia. Esto continuará hasta haber completado la secuencia del grupo. Las transiciones se indicarán mediante una flecha curva que une a las dos letras en la parte superior, la flecha debe tener una etiqueta que indique que sensor detectó la finalización de la acción del cilindro.

2.4.2. *Transición entre grupos*

Cuando se presenta un cambio de grupo, se evalúa si la última condición se cumplió mediante un sensor, y se coloca una flecha para unir a los grupos donde se presenta la transición. La transición del último grupo al primer grupo se indicará con una flecha que indica el reinicio de la secuencia.

2.5. *Construcción del diagrama de escalera con simbología europea*

La construcción del diagrama de escalera para el método paso a paso está conformada por dos configuraciones, la primera es utilizada para sincronizar los cambios entre grupos con base a las transiciones establecidas en la secuencia de movimientos, mientras que la segunda es empleada para realizar las transiciones entre los elementos del mismo grupo y la activación de sus propios cilindros.

2.5.1. *Descripción de la estrutura del diagrama de escalera para la transición entre grupos*

La estructura del escalón se muestra en la Figura 1, el cual se repetirá con base al número de grupos que existen en la secuencia de movimientos. Los elementos principales del escalón son un relevador eléctrico y los interruptores de límite o sensores. Los dos primeros contactos del escalón (viendo la Figura 1 de arriba hacia abajo, y de derecha a izquierda) tienen la función de activar a la bobina del relevador, el primer contacto controla la secuencia de activación de cada grupo, este pertenece al contacto del relevador del grupo anterior el cual es un contacto del relevador anterior; mientras que el segundo contacto indica en que momento debe iniciar el grupo, y depende del sensor que da paso al inicio del grupo. Cabe aclarar que el primer contacto de inicio del grupo 1 es un boton de marcha.

Prosiguiendo con la descripción, el contacto que está en paralelo, con dos contactos en serie, pertenece a la bobina del relevador del grupo, su función es mantener activado a la bobina mientras se ejecuta la secuencia del primer grupo. Por último, el contacto normalmente cerrado, conectado en serie con la bobina, es de la bobina del siguiente grupo y su función es apagar a la bobina para desactivar a este grupo.

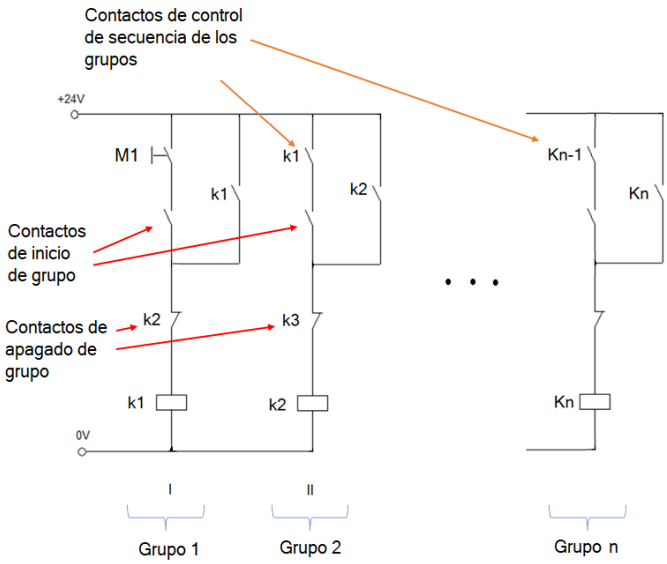

Figura 1: Diagrama de escalera para la sincronización de grupos.

2.5.2. *Descripción de la estrutura de activación de las electroválvulas*

Se procederá a agrupar a los solenoides de las electroválvulas con base a las letras que conforman a cada grupo de la secuencia de movimiento con el propósito de sincronizar la activación de cada cilindro (recuerde que cada letra de la secuencia de movimiento representa un cilindro del sistema y que estos son activados por electroválvulas biestables, en consecuencia, se tienen dos solenoides para hacer que el vástago entre o salga). Además, para explicar el circuito eléctrico, se abusará un poco del lenguaje al referirse a letra junto con el signo (+,-), para solo llamarla letra.

Todos los solenoides del grupo se conectan a un contacto del relevador que activa a dicho grupo, el cual se nombra contacto del grupo. Para activar a la siguiente letra, se conecta a un interruptor entre el contacto de grupo y el solenoide de la letra, este interruptor pertenece al interruptor de límite indicado en la secuencia de movimiento. Esta misma acción se realiza para cada una de las siguientes letras del grupo; mientras que la configuración de los grupos restantes es de la misma forma antes descrita.

El solenoide de la primera letra se conecta en serie al contacto del grupo. En las letras restantes, se conecta, entre el contacto de grupo y los solenoides de la secuencia, a los interruptores de final de carrera con base a lo indicado en la secuencia de movimiento. En las letras restantes, se conecta a un interruptor entre el contacto de grupo y los solenoides. Este interruptor pertenece a un interruptor de límite o a un sensor con base a lo indicado en la secuencia de movimiento.

Esta estructura se basa en la secuencia de los grupos y las condiciones de transición, representadas por los sensores. Primeramente, los solenoides de las electroválvulas se agrupan con base en los grupos de la secuencia de movimiento. Después, todos los solenoides de los grupos se conectan en serie con el contacto del grupo (el contacto del grupo pertenece al relevador de su respectivo grupo).

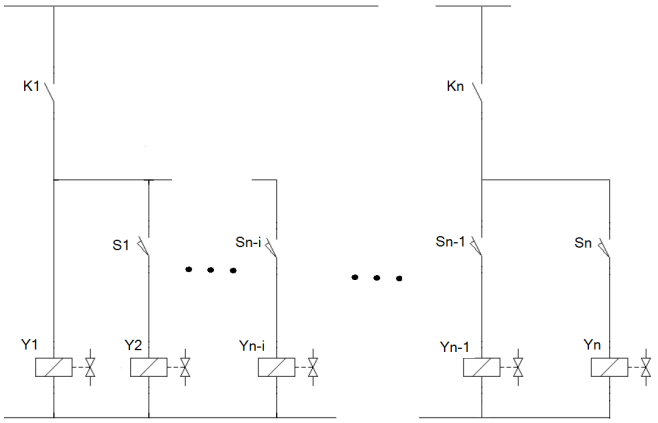

Figura 2: Diagrama de activación de solenoides.

2.6. *Ejemplo 1. Sistema electroneumático de tres cilindros*

Se tiene tres cilindros neumáticos de doble efecto y cada uno tiene dos interruptores de límite para detectar la entrada y salida del vástago. La secuencia de movimiento de sistema es A+B+A-C+B-C- para lograr la automatización de un proceso (El proceso no es detallado, ya que solo se busca explicar el método a partir de la secuencia de movimiento). La primera letra indica que el vástago del cilindro sale, después sale el vástago B, entra el vástago A, sale el vástago C, regresa el B y por último, el vástago C vuelve a su posición inicial.

De acuerdo con el paso 3, se tiene que leer la secuencia de izquierda a derecha y separar las letras en el momento que aparezca una letra que ya haya parecido dentro del mismo grupo; entonces, se observa que la primera y segunda letra son diferentes, pero la tercera letra se repite con la primera, por lo tanto, el primer grupo es A+B+. Con las letras restantes, se prosigue con el procedimiento, viendo que la tercera letra y la cuarta son diferente, la quinta letra es diferente con la tercera y cuarta letra; la sexta letra se repite con la cuarta letra, por tal motivo, se separa a la sexta letra de las otras, obteniendo así al segundo grupo conformado por A-C+B- y el tercer grupo sólo será C-. Dando por resultado que la secuencia de movimientos está dividida en tres grupos, de la siguiente forma:

A+B+/A-C+B-/C-.

