Ingeniería Investigación y Tecnología
volumen XX (número 4), octubre-diciembre 2019 1-12
ISSN 2594-0732 FI-UNAM artículo arbitrado
Información del artículo: Recibido: 12 de junio de 2018, aceptado: 19 de agosto de 2019
Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license
http://dx.doi.org/10.22201/fi.25940732e.2019.20n4.041

Ingeniería
Investigación y
Tecnología

Automatización multinivel de celda de manufactura robotizada utilizando máquinas de estados finitos

Multilevel automation of a manufacturing robotic cell using finite state machines

Sánchez-Chávez Irma Yolanda
Tecnológico de Monterrey
Campus Aguascalientes, México
Correo: isanchez@itesm.mx
https://orcid.org/0000-0002-6710-9049

Martell-Chávez Fernando
Centro de Investigaciones en Óptica, A.C.
Unidad Aguascalientes, México
Correo: fmartell@cio.mx
https://orcid.org/0000-0002-2741-0689

Resumen

Se presenta un caso práctico de automatización de una celda de manufactura robotizada, el cual se realiza mediante máquinas de estados que resuelven la secuencia de operación de cada unidad de la celda, así como la integración de todas las unidades. En muchos casos, la automatización se realiza sin una metodología para plantear la lógica de control y mucho menos para programarla. Ante esto, se muestra la máquina de estados como una estrategia de diseño y programación estructurada aplicada a controladores lógicos programables. Se demuestra la generación de código a partir de las máquinas de estado planteadas. La automatización multinivel logra la ejecución en modo automático del funcionamiento total de la celda, y, además, integra la operación manual para permitir la manipulación directa de los actuadores, así como un modo de operación semiautomático para probar el procesamiento de las unidades individualmente y dar flexibilidad ante posibles interrupciones del ciclo de trabajo de la celda. El diseño resultante cumple con las especificaciones funcionales y facilita la comprensión, mantenimiento y modificación del sistema automatizado. Este trabajo contribuye con un caso de automatización utilizando herramientas metodológicas, útil para la capacitación en esta área y para el desarrollo de competencias en el contexto de la industria 4.0.
Descriptores: Máquinas de estados finitos, control supervisorio, generación de código para PLC.

Abstract

A practical case of automation of a manufacturing robotic cell is presented, which is solved by finite state machines that implement the operation sequence of every unit of the cell, as well the integration of all the units. In many cases, no methodology is used in automation for the design of control logic and even less for programming. For this reason the finite state machine is shown as a strategy for the design and structured programming applied to programmable logic controllers. Code generation is demonstrated from the proposed finite state machines. Multilevel automation achieves the automated execution of the total functioning of the cell, and also integrates the manual operation mode to allow the direct manipulation of the actuators, as well as a semiautomated operation mode to test or execute the processing of the units individually, and give flexibility before possible interruptions of the cell work cycle. The resultant design satisfies the functional specifications and facilitates the comprehension, maintenance and modification of the automated system. This work contributes with an automation case using methodologic tools, useful for the capacitation in this area, and for the development of competences in the context of industry 4.0.
Keywords: Finite state machines, supervisory control, PLC code generation.

Introducción

México en la industria de la manufactura ocupa un alto lugar en competitividad, el octavo de acuerdo con el Global Manufacturing Competitiveness Index (GMCI) de 2016 (Deloitte Touche Tohmatsu Limited y US Council on Competitiveness, 2016), y se prevé que mejore en los próximos años. De acuerdo con el estudio mencionado, la manufactura contribuye al PIB total de México en forma similar a la de otros países competidores (en 2013 con 17.6 %, un punto porcentual por arriba del promedio reportado por GMCI, 2016), genera más exportaciones que el promedio (77.7 % de exportaciones totales en 2014; 17 puntos arriba del promedio) y tiene costos laborales altamente ventajosos (6.2 dólares por hora en 2015 contra un promedio de 18.7 dólares por hora). Dentro de la gran variedad de factores que determinan la posición competitiva de México en el ramo de la manufactura, el factor tecnológico es básico y estratégico. En la industria 4.0, la accesibilidad de la información es determinante (Martell, 2017), e implica grados más avanzados de automatización de diversas funciones. La automatización de los procesos de transformación de materia y energía es esencial, por lo que realizarla localmente en forma sistemática es muy importante para la eficiencia y productividad. En la medida en que se utilicen metodologías estructuradas para el diseño de sistemas automáticos se estará impactando en la eficiencia de este tipo de industria, lo cual es uno de los objetivos de la industria 4.0.

Las operaciones básicas en la manufactura incluyen operaciones de procesamiento y ensamble, manejo de materiales, inspección y pruebas, así como coordinación y control. En este trabajo se aborda el problema del manejo de materiales para realizar una operación sencilla de empacado, con el objetivo de solucionar su automatización de manera formal o estructurada, permitiendo que pueda leerse y modificarse fácilmente, de acuerdo con nuevas necesidades, en lugar de ser reelaborada.

Niveles de automatización

El concepto de automatización puede aplicarse en distintos niveles que implementan agrupamientos progresivos (Groover, 2015). En el nivel uno o de dispositivos se operan sensores y actuadores en lazos de control. En el nivel dos o nivel de máquina se integra el funcionamiento de diversos dispositivos. En el nivel tres o nivel de sistema o celda, se define y opera una celda de manufactura como un grupo de máquinas o estaciones de trabajo interconectadas por un sistema de manejo de materiales; este nivel incluye líneas de producción y ensamble. El cuarto nivel o nivel de planta produce planes operacionales tales como planes de procesamiento, control de inventarios, compras, control de calidad, etcétera. El quinto nivel es el nivel de empresa; maneja la información corporativa relativa a mercadeo, ventas, contabilidad, investigación y diseño, calendarización de producción, etcétera. La automatización propuesta en este trabajo corresponde al nivel tres, dado que deben coordinarse máquinas o unidades entre las cuales deben trasladarse piezas con el objetivo de producir piezas terminadas en forma de piezas empacadas. La coordinación de las unidades corresponde a un nivel supervisorio, en tanto que el control de cada unidad concierne al nivel de máquina, y la operación de cada actuador corresponde al nivel de dispositivo.

