

# Diseño del programa de control para una celda de manufactura flexible didáctica

*Control program design for a flexible manufacturing didactic cell*

*Luis Diego Murillo-Soto<sup>1</sup>*

*Fecha de recepción: 27 de enero del 2014  
Fecha de aprobación: 05 de mayo del 2014*

Murillo-Soto, L. Diseño del programa de control para una celda de manufactura flexible didáctica. *Tecnología en Marcha*. Vol. 27, N° 3, Julio-Setiembre 2014. Pág 41-.52

<sup>1</sup> Profesor Asociado e Investigador; Instituto Tecnológico de Costa Rica, Escuela de Ingeniería Electromecánica, Cartago, Costa Rica, Correo electrónico: [lmurillo@tec.ac.cr](mailto:lmurillo@tec.ac.cr)

## Palabras clave

Celdas de manufactura; sistemas de eventos discretos; PLC; políticas de despacho; aproximación basada en matrices.

## Resumen

Una forma alternativa de programar el controlador de una Celda de Manufactura Flexible (CMF) es utilizar el enfoque propuesto en Huang, Pastravanu y Gurel (1995), con la teoría de Aproximación Basada en Matrices (ABM) para sistemas de eventos discretos (DE). Esta teoría se diseñó para su implementación en computadores industriales en las cuales la ejecución de operaciones matriciales binarias en tiempo real no representa problema alguno, sin embargo, la implementación de dichas operaciones en controladores programables (PLC) comerciales no se puede realizar en forma directa ya que la mayoría de PLC se programan mediante lenguajes gráficos, como el llamado Lógica en Escalera (*Ladder Logic Diagram, LLD*). Dicha restricción provoca que este poderoso enfoque de programación de DE excluya la mayoría de autómatas programables de uso comercial.

El presente trabajo describe la interpretación de la teoría ABM con la finalidad de implementarla en un PLC comercial. El programa de control diseñado se probó con tres políticas de producción, en una celda de manufactura flexible multireentrantante (MRF) ubicada en el Instituto Tecnológico de Costa Rica; el objeto de las pruebas fue observar el rendimiento de la celda de manufactura ante distintas políticas de despacho en la producción.

## Key words

Manufacturing cells; discrete event systems; PLC; dispatch policies; matrix based approach.

## Abstract

An alternative way to program the controller of a Flexible Manufacturing Cell (FMC) is to use the proposed theory by Huang, Lewis, Pastravanu & Gurel (1995) called the matrix based approach for discrete event systems. This theory was designed to run on computers, where execution of binary matrix operations in real time is not a problem; however the implementation of these operations in commercial programmable controllers (PLC) is not realizable in direct way, because most of the PLCs are programmed using graphics language as the called Ladder Logic Diagram (LLD). The above restriction causes this powerful programming tool, excludes the majority of the PLC.

This paper describes the interpretation of the Matrix Based Approach to be able to implement on a commercial PLC. The controller designed was tested with three production policies in a flexible manufacturing cell multi reentrant (MRF) located at the Technological Institute of Costa Rica, the purpose of the tests was to observe the performance of the manufacturing cell with different dispatch policies production.

---

## Introducción

Se entiende como Celda de Manufactura Flexible (CMF) un grupo de estaciones de trabajo interconectadas mediante sistemas automatizados de manejo y almacenamiento de materiales, todo esto controlado por un sistema computarizado. Las CMF se clasifican de acuerdo con su flexibilidad y según Bogdan, Lewis, Kovacic & Mireles (2006), los aspectos más comunes que se valoran son:

- Flexibilidad de máquina: consiste en la facilidad para realizar cambios en máquinas para producir distintos tipos de partes.
- Flexibilidad de proceso: habilidad de producir un conjunto de partes de forma distinta.
- Flexibilidad de producto: habilidad de cambiar a nuevos productos en forma económica y rápida.
- Flexibilidad de direccionamiento: habilidad de manejo de paros y fallas, con la finalidad de continuar produciendo distintos tipos de partes.

- Flexibilidad de volumen: habilidad de operar a distintos volúmenes de producción.
- Expansión de flexibilidad: habilidad de expandir el sistema en forma modular y de manera sencilla.
- Flexibilidad de operaciones: habilidad de intercambiar el ordenamiento de algunas operaciones para cada tipo de producto.
- Flexibilidad de producción: total de partes que un sistema de manufactura flexible puede producir.

Uno de los problemas principales en la construcción de CMF radica en desarrollar algoritmos de control suficientemente robustos, que puedan adaptarse en forma simple a los cambios en la dinámica del sistema productivo debido a la búsqueda de la flexibilidad. En este contexto, la elaboración de algoritmos de control para establecer los itinerarios de las operaciones representa un problema central de investigación en el área de sistemas dinámicos de eventos discretos. La necesidad de comprender la dinámica del sistema, su análisis y su control ha conducido a la investigación de mejores metodologías y herramientas de diseño, tal es el caso de la teoría llamada Aproximación Basada en Matrices (ABM). Sin embargo, como se mencionó, utilizar esta teoría requiere de computadores industriales que puedan realizar operaciones matriciales booleanas, tal y como se muestra en el trabajo de Mireles y Lewis (2001), y debido a su naturaleza matricial es de difícil aplicación en controladores programables (PLC) comerciales que se programen en lenguajes gráficos, por tal razón este trabajo muestra una implementación para este tipo de PLC.