El siguiente paso es establecer la transición entre los elementos de la secuencia. Primeramente, para pasar de la primera a la segunda letra, debe haberse cumplido la acción de la primera, es decir, cuando del vástago A sale por completo, el interruptor A1 detecta la salida y manda una señal para activar al cilindro B (ver el diagrama neumático ubicado en la parte superior de la Figura 4), haciendo que su vástago salga, ocasionado el cierre del interruptor B1. La señal de B1 es utilizada para activar al grupo 2. Una vez activado el grupo 2, se enviará una señal para el regreso del vástago A. El interruptor A0 detecta el regreso del vástago y manda la señal para accionar a C y su vástago al salir por completo realizará el cierre del interruptor C1, mandando una señal para hacer que regrese el vástago B. Al retornar B a su posición inicial, B0 manda la señal para encender al grupo 3 y esta manda una señal para hacer regresar al cilindro C. Enseguida, el C0 manda una señal para accionar al grupo 1. La secuencia de transiciones queda de la forma

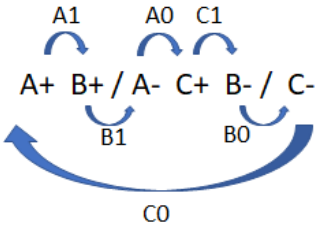

Figura 3: La secuencia de transiciones.

Se procede a la construcción del diagrama de escalera para continuar con el paso cinco del método. Se establecen tres configuraciones para la transición de los grupos debido a que la secuencia de movimiento está separada en tres grupos. Las bobinas de los relevadores 1, 2 y 3 se colocan en los escalones 1, 2 y 3 respectivamente, junto con sus etiquetas k1, k2 y k3.

Un botón de enclavamiento se coloca en el primer escalón para iniciar la secuencia; mientras que en los contactos de control de secuencia se coloca k1 y k2 para el escalón 2 y 3 respectivamente, con el propósito de coordinar la activación de los grupos. En el segundo contacto de inicio de grupo, se coloca a los interruptores indicados en la secuencia de transición; es decir, los interruptores utilizados para cambiar de un grupo a otro que son los interruptores C0, B1 y B0. Los contactos en paralelo son del relevador de su respectivo grupo; mientras que los contactos normalmente cerrados serán del relevador del siguiente grupo (en el último grupo se puede colocar a k1 o poner al último interruptor de la secuencia con el propósito de apagar al escalón en caso de que solo se realice una sola repetición de la secuencia).

La segunda parte del diagrama es la estructura de activación de las electroválvulas, en ella se agrupan a los solenoides de las electroválvulas conforme a los grupos. Los solenoides de un mismo grupo se conectan a un contacto de su respectivo relevador. El primer solenoide de cada grupo queda conectado directamente al contacto de grupo, como es el caso de los so-

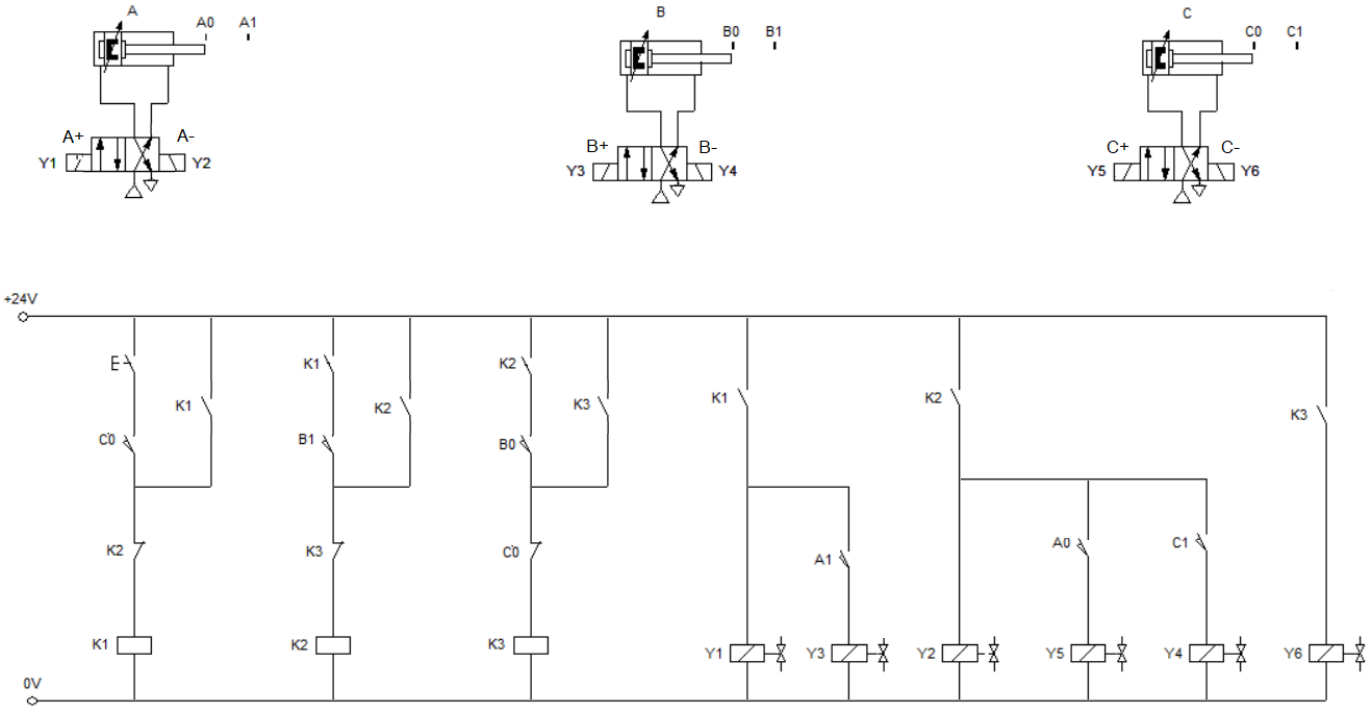

Figura 4: Diagrama electroneumático de la secuencia A+B+A-C+B-C-.

lenoides Y1, Y2 y Y6; mientras que los restantes solenoides se conectan a su respectivo interruptor con base a la secuencia de transición. Es decir, para el grupo 1, el solenoide Y3 se conecta a A1, para el grupo 2, los solenoides Y5 y Y4 se conectan a los interruptores A0 y C1, respectivamente. Terminando así la construcción del diagrama debido a que el grupo tres solo existe una electroválvula, la cual ya fue conectada.

2.7. Descripción del funcionamiento del ejemplo 1

El comienzo del sistema inicia cuando se presiona el botón de marcha y el interruptor C0 está cerrado, esto ocasionará la energización del relevador K1; resultando el cambio de estado de los contactos de K1, pasando de normalmente abierto a cerrado y los contactos cerrados a abiertos, provocando el enclavamiento del grupo uno. El primer solenoide en activarse es Y1 ocasionado que salga el vástago A, el cual presiona al interruptor A1 y este activa a Y3 con el propósito que salga el vástago de B. El interruptor B1 es activado dando paso a la energización de K2, cuyos contactos cambiarán de posición, haciendo que el grupo 1 sea desactivado, el grupo 2 se active; el contacto K2 ubicado en el grupo 3 se cerrará. El solenoide Y2 se activa cuando se cierra el contacto K2, provocando el regreso del vástago A, el interruptor A0 se cierra y energiza a Y5, originando que C salga. C1 se cierra y activa a Y4 para el regreso de B. El interruptor B0 se cierra y activa al grupo 3 mediante el relevador K3; este genera la desactivación de K2, el enclavamiento del grupo 3 y al mismo tiempo la activación de Y6 para el regreso del vástago 3, finalizando así la secuencia.

3. Lenguaje de programación de un robot manipulador industrial

La Asociación de Industrias Robóticas (RIA) define a un robot industrial como: “Un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas”. Existen otras definiciones sobre el robot industrial dados por la Organización Internacional de Estándares (ISO), la European Standard En775, entre otras, las cuales describen al robot de forma semejante.

Un robot manipulador industrial está compuesto por varias elementos (ver Figura 5) que le permiten realizar sus funciones de manera eficaz (Mendoza, 2017), dentro de sus partes principales están:

- Brazo o unidad mecánica._ Es el componente más destacado del robot y está formado por una serie de elementos estructurales sólidos o eslabones unidos mediante articulaciones que permiten el movimiento relativo entre cada dos eslabones consecutivos.