Máquina de estados

Entre las herramientas desarrolladas para el análisis de sistemas discretos se encuentran la automatización temporizada, álgebra de proceso, redes de Petri y máquinas de estado (Popescu y Martinez, 2010). La automatización temporizada se basa esencialmente en autómatas individuales para cada dispositivo con transiciones producidas por temporizadores y canales de sincronización; demandan un gran espacio de memoria y tiempo computacional. El álgebra de procesos describe un sistema mediante expresiones algebraicas o axiomáticas. Existen diferentes tipos de álgebra de procesos (para cálculo de sistemas de comunicación y de procesos secuenciales, por ejemplo), y difieren en la definición de sus operadores. El formalismo del álgebra de procesos no maneja acciones en paralelo, sino que permite un orden arbitrario a eventos que deberían ocurrir simultáneamente. Las redes de Petri permiten la representación gráfica de un sistema secuencial utilizando lugares, transiciones, arcos y tokens (marcas en cada lugar); los tokens regulan el flujo de la secuencia, es decir, activan las transiciones, y estas se disparan realizando una tarea, eliminando los tokens de entrada y produciendo tokens en cada lugar de salida, permitiendo la simultaneidad de acciones. La definición de los tokens tiene una representación matemática. La máquina de estados finitos surge como un modelo computacional para el diseño de lenguajes de programación (Mealy, 1955; Moore, 1956). Las máquinas de estado son un tipo particular de redes de Petri, que se diferencian por una simplificación de la representación de la secuencia utilizando estructuras lineales, prescindiendo del uso de tokens, reduciendo los elementos gráficos a círculos para estados a los que se asocian acciones, y

flechas para transiciones asociadas a consecuencias de acciones del estado previo y al mismo tiempo requisitos para el siguiente estado. La concurrencia de eventos se puede resolver con la combinación de máquinas de estados.

El análisis y diseño de un sistema de eventos discretos puede realizarse en forma estructurada utilizando máquinas de estado. Típicamente, en el ambiente industrial no se aplica esta herramienta (salvo por algunos fabricantes de equipo original, OEM, por sus siglas en inglés), y los programas se elaboran sin ninguna metodología formal que facilite su revisión, modificación o transferencia. Es decir, el software desarrollado frecuentemente resulta inflexible y poco comprensible para otros programadores, por lo que el mantenimiento de los sistemas frecuentemente requiere su reelaboración.

La máquina de estados describe paso a paso, la secuencia o el proceso a implementar, incluyendo las acciones a realizar, así como las condiciones que permiten determinar que el paso ha terminado y se puede proseguir con el siguiente (transiciones entre etapas o estados). Puede utilizarse una representación gráfica o tabular de la máquina de estados. A partir de dicha representación se deriva directamente la implementación del sistema: ecuaciones lógicas para programación de estados y de salidas en función de los estados activos. De esta manera, la máquina de estados asiste tanto el diseño como la programación del sistema de control secuencial.

La desventaja de la máquina de estados puede ser el gran número de estados que deban definirse para un proceso muy largo. Diversos trabajos han propuesto métodos para simplificar o reducir la máquina de estados (Possan, 2012). La estrategia de este trabajo es la modularidad: se propone una máquina de estados para cada subsistema, y se diseña la coordinación de las máquinas individuales mediante otra máquina de estado de nivel superior. De esta manera, pueden crearse varios niveles. En el nivel más alto se logra la coordinación y supervisión de todos los subsistemas. La ventaja de esta estrategia es la posibilidad de mantener máquinas de estados relativamente pequeñas que faciliten la visualización del proceso.

Programación de PLC

Los PLC son una plataforma de automatización ampliamente utilizada, sin embargo, su programación no solo está sujeta al dominio de cierto lenguaje de programación, sino también de prácticas y estilos de cada programador. El estándar internacional IEC 61131-3 (IEC, International Electrotechnical Commission, 2013) busca establecer ciertos lineamientos en la programación en lista de instrucciones, texto estructurado, diagramas de bloques de función, diagramas escalera y gráficos de funciones secuenciales (SFC). El desarrollo de este estándar es una muestra del énfasis en la programación, más no en la lógica de control (Music y Matko, 2002). Incluso se han desarrollado varias técnicas para verificar el funcionamiento y corrección de la lógica de control (Frey y Litz, 2000), más no metodologías para el diseño de la lógica de control.

El uso de máquinas de estados provee una herramienta de análisis del problema y de síntesis de la lógica de control para su solución, y conduce a una programación directa y estructurada al mismo tiempo. El concepto de control supervisorio, tradicionalmente se ha asociado al control de eventos discretos mediante máquinas de estados bajo la intención de dar una robustez a los sistemas de control para mantener un comportamiento o desempeño aceptable (Ramadge y Wonham, 1987). Bajo este enfoque de supervisión y con una metodología de diseño y codificación correcta de la lógica, se reporta un caso simulado de un manipulador en un sistema de producción (Music y Matko, 2002). A diferencia de este antecedente, en este trabajo las máquinas de estado se utilizan para automatizar dispositivos individuales, así como máquinas completas y para el control supervisorio desde el punto de vista de coordinación de máquinas. La aplicación que aquí se detalla demuestra el potencial de las máquinas de estado para generación de lógica de control, así como de código. Además, se estructura no solo la programación de una secuencia automática, sino también la lógica para la operación manual, necesaria para intervenir en situaciones de emergencia, de preparación o revisión del sistema.

Desarrollo

Descripción de la celda

La celda de manufactura consiste en un almacén que se utiliza para alojar tanto piezas por procesar como piezas procesadas, un robot cartesiano que permite alcanzar las distintas localidades del almacén, un sistema de transporte que permite trasladar las piezas hacia el robot y de regreso hacia el almacén, y un manipulador robótico articulado que realiza un procesamiento (Figura 1).

El almacén es una estructura que consta de dos columnas con distintos niveles, en los cuales puede colocarse una pieza sobre un pallet que permite sujetarla para facilitar su traslado. El subsistema del almacén tam-

bién consta de un robot cartesiano que puede moverse horizontalmente de una columna a otra, verticalmente hacia los distintos niveles, meter y sacar la pieza de su localidad, y finalmente, girar 180 grados entre el almacén y la banda. El robot cartesiano interactúa con la banda en solo una posición (carril 1 a nivel de banda) tanto para colocar una pieza como para tomarla. La pieza sobre la banda y en el almacén siempre se encuentra fija sobre un pallet. Los actuadores lineales, así como el actuador giratorio del robot cartesiano cuentan con sensores de final de carrera, y están conectados a un PLC.