### Controlador basado en matrices

La ABM se entiende como un conjunto de reglas lógicas que plasman la dinámica del sistema discreto, de tal forma que se logra describir la relación entre los distintos recursos, las operaciones que realizan, las partes dentro del sistema y las señales de sensores y los actuadores. Este conjunto de reglas lógicas de la CMF que se establecen en forma de matrices booleanas y vectores de señales describen la naturaleza del sistema de eventos discretos a través de un vector de estados llamado  $x$ . A continuación se muestran las ecuaciones para el desarrollo del

algoritmo del controlador de la CMF y se describen ampliamente en los trabajos de Bogdan, Lewis, Kovacic y Mireles (2006), Huang (2001a, 2001b) y Lewis, Gurel, Bogdan, Doganalp y Pastravanu (1998):

Vector de estado del sistema

$$\bar{x}(k) = F_v \Delta \bar{v}_c(k-1) \nabla F_r \Delta \bar{r}_c(k-1) \nabla F_u \Delta \bar{u}(k-1) \nabla F_D \Delta \bar{u}_D(k-1) (1)$$

Vector de inicio de trabajos:

$$v_s(k) = S_v \Delta x(k) \quad (2)$$

Vector de liberación de recursos:

$$r_s(k) = S_r \Delta x(k) \quad (3)$$

Vector de productos salientes de la celda:

$$y(k) = S_y \Delta x(k) \quad (4)$$

El significado de cada matriz es el siguiente:

- $F_v$  matriz que establece la secuencia de trabajo; muestra el orden de las tareas del proceso de ensamble.
- $F_r$  matriz que establece los requerimientos de recursos; muestra los recursos necesarios para llevar a cabo cada tarea, por lo tanto, relaciona las tareas ejecutadas por cada recurso.
- $S_v$  matriz de inicio de tareas; indica para cada tarea cuales son sus tareas previas.
- $S_r$  matriz de liberación de recursos; indica cuales tareas liberan un determinado recurso una vez que finalizan.
- $F_u$  matriz que establece el ingreso de partes al sistema.
- $S_y$  matriz que declara la lógica de salida para los productos.
- $F_D$  matriz de control de despacho.

Los vectores binarios presentes en las ecuaciones significan lo siguiente:

- $x$  es el vector de estado del sistema y describe las reglas lógicas que están activas.
- $v_c$  es el vector que indica cuales tareas se han completado.
- $r_c$  es el vector que indica los recursos disponibles para ser utilizados para una tarea.
- $v_s$  es el vector que indica cuando iniciar una tarea.

- $r_s$  es el vector que indica en que estados se deben liberar los recursos.
- $u$  es el vector que indica el producto entrante en la celda CMF.
- $u_D$  es el vector que establece las políticas de despacho para la parte o subproducto.

En la ecuación (1) el símbolo  $\Delta$  representa la conectiva lógica AND entre dos matrices booleanas; por otra parte, el símbolo  $\vee$  significa la conectiva lógica OR entre dos matrices booleanas. Por ejemplo, si A y B son matrices booleanas, de tamaño  $n \times m$  y  $m \times n$  respectivamente, la operación  $C=A\Delta B$  se resuelve para cada celda como  $c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2j}) \vee \dots \vee (a_{im} \wedge b_{mj})$ . Además, la operación  $C=AB$ , se resuelve para la cada celda  $c_{ij} = (a_{i1} \vee b_{1j})$ . Donde los símbolos  $\wedge$  y  $\vee$  indican la conectiva lógica O y la conectiva lógica Y respectivamente. La barra superior sobre un vector denota la negación de todas las variables del vector.

La figura 1 muestra el esquema básico entre el controlador y la celda de manufactura, el cual fue propuesto por Huang, Lewis, Pastravanu y Gurel (1995). El controlador debe ser programado de acuerdo a las ecuaciones (1-4), con la finalidad de obtener un algoritmo que permita administrar adecuadamente las tareas y recursos de la celda, con el propósito de evitar las siguientes dos condiciones: conflictos por un mismo recurso o bloqueo de recursos en el sistema.

La construcción de las matrices  $F_v, F_r, S_v, S_r$ , depende de la información que se extrae de la estructura de la celda, específicamente de la secuenciación de las tareas y asignación de los recursos. Para el presente trabajo se usará el enfoque propuesto (Lewis, Gurel, Bogdan, Doganalp y Pastravanu, 1998; Mireles y Lewis, 2001 y Zhou y Fanti, 2004), en donde las matrices se obtienen del modelo de Red de Petri (RdP) de la celda de manufactura flexible. Una Red de Petri binaria es un grafo dirigido que posee dos tipos de nodos llamados *lugares* y *transiciones*. Los nodos *lugares* se representan por círculos e indican las tareas o los recursos de la celda. Por otro lado, los nodos *transiciones* están representados por barras rectangulares e indican los eventos de la celda, entendiéndose como el inicio o finalización de tareas, liberación de recursos, etc. Entre estos dos tipos de nodos se ubican los arcos dirigidos, los cuales se encargan de unir las transiciones con los lugares y viceversa. Cada arco dirigido posee un número que indica su peso; este peso determina la cantidad de *marcas* que consumen o depositan en un nodo lugar. Las marcas representan la cantidad de partes que se consumen en una operación. Las marcas se representan por puntos dentro de los nodos lugar. De acuerdo con Murata (1989), la RdP se define como una quintupla:

$$RdP = (P, T, F, W, M_0) \quad (5)$$

Donde  $P = \{p_1, p_2, \dots, p_m\}$  es el conjunto finito de lugares,  $T = \{t_1, t_2, \dots, t_n\}$  es el conjunto finito de transiciones,  $F \subseteq (P \times T) \cup (T \times P)$  es el conjunto de arcos