- Controlador._ Es un dispositivo electrónico y eléctrico que controla los movimientos del brazo mecánico determinando la posición, velocidad, precisión con respecto a

un marco de referencia. Además, almacena los programas, realiza cálculos y procesamiento de información, envía y recibe información a través de diferentes protocolos de comunicación; así como puede enviar y recibir señales digitales a través de un tarjeta de adquisición de señales o un controlador lógico programable (PLC).

- Dispositivo de entrada y salida de datos o caja de enseñanza.- Este dispositivo es comúnmente llamado como teach-pendant o flex-pendant, según sea la marca del robot y sus características de funcionalidad y diseño. Es un dispositivo portátil utilizado para programar, operar y supervisar al robot industrial. Es esencial para la interacción entre el usuario y el robot

Figura 5: Partes principales de un robot (imágen de la página web de Fanuc).

Al enfocarse en los lenguajes de programación, se tiene una variedad de lenguajes, que varían según el fabricante y la aplicación específica. A continuación, se mencionan algunos de los lenguajes de programación más comunes para robots industriales:

- RAPID.- Es el lenguaje de programación para los robots de la empresa ABB.
- KRL. - Es el lenguaje utilizado por los robots de la marca KUKA.
- KAREL.- Es el lenguaje básico de los robots FANUC.
- AS.- Es el lenguaje desarrollado por Kawasaki

Existen otros lenguajes de programación de otros fabricantes; además, cada fabricante proporciona librerías o paquetes para realizar tareas en específico como paletización, empaquetado, soldadura, pintura, entre otras tareas. Básicamente todos los lenguajes son parecidos entre sí.

Generalmente existen dos instrucciones básicas de movimiento de un robot que son el movimiento punto a punto y el movimiento lineal. El movimiento punto a punto (PTP) mueve el robot a través del espacio de manera rápida, pero no necesariamente en línea recta. Esta es una instrucción básica para mover el robot de una posición a otra. El movimiento lineal (LIN) mueve el robot en una línea recta entre dos puntos. Es útil cuando se necesita un control preciso sobre la trayectoria del robot.

El formato de instrucción de movimiento inicia con definir que tipo de desplazamiento se va a realizar, ya sea punto a punto o lineal, después se indica el punto a donde se va a mover el robot (Este punto contiene las coordenadas de posición y orientación), enseguida se coloca la velocidad de desplazamiento y por último, la precisión con la que va a llegar al punto final, en otras palabras, que tan alejado va a estar del punto deseado, esto da una flexibilidad en los cálculos para generar de trayectoria entre puntos. A continuación, se describen estas instrucciones para los lenguajes de programación de Kuka, Fanuc y ABB.

Instrucciones de movimiento para un robot de la marca Kuka

Movimiento PTP:

- PTP XHOME VEL=50 APO_DIST=10 mueve el robot a la posición XHOME con el 50 % de la velocidad máxima y una distancia de aproximación de 10 mm.
- PTP XP1 VEL=75 APO_DIST=5 mueve el robot a la posición XP1 con el 75 % de la velocidad máxima y una distancia de aproximación de 5 mm

Movimiento Lineal:

- LIN XHOME VEL=2.0 C_DIS=3.0 mueve el robot a la posición XHOME con una velocidad lineal de 2 m/s y una distancia de contorneado de 3 mm.
- LIN XP1 VEL=1.0 C_DIS=1.0 mueve el robot a la posición XP1 con una velocidad lineal de 1 m/s y una distancia de contorneado de 1 mm.

Instrucciones de movimiento para un robot de la marca Fanuc

Movimiento PTP:

- J PR[1:Home] 50 % FINE mueve el robot a la posición Home con el 50 % de la velocidad máxima y precisión fina.

- J PR[2:Point1] 75 % CNT50 mueve el robot a la posición Point1 con el 75 % de la velocidad máxima y una continuidad de 50, lo que significa que no se detendrá completamente en el punto.

Movimiento Lineal:

- L PR[1:Home] 2000mm/sec CNT50 mueve el robot a la posición Home con una velocidad lineal de 2000 mm/s y una continuidad de 50.

- L PR[2:Point1] 1000mm/sec FINE mueve el robot a la posición Point1 con una velocidad lineal de 1000 mm/s y precisión fina.

Instrucciones de movimiento para un robot de la marca ABB

Movimiento PTP:

- MoveJ Home, v100, z10, tool0; mueve el robot a la posición Home con una velocidad de 100 mm/s y una zona de precisión de 10 mm.

- MoveJ Point1, v500, z50, tool0; mueve el robot a la posición Point1 con una velocidad de 500 mm/s y una zona de precisión de 50 mm.

Movimiento Lineal:

- MoveL Home, v2000, z50, tool0; mueve el robot a la posición Home con una velocidad lineal de 2000 mm/s y una zona de precisión de 50 mm.

- MoveL Point1, v1000, z0, tool0; mueve el robot a la posición Point1 con una velocidad lineal de 1000 mm/s y precisión fina (z0).

Cabe hacer mención que estos lenguajes están basadas en Basic, Pascal y C, dependiendo del fabricante, por consecuencia los lenguajes de programación para robots contienen instrucciones bucle como while, for, y goto. Instrucciones de decisión como el if y if else, instrucciones de salida y entrada de señales digitales, creación y llamado de subrutinas, entre otras funciones.

4. Método paso a paso aplicado a los robot manipulasores industriales

En esta sección se aplica el método paso a paso para la programación de un robot. Además, se indican las adecuaciones y cambios al método debido a la flexibilidad que tiene un robot para realizar diferentes tipos de tareas.

4.1. Identificar los equipos y dispositivos necesarios para la automatización de sistemas.

Al aplicar el método se debe considerar la naturaleza del proceso, la tarea a automatizar, la selección de los equipos que intervendrán en el proceso, así como su distribución en el espacio de trabajo.

4.2. Establecer la secuencia de movimiento

Una vez definidos los elementos y la distribución de los equipos y piezas, se procede a establecer la secuencia del proceso; es decir, se determina que máquina o dispositivo empieza el proceso, cual enciende después y así sucesivamente.

Por otra parte, considere que una máquina puede ejecutar tareas más complejas que un cilindro. Además, al contar con varias máquinas en un proceso automatizado, estas pueden interactuar entre sí, para lograr la construcción de un dispositivo, elemento y/o sustancia. Por lo tanto, se plantean dos casos para la asignación de las letras para una máquina, dispositivo o elemento, el primer caso es cuando el robot está trabajando de manera aislada, mientras que el segundo caso es cuando el robot interactúa con otros equipos.

4.2.1. Caso 1

A diferencia del método paso a paso aplicado a los sistemas electroneumáticos, donde la ida de un cilindro se asigna la letra A+ y el regreso la letra A-, aquí se emplea una letra junto con un número debido a que un robot puede tener una ruta de ida y vuelta con diferentes trayectorias. Entonces a la ruta de ida se le asigna la etiqueta A1 y a la ruta de regreso se indica como A2, ver Figura 6.

Además, un robot tiene la flexibilidad de realizar varias tareas dentro de un proceso. Por lo tanto, se propone colocar un número para cada acción, siempre y cuando se considere acciones distintas dentro del proceso. La asignación de los números se realiza por pajeras para cada acción, es decir, si el robot viaja de una posición inicial hacia un destino, a la ruta se le nombra como A1, y después regresa a su misma posición inicial, se le etiqueta como A2. Ahora, la siguiente actividad del robot es partir de su posición inicial a un segundo destino, entonces se nombra como A3; mientras que el regreso será A4. Si se tiene un tercer punto de destino y regreso, se utilizarán las etiquetas A5 y A6, y así sucesivamente. Note que el número impar se utiliza para etiquetar a la ruta de ida, mientras que el número par es usado para la ruta de regreso.