La banda transportadora tiene una configuración circular. Los dos tramos lineales se designan como carril 1, por el que se transporta la pieza del almacén hacia el robot del lado opuesto, y carril 2, por donde se regresa la pieza hacia el lado del almacén. Hay dos cilindros lineales de simple efecto que empujan la pieza del carril 1 al 2 del lado del robot articulado, y del carril 2 al 1 del lado del almacén. Se cuenta con sensores de posición de pieza en ambos extremos de ambos carriles. Se tiene un PLC para la operación de la banda.

El manipulador robótico encargado del procesamiento de las piezas consta de 6 articulaciones. El robot articulado cuenta con su propio sistema de control (para controlar el movimiento y la posición de las 6 articulaciones) y es programado en un lenguaje especial de alto nivel. Como complemento de este subsistema se tiene una mesa de trabajo con dos sensores de posición. En el PLC de la mesa de trabajo está conectada una salida y una entrada digitales del robot y los sensores de posición de la mesa.

La celda debe automatizarse según la siguiente especificación funcional: el robot cartesiano toma una pieza del lado izquierdo del almacén de una altura predeterminada y la coloca sobre el carril 1 de la banda; la banda traslada la pieza frente al robot y se detiene; el robot toma la pieza (dejando el pallet sobre la banda), realiza una operación de empacado y regresa la pieza procesada o empacada a la banda (en el pallet); la pieza se empuja hacia el carril 2, la banda la traslada hasta el lado del almacén y se detiene, y el cilindro empuja la pieza hacia el carril 1; el robot cartesiano toma la pieza de la banda, la coloca en la columna derecha de la estructura en el nivel predeterminado, y regresa a la posición de inicio: carril 1, nivel de banda y girado hacia la banda. Adicionalmente a la operación automática descrita de la celda, debe contemplarse una operación en modo semiautomático en el que pueda ejecutarse por separado la secuencia de cada unidad o elemento de la celda, y un modo manual que permita operar cada actuador individualmente.

Solución de automatización

En el proceso de solución de esta automatización se utiliza la representación gráfica de la máquina de estados: cada etapa o estado se representa con un círculo numerado, a lado del círculo o etapa se especifica la acción correspondiente, la flecha representa el flujo hacia la

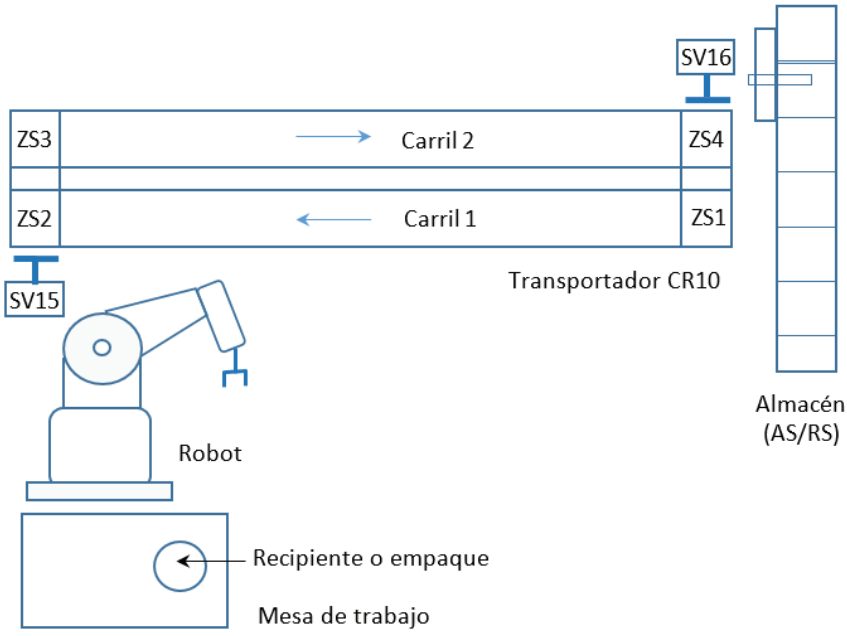

Figura 1. Celda de manufactura robotizada: almacén (AS/RS), transportador de dos carriles (CR10) con pistones para cambio de carril (SV15, SV16), robot, mesa de trabajo, sensores para detección de pieza (ZS1, ZS2, ZS3, ZS4) y de recipiente en posición sobre mesa de trabajo

siguiente etapa y se indica a un lado de esta la transición que lo permite. Una transición incluye condiciones que son consecuencia de la acción tomada en la etapa flujo arriba y requerimiento de la acción asociada a la etapa flujo abajo. El anexo presenta el listado de señales con sus símbolos para cada unidad (Tablas 1, 2 y 3).

Sistema de banda transportadora

Colocación de pieza frente a robot

La Figura 2 representa la secuencia de transporte de la pieza hacia una posición al alcance del robot. Se verifica que no se haya solicitado ningún paro de emergencia (*SYS_OK*) y que se haya dado la orden de arranque (ya sea en modo automático o semiautomático). El motor de la banda se enciende para trasladar la pieza frente al robot. En cuanto se detecta la pieza frente al robot, la banda se detiene y se finaliza la secuencia. La señal Arranque1Banda se produce modo automático, cuando se detecta pieza colocada en carril 1 (*ZS1*), o bien, cuando no está activo el modo automático y se ha dado el arranque de modo semiautomático (*Arrsnque*1*Banda* = *Arranque Auto*1*Banda* & *ZS*1 + *ArranqueSemiauto*1*Banda* & *NOT*(*MODO_AUTO*)).

Retirar pieza frente al robot de regreso al almacén

La Figura 3 detalla la secuencia de transporte de la pieza empacada hacia el almacén. Se verifica el estatus de listo (*SYS_OK*), es decir, de ausencia de paros de emergencia, y la señal de arranque (en modo automático o semiautomático) para activar el cilindro del lado del robot que empuja la pieza hacia el carril 2, se activa el motor de la banda para llevar la pieza hacia el lado del almacén, cuando se detecta la pieza en el lado de almacén y carril 2 se detiene el motor de la banda, y se empuja la pieza hacia el carril 1 alistando la pieza para ser recogida.

La señal *Arranque2Banda* se produce en modo automático cuando se detecta pieza en carril 1 del lado del robot (*ZS2*), o bien, cuando no está activo el modo automático y se ha dado el arranque en modo semiautomático (*Arranque*2*Banda* = *ArranqueAuto*2*Banda* & *ZS*2 + *ArranueSemiAuto*2*Banda* & *NOT*(*MODO_AUTO*)).

