

Modelado y Control Supervisor de un Prototipo de Celda de Manufactura Flexible

J. A. Rodríguez-Zúñiga* E. Aranda-Bricaire*

* *Departamento de Ingeniería Eléctrica, Sección de Mecatrónica, CINVESTAV, Av. IPN 2508, San Pedro Zacatenco, C.P. 07360, Ciudad de México, México (e-mail: jantonio.rodriguez@cinvestav.mx, earanda@cinvestav.mx).*

Resumen: Este trabajo presenta un caso de estudio de la aplicación de la Teoría de Control Supervisor usando Autómatas de Estado Finito a un prototipo de Celda de Manufactura Flexible. Se presenta el modelado de la celda para luego sintetizar un control supervisor, el cual permite que el lenguaje generado por el sistema en lazo cerrado sea confinado al lenguaje dado en las especificaciones, sin restringir las capacidades de la celda. Posteriormente el supervisor resultante es traducido lenguaje escalera e implementado en tiempo real en un Controlador Lógico Programable (PLC).

Palabras clave: Control supervisor, Autómatas de estado finito, Celda de manufactura flexible.

1. INTRODUCCIÓN

Los *Sistemas de Manufactura Flexible* (SMF) han recibido gran atención en las últimas décadas debido a la necesidad de las industrias de elevar su productividad, con mejor calidad y empleando la menor cantidad de recursos.

Un SMF es un sistema controlado por computadora compuesto por un grupo de estaciones de trabajo automatizadas e interconectadas por sistemas automáticos de manejo y almacenamiento de materiales (Groover 2000).

El control de los SMF debe permitir la realización de tareas flexibles y simultáneas, reducir el tiempo de producción y el requerido para modificar el proceso en caso de ser necesario.

Desde el punto de vista del modelado, los SMF pueden ser vistos como sistemas de eventos discretos debido a que su evolución depende de acciones que ocurren en instantes asíncronos en el tiempo, tales como señales de sensores, accionamiento de botones, fallas de equipos, etc.

Los sistemas de eventos discretos pueden ser estudiados a través de diferentes formalismos, tales como los autómatas de estado finito, las redes de Petri, lenguajes formales, lógica proposicional, etc (Cassandras and Lafortune 2009).

Una ventaja del modelado de SMF mediante autómatas de estado finito es que existen herramientas sistemáticas para construir modelos complejos a partir de modelos de componentes individuales. Además, es posible verificar la controlabilidad del sistema así modelado y construir controles supervisores de manera directa a partir de los modelos y especificaciones del sistema (Jaimes 2004) (Hernández 2005) (Wonham 2017).

El caso de estudio abordado en este trabajo es el prototipo de sistema de manufactura flexible conocido como *Sistema Mecatrónico I* de la compañía Fischertechnik. El producto final obtenido es un control supervisor para el centro neumático. Posteriormente, este controlador es traducido a lenguaje escalera para su implementación en un PLC.

2. AUTÓMATAS DE ESTADO FINITO Y TEORÍA DE CONTROL SUPERVISOR

Un *evento* se define como una ocurrencia instantánea dictada por naturaleza en un sistema, cuya ejecución genera un cambio abrupto en el estado del sistema. Un *Sistema de Eventos Discretos* (SED) es un sistema en el que la evolución de su estado depende completamente de la ocurrencia asíncrona de eventos a través del tiempo (Cassandras and Lafortune 2009) (Wonham 2017).

2.1 Autómatas

Un *Autómata* es una entidad capaz de representar un lenguaje de acuerdo a reglas bien definidas (Cassandras and Lafortune 2009) (Wonham 2017). Se puede dar una definición formal por medio de una gráfica directa o un diagrama de transición de estados.

Definición. Un *Autómata determinista* denotado por G se representa como una séxtupla

$$G = \{X, E, f, \Gamma, x_0, X_m\} \quad (1)$$

donde:

- X es el conjunto de *estados*
- E es el conjunto de *eventos*

- $f : X \times E \rightarrow X$ es la *función de transición*.
 $f(x, e) = y$ significa que estando en el estado x , si llega el evento e , entonces el autómata pasa al estado y . En general, f es una función parcial en su dominio
- $\Gamma : X \rightarrow E$ es la *función de eventos activos*. $\Gamma(x)$ es el conjunto de todos los eventos e para los cuales $f(x, e)$ está definida y es llamada *conjunto de eventos activos* de G en x
- x_0 es el estado inicial
- $X_m \subseteq X$ es el conjunto de estados marcados.

