



Modelado y Control de una Celda Automatizada de Manufactura Flexible basado en Redes de Petri

Sergio A. Foyo Valdés, Erika S. Puga Velazquez, Eduardo G. Hernández Martínez
División de Ingeniería Mecatrónica -TESCo, Estado de México, 55700 Coacalco, México.

sergiofoyo@live.com.mx, erika_selenep@hotmail.com, eghm2@yahoo.com.mx

Teléfono: 01(55)-21-594324

Resumen- En este trabajo, se propone una metodología que: (1) modela una celda de un sistema de manufactura flexible utilizando redes de Petri y (2) realiza la conversión del modelo obtenido a la programación de hardware de bajo nivel. La propuesta representa una solución al problema que presentan las redes de Petri para su implementación en sistemas dinámicos de eventos discretos, como lo son las celdas automatizadas de manufactura flexible.

Palabras clave: Sistemas dinámicos de eventos discretos, automatización, sistemas de manufactura, modelos de seguimiento de control, controladores lógicos

I. INTRODUCCIÓN

El proceso de diseño y control de celdas pertenecientes a sistemas de manufactura flexible debe considerar varios aspectos, desde la complejidad de la configuración del sistema, hasta el cambio de planes de trabajo (Chandra and Kumar, 2001). Un sistema automatizado de manufactura flexible (FMS, por sus siglas en inglés) integra todos los elementos importantes de la manufactura en un sistema altamente automatizado. Un FMS, consta de un robot industrial con servicio a máquinas de control numérico computarizado (CNC) y un sistema de manejo y almacenamiento de materiales, todo controlado por una computadora central (Kalpakjian y Schmid, 2008). La obtención de un modelo dinámico para el sistema es necesario para iniciar su análisis, detectar comportamientos no deseados como bloqueos o de conflicto y obtener esquemas de control para establecer una dinámica deseada en la ejecución de tareas concurrentes y flexibles (Chandra et al, 2003). Comúnmente, la coordinación de tareas de estos sistemas está basado en la ocurrencia de eventos, por lo que es posible modelar mediante la teoría de sistemas dinámicos de eventos discretos (Cassandras, and Lafortune, 1999). Este modelo dinámico debe ser capaz de incorporar situaciones de concurrencia, asincronía, bucles muertos, conflictos, exclusión mutua, etc. Las redes de Petri (rdP) (Murata, 1989) pueden ser usadas para modelar estas propiedades que caracterizan a los sistemas dinámicos de eventos discretos (Wang, 1998 y Zhou and Venkatesh, 1999).

En (Sánchez, 2004) se expone el modelado, simulación y control de FMS, apoyándose en redes de Petri, como herramienta de diseño de metodologías aplicadas en FMS, proponiendo modelos temporizados, principalmente electro-neumáticos. Algunos enfoques se centran en el estudio de las redes temporizadas de Petri, mostrando teoría y

aplicaciones (Desrochers and Jaar, 1995). Sin embargo, a pesar de que las rdP tienen un poder de representación gráfico y analítico importante para esta clase de sistemas dinámicos, se ha visto limitada su implementación física en sistemas automáticos de manufactura, debido a que no existe una metodología general para traducir la rdP al código de programación del hardware que controla los distintos elementos del sistema. (Bulach et al, 2002). Esto ha traído en consecuencia que únicamente se lleve a cabo una programación modular (Piedrafita, 2004) teniendo un detrimento en el desempeño global del sistema.

En este trabajo, se propone una metodología para modelar una celda de manufactura flexible y realizar la conversión del modelo obtenido a la programación del hardware de bajo nivel. Para ilustrarlo se aplica en el FMS de propósito académico.

El resto del documento está organizado de la siguiente manera. La sección II presenta una breve introducción de las rdP. La sección III describe el planteamiento del proceso y el FMS a utilizar, así como las características principales de sus elementos. En la sección IV se obtiene el modelo dinámico para la celda de manufactura considerada; adicionalmente se muestra el análisis basado en una analogía a la técnica de árboles de alcanzabilidad representada mediante matrices. La sección V describe la metodología de traducción a programación de bajo nivel (diagrama de escalera), basado en el modelo de rdP obtenido. Finalmente, en la sección VI se presentan algunas conclusiones y perspectivas.