Sistema de robot

Cuando se recibe señal de arranque (automático o, semiautomático), el robot se mueve de su posición de inicio hacia la pieza que ha llegado al carril 1 del lado del robot, toma la pieza, la traslada a la mesa de trabajo y la empaca, toma pieza empacada, la coloca en la banda, regresa a su posición inicial y se envía señal de terminación de tarea del manipulador. La Figura 4 contiene la máquina de estados finitos que controla la ejecución de la operación de empacado.

La señal *ArranqueRobot* se produce de modo automático, cuando se detecta pieza en la banda junto a la ubicación del robot (*ZS2*) y se verifica sensor de posición en mesa de trabajo (*di1*) activado por recipiente o empaque, o bien, en modo semiautomático siempre y cuando no esté activo el modo automático (*ArranqueRobot* = *ArranqueAutoRobot* & *ZS*2 & *di*1 + *ArranqueSemiAuto*Robot & *NOT*(*MODO_AUTO*)).

El programa a ejecutar en el estado 1 de la máquina de estados se ejemplifica con un lenguaje de alto nivel para un robot comercial articulado de 6 grados de liber-

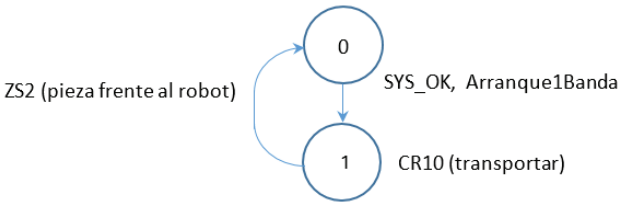

Figura 2. Máquina de estados finitos para transporte de pieza del almacén hacia el robot

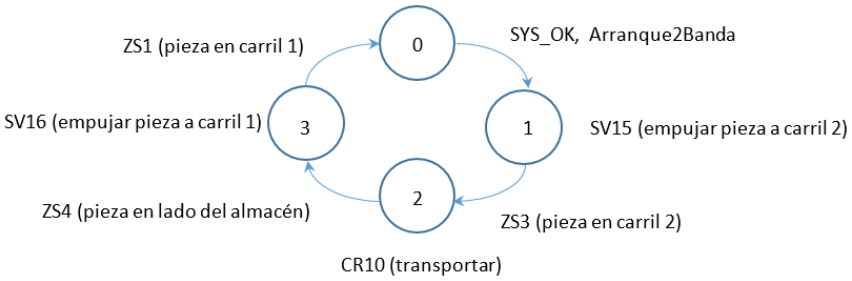

Figura 3. Máquina de estados finitos para transporte de pieza del lado del robot hacia el almacén

http://dx.doi.org/10.22201/fi.25940732e.2019.20n4.041

tad en la Figura 5. *MoveJ* mueve las articulaciones del robot hacia una posición objetivo del efector final con determinada velocidad (*v####*), tolerancia (*z####*) y herramienta (*tool#*). *MoveL* permite mover el robot siguiendo una trayectoria lineal hacia la posición objetivo, generalmente con una velocidad y tolerancia menores para interactuar adecuadamente con el objeto a manipular. *WaitTime* permite introducir un tiempo de espera para asegurar la colocación estable del objeto manipulado al dejarlo, o la correcta sujeción al tomarlo. *SetDO* permite asignar un valor a la variable indicada. La señal *di5* es la señal *ArranqueRobot* en la transición de la etapa 0 a la 1, o señal proveniente del PLC hacia el controlador del robot para iniciar su rutina. La señal *do5* es la señal de terminación de tarea del manipulador que envía el controlador del robot al PLC, o transición de la etapa 1 a la 0. La señal *do1* es la señal de salida del controlador del robot hacia el efector final o pinza.

Sistema de almacén

Toma de pieza del almacén

La Figura 6 muestra la máquina de estados finitos que implementa la secuencia para tomar una pieza del almacén. Una vez recibida la señal de arranque (automático o semiautomático) etiquetada como *Arranque1Alm* (*Arranque1Alm = ArranqueAuto1Alm + ArranqueSemiAuto1Alm & NOT(MODO_AUTO)*), se retrae el cilindro en la dirección *Y* por seguridad, se gira el robot o elevador hacia el almacén, se mueve hacia el lado izquierdo (dirección *X*), y luego se mueve verticalmente (dirección *Z*) hasta la posición deseada (*LOC_PZA*). Se abre la pinza, se extiende en la dirección *Y* para entrar y alcanzar el pallet, se cierra la pinza, se retrae en *Y* para extraer la pieza de la estructura del almacén, se mueve el elevador hacia la derecha, y se coloca verticalmente a la altura de la banda (*BANDA*). Se gira hacia la banda, se

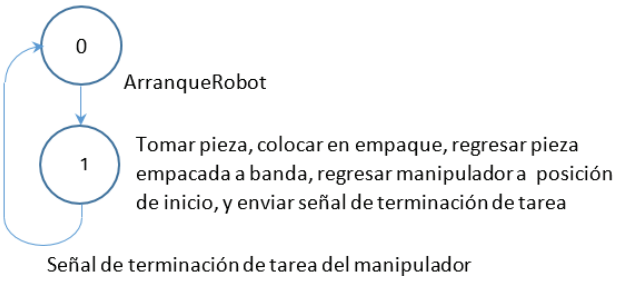

Figura 4. Máquina de estados finitos para ejecución de tarea del robot

```
PROC main()
        MoveJ (inicio, v1000, z100, tool1)
        SetDO do5, 0  ! desactivar señal interlock hacia el PLC o señal de terminación de rutina del robot
        SetDO do1, 1 ! abrir pinza
        IF (di5=1)
                MoveJ(cerca_pza_banda, v1000, z100, tool1)
                MoveL(pza_banda, v100, fine, tool1)
                SetDO do1, 0 !cerrar pinza
                WaitTime 0.5
                MoveL(cerca_pza_banda, v100, fine, tool1)
                MoveJ(pos_intermedia, v1000, z100, tool1) !ir a posición intermedia para trayectoria segura
                MoveJ(cerca_caja, v1000, z100, tool1) !alcanzar caja en mesa de trabajo
                MoveL(caja_dentro, v100, fine, tool1) !introducir pieza dentro de la caja
                SetDO do1, 1 !abrir pinza
                WaitTime 0.5
                MoveL(cerca_caja, v100, fine, tool1)
                MoveL(caja_pared, v100, fine, tool1) !alcanzar posición para sujetar pared de la caja
                SetDO do1, 0 !cerrar pinza para tomar caja con pieza adentro
                MoveL(cerca_caja, v100, fine, tool1)
                MoveJ(pos_intermedia, v1000, z100, tool1)
                MoveJ(cerca_caja_banda, v1000, z100, tool1) !aproximarse a la banda
                MoveL(caja_banda, v100, fine, tool1) !alcanzar posición para dejar pieza en caja sobre banda
                SetDO do1, 1 !abrir pinza
                WaitTime 0.5
                MoveL(cerca_caja_banda, v100, fine, tool1) !alejarse de banda
                MoveJ (inicio, v1000, z100, tool1)
                SetDO(do5, 1)
        ENDIF
ENDPROC
```