El conjunto de eventos puede descomponerse en $E = E_C \cup E_U$, $E_C \cap E_U = \emptyset$ donde E_C , E_U son los conjuntos de eventos controlables e incontrolables, respectivamente. Un evento controlable puede ser inhibido por el control supervisor. Un evento incontrolable no puede ser inhabilitado y puede ocurrir espontáneamente; por ejemplo, la señal de un sensor, la falla de una máquina o el fin de una tarea. Si X es finito, decimos que G es un *autómata determinista de estado finito*, abreviado como AEF.

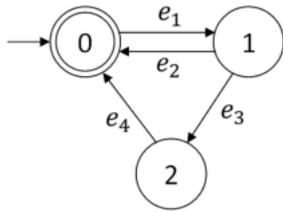


Fig. 1. Ejemplo de un AEF

Considere el AEF de la Figura 1, este autómata es un ejemplo clásico de la operación lógica de una máquina. Los estados son 0: apagado, 1: encendido y 2: descompuesto. Los eventos e_1 y e_2 indican el arranque y paro respectivamente. El evento e_3 representa una falla y e_4 indica que la máquina es reparada. Los elementos de este ejemplo son:

- Conjunto de estados y eventos:
 $X = \{0, 1, 2\}$, $x_0 = 0$, $X_m = \{0\}$
 $E = \{e_1, e_2, e_3, e_4\}$, $E_C = \{e_1, e_4\}$, $E_U = \{e_2, e_3\}$
- Función de transición:
 $f(0, e_1) = 1$, $f(1, e_2) = 0$, $f(1, e_3) = 2$, $f(2, e_4) = 0$
- Función de eventos activos:
 $\Gamma(0) = \{e_1\}$, $\Gamma(1) = \{e_2, e_3\}$, $\Gamma(2) = \{e_4\}$

Los estados marcados se representan con doble círculo y permiten distinguir algunos estados que tienen un significado especial, por ejemplo, el final de un proceso o el arranque de una estación de trabajo.

2.2 Operaciones con Autómatas

Resulta conveniente definir operaciones que nos permitan combinar o componer dos o más autómatas, de tal manera que los modelos de sistemas completos puedan ser construidos a partir de modelos de sistemas individuales. Definimos dos operaciones: el *producto* denotado por \times , y la *composición paralela* denotada por \parallel . La *composición paralela* a menudo se llama producto sincrónico y el *producto* es a veces llamado producto completamente

sincrónico. Estas operaciones modelan dos formas de componer el comportamiento de un grupo de autómatas que operan concurrentemente.

Considere los dos autómatas G_1 y G_2

$$\begin{aligned} G_1 &= \{X_1, E_1, f_1, \Gamma_1, x_{01}, X_{m1}\} \\ G_2 &= \{X_2, E_2, f_2, \Gamma_2, x_{02}, X_{m2}\} \end{aligned} \quad (2)$$

El *producto* de G_1 y G_2 es el autómata

$$G_1 \times G_2 := Ac(X_1 \times X_2, E_1 \cup E_2, f, \Gamma_{1 \times 2}, (x_{01}, x_{02}), X_{m1} \times X_{m2}) \quad (3)$$

donde

$$f((x_1, x_2), e) := \begin{cases} (f_1(x_1, e), f_2(x_2, e)) & \text{si } e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ \text{no definido} & \text{de otra manera} \end{cases} \quad (4)$$

En el producto, las transiciones de dos autómatas deben siempre estar sincronizadas en un evento común, esto es, un evento en $E_1 \cap E_2$. En particular el producto bloquea aquellos eventos que no ocurren en ambos autómatas. Corresponde a una sincronización total.

La *composición paralela* de G_1 y G_2 es el autómata

$$G_1 \parallel G_2 := Ac(X_1 \times X_2, E_1 \cup E_2, f, \Gamma_{1 \parallel 2}, (x_{01}, x_{02}), X_{m1} \times X_{m2}) \quad (5)$$

donde

$$f((x_1, x_2), e) := \begin{cases} (f_1(x_1, e), f_2(x_2, e)) & \text{si } e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ (f_1(x_1, e), x_2) & \text{si } e \in \Gamma_1(x_1) \setminus E_2 \\ (x_1, f_2(x_2, e)) & \text{si } e \in \Gamma_2(x_2) \setminus E_1 \\ \text{no definido} & \text{de otra manera} \end{cases} \quad (6)$$

En la composición paralela, un evento común, es decir, un evento en $E_1 \cap E_2$, sólo puede ser ejecutado si los dos autómatas lo ejecutan simultáneamente. Los eventos no comunes no están sujetos a restricciones y pueden ser ejecutados cuando sea posible.