II. INTRODUCCIÓN A REDES DE PETRI

Definición 1: Una red de Petri (rdP) es una estructura algebraica que se compone de un conjunto finito de *sitios* $P = \{p_1, p_2, \dots, p_n\}$, un conjunto finito de *transiciones* $T = \{t_1, t_2, \dots, t_m\}$, una *función de entrada* $I: P \times T \rightarrow \mathbb{N}$, con $\mathbb{N} = 1, 2, \dots$, que define el conjunto de arcos dirigidos de P a T y una *función de salida* $O: P \times T \rightarrow \mathbb{N}$ que define el conjunto de arcos dirigidos de T a P . Las funciones I y O forman dos matrices no negativas. La matriz $C: O - I$ se conoce como *matriz de incidencia*.

La estructura de la red de Petri $Z = (P, T, I, O, m)$ puede representarse gráficamente: cada *sitio* se denota por un círculo y cada *transición* se denota por una barra. Una red de Petri marcada $Z = (P, T, I, O, m)$, tiene además una marca, cuyo i -ésimo componente representa el número de fichas en el i -ésimo *sitio*. La marca inicial en la red se denota por m_0 .

En una rdP, los estados o las marcas cambian de acuerdo con las siguientes reglas de habilitación/activación [2].

Definición 2: Una transición $t \in T$ está habilitada si

$$m(p) \geq I(p,t) \quad \forall p \in P \quad (1)$$

Definición 3: una transición t habilitada, se activa y resulta en la nueva marca

$$m'(p) = m(p) - I(p,t) + O(p,t) \quad \forall p \in P \quad (2)$$

1) *Aspectos de modelado:* Las redes de Petri modelan el comportamiento dinámico de un sistema de eventos discretos, el conjunto de fichas y marcas pueden cambiar durante la ejecución de la red, describiendo las características dinámicas del sistema modelado.

III. PLANTEAMIENTO DEL PROCESO

A. Descripción del FMS

El FMS a modelar se muestra en la figura 1, en donde se identifican los componentes de interés para este trabajo: (1) almacén hexagonal giratorio, (2) centro de maquinado vertical (VMC, del acrónimo en inglés) de CNC y (3) robot transportador gantry, los cuales son controlados mediante los PLC's Siemens S7-200 en forma local y S7-300 en función de enlace maestro.

Principales características del almacén hexagonal giratorio:

- 42 bahías distribuidas en 7 niveles y 6 columnas, cada una con sensores micro-interruptores mecánicos
- 6 sensores inductivos de posicionamiento
- Almacenamiento temporal configurable de charolas para pallets (transportadores)
- Inversión de giro de motor de CD a 24v

Principales características del centro de maquinado vertical VMC-CNC:

- Posee 6 herramientas de corte en torreta
- Movimiento en 3 ejes
- Programación G&M y Diseño Asistido por Computadora

Principales características del robot transportador gantry:

- 6 grados de libertad (incluyendo el riel horizontal)
- Carga de 1.0 kg y alcance de 660 mm. sin mordazas
- Repetibilidad +/- 0.005 mm



Figura 1. Fotografía del FMS a modelar

B. Proceso General

El almacén hexagonal girará libremente hasta encontrar la materia prima en la columna y bahías adecuadas (previamente elegidas como almacenamiento de materia prima, columna 1 y bahías 5, 4 ó 3), por lo que al localizarlas, activa el robot transportador que se dirige a éstas con la prioridad establecida (de mayor a menor). Al recoger el material lo lleva a la máquina VMC-CNC. Terminado el proceso de manufactura del VMC, el robot transportador recoge el producto terminado de la máquina y lo regresa al almacén hexagonal en la columna de producto terminado (a la cual se le asigno la columna 4 y las mismas bahías). Después de 5 vueltas sin detectar materia prima en columna 1, el giro del almacén se detiene y espera reinicio.

IV. MODELADO Y ANÁLISIS

A. Modelado de FMS utilizando rdP

A continuación se enlistan los criterios utilizados para diseñar la rdP.