Figura 5. Programa de tarea de robot en lenguaje de alto nivel

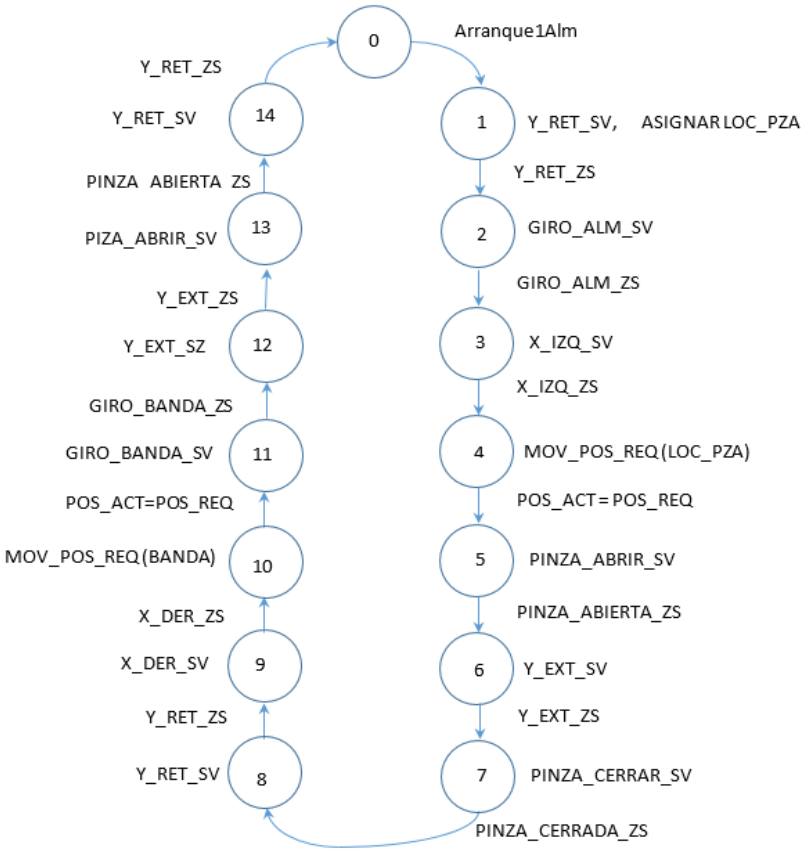

Figura 6. Máquina de estados finitos para toma de pieza del almacén

extiende para colocar pallet con pieza, se abre pinza, se retrae el cilindro volviendo así a la posición de inicio. Se decide mantener pinza abierta excepto cuando se carga el pallet que contiene la pieza.

Toma de pieza de la banda para almacenamiento

Para tomar la pieza empacada de la banda y colocarla en el almacén se ejecuta la secuencia representada en la Figura 7. Una vez recibida la señal de arranque (automático o semiautomático), se retrae el cilindro en la dirección *Y*, se mueve el robot cartesiano hacia el carril derecho (dirección *X*), se coloca en el nivel de banda (posición *BANDA* sobre dirección *Z*), se gira el robot o elevador hacia la banda. Se abre la pinza, se extiende en la dirección *Y* para tomar el pallet con pieza procesada, se cierra la pinza, se retrae en *Y*, se gira hacia el almacén, se mueve el elevador verticalmente a la altura deseada (*LOC_PZA*). Se extiende para colocar pallet con pieza dentro de la estructura, se abre pinza, se retrae el cilindro, se regresa a la altura de la banda y se gira hacia esta volviendo así a la posición de inicio.

La señal *Arranque2Alm* se produce en modo automático cuando se detecta pieza colocada en carril 1 (*ZS1*), o bien, cuando no está activo el modo automático y se ha dado el arranque en modo semiautomático (*Arranque2Alm = ArranqueAuto2Alm & ZS1 + ArranqueSemiAuto2Alm & NOT(MODO_AUTO)*).

Movimiento vertical del elevador del almacén

El movimiento vertical del robot cartesiano del almacén se realiza a velocidad normal hasta aproximarse a la posición requerida en un cierto margen o tolerancia, dentro de dicho margen, el movimiento deberá continuarse a baja velocidad hasta alcanzar la altura deseada, como se indica en la máquina de estados finitos de la Figura 8. Ante una sobrecarga del motor debe activarse la etapa 0 para detener el motor. Adicionalmente, debe considerarse que al arrancar el motor (en las etapas 1 o 3) debe mantenerse el freno por un tiempo corto (puede ser de 100 a 200 milisegundos), y luego liberarlo para que el motor continúe su operación y produzca el movimiento deseado, esto con el fin de asegurar una correcta energización de las bobinas del motor. Las posiciones posibles se definen según los niveles de la estructura del almacén y las respectivas cuentas de pulsos del codificador óptico (encoder).