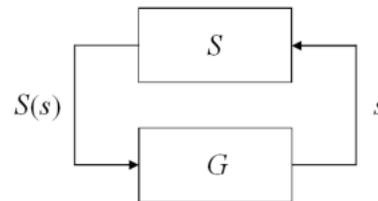


Fig. 2. Esquema retroalimentación de un control supervisor, G el sistema sin control y S el supervisor

2.3 Teoría de Control Supervisor

La *Teoría de Control Supervisor* (TCS) desarrollada por W. M. Wonham (Wonham 2017) es una metodología

general para la verificación y síntesis de supervisores de SED. Asumimos que un SED es modelado por el autómata G , este autómata describe el lenguaje generado por la planta. La premisa es que este comportamiento no es satisfactorio y debe ser “modificado” por un control. Se entiende por modificar, el restringir su comportamiento a un subconjunto del lenguaje generado por el sistema $\mathcal{L}(G)$. A fin de alterar el comportamiento de G introducimos un supervisor denotado por S .

Asuma que todos los eventos en E ejecutados por G son observados por el supervisor S . Por lo tanto, en la Figura 2, s es la cadena de todos los eventos ejecutados por G y s es visto completamente por S . La función de transición de G puede ser controlada por S en el sentido de que los eventos controlables de G pueden ser deshabilitados de manera dinámica por S .



Fig. 3. Celda de Manufactura Fischertechnik

3. DESCRIPCIÓN CELDA DE MANUFACTURA

El resto del trabajo se centra en el caso de estudio, el prototipo de celda de manufactura de la marca alemana Fischertechnik constituida por 4 módulos que corresponden a diferentes procesos, los cuales permiten simular un entorno industrial real para realizar prácticas de control de procesos, ver Figura 3, los módulos son:

- Centro Neumático
- Centro de Maquinados
- Centro de Prensado
- Robot Cilíndrico de 3 GDL

El centro neumático se encarga de recibir la materia prima, almacenarla en forma apilada y distribuirla a través de una mesa giratoria. La mesa lleva la pieza a un proceso de maquinado o a una banda transportadora la cual a su vez la lleva a otros procesos. La estación cuenta con su propio compresor neumático encargado de alimentar con aire las electro-válvulas que controlan los pistones.

El centro de maquinados es el siguiente paso en el proceso de la celda. Los materiales llegan a esta estación mediante la banda transportadora que une este módulo con el centro neumático. La pieza es redireccionada mediante un pistón que la empuja a otra banda en la cual hay dos estaciones de proceso (fresado y torneado), pudiendo ser elegida cualquiera de ellas. Finalmente la pieza es cambiada de dirección mediante un pistón de empuje a

la última banda con lo cual termina su proceso en la estación.

La estación de prensado se encarga de recibir las piezas que ya fueron procesadas por los dos módulos anteriores, a través de un manipulador robótico. El módulo tiene una banda que lleva la pieza hacia la máquina de prensado. Una vez que el proceso termina, la banda regresa la pieza para que el manipulador la pueda llevar al almacén de producto terminado.

El brazo robótico de 3 GDL es el encargado de las tareas de manipulación y transporte de las piezas desde el centro de maquinados hasta el centro de prensado, y de éste último al almacén de producto final. El robot cuenta con una pinza en su extremo final.

Cada módulo del prototipo de celda de manufactura cuentan con un conector $DB37$. Mediante dicho conector podemos comunicar los sensores y actuadores del sistema con las entradas y salidas del PLC.

4. MODELADO DEL CENTRO NEUMÁTICO

Para propósitos de la lógica de control, los procesos de la celda de manufactura pueden ser abstraídos en el proceso de modelado a sistemas de eventos discretos, que en este caso describen el comportamiento del sistema en lazo abierto.

Para el modelado del módulo del centro neumático mediante AEF, se identifican los siguientes subsistemas:

- Pistón de entrada
- Máquina de prensado
- Pistón de salida
- Mesa giratoria

No se incluye el modelo de la banda transportadora, ya que se pretende que funcione continuamente. La síntesis de los otros autómatas se muestra a continuación.