- El sistema que se modela es un sistema dinámico de eventos discretos.
- Las transiciones implican eventos (sensores) I y temporizaciones, los sitios implican acción (actuador) Q.
- Todo sitio debe tener sus dos estados, con uno de ellos activos y nunca ambos a la vez. Las transiciones se activarán en el momento adecuado marcado en la red.
- Colocar una marca en el estado inicial, éste puede ser "0" o "1".
- Basándose en redes de Petri, no se podrán conectar directamente 2 sitios o 2 transiciones.
- Analizar discretamente el proceso a modelar y efectuar el avance de eventos con posibilidad a retornar para establecer algunas futuras restricciones.

B. Modelado Almacén Hexagonal

El planteamiento de esta sección consiste en lograr el modelado y posterior análisis del almacén hexagonal. Apoyándose en los criterios que se menciona, se procede al diseño de la rdP como se muestra en la figura 2, la cual representa gráficamente la red del proceso principal del almacén hexagonal, donde se utilizan subrutinas, lo cual le brinda mayor claridad visual para su posterior análisis en el proceso de prioridad en bahías. El proceso completo obedece a la siguiente lógica:

- Las condiciones iniciales indican la activación de los sitios $Q2.0=1$ (giro del almacén), Flag, Inicio, $Input16=0$, así como el sitio No. Vueltas que representa un almacenamiento con 5 fichas que realiza conteo.
- Comienza con el almacén hexagonal girando en búsqueda de la columna 1 y materia prima en las

subrutina Sub MP Col1. Igualmente se observa que la marca 5 es la única que presenta activo el sitio $input8=1$ (indicando prioridad en bahía5), análogamente la marca 7 para la $input9=1$ y la marca 9 para la $input10=1$.

sitio/ Marca	Sub mp coll	bahía 5	bahía 4	bahía 3	input 8=0	input 8=1	input 9=0	input 9=1	input 10=0	input 10=1
m3	1	0	0	0	1	0	1	0	1	0
m4	0	1	0	0	1	0	1	0	1	0
m5	0	0	0	0	0	1	1	0	1	0
m6	0	0	1	0	1	0	1	0	1	0
m7	0	0	0	0	1	0	0	1	1	0
m8	0	0	0	1	1	0	1	0	1	0
m9	0	0	0	0	1	0	1	0	0	1

Tabla 3. Marca/sitio subrutina tomar materia prima.

La restante red del almacén hexagonal y subrutinas obedecen a un modelado y análisis que cumple con los criterios de creación establecidos.

D. Modelado de robot transportador gantry

El robot transportador espera la señal proporcionada por la $input\ 16=1$, proveniente del modelo del almacén hexagonal. Sucedido esto, el robot transportador se coloca en la posición listo, y llama a la subrutina SUB-VMC (ver figura 4). Se detalla el proceso.

- Si cualquiera de las entradas 8, 9 ó 10 (figura 2) está activa, el robot transportador realizará la secuencia de puntos correspondiente a dicha entrada y con esto recogerá la materia prima, la cual dependerá de la prioridad que se modeló (figura 3), cabe mencionar que de esta forma sólo puede estar una de las entradas activas.
- Se realiza una secuencia de puntos para que el robot transportador lleve la materia prima hacia la plataforma intermedia (se encuentra entre las dos máquinas de CNC), donde deja la charola para manipular el material e introducirlo en el VMC.
- La siguiente secuencia es la coordinación entre la máquina de CNC y el robot gantry, esta involucra activaciones, tiempos de espera y desactivación de puertas, mordazas y maquinado (previamente programado).
- Terminado el maquinado en el VMC, el robot transportador recoge el producto terminado, y regresa a la plataforma intermedia, colocando el producto terminado en la charola y llevándolo al almacén hexagonal, para colocarlo en la columna 4 de producto terminado. El orden lógico involucra las secuencias de puntos y la espera de las señales provenientes del PLC.

V. METODOLOGIA DE TRADUCCION

A. Diagramas espacio-fase

Estos diagramas indican gráficamente el comportamiento que sigue la rDP y son un paso necesario para realizar la implementación física de las secuencias de disparo. Su metodología es la siguiente.