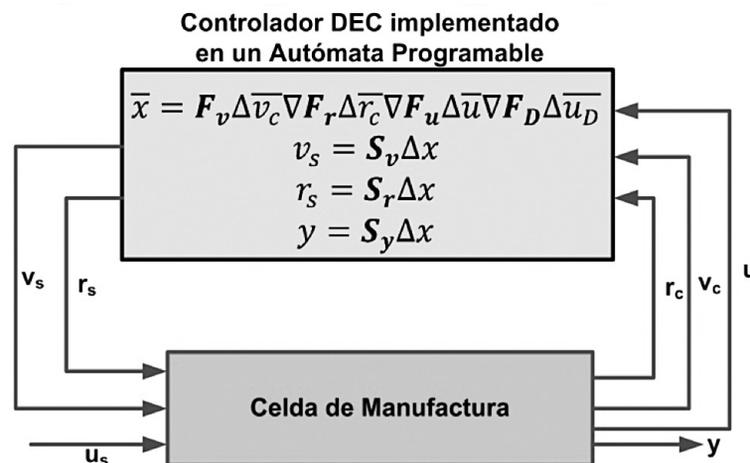


Figura 1. Relación entre el autómata programable y la celda flexible de manufactura.

dirigidos,  $W:F \rightarrow \{1, 2, 3, \dots\}$  es la función de pesos de los arcos,  $M_0:P \rightarrow \{1, 2, 3, \dots\}$  es el marcado inicial de la red. Finalmente, los lugares y transiciones son dos conjuntos independientes que no se intersecan,  $P \cap T = \emptyset$  y  $P \cup T \neq \emptyset$ . El modelado de celdas de manufactura es un tópico ampliamente estudiado. Para más información sobre el modelado con RdP, ver Silva (1998), Zhou (2000), Zhou y Kurapati (2000) y Hruz y Zhou (2007).

Para el diseño del controlador de eventos discretos (DEC), este recibe los vectores  $v_c, r_c$  y  $u_D$  y las salidas del controlador son  $v_s, r_s$ . Estos vectores son vectores de señales digitales entre el DEC y los recursos de la CMF. Sin embargo, el vector  $u_D$  es el responsable de indicar cuando un elemento  $x_i$  del vector de estados debe habilitarse; este vector se construye a partir de la política de despacho implementada y su función es asignar recursos a tareas sin que la CMF se bloquee.

### Bloqueos en CMF y políticas de despacho

Las celdas de manufactura flexible donde al menos una máquina realiza más de una tarea distinta sobre las piezas se conocen como celdas de flujo multi-reentrantes (MRF). Las celdas de manufactura MRF son susceptibles al bloqueo si se realiza una mala asignación de los recursos disponibles. Una forma de asignar los recursos y construir el vector  $u_D$ , es mediante políticas de despacho, las cuales buscan evitar que la celda entre en estado de bloqueo. Las políticas más conocidas son las llamadas *Last Buffered First Served* (LBFS), *First Buffered First Served* (FBFS) y *Maximum Work In Progress* (MAXWIP), descritas por Mireles et al. (2001), Mireles y Lewis (2001) y Bogdan et al. (2006). A partir del estudio topológico de la red de Petri, Murillo-Soto (2013) plantea la construcción de la política de despacho.

La política LBFS sirve primero la última parte almacenada en una pila y consiste en que cada vez que haya un conflicto con un recurso, primero se debe realizar la operación más cercana al producto terminado, con lo cual se garantiza evitar los bloqueos en el sistema; es decir, se da prioridad a las tareas que finalizan producto sobre el ingreso de nuevas partes. Por otra parte, la política FBFS sirve primero al primer subproducto almacenado y consiste en que en caso de conflicto se le da prioridad a las operaciones que ingresan producto a la celda, siempre y cuando la realización de esas operaciones no cause

un bloqueo. La política MAXWIP es una combinación de FBFS y LBFS, y busca que la cantidad de producto dentro de la celda sea lo más cercano al máximo, para lograr una alta utilización de los equipos. La posibilidad de bloqueo en estas dos últimas políticas de control es mayor. La implementación de estas tres políticas para PLC se encuentra en Bogdan et al. (2002).

Para construir de manera adecuada dicha política se debe realizar un estudio de las esperas circulares (EC) de la celda de manufactura. Las EC son el conjunto de recursos  $R$  de la celda, que están en una cadena de relaciones de espera, y esta cadena de relaciones es circular. La relación de espera entre dos recursos se denota como  $(r_i \rightarrow r_j)$  y significa que la tarea ejecutada por  $r_i$  no se inicia hasta que se libere el recurso  $r_j$ . Una espera circular simple (ECs) se denota como  $(r_i \rightarrow r_j \rightarrow r_k \rightarrow r_p \rightarrow r_i)$ , donde los recursos necesarios para la espera deben ser al menos dos (ver Lewis, Gurel, Bogdan, Doganalp y Pastravanu, 1998). Para construir una política que utiliza las esperas circulares, se hace referencia a los teoremas propuestos por Huang et al. (1996), Huang, Lewis y Tacconi (1996) y Lewis et al. (1998), en donde se establece el concepto de sifones críticos ( $Sc$ ) y se indica que limitando las tareas del sifón, se evita el bloqueo. A continuación los teoremas:

*Teorema 1:* Teorema principal para evitar bloqueos. Sea  $C$  una espera circular y  $Sc$  su sifón crítico. Entonces,  $C$  está en bloqueo circular (BC) si y solo si el sifón crítico  $Sc$  está vacío.

*Teorema 2:* Definición y construcción del sifón crítico

$$Sc = C \cap J(C)_+ = C + \sum_{i=0}^n \{ \bullet \{ r_{si} \} + \cap J \} \quad (6)$$

Donde:

$Sc$  es el conjunto de recursos y tareas de la celda que forma el sifón crítico.

$C$  son los recursos que componen una espera circular.

$J$  tareas de la celda de manufactura.

$J(C)_+$  conjunto de tareas que realizan los recursos  $C$ , cuyos arcos de salida no apuntan a nodos ubicados dentro de la espera circular  $C$ .

$r_{si}$  es el conjunto de recursos compartidos dentro de  $C$ .

$\{\bullet r_{si}\}_+$  es el conjunto de transiciones cuyos arcos apuntan a  $r_{si}$  y no tienen arcos de entrada de otros recursos dentro de la espera circular C.