Consideramos que la operación de los subsistemas, comienzan con un evento controlable a_i y terminan con evento incontrolable b_i . Esta abstracción permite usar modelos simples para un control supervisor.

Table 1. Eventos de los subsistemas del módulo neumático

i	a_i	b_i	Subsistema
1	111	110	Pis1
2	121	120	Maq1
3	131	130	Pis2
4	141	140	Mesa

La Figura 4 muestra un autómata genérico para los subsistemas modelados del centro neumático. La tabla 1 muestra los eventos controlables e incontrolables para cada uno de los subsistemas. Los subsistemas modelados mediante la Figura 4 son *Pis1*, *Maq1*, *Pis2* y *Mesa*, a continuación se explica a detalle cada uno de ellos.

Modelo del Pistón de Entrada (Pis1). El pistón de entrada es el encargado de llevar la pieza desde el almacén hacia la mesa giratoria. Definimos su comportamiento con dos estados, 0: *pistón contraído* y 1: *pistón extendido* y las transiciones 111: *extender pistón* y 110: *contraer pistón*.

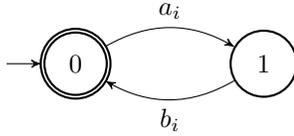


Fig. 4. Autómata genérico para subsistemas del centro neumático

Modelo de la Máquina de Prensado (Maq1). La máquina de prensado es modelada con dos estados, 0: *máquina en espera* y 1: *máquina trabajando* y las transiciones 121: *inicio de maquinado* y 120: *fin de maquinado*.

Modelo del Pistón de salida (Pis2). El pistón de salida es el encargado de llevar la pieza de la mesa giratoria hacia la banda. Para su modelo definimos los estados 0: *pistón contraído* y 1: *pistón extendido* y las transiciones 131: *extender pistón* y 130: *contraer pistón*.

Modelo de la Mesa giratoria (Mesa). El modelo de la mesa giratoria es representado por dos estados 0: *mesa detenida* y 1: *mesa girando* y la transiciones 141: *girar mesa* y 140: *detener mesa*.

Modelo de la Planta. Para obtener el modelo de la planta se debe realizar la composición paralela con los modelos de los subsistemas anteriores.

$$Plant = Pis1 \parallel Maq1 \parallel Pis2 \parallel Mesa \quad (7)$$

El autómata de la planta tiene 16 estados y 64 transiciones. No se presenta aquí debido a su tamaño. Dicho autómata representa el comportamiento del módulo neumático sin tener ninguna restricción, es decir en lazo abierto.

5. CONTROLADOR DEL CENTRO NEUMÁTICO

Partiendo de la teoría del control supervisor se definen una serie de especificaciones de seguridad para no violar ninguna regla de seguridad y limitar mínimamente las capacidades de la celda. A continuación se muestran las especificaciones para el centro neumático. Dichas especificaciones se abstraen como AEF.

En el resto de la Sección, la condición *selfloop* se incluye para evitar bloqueos de eventos que no están involucrados en la especificación pero que se pueden ejecutar en la planta.

Especificación 1 (Spec1). Permite evitar que la *Mesa* se encuentre girando mientras el pistón de entrada (*Pis1*) se encuentre extendido. Su autómata se muestra en la Figura 5. Esta es una especificación de seguridad, ya que al estar el *Pis1* extendido y la *Mesa* girando al mismo tiempo puede ocurrir un choque entre estos dos componentes.

Especificación 2 (Spec2). Permite evitar que la *Mesa* se encuentre girando mientras la máquina de prensado (*Maq1*) se encuentre trabajando, es decir mientras su pistón se encuentre extendido. La Figura 6 muestra el autómata que la describe. Esta es también una especificación de seguridad ya que al estar el *Maq1* trabajando y

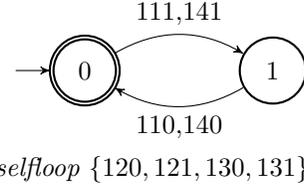


Fig. 5. Autómata de la especificación 1

la *Mesa* girando al mismo tiempo puede ocurrir un choque entre estos dos componentes.