- Analizar o simular la red de Petri paso a paso (lo cual automáticamente se llevará de acuerdo a la secuencia lógica que se le dio al crearla).
- Realizar los diagramas espacio-fase conforme marca la secuencia.
- Considerar todas las condiciones iniciales que se marcan en el modelo.
- Trazar las gráficas parciales conforme al orden de aparición, tanto sensores como actuadores.
- Las gráficas son parciales hasta no completar toda la simulación o análisis completo.

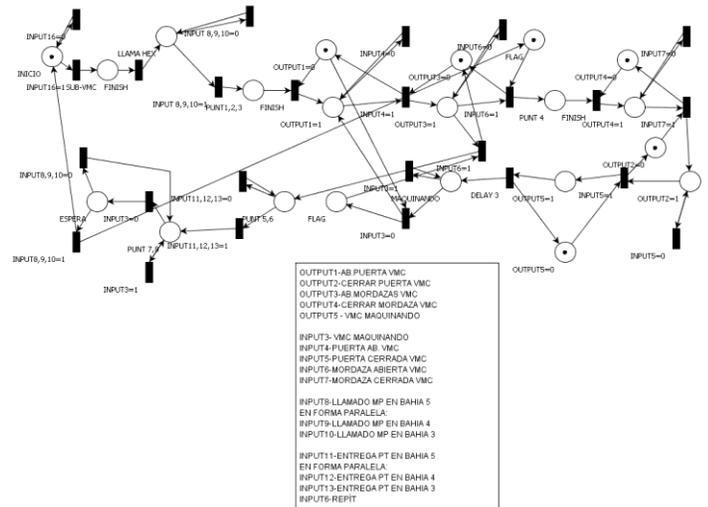


Figura 4. Modelo del robot transportador

- Las gráficas deben de apilarse con el mismo origen (para lograr su asociación).
- Identificar la secuencia de activación.
- Marcar externamente si el inicio o culminación de un evento o transición repercute en alguna acción de los actuadores o sensores (inclusive ya analizados anteriormente).
- Completar todo el análisis obteniendo los diagramas finales y trazarlos en la misma escala de tiempo.
- Analizar detenidamente la relación de eventos marcados anteriormente en los diagramas espacio-fase, para su fácil interpretación a otro lenguaje de programación.

El diagrama espacio-fase del almacén hexagonal se muestra en el lado derecho de la figura 5.

B. Programación KOP (Diagramas de Escalera)

Los diagramas espacio-fase son convertidos a programación KOP (diagramas de escalera) para que puedan ser implementados en los PLC's. Para ello, se tienen las siguientes consideraciones.

5.- La detección de la columna 4 activa la señal Q3.7 comunicada con el robot transportador y en conjunto con la marca M1.0 detiene el giro del almacén hexagonal, para la entrega del producto terminado, que al igual que en el anterior proceso y de acuerdo con el modelado análogo de la subrutina, se tendrán prioridades para la entrega del producto terminado.

La lógica marcada para el producto terminado se invierte respecto a la de materia prima, dado que ahora interesa que las bahías estén libres.

Las salidas input 11, input 12, input 13, corresponden a la salida física del PLC, Q3.7, Q1.2, Q1.3 respectivamente. Adicionalmente se le agrega un contacto normalmente cerrado asociado al giro del motor a cada segmento de salidas 11,12 y 13; es decir, estas señales no pueden ser activadas si es que el motor se encuentra girando, lo que evita un posible error al poder activarse sólo cuando detecta la columna.

6.- El robot transportador entrega el producto terminado y es necesario volver a activar el proceso para hacerlo cíclico, para esto se utilizará un temporizador T40 tipo ON-delay activado por la entradas 11,12 o 13, lo cual permitirá el libre giro del almacén.

Se aprovechan las propiedades de conteo inicial del T40 logrando no reiniciar el sistema inmediatamente, sino que le proporciona un tiempo a que el robot transportador entregue el producto terminado, y segundos después (8 segundos) reiniciará.

Con un contacto normalmente cerrado asociado al T40 se desenchava la marca M0.0 (figura 9) y por consiguiente la marca M1.0 activando nuevamente el giro del almacén y quedando en espera del reinicio del proceso.