$\bullet\{r_{si}\}_+$  es el conjunto de tareas cuyos arcos de salida apuntan a las transiciones definidas por  $\{\bullet r_{si}\}_+$ .

### Celda de manufactura

La celda de manufactura utilizada para probar el programa de control consta de cuatro entradas de partes: un robot, una máquina CNC y un almacén de tamaño uno. La distribución de planta de la celda de manufactura, así como una fotografía se muestra en la figura 2.

La celda de manufactura realiza un ensamble de una caja de engranes plásticos de uso didáctico. La caja de engranajes consiste de cuatro partes: la base de la caja o carcasa, dos engranes plásticos y la tapa del conjunto, estas partes se aprecian en la figura 3a. El proceso de ensamble consiste en tomar un acrílico de la banda transportadora y colocarlo en la CNC donde se realizan unos agujeros, cuando se finaliza esta tarea, se coloca el acrílico en un *buffer* para liberar la CNC. De manera paralela, se debe colocar la caja en la mesa de ensamble y después colocarle

los dos engranes. Finalmente, se toma el acrílico del *buffer* para colocarlo en el subensamble, una vez hecho esto, el robot se lleva el producto terminado hacia la banda transportadora para su salida. El árbol de ensamble se muestra en la figura 3b.

### Equipo de la celda de manufactura

La celda de manufactura se controla con un PLC CompactLogix 5330 de Allen-Bradley, que cuenta con dos módulos de entradas digitales, dos módulos de salidas digitales, un módulo de Controlnet y un módulo de comunicación por Ethernet. La celda se compone además de un robot de ensamble marca Amatrol Pegasus, una máquina CNC Micromill IST, un almacén y una estación de ensamble. La celda cuenta con banda transportadora para la salida de productos, así como dispensadores de las partes a ensamblar:

### Software de simulación y programación

La simulación de los algoritmos de control se realizó con el software Petri.Net. Este paquete permite simular sistemas de eventos discretos a través de RdP binarias temporizadas. El software fue desarrollado por el Departamento de Control y Ingeniería de Computación de la Facultad de Electrotecnia y Computación de la Universidad de Zagreb. Para más detalles del paquete, ver Bogdan et al. (2004).

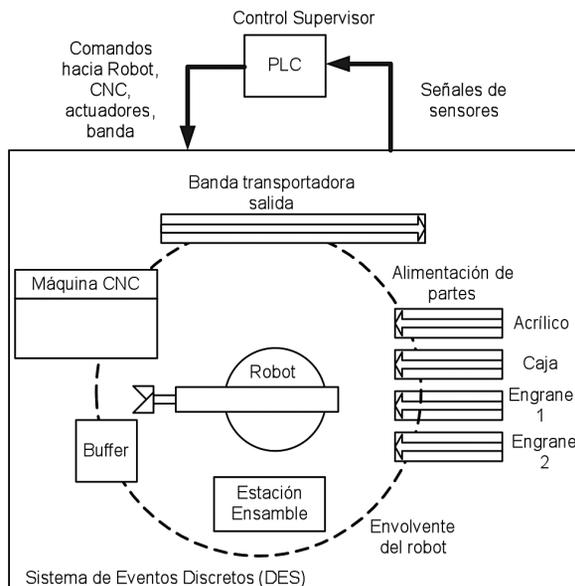


Figura 2. Layout de la celda de manufactura y una fotografía de la celda.

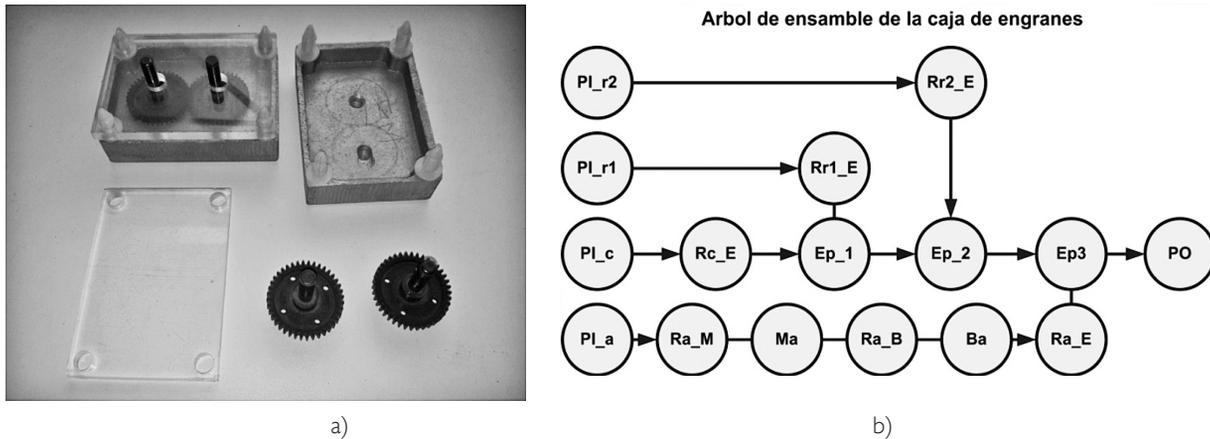


Figura 3. a) Caja de engranes ensambladas, caja metálica, engranes y acrílico. b) árbol de ensamble de la caja de engranes

Cuadro 1. Operaciones de la celda de laboratorio.