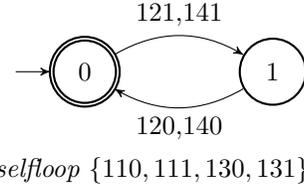


Fig. 6. Autómata de la especificación 2

Especificación 3 (Spec3). Permite evitar que la *Mesa* se encuentre girando mientras el pistón de salida (*Pis2*) se encuentre extendido. El autómata de la especificación se muestra en la Figura 7. Esta al igual que las especificaciones anteriores es de seguridad, ya que al estar el *Pis2* extendido y la *Mesa* girando al mismo tiempo puede ocurrir un choque entre dichos componentes.

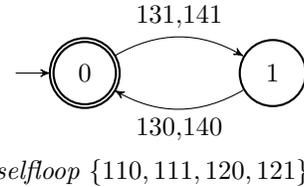


Fig. 7. Autómata de la especificación 3

Especificación 4 (Spec4). Indica que la *Mesa* puede girar hasta que alguno de los subsistemas *Pis1*, *Maq1* o *Pis2* haya regresado a su posición de inicio. Podemos interpretarlo como que la *Mesa* no gira hasta que los elementos mencionados terminen su "tarea". En la Figura 8 se muestra el autómata que describe la especificación.

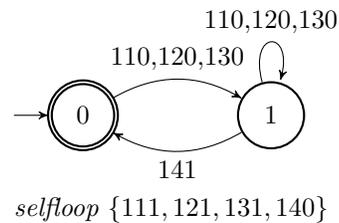


Fig. 8. Autómata de la especificación 4

Especificación 5 (Spec5). Su intención es que el *Pis1* no envíe más de una pieza a la *Mesa* hasta que ésta haya girado al menos una vez, ya que de lo contrario el espacio donde debe llegar la pieza ya se encuentra ocupado y eso ocasionaría un mal funcionamiento del sistema. La Figura 9 muestra el autómata que describe la especificación.

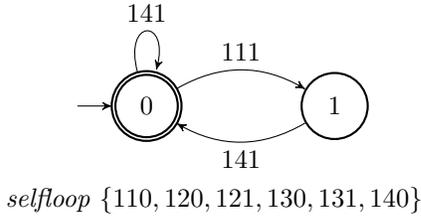


Fig. 9. Automata de la especificación 5

Modelo de las Especificaciones. Para obtener el autómata completo de las especificaciones se realiza la composición paralela de todas las especificaciones individuales.

$$Espec = Spec1 \parallel Spec2 \parallel Spec3 \parallel Spec4 \parallel Spec5 \quad (8)$$

El autómata resultante tiene 29 estados y 85 transiciones. No se presenta aquí debido a su tamaño.

Síntesis del supervisor. Para obtener el supervisor que garantiza cumplir con las especificaciones dadas, primero se realizan los productos síncronos mencionados anteriormente. A partir de ellos se obtiene un autómata para la planta (*Plant*) y otro para las especificaciones (*Espec*). Una vez obtenidos los autómatas mencionados se puede calcular el supervisor. A fin de simplificar el cálculo requerido nos apoyamos en la herramienta computacional *TCT*. El software *TCT* es un programa para la síntesis del control supervisor para sistemas de eventos discretos y que fue desarrollado por el grupo de Sistemas de Control de la Universidad de Toronto, bajo la dirección del célebre Profesor W.M. Wonham (Wonham 2017).

Las operaciones anteriores de composición paralela usadas para obtener el modelo de la planta y el modelo de las especificaciones se realizan en *TCT* utilizando el comando *Sync*. Para el caso de sintetizar el controlador se utiliza el comando *Supcon* el cual calcula el mayor lenguaje supremo controlable, es decir, nos proporciona un autómata que describe el funcionamiento de la planta en lazo cerrado satisfaciendo todas las especificaciones. Las operaciones en *TCT* son las siguientes.

$$\begin{aligned} Plant &= Sync(Pis1, Maq1, Pis2, Mesa) \\ Espec &= Sync(Spec1, Spec2, Spec3, Spec4, Spec5) \\ Super &= Supcon(Plant, Espec) \end{aligned}$$

El autómata del lenguaje supremo controlable tiene 21 estados y 59 transiciones. No se muestra aquí debido a su tamaño. Posteriormente se verifica la controlabilidad del supervisor mediante el comando *Condat* el cual además de decirnos si el sistema es controlable nos da una lista de los eventos inhabilitados en cada uno de los estados de la planta. Debido a que el autómata *Super* es demasiado grande y complica su programación en el dispositivo de control (PLC), se procede a usar el comando *Supreduce* de *TCT* para calcular un autómata reducido cuyo lenguaje es equivalente al del supremo controlable. Los comandos descritos anteriormente son los siguientes:

$$\begin{aligned} Ctrl &= Condat(Plant, Super) \\ Simsup &= Supreduce(Plant, Super, Ctrl) \end{aligned}$$

El autómata denominado *Simsup* y que se muestra en la Figura 10 describe el lenguaje supremo controlable de manera reducida. Éste autómata cuenta con 7 estados y 32 transiciones, debido a su tamaño el autómata se puede programar más fácilmente en el PLC.