http://dx.doi.org/10.22201/fi.25940732e.2019.20n4.041

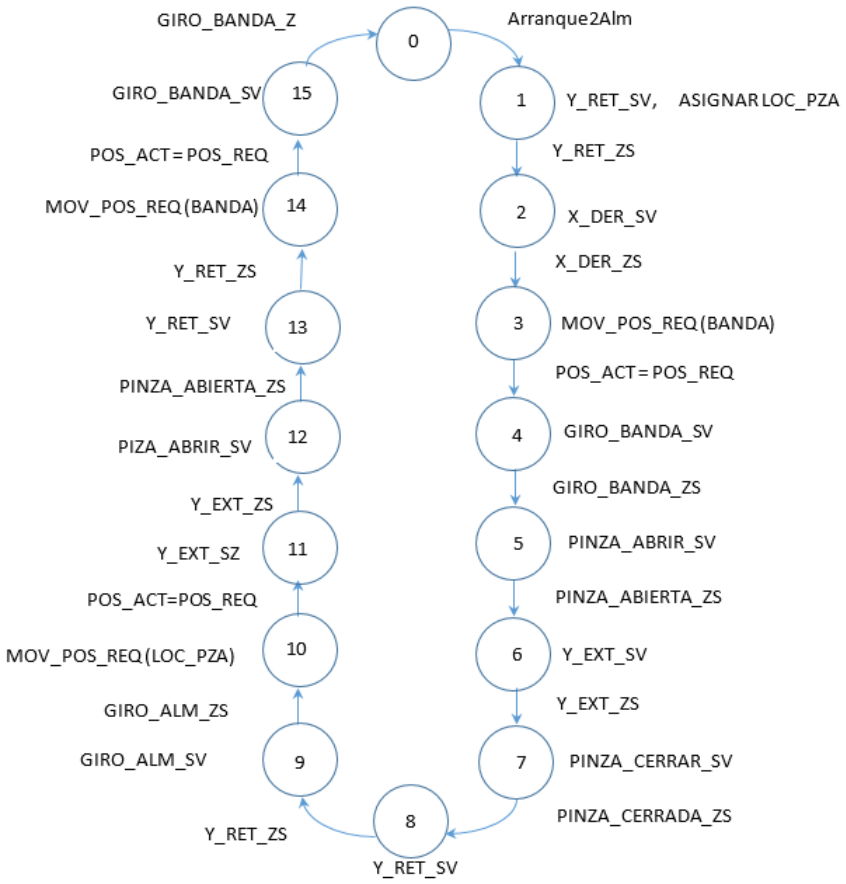

Figura 7. Máquina de estados finitos para toma de pieza de la banda para su almacenamiento

Integración de celda de manufactura

La coordinación de la operación de las unidades individuales de la celda constituye la automatización de un nivel supervisorio. La máquina de estados correspondiente a este nivel (Figura 9) activa en forma ordenada cada uno de los procesamientos de cada elemento de la celda y espera la señal de terminación correspondiente generada en cada uno de estos. La integración de la celda se realiza desde la interfaz humano-máquina desarrollada en una estación de trabajo computacional con comunicación hacia los PLC de las unidades vía Ethernet. Con la orden de inicio en modo automático, se genera la señal de arranque de la secuencia de toma de pieza del almacén, hasta que el PLC del almacén genera y envía la señal de terminación de esta tarea; entonces se envía el arranque a la banda transportadora para colocar la pieza frente el robot, y la máquina supervisoria espera la señal del PLC al completar la secuencia correspondiente. Luego se envía la señal de arranque a la estación de trabajo para que el robot articulado realice el empacado de la pieza; cuando el robot regresa la pieza a la banda y adquiere la posición de inicio lejos de la banda, genera la señal de finalización de tarea que permite proseguir con la máquina de estados supervisoria. Se activa la banda para regresar la pieza empacada a la posición donde será recogida, y se espera la señal de terminación de este proceso. Se activa el sistema de almacén para recoger y guardar la pieza procesada y se espera la señal de finalización para volver a iniciar la secuencia maestra o de integración.

El arranque del modo automático de la celda (*Arranque Automático*) se produce mediante un botón de inicio, y dicho modo de operación automático (*MODO_AUTO*), se termina con un botón de paro (ambos botones de la interfaz humano-máquina).

La señal de arranque de la máquina de estado de cada unidad en modo automático se produce por la máquina de estado supervisoria al detectar el flanco ascendente (o positivo, *P*) de la activación de cada una de las etapas de esta última. Por ejemplo, si el registro entero asignado a la máquina de estados supervisoria se etiqueta como *MS*, el arranque automático para tomar una pieza del almacén se produce según la lógica escalera de la Figura 10.

http://dx.doi.org/10.22201/fi.25940732e.2019.20n4.041

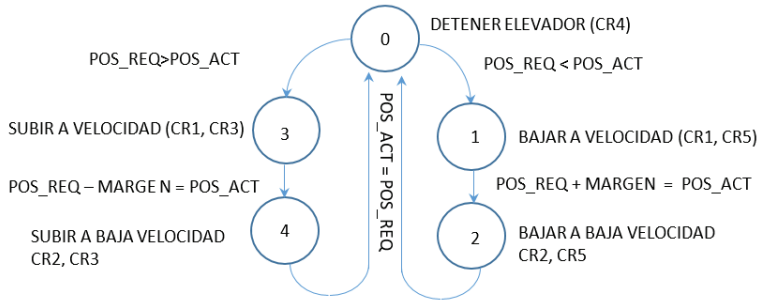

Figura 8. Máquina de estados finitos para movimiento vertical del elevador del almacén