Notación	Tareas
PI_a	Ingreso de las láminas de acrílico a la celda por el alimentador A.
PI_c	Ingreso de las cajas de aluminio por el alimentador B.
PI_r1	Ingreso del engrane r1 por el alimentador C.
PI_r2	Ingreso del engrane r2 por el alimentador D.
Ra_M	El robot toma un acrílico de la banda y lo pone en la máquina CNC.
Ma	La máquina CNC desbasta el acrílico, realiza agujeros.
Ra_B	El robot toma el acrílico recién maquinado y lo pone en el <i>buffer</i> .
Rc_E	El robot toma una caja de la entrada y la coloca en la mesa de ensamble.
Ba	<i>Buffer</i> en uso.
Ep_1	Subensamble 1 terminado.
Rr1_E	El robot ensambla la rueda 1.
Ep_2	Subensamble 2 terminado.
Rr2_E	El robot ensambla la rueda 2.
Ep_3	Subensamble 3 terminado.
Ra_E	El robot toma un acrílico del <i>buffer</i> , lo ensambla y se lleva el producto terminado a la banda.
PO	Salida del producto terminado por la banda transportadora.

Para programar el controlador RS5330 se utilizó el software *RSLogix5000 Pro*, Versión 16.00 de Allen-Bradley. El robot se programó con *Pegasus II Control Software*, Versión 1.1.4.

## Metodología

La elaboración y prueba de un controlador matricial en la celda de manufactura se realizó en las siguientes cuatro etapas:

- Modelado de la celda de manufactura.
- Simulación del controlador.
- Implementación del controlador.
- Pruebas de ejecución.

## Modelado de la celda de manufactura

En esta etapa se construyó una red de Petri que modela los recursos, las operaciones, las partes a ensamblar y el flujo de estas, con la finalidad de

comprender la dinámica de la celda de manufactura. La figura 4 muestra la RdP de la celda de ensamble.

Los recursos de la celda son el robot R, la máquina CNC M, el almacén B y la mesa de ensamble E; estos recursos se representan en la RdP como nodos tipo lugar llamados  $R = \{R, M, B, E\}$ . Las tareas que ejecutan estos recursos se enlistan en el cuadro 1 y conforman los nodos lugar que representan las tareas  $J = \{Ra\_M, Ra\_B, Ra\_E, Rc\_E, Rr1\_E, Rr2\_E, Ma, Ep\_1, Ep\_2, Ep\_3, Ba\}$ . Los recursos, tareas y la secuencia de actividades se determinan los nodos tipo transición (barras verticales), y además se determinan cuales transiciones se encuentran en conflicto (barras claras en figura 4). Para cada transición en conflicto se coloca un nodo lugar de control, este nodo se representa con doble círculo y se nombra  $u_{Di}$ . La función de estos nodos es facilitar la implementación de la política de control. También se establecieron las cadenas de recursos que pertenecen a esperas circulares (EC), para esta RdP existen dos EC simples,  $C1 = \{M, R\}$  y  $C2 = \{B, R\}$ , para cada espera circular se buscan los recursos y tareas que constituyen un sifón crítico. El cálculo de los sifones críticos se realiza de acuerdo a la ecuación (6); el sifón crítico uno (Sc1) se compone de los siguientes lugares  $\{M, R, Ra\_B, Rc\_E, Rr1\_E, Rr2\_E, Ra\_E\}$  y el

sifón crítico dos (Sc2) se compone de  $\{B, R, Ra\_E, Ra\_M, Rc\_E, Rr1\_E, Rr2\_E\}$ .

La estrategia de control consiste en hacer cumplir el teorema 1 de la sección 2.1. Para que no se bloquee la celda de manufactura, tanto las políticas LBFS como FBFS deben lograr en todo momento al menos una marca en algún nodo lugar, ya sea Sc1 o Sc2, en otras palabras, nunca se deben vaciar los sifones críticos. Para lograr cumplir con el teorema se utilizan los nodos de control  $u_{Di}$  como "semáforos" que controlan el flujo de marcas en la red.

### Simulación del controlador

Una vez terminado el análisis estructural de la celda de manufactura, se procede a simular el comportamiento y rendimiento de la celda ante distintas políticas de producción. La celda de manufactura se probó primero sin ninguna regla de control, y luego con tres tipos de reglas de despacho de operaciones. Primero se probó la celda con una política secuencial sin prioridades, luego se probó la política FBFS y por último la política LBFS.

El oscilograma de la simulación que se muestra en la figura 5 indica que la CMF se puede bloquear en tan solo el octavo paso de simulación cuando no hay política de despacho implementada, por lo que se

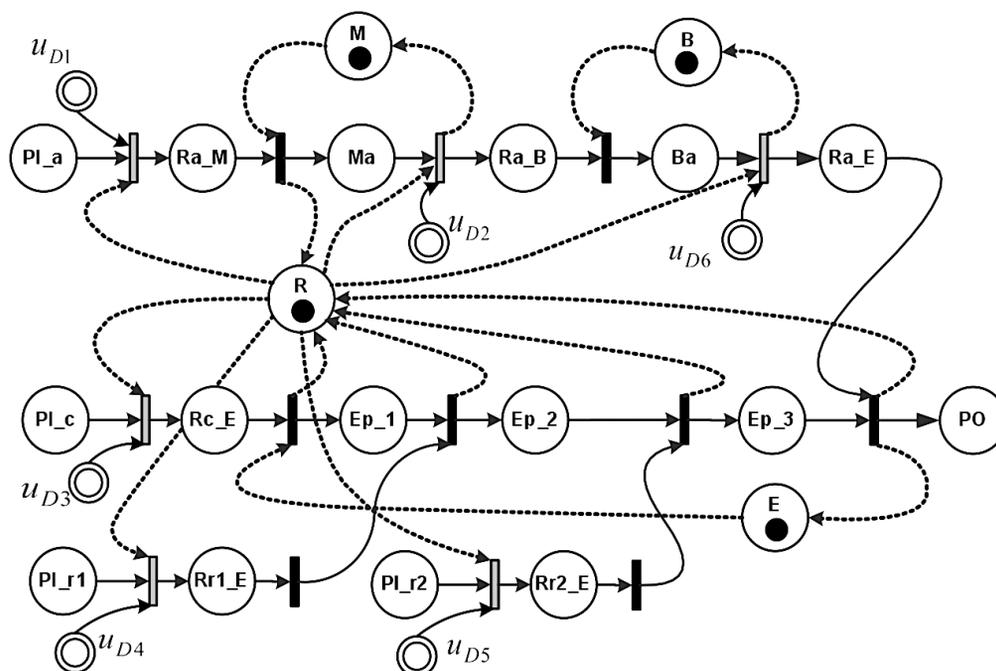


Figura 4. Modelo de la celda de manufactura usando una Red de Petri controlable.