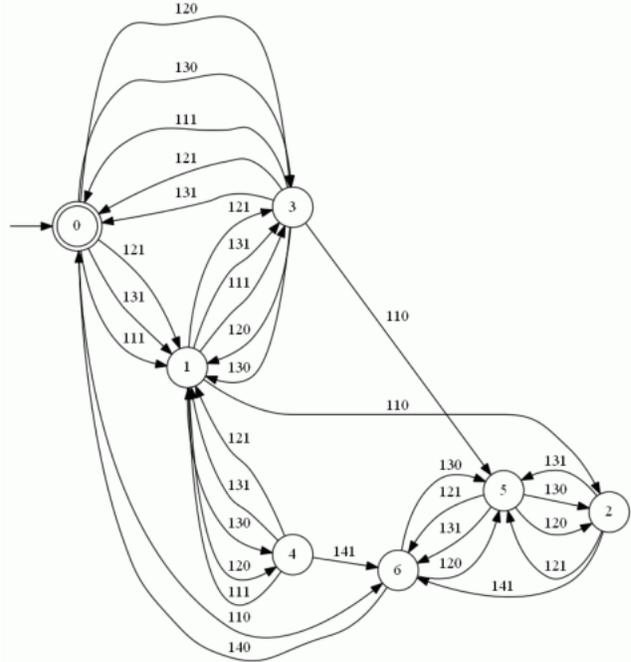


Fig. 10. Automata del supervisor reducido

Podemos hacer algunas observaciones respecto al autómata del supervisor reducido mostrado en la Figura 10. La primera de ellas es que el evento 141 sólo se encuentra habilitado en los estados 2 y 6 en donde llega después de la ocurrencia de los eventos 110, 120 o 130 tal como lo define la especificación 4. Además se puede notar que mientras ocurra alguno de los eventos 111, 121 o 131 el evento 141 está bloqueado conforme a las especificaciones 1, 2 y 3. Observe que después de que ocurre un evento 110 llegamos a los estados 2, 5 y 6, en dichos estados ya no puede ocurrir un evento 111 lo que implicaría enviar más piezas a la mesa, esto sigue la lógica descrita por la especificación 5.

6. TRADUCCIÓN DEL SUPERVISOR A PLC

Para la implementación en tiempo real del controlador monolítico obtenido para el centro neumático, es necesario traducir el autómata de la Figura 10 a lenguaje escalera (LAD). La traducción del autómata del supervisor se realiza empleando el método propuesto por (Hasdemir et al. 2008) el cual evita la ocurrencia del problema llamado “efecto avalancha” (Fabian and Hellgren 1998).

El método se basa en definir las funciones *SET* y *RESET* y que son realizadas por operaciones de lógica Booleana. Denotamos la condición *SET* de una variable lógica x por S , y la condición *RESET* por R . Representamos el valor actual de x y el siguiente por $x(k)$ y $x(k + 1)$ respectivamente. Entonces,

$$x(k+1) = S + x(k) \cdot R' \quad (9)$$

Donde (+) corresponde a OR, (\cdot) corresponde a la operación AND y R' significa la negación de R . Esta función lógica es usada para la realización de las reglas definidas por una función de transición para un autómata dado. Las expresiones lógicas para el estado x_i se definen mediante

$$x_i(k+1) = S_i + x_i(k) \cdot R'_i(k) \quad (10)$$

La función lógica dada por la Ecuación (10) puede ser programada fácilmente en lenguaje LAD. Representando las variables lógicas $x_i(k+1)$ y $x_i(k)$ por los bits de memoria X_i y x_i , respectivamente, el diagrama mostrado en la Figura 11 es usado para implementar la función mostrada en la Ecuación (10).