Aunado a esto, la ruta, generalmente, está conformada por varios puntos a seguir para desplazarse de una posición inicial hasta el punto final con la finalidad de evitar puntos de singularidad y colisión con obstáculos; logrando así que el robot viaje en una ruta segura. Entonces a este grupo de puntos se les asigna una letra con un número. Por ejemplo, la Figura 6 se muestra que la ruta A1 está conformada por dos puntos p2 y p3, mientras que la ruta A3 la constituye los puntos p4 y p5.

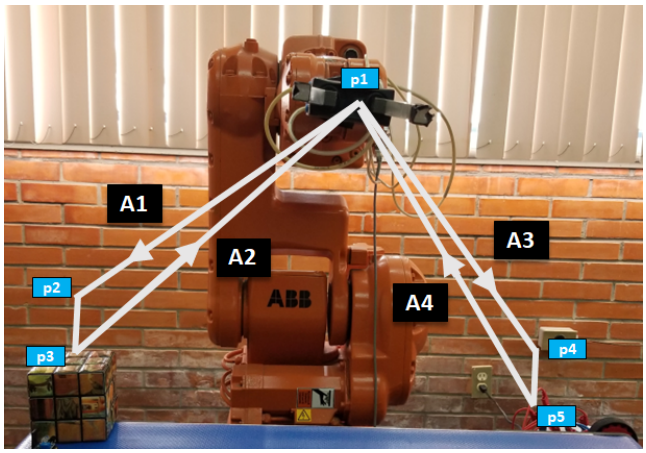

Figura 6: Rutas de un robot.

Otro caso distinto es cuando el robot no regresa a sus puntos de partida, por ejemplo, un robot va a un punto de trabajo para tomar unas piezas, luego viaja a un segundo punto de trabajo con el fin ensamblar esa pieza con otra, después el robot se traslada a un tercer punto de trabajo para almacenar a las piezas. La etiqueta A1 es para la primera ruta; mientras que A3 y A5 se colocan para las rutas de ensamblado y almacenado de la pieza, respectivamente. El regreso del robot a su posición inicial (home) se etiqueta como A2 debido a que es la única ruta de regreso. Cabe aclarar que las etiquetas A4 y A6 no se utilizan debido a que el robot no regresa del punto de ensamble a la zona de tomar la pieza, así como tampoco viaja de la zona de almacenamiento al punto de ensamble.

4.2.2. Caso 2

En este caso se toma en cuenta que el robot interactúa con otros equipos, tales como un torno CNC, una fresa CNC, un centro de maquinado, una banda transportadora, un posicionador, una máquina especializada u otro robot.

La asignación de letras se establece de la siguiente manera: a cada máquina se le asigna una letra junto con un número. El número uno indica el encendido de la máquina y el número dos señala el apagado de esta. Además, en el caso que una máquina realice dos o más tareas diferentes, se les añadirá un par de números por cada tarea, en otras palabras, si una máquina realiza tres tareas, las parejas 1 y 2, 3 y 4, 5 y 6 son las etiquetas de la primera, segunda y tercera actividad respectivamente donde el número impar denota el encendido y el par el apagado de la máquina.

Por ejemplo, en una línea de producción se fabrican dos productos diferentes, en dicho proceso se comparte un CNC para realizar un maquinado para el primer producto y un maquinado diferente para el segundo producto. Entonces, el inicio del primer maquinado se indica con la letra A1 y su final con A2; mientras que el comienzo del maquinado de la segunda pieza se indica con la letra A3 y su fin con A4.

Otro ejemplo es la interacción de varias máquinas, considere un proceso para el maquinado de una pieza cilíndrica. El proceso comienza con el encendido de una banda transportadora que transporta una pieza cilíndrica hasta el lugar de un torno CNC, cuando la pieza llega a su destino, la banda se detiene, y un robot parte de su posición inicial para tomar la pieza y colocarla en el área de trabajo del CNC, el brazo mecánico sale del torno y el torno empieza a trabajar. Al terminar el maquinado, el robot toma la pieza del CNC y la coloca en una segunda banda transportadora para llevarla a un área de almacenamiento.

4.3. Formación de grupos

El siguiente paso es dividir la secuencia en grupos de tal forma que en un grupo no debe haber una letra repetida dos veces, tal como se expone en la sección 2.3. Sin embargo, por las adecuaciones realizadas en la asignación de etiquetas para la secuencia de movimientos de un robot o una máquina. Se tiene que cambiar el criterio para separar a la secuencia en grupos.

> **Criterio 1:** *Las parejas de letras con número impar y par, que sean sucesivos y que el número impar sea menor al numero par, se consideran como un conjunto de letras que no puede estar en un mismo grupo.*

Esto da cabida a que en un grupo pueda haber la misma letra con un orden de números distintos al indicado anteriormente. Por ejemplo: consideré la secuencia A1A2A3A4, esta se puede agrupar como

A1/A2A3/A4.

Observe que la secuencia está formada por la misma letra. El par de etiquetas A1 y A2 no pueden estar en el mismo grupo, así como las etiquetas A3 y A4; en cambio, las etiquetas A2 y A3 conforman a un grupo; ya que el número par es menor que el número impar.

Se presenta la secuencia A1A2B1B3B5B6B7B2D1D2 para realizar la conformación de grupos. Debido a que A1 y A2 no pueden estar en el mismo grupo, se procede a separa a la primera etiqueta con la segunda etiqueta. Luego se observa que A2, B1, B3, B5 forman al segundo grupo, puesto que B5 y B6 son números consecutivos y el número impar es menor al par; dando por consecuencia, que se tenga que dividir en esta parte al grupo. La última separación de grupos se presenta con las etiquetas D1 y D2, quedando los grupos como

A1/A2B1B3B5/B6B7B2D1/D2.

4.4. Asignación de variables para la transición entre los elementos de secuencia

Ahora, se procede a determina a las transiciones entre las actividades, con el propósito de indicar qué elemento ocasiona el salto entre actividades. El procedimieno se describe en la sección 2.4. Las transiciones pueden ser cualquiera de las siguientes variables:

- Una señal digital proveniente de un sensor, interruptor, un PLC, un sistema embebido, un CNC u otro tipo de dispositivo o máquina.
- La señal de un temporizador al finalizar su tiempo establecido, o la señal de un contador al terminar su cuenta.
- La activación de una variable de bandera que indique el inicio o fin de una actividad.
- El accionamiento de un botón o un sensor por parte del operario.

Las condiciones de transiciones pueden ser una combinación de las señales anteriormente indicadas.

4.5. Etapa de programación de la secuencia de movimientos junto con sus condiciones de transición

El último paso es implementar la secuencia de movimiento junto con sus respectivas transiciones hacia un lenguaje de programación para el robot. La estructura de activación de grupos mostrada en la sección 2.5 consta de un grupo de contactos que condicionan la activación de la bobina del relevador del grupo. Al representar a un grupo del diagrama de escalera mediante una ecuación booleana, se tiene lo siguiente (Soria-Tello, 2017)

$$k1=(((M1\cdot S1)+k1)\cdot \overline{k2}),$$

donde M1 representa la condición de marcha o inicio del sistema a través de un botón, S1 es el interruptor de un sensor, k1 y k2 son las bobinas del primero y segundo relevador, respectivamente. Esta ecuación se reescribe mediante operadores lógicos como:

k1=((M1 AND S1) OR k1) AND NOT(k2).

Transformando a la ecuación a un código de programación, se obtiene como resultado a:

```
If (((M1=1 and S1=1) or k1=1) and k2<>1)
{
    k1=1;                                    .
}
```

Básicamente la estructura de activación de un grupo es mediante la instrucción IF, la cual tiene el propósito de verificar que todas las condiciones de transición se cumplan para la activación del grupo. Note que las tres primeras condiciones son de transición y enclavamiento, mientras que la última condición es de desenclavamiento; aunado a esto, si se cumplen las condiciones se activa una bandera o variable para indicar que el grupo se activó ($k_i = 1$), y se agrega otra variable para desactivar al grupo anterior ($k_{i-1} = 0$), ver Tabla 1.