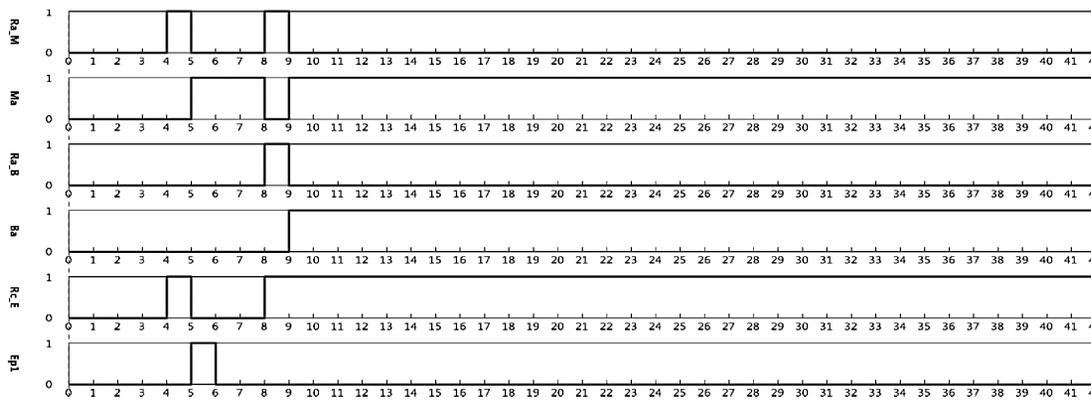


Figura 5. Oscilograma de la CMF que muestra el bloqueo de la celda cuando no hay una política definida.

demuestra que se requiere de un control supervisor para la celda de manufactura.

Las simulaciones para una política tipo LBFS y tipo FBFS arrojaron que la política LBFS tiene un rendimiento menor que la FBFS, es decir, la política FBFS logra que la celda de manufactura brinde una mayor cantidad de piezas por unidad de tiempo o por pasos de simulación. El cuadro 2 muestra la forma en que se programaron las dos políticas en el simulador.

### Implementación del controlador en el PLC

El modelo matricial se programó usando el lenguaje escalera del PLC RSLogix 5330. El propósito de este controlador es coordinar todos los trabajos de la CMF entre PLC, CNC, Robot, Almacén y Banda transportadora. El PLC es con controlador-

supervisor centralizado y para construirlo según la ecuación (1) se requiere de un vector de estados del sistema  $x$  que a su vez requiere de tres vectores de señales, las cuales son: el vector de tareas completadas  $v_c$ , el vector de recursos disponibles  $r_c$  y el vector de productos en la entrada  $u$ . Además, se requiere de una política de control o “ley de control”, que define cuándo se asigna una marca al nodo de control  $u_{D_i}$ . Con la información anterior, el autómata calcula el estado del sistema  $x$ , el vector de comandos para el inicio de tareas llamado  $v_s$ , así como el vector de comandos para la liberación de recursos  $r_s$ .

La figura 6 muestra la interface gráfica del RSLogix5000; al lado izquierdo se muestra la rutina principal (*main*) que invoca a cada uno de los módulos programados. Por cada matriz  $F_v$ ,  $F_r$ ,  $S_v$ ,  $S_r$

Cuadro 2. Políticas de despacho para la RdP de la celda.

Política LBFS	Política FBFS
<p>If (I==1) THEN ((Ud1=0 AND Ud2=0 AND Ud3=0 AND Ud4=0 AND Ud5=0 AND Ud6=0))</p> <p>If(PI_c&gt;0 AND E==1) THEN Ud3=1</p> <p>If (PI_r1&gt;0 AND Ep_1==1) THEN Ud4=1</p> <p>If (PI_r2&gt;0 AND Ep_2==1) THEN Ud5=1</p> <p>If ( Ba==1 AND Ep_3==1) THEN Ud6=1</p> <p>If (Ma==1 AND Ba==1 AND Ep_3==1) THEN Ud6=1</p> <p>If(PI_a&gt;0 AND Ma==1 AND Ba==1 AND Ep_3==1) THEN Ud6=1</p> <p>If( Ma==1 AND Ba==0 AND Ep_3==1) THEN Ud2=1</p> <p>If (PI_a&gt;0 AND Ma==0 AND Ba==0 AND Ep_3==1) THEN (Ud1=1)</p>	<p>If (I==1) THEN ((Ud1=0 AND Ud2=0 AND Ud3=0 AND Ud4=0 AND Ud5=0 AND Ud6=0))</p> <p>If ( Ba==1 AND Ep_3==1) THEN Ud6=1</p> <p>If (Ma==1 AND Ba==1 AND Ep_3==1) THEN Ud6=1</p> <p>If(PI_a&gt;0 AND Ma==1 AND Ba==0) THEN Ud2=1</p> <p>If (PI_a&gt;0 AND Ma==0) THEN (Ud1=1)</p> <p>If(PI_c&gt;0 AND E==1 AND Ud1==0 AND Ud2==0) THEN Ud3=1</p> <p>If (PI_r1&gt;0 AND Ep_1==1) THEN Ud4=1</p> <p>If (PI_r2&gt;0 AND Ep_2==1) THEN Ud5=1</p>