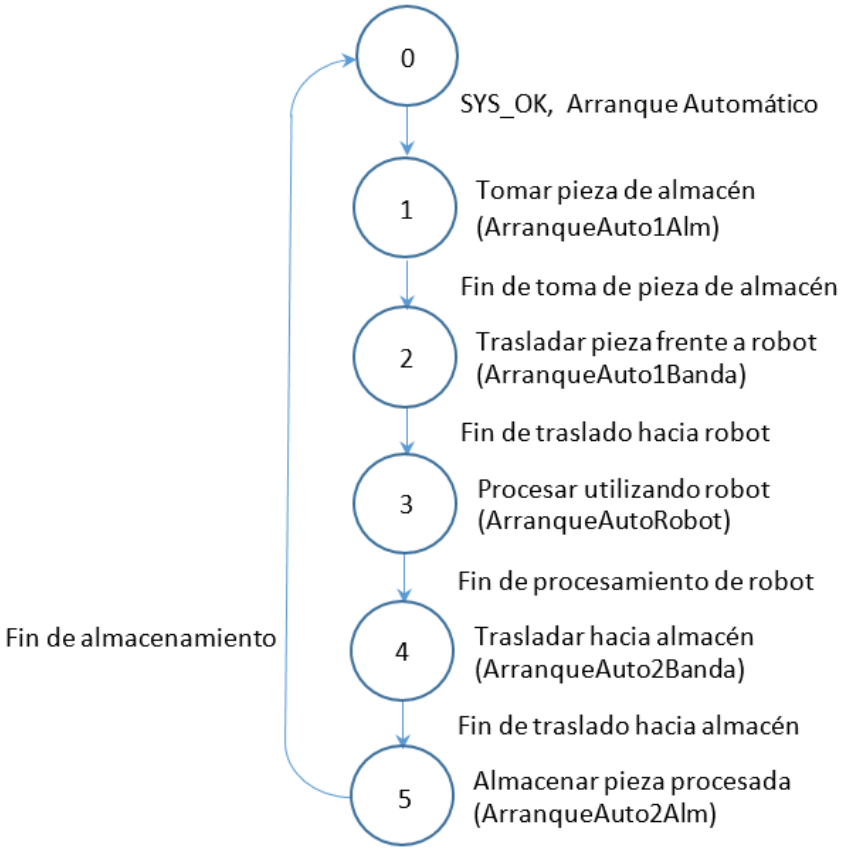

Figura 9. Máquina de estados finitos supervisoria

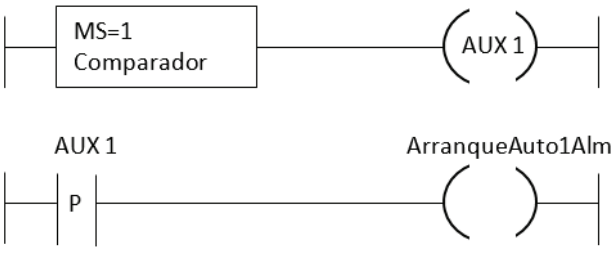

Figura 10. Señal de arranque generada por máquina de estados supervisoria

http://dx.doi.org/10.22201/fi.25940732e.2019.20n4.041

La señal de finalización de tarea de cada unidad de la celda se produce con la misma transición hacia el estado cero o inicio de la máquina correspondiente. El cambio de estado en la máquina supervisoria debe programarse antes que el cambio al estado cero de la máquina de una unidad. Por ejemplo, si la máquina de estados para tomar pieza del almacén se controla con un registro entero etiquetado como *ALM1*, al terminar esta tarea se incrementa el valor del registro *MS* de la máquina supervisoria y se regresa a cero el valor de *ALM1*, como se observa en la Figura 11.

Cuando no se está en modo automático, ni la unidad específica se está probando en modo semiautomático, entonces cada actuador puede operarse manualmente con los botones virtuales designados. Por ejemplo, para activar la solenoide que produce el giro del robot cartesiano hacia la banda mediante un botón virtual se tendría la programación mostrada en la Figura 12 (notar que el botón puede sustituirse por dos botones en un circuito *set-reset*).

Resultados

Se lograron dos resultados importantes del desarrollo mostrado de la automatización e integración de la celda de manufactura mediante máquinas de estados: código de programación y funcionamiento deseado. Para ejemplificar la programación estructurada que se deriva del planteamiento de la máquina de estado, se utiliza la máquina de estados para retirar la pieza empacada frente al robot de regreso al almacén (Figura 3). La Figura 13 muestra el programa en diagrama escalera. Primeramente se define una variable de tipo registro numérico (*R0*) para controlar el número de estado activo. El valor de la etapa actual y la satisfacción de las condiciones de transición de salida, es decir, la detección de terminación de ejecución de las acciones correspondientes a la etapa actual, permiten incrementar o avanzar el número de etapa. Finalmente, se activan las cargas al detectarse el número de etapa correspondiente, es decir, el valor específico del registro asociado al estado donde se toma cada acción. El código de PLC para el resto de las unidades y para su integración se obtiene de forma similar.

En cuanto al funcionamiento de la celda se lograron las especificaciones planteadas: activación de actuadores (modo manual), operación individual de cada unidad (modo semiautomático), e integración de unidades (modo automático). Un logro importante es el desarrollo estructurado de la automatización en los distintos niveles planteados, tanto para la programación de los PLC como para la programación de la HMI que instrumenta la coordinación de las unidades de la celda.

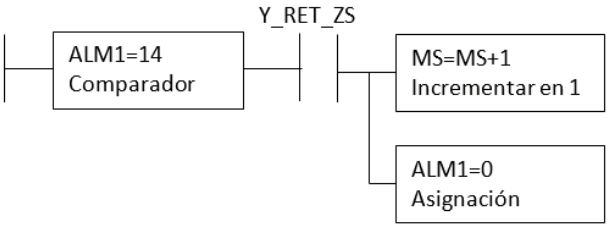

Figura 11. Señal de finalización de tarea de unidad como transición en máquina de estados supervisoria

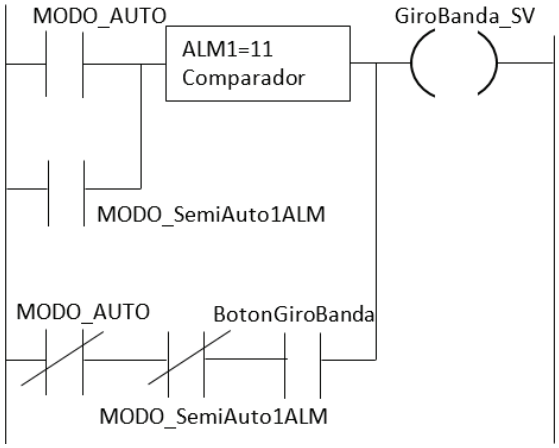

Figura 12. Activación de actuador en modo automático, semiautomático y manual

http://dx.doi.org/10.22201/fi.25940732e.2019.20n4.041

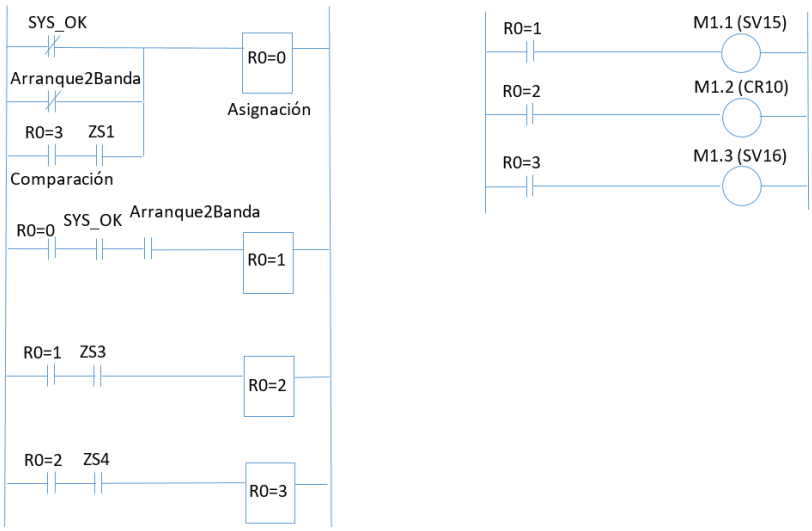

Figura 13. Ejemplificación de generación de código a partir de máquina de estados de Figura 3

Conclusiones

Este trabajo ilustra el uso de una metodología basada en máquinas de estados para la automatización de una celda de manufactura. Las máquinas de estado, como se muestra, no solo son una herramienta analítica, sino también de modelación de la solución de control y programación estructurada de PLC. La metodología se complementa con la definición de modos de operación, lo cual implica el desarrollo de una interfaz humano-máquina que provea la instrumentación virtual adecuada.