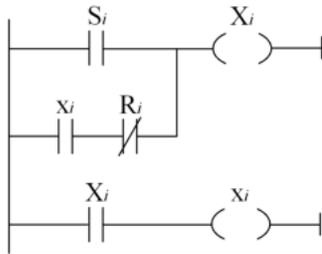


Fig. 11. Diagrama LAD correspondiente a la Ecuación 10

Para finalizar la implementación del supervisor se procede a traducir los autómatas de los subsistemas. Se definen dos niveles en la lógica de programación, a dichos niveles los vamos a llamar alto y bajo. En el nivel alto se programa la inicialización de la celda por medio de un botón de arranque, la interrupción del programa con un botón de paro de emergencia y el autómata del supervisor. En el nivel alto también se inicializan todos los estados de los AEF de los subsistemas. En el nivel bajo se programan los AEF de los subsistemas empleando la traducción propuesta por (Bucio 2011), éstos se programan en bloques de funciones para ser llamados por el programa principal (nivel alto) (De Queiroz and Cury 2002).

Los modelos que corresponden a los subsistemas en su traducción a LAD utilizan solamente bobinas de enganche (*latch coils* o *set coils* en inglés), bobinas de liberación (*unlatch coils* o *reset coils* en inglés), bobinas regulares y contactos normalmente abiertos. La Figura 12 muestra un ejemplo de la traducción de un autómata simple a lenguaje escalera (LAD) usada para los subsistemas (Bucio 2011).

7. CONCLUSIÓN

Se llevó a cabo el modelado y la síntesis de un supervisor para el módulo de centro neumático del prototipo de celda de manufactura. Se obtuvieron los modelos para los subsistemas del centro en AEF. Se definieron especificaciones de seguridad para limitar al sistema y evitar llegar a estados que pongan en riesgo la integridad del prototipo. Se obtuvo un supervisor cuyo comportamiento se encuentra restringido por medio de especificaciones de seguridad. Mediante una metodología se realizó una traducción del supervisor a lenguaje escalera (LAD) para

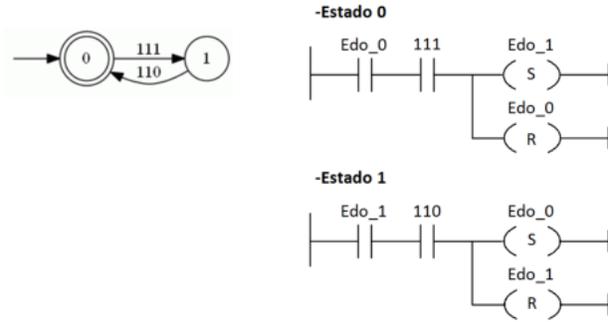


Fig. 12. Ejemplo de traducción de un AEF

su implementación en PLC. El comportamiento obtenido en las pruebas experimentales es satisfactorio y congruente con el supervisor obtenido ya que el funcionamiento del módulo se encuentra restringido a las especificaciones definidas.

REFERENCIAS

- Bucio, J. (2011). *Aplicación de control supervisor jerárquico-modular*. Depto. de Control Automático, CINVESTAV Guadalajara.
- Cassandras, C.G. and Lafortune, S. (2009). *Introduction to discrete event systems*. Springer Science & Business Media.
- De Queiroz, M.H. and Cury, J.E. (2002). Synthesis and implementation of local modular supervisory control for a manufacturing cell. In *Discrete Event Systems, 2002. Proceedings. Sixth International Workshop on*, 377–382. IEEE.
- Fabian, M. and Hellgren, A. (1998). Plc-based implementation of supervisory control for discrete event systems. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 3, 3305–3310. IEEE.
- Groover, M.P. (2000). *Automation, Production Systems, and Computer-integrated Manufacturing*. Prentice Hall.
- Hasdemir, İ.T., Kurtulan, S., and Gören, L. (2008). An implementation methodology for supervisory control theory. *The International Journal of Advanced Manufacturing Technology*, 36(3-4), 373–385.
- Hernández, E.G. (2005). *Modelado y control jerárquico-modular de la coordinación de recursos de sistemas de manufactura flexible*. Tesis de Maestría, Depto. de Ingeniería Eléctrica, CINVESTAV.
- Jaimes, F. (2004). *Modelado y control de sistemas automatizados de manufactura*. Tesis de Maestría, Depto. de Ingeniería Eléctrica, CINVESTAV.
- Wonham, W.M. (2017). *Notes on control of discrete event systems*. Dept. of Elec. and Comp. Eng., University of Toronto.