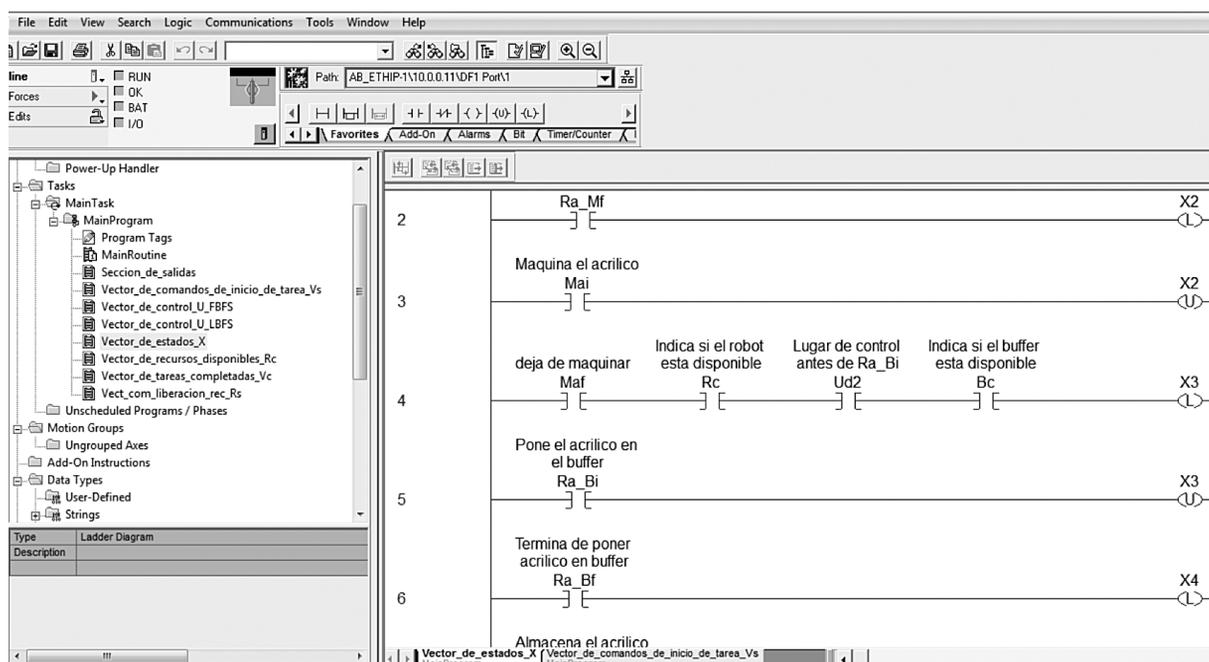


Figura 6. Pantalla del software RSLogix5000. A la izquierda se muestran, en el árbol de desarrollo del proyecto, los módulos programados. El código LLD mostrado calcula el vector de estados  $x$ .

$S_v$ , y vectores boléanos correspondientes que son manipulados por un operador matricial, se forma un nuevo vector, el cual se programa como módulo en el PLC. Por tanto, se crean cinco módulos, las señales de los sensores son mapeadas a marcas de memoria internas por módulo. Dos módulos adicionales surgen de la implementación de las dos políticas de control LBFS y FBFS. Por último, está el módulo que calcula el vector de estados  $x$ ; la forma en que se calcula este vector es construyendo un diagrama escalera LLD, que surge al desarrollar la ecuación (1); para cada estado se construye una rama de diagrama LLD, este diagrama siempre será una AND entre elementos de distintos vectores involucrados. Los módulos implementados son:

- Módulo que calcula el vector de recursos disponibles  $r_c$ , de acuerdo a sensores o señales del robot y CNC.
- Módulo que calcula el vector de tareas finalizadas.
- Módulo que calcula las instrucciones de liberación de recursos de acuerdo al vector de estado de la celda  $r_s$ .
- Módulo de comandos para el inicio de tareas de acuerdo con las transiciones disparadas  $v_s$ , señales al robot, CNC, pistones neumáticos, banda transportadora.

- Módulo de salidas de producto.
- Dos módulos que definen las políticas de control de la celda. La ley de control utiliza la información del estado de las entradas, recursos y tareas completadas, para definir la asignación del nuevo recurso.
- Módulo que calcula el estado del sistema, con la finalidad de ver si se dispara alguna transición  $x$ .

La CMF posee dos máquinas adicionales, la máquina CNC y el Robot, que requieren programación. La máquina CNC posee un programa en código G, que cuando el PLC le indica a través de una señal binaria, le hace dos agujeros al acrílico. En cuanto al robot, este se programó con las seis tareas que se indican en la Tabla 1, el PLC le ordena al *driver* del robot mediante un código binario de tres cifras cual tarea se debe ejecutar.

### Pruebas de ejecución

En esta etapa se realizaron las mediciones de los tiempos de fabricación para cada una de las políticas de despacho. En el caso del controlador con política LBFS, el sistema también se comporta como una rutina rígida o secuencial. Por otra parte, cuando se corre el controlador con política FBFS, el orden de tareas se vuelve mucho más flexible y varía según

la combinación de entradas, tareas completadas y recursos disponibles que tenga la celda. Es importante mencionar que ninguna política programada generó bloqueos de la celda. Esto ratifica la teoría de que los sistemas MRF son fácilmente controlables con estas dos políticas de control (Bogdan, Lewis, Kovacic, Gurel y Stajdohar, 2002).

En cuanto a resultados medibles, existe un pequeño aumento del tiempo de fabricación al utilizar el controlador basado en matrices con política LBFS, por dos razones. La primera es la baja velocidad del robot (uso didáctico no industrial) y la segunda razón es debido a que al inicio de cada tarea el robot regresa al origen de referencia (*home*), incrementando los tiempos de la rutina en forma innecesaria. Con la implementación de la política FBFS se mejoró el tiempo respecto a la política LBFS y a la programación lineal; a continuación se muestran los tiempos:

1. Con la secuencia preprogramada le tomó en promedio 17:39 minutos para ensamblar 3 piezas.
2. Con el algoritmo de control LBFS le tomó en promedio 19:30 minutos para ensamblar 3 piezas.
3. Con el algoritmo de control FBFS le tomó en promedio 14:56 minutos para ensamblar 3 piezas.