Con una metodología de programación se facilita la integración de las distintas unidades y procesamientos de la celda, así como la revisión y corrección de los programas. La metodología al incluir el diseño de una máquina de estados supervisoria permite una planeación de las comunicaciones necesarias entre distintos PLC a cargo de unidades separadas de la celda. Como trabajo futuro se plantea la incorporación de evaluación de métricas de producción en cada unidad de la celda utilizando la estructura de las máquinas de estado, así como el desarrollo de modelos virtuales de la celda y sus componentes en que pueda probarse la automatización.

La experiencia práctica en la aplicación de máquinas de estados desarrolla una competencia tanto de análisis y síntesis, como de diseño e implementación de soluciones avanzadas de automatización, contribuyendo a la capacitación del ingeniero para soportar sistemas de manufactura e incorporar nuevas tendencias tecnológicas como la industria 4.0.

Anexos

Listados de señales de entrada y salida de equipos PLC.

Tabla 1. Sistema de transporte

Entradas PLC	Símbolo	Salidas PLC	Símbolo
Sensor de posición carril 1, lado almacén	ZS1	Válvula solenoide de cilindro lado del robot	SV15
Sensor de posición carril 1, lado robot	ZS2	Válvula solenoide de cilindro lado del almacén	SV16
Sensor de posición carril 2, lado robot	ZS3	Motor de banda transportadora	CR10
Sensor de posición carril 2, lado almacén	ZS4		
Botón pulsador paro de emergencia	PB_TRA		

Tabla 2. Sistema de almacén automático (AS/RS)

Entradas PLC	Símbolo	Salidas PLC	Símbolo
Sensor de posición eje X retraído (en lado izquierdo)	X_IZQ_ZS	Válvula solenoide para retraer en X (a la izquierda)	X_IZQ_SV
Sensor de posición eje X extendido (en lado derecho)	X_DER_ZS	Válvula solenoide para avanzar en X (a la derecha)	X_DER_SV
Sensor de posición eje Y retraído (fuera de almacén)	Y_RET_ZS	Válvula solenoide para retraer en Y (hacia afuera)	Y_RET_SV
Sensor de posición eje Y extendido (dentro de almacén)	Y_EXT_ZS	Válvula solenoide para avanzar en Y (hacia dentro)	Y_EXT_SV
Sensor de posición en ángulo de 0 ° (hacia el almacén)	GIRO_ALM_ZS	Válvula solenoide para girar a 0 ° (hacia almacén)	GIRO_ALM_SV
Sensor de posición en ángulo de 180 ° (hacia la banda)	GIRO_BANDA_ZS	Válvula solenoide para girar a 180 ° (hacia banda)	GIRO_BANDA_SV
Sensor de pinza en posición abierta	PINZA_ABIERTA_ZS	Válvula solenoide para abrir pinza	PINZA_ABRIR_SV
Sensor de pinza en posición cerrada	PINZA_CERRADA_ZS	Válvula solenoide para cerrar Pinza	PINZA_CERRAR_SV
Botón pulsador paro de emergencia	PB_ALM	Elevador - libera freno	CR1
Entrada pulso del encoder	ZT1	Elevador – veloc. baja	CR2
		Elevador – subir	CR3
		Elevador – parar	CR4
		Elevador – bajar	CR5

Tabla 3. Robot y mesa de trabajo

Entradas a Robot	Símbolo	Salidas del Robot	Símbolo
Sensor de posición para detectar contenedor	di1	Activación de efector final del robot: pinza	do1
Señal de inicio de tarea	di5	Señal de fin de tarea	do5

Referencias

Deloitte Touche Tohmatsu Limited and US Council on Competitiveness (2016). 2016 Global Manufacturing Competitiveness Index.

Frey, G. & Litz, L. (2000). *Formal methods in PLC programming*. Nashville TN: Proc. of the IEEE SMC.

Groover, M.P. (2015). *Automation, production systems, and computer-integrated manufacturing*, 4a ed., Pearson Education.

IEC, International Electrotechnical Commission (2013). Programmable controllers, Parte 3: Programming Languages.

Martell-Chávez, F. (2017). Automatización robótica y visión artificial en la industria 4.0. *(Cruz, Ed.).* (15), 61-67. Recuperado de http://www.cio.mx/noticio.php

Mealy, G. (1955). A method for synthesizing sequential circuits. *Bell System Technical Journal*, 1045-1079. https://doi.org/10.1002/j.1538-7305.1955.tb03788.x

Moore, E. (1956). Gedanken-experiments on sequential machines. *Ann. Math. Stud.*, 34, 129-153.

Music, G. & Matko, D. (2002). Discrete event control theory applied to PLC programming. *Automatika*, 43(1-2), 21-28.

Popescu, C. & Martinez-Lastra, J.L. (2010). Formal methods in factory automation. En Silvestre-Blanes J. (Ed.), *Factory automation*, 465-476. InTech. Recuperado de http://www.intechopen.com/books/factory-automaton/fromal-methods-in-factory-automation

Possan, M.C. (2012). Modelling and implementation of supervisory control systems using state machines with outputs. En Aziz F.A. (Ed.). *Manufacturing System*, 285-306. InTech. Recuperado de http://www.intechopen.com/books/manufacturing-system/modelling-and-implementation-of-supervisory-control-systems-using-state-machines-with-outputs

Ramadge, P.J. & Wonham, W.M. (1987). Supervisory control of a class of discrete event processes. *SIAM J. Control and Optimization*, 25(1), 206-230.