Tabla 1: Estrutura de programación de un grupo.

```
IF ((condiciones de transición y enclavamiento)
AND (Condición de desenclavamiento) THEN
    kn:=0;              !desactivación del grupo anterior
    k1:=1;              !activación del grupo actual
ENDIF
```

Dependiendo del número de grupos, es las veces que se repite la estructura. La Tabla 2 muestra la programación de tres grupos, note que las banderas k1, k2, k3 controlan la secuencia de activación para cada grupo, independientemente de las condiciones de transición.

Ya que se estableció la forma de accionamiento de cada grupo, se procede a realizar la adecuación de la segunda parte del diagrama de escalera que es la activación de solenoides. Debido a que se trata de un robot que puede interactuar con diferentes máquinas, la tarea de activación de elementos es más complejo que solo activar solenoides; ya que el sistema puede incluir la comunicación con motores, bandas transportadoras, CNC, entre otras máquinas o dispositivos para enviar o solicitar información de mando (encendido y apagado de una máquina) y/o monitoreo; Aunado a esto, se debe tener en cuenta que las rutas de movimiento de un robot puede estar conformada por varios puntos para alcanzar su posición final, en este caso se recomienda emplear rutinas que almacene a las instrucciones de movimiento con el propósito de simplificar la estructura del programa.

Tabla 2: Estrutura de activación de tres grupos.

```
!activación del grupo 1
IF ((condiciones de transición 1 y enclavamiento) AND (k2<>1)
THEN
      k3:=0;                    !desactivación del grupo anterior
      k1:=1;                    !activación del grupo actual
ENDIF

!activación del grupo 2
IF ((condiciones de transición 2 y enclavamiento)
AND (Condición de desenclavamiento 2) THEN
      k1:=0;                    !desactivación del grupo anterior
      k2:=1;                    !activación del grupo actual
ENDIF

!activación del grupo 3
IF ((condiciones de transición 3 y enclavamiento)
AND (Condición de desenclavamiento 2) THEN
      k2:=0;                    !desactivación del grupo anterior
      k3:=1;                    !activación del grupo actual
ENDIF
```

Tomando en cuenta estas consideraciones, se procede a trasladar el diagrama de activación de los solenoides al lenguaje de programación del robot. Observe que el primer escalón de la Figura 1, muestra que todos los solenoides del grupo 1 están conectados al contacto del relevador 1; esto mismo sucede para los otros escalones de los otros grupos. Entonces, las rutinas de movimiento del robot como los elementos a activar se agrupan en a través de la condición IF cuya condición depende del grupo que esté encendido, ver Tabla 3.

Prosiguiendo con el diagrama de escalera de los solenoides, se tiene que el primer solenoide está conectado directamente al contacto del grupo, mientras que los otros solenoide está conectados a sus respectivos interruptores. Por lo tanto, se procede a colocar a la primera acción dentro del IF, mientras que las otras acciones que dependen de la condición de un interruptor o un sensor, se colocaran dentro de otro IF, ver grupo 2 de la Tabla 3.

Tabla 3: Estrutura del programa de activación de elementos.

```
IF (k1=1) THEN
      A1
ENDIF

!activación del grupo 2
IF (k2=1) THEN
      A2
      IF (Condición) THEN
            B1
      ENDIF
ENDIF

!activación del grupo 3
IF (k3=1) THEN
       B2
ENDIF
```

A continuación se darán algunos ejemplos para especificar el uso del método.

5. Descripción de la plataforma experimental.

La plataforma experimental es un robot IRB 140 de la marca ABB con un controlador S4C que contiene una tarjeta DAQ, modelo DSQC 328. Dicha tarjeta proporciona 16 entradas y 16 salidas divididas en 2 grupos de 8. Las cuales están aisladas y pueden ser alimentadas desde fuentes de 24v. La herramienta del robot es una pinza neumática; también, se utilizó una banda transportadora marca Dorner, modelo 2200, un sensor de infrarrojo, y una parrilla, ver Figura 7.

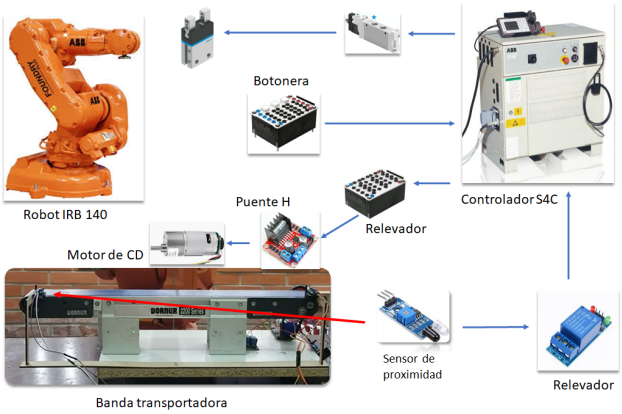

Figura 7: Esquema de la plataforma experimental.

Un motor de CD se acopló a la banda transportadora para transmitir el par requerido para mover el mecanismo de la banda. El motor es alimentado a 5 v a través de una etapa de potencia que consistió en puente H, modelo L298 y un relevador. Cuando el relevador recibe una señal de 24 v proveniente de la terminal D02 de la DAQ del controlador del robot, éste activa sus contactos para transmitir un voltaje de 5 v al puente H con el propósito que active sus transistores para proporcionar el voltaje de alimentación al motor; y así empiece a girar su flecha.

La plataforma tiene un sensor de proximidad infrarrojo, modelo FC-51, ubicado en la salida de la banda transportadora con la finalidad de detectar los objetos empleados en las pruebas del algoritmo. La señal de salida del sensor se conecta a un relevador para emplearlo como interfaz de amplificación debido a que el voltaje de la señal del sensor es de 5 v y las entradas digitales del DAQ son a 24 V. Entonces, la señal de sensor alimenta a la bobina del relevador, para activar los contactos del relevador. Una de las terminales del contacto es conectado a 24 V mientras que la otra terminal es conectada a terminal DI1 del DAQ.

Por otro lado, un botón de enclavamiento es utilizado como el botón marcha para dar inicio a la tarea del proceso. El botón se conecta directamente a la terminal DI2 de la DAQ.

Una parrilla eléctrica es conectada a una fuente de alimentación de 127V a través de un relavador. La bobina del relevador está conecata a la terminal DO3 del DAQ para encender o apagar la parrilla.

Por otra parte, la pinza neumática es activada mediante una electroválvula 4/2, con retorno de muelle. El solenoide de la electroválvula se conecta directamente a la terminal DO1 de la DAQ.

Se asignan unas etiquetas a las terminales de la DAQ con base a la conexión que tienen con los sensores y actuadores con el propósito de simplificar la explicación del método paso a paso, tal como se muestra en la Tabla 4.

Tabla 4: Asignación de etiquetas a las terminales de la DAQ

Etiqueta	Terminal de la DAQ
S1	DI1
Marcha	DI2
Pinzas	DO1
Motor (banda transportadora)	DO2
Parrilla	D03

6. Rutina de tomar y colocar una pieza mediante un robot.

Se presentar un ejemplo básico con la finalidad de empezar a introducir al lector en la implementación del método. La tarea del robot es mover una pieza de una posición a otra mediante una pinza neumática.

6.1. Identificación de los elementos a automatizar

La tarea de mover una pieza de un lugar a otro, requiere el uso de una pinza neumática como herramienta del robot para tomar y soltar al objeto, que en este caso es un cubo de rubik. El sensor de proximidad es colocado para detectar que el objeto esté presente en la posición inicial. Por último, se usa un botón de marcha para iniciar el proceso.

6.2. Establecimiento de la secuencia de movimiento

La secuencia de movimiento inicia con el robot en la posición de home (posición inicial) y éste procede a tomar la pieza, A1. La segunda acción es regresar a la posición de home, A2. Después, la tercera acción del robot es colocar la pieza al lugar de destino, A3. La cuarta es el regreso del robot a la posición inicial, A4, ver Figura 6. La ecuación de movimiento queda de la forma:

A1A2A3A4.

6.3. Formación de grupos

Separando a la secuencia de movimiento con base al criterio 1 y la explicación dada en la subsección 4.3, la secuencia queda dividida como:

A1/A2A3/A4.