### Análisis de resultados

Se considera que se cumplió el objetivo de implementar un controlador de eventos discretos en un PLC a partir del controlador matricial propuesto por Huang et al. (1995). Entre los resultados concretos que se destacan están:

- Se desarrollaron dos algoritmos de control para una celda de manufactura flexible (CMF) de laboratorio, basados en políticas LBFS y FBFS. La política FBFS es más eficiente que la LBFS, presentando una mejora en el tiempo de producción del 24,6%
- La forma en que se programó constituye una alternativa diferente de plantear el algoritmo de control, debido a que es fácil cambiar la secuencia o el control sin necesidad de cambiar todo el programa. Una vez que se tiene el

modelo en el PLC, es suficiente con cambiar la función de la política de control para que el sistema se comporte de manera diferente.

- Se determinó que la CMF es susceptible a bloqueos, ya que la ecuación (1) no contempla ningún manejo de faltas o fallas en las máquinas. Aquí se abre un posible trabajo de investigación, que consiste en ampliar el modelo matricial propuesto (Huang, Lewis, Pastravanu y Gurel, 1995) e incorporar el manejo de fallas y faltas de la celda a través de una segunda política de administración de los equipos.

### Reconocimientos

Se agradece a la Vicerrectoría de Investigación y Extensión del Instituto Tecnológico de Costa Rica por el apoyo brindado al proyecto VIE 5402-1341-1101, así como a la Escuela de Producción Industrial por el préstamo del laboratorio.

### Bibliografía

- Ballal, P., Lewis, F., Mireles, J. & Sreenath, K. (2007). Deadlock avoidance for free choice multi-reentrant flow lines: Critical siphons and critical subsystems. *IEEE Proc. Mediterranean Conf. Control & Automation*.
- Bogdan, S., Lewis, F. L., Kovacic, Z., Gurel, A. & Stajdohar, M. (2002). An Implementation of Matrix-Based Supervisory Control of Flexible Manufacturing Systems. *IEEE Transactions on Control Systems Technology*, 10(5), 709-725.
- Bogdan, S., Kovacic, Z., Smolic-Rocak, N. & Birgmajer, B. (2004). A matrix approach to an FMS control design. *IEEE Robotics & Automation Magazine*, 11(4), 92-109.
- Bogdan, S., Lewis, F. L., Kovacic, Z. & Mireles, J. (2006). *Manufacturing System Control Design: A Matrix-based Approach*. London: Springer-Verlag.
- Groover, M. (1987). *Automation, Production System and Computer Integrated Manufacturing*. New Jersey: Prentice-Hall.
- Huang, H. H., Lewis, F., Pastravanu, O. y Gurel, A. (1995). Flow-shop scheduling design in an FMS matrix framework. *Control Eng. Practice*, 3(4), 562-568.
- Huang, H. H., Lewis, F. L. & Tacconi, D. (1996). Deadlock analysis using a new matrix-based controller for reentrant flow line design. *Proceedings IEEE IECON 22nd International Conference*. Taiwán: IEEE Industrial Electronics, Control, and Instrumentation.
- Huang, H. H. (2001a). Matrix Controller Design and Deadlock Analysis of Automated Manufacturing Systems. Part I. Designing the Matrix-Based Controller. *The International Journal of Advanced Manufacturing Technology*, 18(6), 434-447.

- Huang, H. H. (2001b). Matrix Controller Design and Deadlock Analysis of Automated Manufacturing Systems. Part 2: Deadlock Avoidance Policy. *The International Journal of Advanced Manufacturing Technology*, 18(7), 490-501.
- Hruz, B. & Zhou, M. (2007). *Modelling and Control of Discrete-event Dynamic Systems - (Advanced textbooks in control and signal processing)*. London: Springer-Verlag.
- Lewis, F. L., Gurel, A., Bogdan, S., Doganalp, A. & Pastravanu, O. C. (1998). Analysis of Deadlock and Circular Waits Using a Matrix Model for Flexible Manufacturing Systems. *Automatica*, 34(9), 1083-1100.
- Mireles, J. & Lewis, F. L. (2001). Intelligent Material Handling: Development and Implementation of Matrix-Based Discrete-Event Controller. *IEEE Transactions on Industrial Electronics*, 48(6), 1087-1097.
- Mireles, J. & Lewis, F. L. (2002). Deadlock analysis and routing on free-choice multipart reentrant flow lines using a matrix-based discrete event controller. *Proceedings of the 41st IEEE Conference on Decision and Control*, 1 (pp. 793-798). US: Las Vegas.
- Murata, T. (1989). Petri nets: Properties, Analysis and Application. *Proceeding of the IEEE*, 77(4), 541-580.
- Murillo-Soto, L. D. (2013). Política de despacho para evitar bloqueos en celdas de manufactura flexible del tipo multi-reentrante. *Tecnología en Marcha*, 26 (4).
- Silva, M. (1998). Petri nets and Production Systems. *LNCS: Lecture on Petri Nets Application II* (1492), 85-124.
- Tacconi, D. A. (1997). A New Matrix Model for Discrete Event Systems: Application to Simulation. *IEEE Control Systems*.
- Zhou, M. & Kurapati, V. (2000). *Modeling, Simulation and Control of Manufacturing Systems: A Petri nets Approach*. New Jersey: World Scientific.
- Zhou, M. & Fanti, M. P. (Eds.). (2004). *Deadlock Resolution in Computer Integrated Systems*. New York: Marcel Dekker.