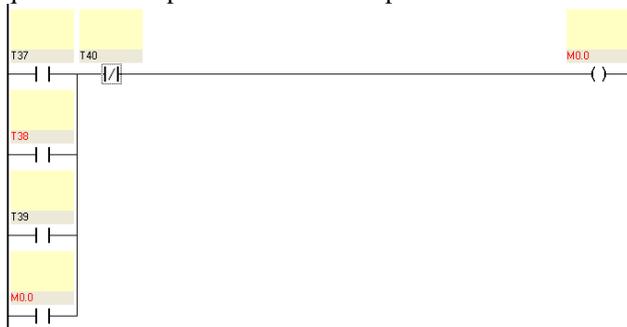


Figura 9. Diagrama con temporizador para búsqueda de materia prima.

Finalmente se contempla el contador (C1) como detector de 5 vueltas sin materia prima, deteniendo el giro del almacén (figura 8) y esperando señal de reinicio con la marca M0.2 en Cont1 (figura 5).

C. Programación para robot transportador gantry

El enfoque se dirige hacia la programación estructurada que requiere el robot transportador para lograr la coordinación adecuada con el PLC y la máquina de CNC.

Esta programación utiliza la lógica básica para la manipulación de los robots. La principal característica es la posibilidad de activar y desactivar entradas y salidas, que

están conectadas al GPIO del controlador, y se comunican directamente con su PLC asociado, así como el poder de marcar rutinas de puntos en el espacio. La codificación a este lenguaje se realiza “directamente” interpretando la red de Petri, asociando comandos y sentencias de programación.

VI. CONCLUSIONES

Se ha propuesto una metodología para modelar y controlar una celda automatizada de manufactura flexible; involucrando el modelado de diferentes tecnologías y analizando la evolución del sistema bajo un esquema de control, logrando así realizar la conversión del modelo obtenido a la programación de hardware de bajo nivel. Aunque no se muestra aquí, los resultados experimentales obtenidos verifican la integración y configuración de los distintos elementos del sistema bajo la propuesta descrita en este trabajo. Se considera que esta puede extrapolarse hacia sistemas de manufactura más complejos.

REFERENCIAS

- Bulach, S., A. Brauchle, H.J. Pfliederer, and Z. Kucerovsky. (2002). Design and implementation of discrete event control systems: a Petri net based hardware approach. *Discrete Event Dynamic Systems: theory and applications*, 12(3):287-309.
- Cassandras, C.G. and S. Lafortune (1999). *Introduction to discrete event systems*. Kluwer academics.
- Chandra, V., Z. Huang and R. Kumar (2003). Automated control synthesis for an assembly line using discrete event systems control theory. *IEEE transactions on systems, man, and cybernetics. Part c: applications and reviews*, 33(2):284-289.
- Chandra, V. and R. Kumar (2001). A discrete event systems modeling formalism base on event occurrence rules and precedences. *IEEE transactions on robotics and automation*, 17(6):785-794.
- Desrochers, A. and A. Jaar (1995). *Applications of Petri Nets in Manufacturing Systems* IEEE Computer Society Press.
- Kalpakjian, S. y S.R. Schmid (2008). *Manufactura, ingeniería y tecnología*. Prentice Hall. México.
- Murata, T. (1989). Petri nets – Properties, analysis and applications. *Proc. of the IEEE*, 7(4):541-580.
- Piedrafita, R. Moreno. (2004). *Ingeniería de la automatización industrial*. Ra-Ma Alfaomega, México.
- Sánchez, B.F.J., R. I. Rivera, R. A. Ramírez y M. E. López (2004). Interpreted petri net based design of a flexible controller for a manufacturing cell. In *proceedings of IEEE International Symposium on Robotics and Automation*. Querétaro, Qro.
- Wang, J.(1998). *Timed Petri nets: theory and applications*. Kluwer Academic Publishers, USA.
- Zhou, M. and K. Venkatesh (1999). *Modeling, simulation, and control of flexible manufacturing systems: a Petri net approach*. World Scientific Publishing.