6.4. Asignación de las transiciones entre los elementos de la secuencia

Lo siguiente es identificar las condiciones de transición entre cada actividad; por lo tanto, la activación de *A1* depende de un botón de inicio o marcha, y que el sensor de proximidad, *S1*, detecte la presencia de la pieza a mover. Después, el cambio de grupo 1 al grupo 2, se realiza cuando el robot toma la pieza mediante sus pinzas, *Pinzas*, permitiendo el regreso del brazo mecánico a home. La transición entre A2 y A3 se llevará a cabo cuando el robot regresa a home, para esto se emplea un variable de bandera, *F1*, en la programación. La transición entre el grupo 2 y 3 se presenta en el momento que el robot abre sus pinzas, *Pinza*. Bajo estas condiciones, se construye a la secuencia de transición, quedando como:

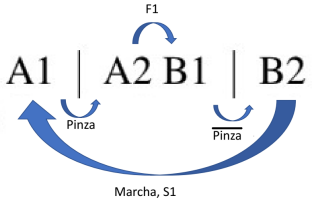

Figura 8: Condiciones de transición de la celda robótica.

6.5. Programación de la estructura del diagrama de escalera

El siguiente paso es trasladar a la secuencia de transición al código del programación del robot con base a las secciones 2.4, 4.4 y 4.5. La programación de la estrutura de activación de grupos queda como:

Tabla 5: Secuencia de activación de grupos de la rutina tomar y colocar.

```
!activación del grupo 1 mediante las condiciones de transición
IF (((Marcha=1 AND S1=1 ) OR k1=1) AND k2<>1) THEN
    k3:=0;                  !desactivación del grupo anterior
    k1:=1;                  !activación del grupo actual
ENDIF

!activación del grupo 2 mediante las condiciones de transición
IF (((k1=1 AND Pinza=1) OR k2=1) AND k3<>1) THEN
    k1:=0;                  !desactivación del grupo anterior
    k2:=1;                  !activación del grupo actual
ENDIF
ENDIF

!activación del grupo 3 mediante las condiciones de transición
IF (((k2=1 AND Pinza<>1) OR k3=1) AND F1<>1) THEN
    k2:=0;                  !desactivación del grupo anterior
    k3:=1;                  !activación del grupo actual
ENDIF
```

Observe que el último grupo es desactivado por medio de una variable F1 para evitar que el último grupo se quede activa y por ende, el robot esté repitiendo la última secuencia.

Tabla 6: Estrutura del programa de activación de elementos de la rutina tomar y dejar.

```
!activación de los elementos del grupo 1
IF (k1=1) THEN
    A1;                          !Llamado del rutina A1
ENDIF

!activación de los elementos del grupo 2
IF (k2=1) THEN
    A2;                          !Llamado del rutina A2
    IF (F1=1) THEN               !Condición de transición entre A2 y A3
        A3;                      !Llamado del rutina A3
    ENDIF
ENDIF

!activación de los elementos del grupo 3
IF (k3=1) THEN
    A4;                          !Llamado del rutina A4
    F1:=0;                       !Variable para apagar el último grupo
ENDIF
```

Los puntos de las trayectorias A1, A2, A3 y A4 se muestran a continuación.

Tabla 7: Lista de instrucciones de movimiento del programa tomar y dejar.

```
A1              !Rutina de desplazarse de home a tomar pieza
{
    MoveL Target10,v1000,z100,ServoWObj:=wobj0;
    MoveL Target40,v1000,z100,ServoWObj:=wobj0;
    MoveL Target20,v100,fine,ServoWObj:=wobj0;
    SetDO Pinza, 1;              !Cerrar pinza
    WaitTime 1;                  !Tiempo de espera de 1 s
}

A2              !Rutina de regreso a home
{
    MoveL Target10,v1000,z100,ServoWObj:=wobj0;
    F1:=1;                       ! Variable de bandera
}

A3              !Rutina de dejar pieza en otra posición
{
    MoveL Target50,v1000,z100,ServoWObj:=wobj0;
    MoveL Target30,v100,fine,ServoWObj:=wobj0;
    WaitTime 1;                  !Tiempo de esperar de 1 s
    SetDO Pinza, 0;              !Abrir pinza
    WaitTime 1;                  !Tiempo de esperar de 1 s
}

A4              !Rutina de regreso a home
{
MoveL Target10,v1000,z100,ServoWObj:=wobj0;
}
```

La secuencia de movimientos del robot se muestra en la Figura A.10. El video de la rutina del robot está en la siguiente enlace: `https://drive.google.com/drive/folders/1bsbLkkU4nR6lnfQALPnIXkG-p0bdNdGg?usp=drive_link`

7. Rutina de generación piezas decorativas a partir de barras de cera.

7.1. Identificación de los elementos a automatizar

El proceso consiste en crear piezas decorativas a través de la fundición de barras de cera. La automatización de la rutina es programada en el controlador S4C del robot IRB 140. Los dispositivos utilizados son: el sensor de proximidad, una parrilla eléctrica, un pocillo y un molde decorativo. Los actuadores y el sensor del proceso se conectan a la tarjeta de adquisición de datos (DAQ) del robot, como se indica en la sección 5.

7.2. Establecimiento de la secuencia de movimiento

El proceso inicia al accionar el botón de Marcha para activar a la banda transportadora con el fin de trasladar la materia prima hacia la zona de fundición. Cuando la materia prima llega hasta el otro extremo de la banda, la banda se detiene. El robot se desplaza para tomar la materia prima de la banda y llevarla hasta la ubicación del pocillo, colocando a las barras dentro de éste. Se enciende la parrilla para iniciar el proceso de fundición de las barras, las barras tardan aproximadamente 120 s en derretirse. Después, el robot toma al pocillo de fundición y lo lleva hasta un molde donde se vierte el líquido. El robot regresa a dejar el pocillo en la posición de la parrilla y finalmente regresa a la posición home.

Bajo este contexto, se procede a etiquetar a cada una de las actividades con base a la descripción dada en la sección 4.2; quedando de la siguiente forma

1. El encendido y apagado de la banda se etiquetan con B1 y B2, respectivamente.
2. El traslado del robot desde su posición de home hasta tomar la pieza de materia prima se etiqueta como A1.
3. El traslado del robot desde la banda hasta la ubicación pocillo, y colocarse en una posición fuera del área de calentado, se etiqueta como A3.
4. El encendido y apagado de la parrilla se etiqueta como C1 y C2, respectivamente.
5. La acción del robot de tomar al pocillo para llevarlo hasta el molde y depositar la cera fundida, se etiqueta como A5.
6. La acción del robot de regresar al pocillo a su posición inicial, se etiqueta como A6.
7. El retorno a la posición del home por parte del robot, se etiqueta como A2.

Esto da como resultado que la secuencia de movimiento quede como:

B1 B2 A1 A3 C1 C2 A5 A6 A2

7.3. Formación de grupos

Enseguida, se procede a dividir a la secuencia de movimientos en grupos con base al criterio 1. La primera separación se presenta entre B1 y B2, ya que son números consecutivos de la misma letra y el número impar es menor al par. Las etiquetas B1, A1, A3, C1 pueden estar en el mismo grupo, pero C1 y C2 no pueden estar en el mismo grupo debido a lo establecido en el criterio 1. Después se observa que C2 y A5 forman al tercer grupo debido a que A5 y A6 no pueden estar en el mismo grupo (criterio 1). Finalmente, el grupo 4 está formado por A6 y A2. Los grupos quedan como:

B1 / B2 A1 A3 C1 / C2 A5 / A6 A2

7.4. Asignación de las transiciones entre los elementos de la secuencia

Las condiciones de transición inician con el botón de marcha, para activar la banda. La señal del sensor de proximidad, S1, se ocupa para pasar del grupo 1 al 2. La transición entre B2 y A1 se presenta al apagar el motor. En la siguiente transición se emplea a una variable de bandera F1, la cual se activa al finalizar el cierre de las pinzas. El paso entre A3 y C1 es mediante la variable de bandera F2, a la cual se le asigna el valor de 1, cuando la pinza se abre. Después, el paso del grupo 2 al 3, es mediante un temporizador con un tiempo de espera de 120 s. Enseguida se apaga la caldera para pasar a la rutina de tomar el pocillo y llevarlo hasta el molde para vaciar la cera en el molde. La transición entre el grupo 3 y 4 es mediante la variable de bandera F3 que indica que la cera fue depositada en el molde. Por último, la transición entre A6 y A2, se realiza con la variable de bandera F4.

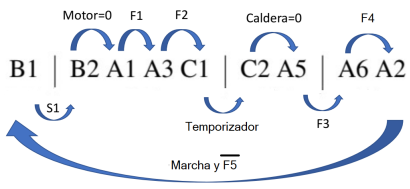

Figura 9: Condiciones de transición de la celda robótica.

7.5. Programación de la estructura del diagrama de escalera

El siguiente paso es trasladar a la secuencia de transición al código del programación del robot con base a las secciones 2.4, 4.4 y 4.5. La programación de la estrutura de activación de grupos queda como:

Tabla 8: Secuencia de activación de grupos de la tarea de fundición.

```
!activación del grupo 1 mediante las condiciones de transición
IF ((Marcha=1 AND F5=0) OR k1=1) AND k2<>1 THEN
    k4:=0;                    !desactivación del grupo anterior
    k1:=1;                    !activación del grupo actual
ENDIF

!activación del grupo 2 mediante las condiciones de transición
IF ((k1=1 AND S1=1) OR k2=1) AND k3<>1 THEN
    k1:=0;                    !desactivación del grupo anterior
    k2:=1;                    !activación del grupo actual
ENDIF

!activación del grupo 3 mediante las condiciones de transición
IF ((k2=1 AND Temporizador=1) OR k3=1) AND k4<>1 THEN
    k2:=0;                    !desactivación del grupo anterior
    k3:=1;                    !activación del grupo actual
ENDIF

!activación del grupo 4 mediante las condiciones de transición
IF ((k3=1 AND F3=1) OR k4=1) AND F5<>1 THEN
    k3:=0;                    !desactivación del grupo anterior
    k4:=1;                    !activación del grupo actual
ENDIF
```

Tabla 9: Secuencia de activación de elementos de la tarea de fundición.

```
!activación de los elementos del grupo 1
IF (k1=1) THEN
    F5:=1;                    !Asiganción de la bandera 1
    Set Motor;                !Encendido del motor
ENDIF

!activación de los elementos del grupo 2
IF (k2=1) THEN
    Reset Motor;              !Apagado del motor
    A1;                       !Llamado del rutina A1
    IF F1=1 THEN;             !Transición de A1 a A3
        A3;                   !Llamado del rutina A3
        IF F2=1 THEN;         !Transición de A1 a A3
            Set Caldera;      !Encendido de Caldera
            WaitTime 120;     !Tiempo de espera de 120s
            Temporizador:=1;  !Condición de transición
        ENDIF
    ENDIF

!activación de los elementos del grupo 3
IF (k3=1) THEN
    Reset Caldera;            !Apagado de Caldera
        IF Caldera=0 THEN;    !Condición de transición
            A5;               !Llamado de la rutina A5
        ENDIF
ENDIF

!activación de los elementos del grupo 4
IF (k4=1) THEN
    A6;                       !Llamado de la rutina A6
    WaitTime 1;               !Tiempo de espera 1 s
    IF F4=1 THEN; ;           !Transicion entre A6 y A2
        A2;                   !Llamado de la rutina A0
    ENDIF
    F1:=0;                    !Variable de bandera
    F2:=0;                    !Variable de bandera
    F3:=0;                    !Variable de bandera
    F4:=0;                    !Variable de bandera
    F5:=0;                    !Variable de bandera
ENDIF
```

Note que el apagado del grupo 4 depende de la variable F5, ya que si se coloca en su lugar a k1, solo funcioná cuando el robot opere en modo un solo

ciclo, pero si el robot está operando en modo continuo, el robot se quedará operando dentro del grupo 4, sin pasar por el grupo 1. Por lo tanto, se utiliza a la variable F5 para indicar que se terminó la operación del último grupo. La Tabla 9 muestra la activación de los actuadores involucrados en el proceso de fundición; así la activación de secuencias.

Se hace mención que no se puso todas las instrucciones de movimiento en la Tabla 10 debido a la longitud del programa, solamente se dejó las instrucciones representativas para el funcionamiento del programa.

Tabla 10: Lista de instrucciones de movimiento de la tarea de fundición.

```
A1              !Rutina de desplazarse de home a tomar pieza
{
    MoveL Target10,v1000,z100,tool0WObj:=wobj0;
    MoveL Target20,v1000,z100,tool0WObj:=wobj0;
    MoveL Target30,v1000,z100,tool0WObj:=wobj0;
    Set Pinza;                              !Cerrar pinza
    WaitTime 2;
    F1:=1;
}

A3              !Rutina de desplazarse de tomar/dejar pieza
{
    MoveL Target40,v1000,z100,tool0WObj:=wobj0;
    MoveL Target60,v1000,z100,tool0WObj:=wobj0;
    Set Caldera;                            !Encender caldera
    Reset Pinza;                            !Abrir pinza
    WaitTime 2;
    F2:=1;
}

A5              !Rutina de vacía liquido en molde
{
    MoveL Target50,v1000,z100,tool0WObj:=wobj0;
    Set Pinza;                              !Cerrar pinza
    WaitTime 1;
    MoveL Target70,v1000,z100,tool0WObj:=wobj0;
    MoveL Target80,v1000,z100,tool0WObj:=wobj0;
    F3:=1;
}

A6              !Rutina de regresar pocillo a parrilla
{
    MoveL Target50,v1000,z100,tool0WObj:=wobj0;
    Reset Pinza;                            !Abrir pinza
    F4:=1;
}

A2              !Rutina de regresar a home
{
    MoveL Target10,v1000,z100,tool0WObj:=wobj0;
}
```

La secuencia de movimientos del robot se muestra en las Figuras A.11 y A.12. Las fotos se tomaron con el pocillo vacío. El video de la rutina del robot con el proceso de fundición. Este se puede apreciar en la siguiente liga: `https://drive.google.com/drive/folders/1bsbLkkU4nR6lnfQALPnIXkG-p0bdNdGg?usp=drive_link`

8. Conclusiones

El método paso a paso aplicado a la programación de robots industriales proporciona una forma sencilla en la configuración del programa, haciendo que sea fácil la capacitación del programador a comparación de otros métodos que son complejos para su comprensión y aplicación. Además de no requerir la adición de sensores o cambios en el software para su implementación como los métodos expuestos en la introducción. Aunado a esto, se puede implementar el método a la programación fuera de línea en simuladores de ambientes virtuales empleando gemelos digitales robóticos.

Aunado a esto, el método paso a paso proporciona una estructura de programación estandarizada con la intención de que cualquier programador identifique rápidamente la organización de las rutinas y las secuencias de transición. Ya que el programa está ordenado en tres partes, que son la activación de grupos para dar seguimiento a una secuencia de grupos, la activación de elementos que realiza el llamado de las rutinas para que el robot se desplace por las rutas programadas, la última parte es la lista de instrucciones de movimiento.

La implementación del método paso a paso a la programación de una rutina de un robot implica realizar adecuaciones al método con respecto al etiquetado de las acciones de la secuencia de movimiento debido a la flexibilidad que tiene el robot para realizar diferentes tareas dentro de un proceso. Esto da como consecuencia, la creación de un nuevo criterio para separar la secuencia de movimientos en grupos.

Respecto a la programación del algoritmo a través del teach pendant, se complica la programación debido a que la pantalla tiene un tamaño reducido, obstaculizando ver una línea de comandos de forma completo, Sin embargo, cuando se tiene programada la estructura para el cambio de grupos, es fácil copiar y pegar, para solamente realizar los cambios en las condiciones de transición. Asimismo, se facilita la implementación del método en la programación fuera de línea empleando simuladores de ambientes virtuales.

Los trabajos a futuro involucran el estudio de este método ante sistemas que presentan eventos concurrentes, ciclos internos dentro de la secuencia de movimientos y divergencia en los procesos.

Agradecimientos

Los autores agradecen a la Universidad La Salle por el apoyo para realizar los experimentos en sus instalaciones; así como al ing. Federico González Rivero, al M. en C. Mario Alberto Tapia Falcon, a la ing. Lirio Melgarejo, y al ing Mario Lozano de la Escuela de Ingeniería.

Apéndice A. Secuencias de movimientos del robot mediante imágenes

En este apartado se muestran las secuencias de movimiento del robot mediante imágenes para el ejemplo de tomar y dejar, y la fundición de cera.

Referencias

Anfindsen O. A., Skouruo C. Petterson T. Pretlove J. (2007), *Method and a System for Programming an Industrial Robot*, U.S. Patent 7 209 801 B2, April 24, 2007.

Berkheim (1980). *Neumática, Iniciación al personal de montaje y mantenimiento, manual de estudio*. FESTO DIDACTIC, Alemania Federal.

Dorantes-González D. J., Manzano-Herrera M., Sandoval-Benítez G., Vásquez-López V. (2004), *Automatización y Control, Prácticas de laboratorio*. McGrawHill, México.

Duran-Moyano J. L. , Martínez-García H , Gómiz-Caro, J. J. Domingo-Peña, A. Grau-Saldes (2019). *Automatismos Eléctricos e Industriales*. Marcombo, España.

Escaño-González J. García-Caballero J. (2019), *Integración de sistemas de automatización industrial*. Ediciones Paraninfo, S. A., España.

Flores I. Guasch A. Piera M. A. (2016). *Modelos de Simulación usando SIMIO y Redes de Petri*. División de Ingeniería Mecánica e Industrial, UNAM, Alemania Federal.

Floyd T. L. (2006). *Fundamentos de Sistemas Digitales*. Pearson/Prentice Hall, España.

Gallo-Piñeros, (2020), *Tesis: Desarrollo de un ambiente de realidad aumentada para la operación y programación de robots industriales*, Universidad Nacional de Colombia, Colombia.

Garbev A. Atanassov A. (2020), *Comparative analysis of RoboDK and Robot Operating System for solving diagnostics task in off-line programming*, 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-5, doi: 10.1109/ICAI50593.2020.9311332.

International Federation of Robotics, (2021), *World Robotics 2021 reports*. IFR, Europa.

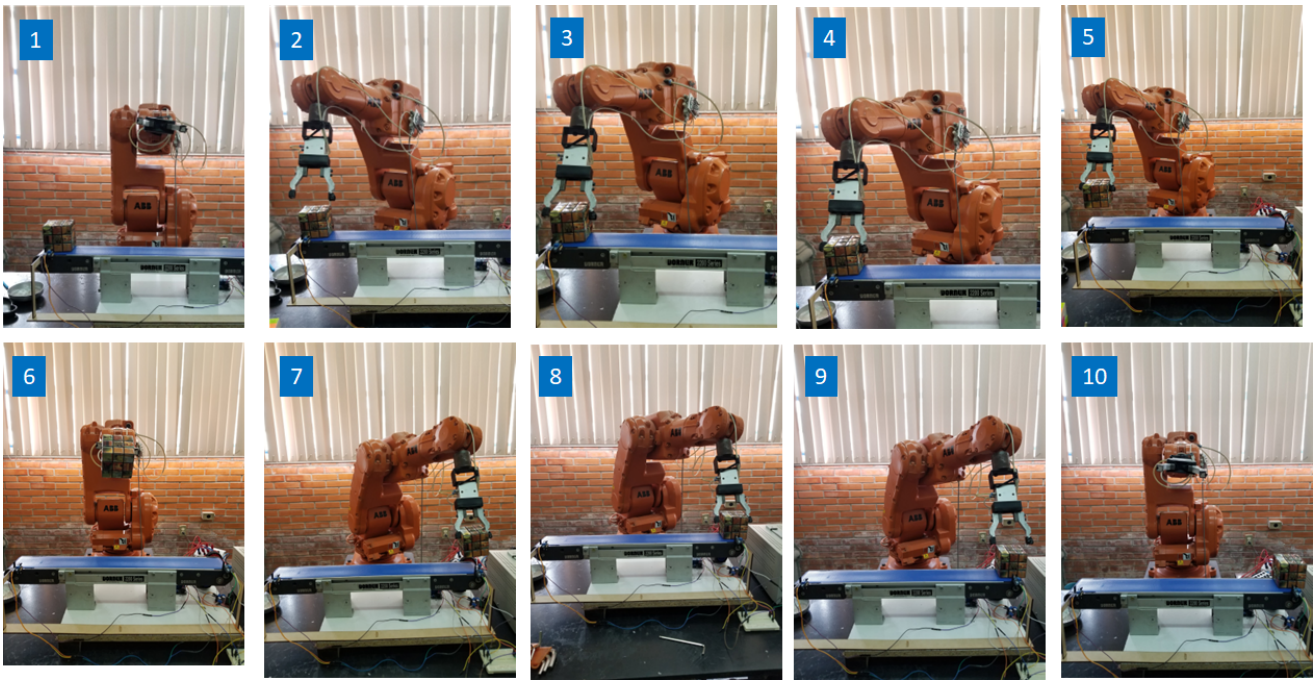

Figura A.10: Secuencia de tomar y colocar una pieza mediante un robot.

Figura A.11: Secuencia de la tarea de fundición, parte 1.

Figura A.12: Secuencia de la tarea de fundición, parte 2.

Jetter J., Eimecke J., Rese A., (2018), *Augmented reality tools for industrial applications: What are potential key performance indicators and who benefits?*, Computers in Human Behavior. doi.org/10.1016/j.chb.2018.04.054.

Mahmound Adbelaal (2019), *A study of robot control programing for an industrial robotic arm*, 2019 6th International Conference on Advanced Control Circuits and Systems (ACCS) & 2019 5th International Conference on New Paradigms in Electronics & information Technology (PEIT), Egipto. doi: 10.1109/ACCS-PEIT48329.2019.9062878.

Mendoza-Garfias P. Sánchez J. C. (2017), *Manual de operación de un robot Fanuc*. Ingeniería de servicios en automatización y robótica, México.

Mendoza-Vargas J. A., Muriel-Escobar J. A., Medina-Aguirre F.B. (2011). Aplicación del método paso a paso en la solución de problemas de electroneumática. *Scientia et Technica*, 47:313–317.

Millan-Teja S. (1996), *Automatización Neumática y Electroneumática*. Marcombo, España.

Nelson V. P., Nagle H. T., C. B. D. I. J. D. (2006). *Análisis y Diseño de Circuitos Lógicos Digitales*. Pearson/Prentice Hall, México.

Nuñez-Alvarez J.R. Benítez-Pina I.F., Acosta-Montoya G.J, Villafruela L. (2023), *Design of an integrated automation and control system using Petri nets: case study*. Journal of Applied Reseach and Technology, México.

Reisig W., (2013). *Understanding Petri Nets, modeling techniques, analysis methods, case studies*. Springer, Alemania Federal.

Sánches Martínez, I. (2019). *La cuarta revolución industrial en México*. Temas Críticos, España.

Soria-Tello S., (2017), *Sistemas Automáticos Industrial de Eventos Discretos*. Alfaomega, México.

Vásquez-Cortés J C, Cardona-Guio J. P., (2017). *Automatización electroneumatica, Métodos Sistematicos*. Editoria de la U., España.

Yongqiang C., Pin T. (2020), *Juvenile Robot Programming*, The 15th International Conference on Computer Science & Education (ICCSE), Holanda. doi: 10.1109/ICCSE49874.2020.9201887

Zhang J., Wang Y., Xiong R. (2016), *Industrial Robot Programming by Demonstration*, International Conference on Advanced Robotics and Mechatronics, China. doi: 10.1109/ICARM.2016.7606936.