Aplicación para control supervisor con redes de Petri y traducción automática a código Ladder para el software RSLogix500™

Application for Petri Net Supervisory Control and Automatic Ladder Code Translation for the RSLogicx500™

CARLOS ALBERTO GAVIRIA-LÓPEZ
Doctor en Automatización Avanzada y Robótica
Universidad del Cauca
cgaviria@unicauca.edu.co
Popayán, Colombia

CRISTHIAN DAVID BUCHELY-MORENO
Ingeniero en Automática Industrial
Universidad del Cauca
cdbm_2@hotmail.com
Popayán, Colombia

FAUSTO RUIZ-COQUE
Ingeniero en Automática Industrial
Universidad del Cauca
fruizco18@hotmail.com
Popayán, Colombia

Fecha de recibido: 08/02/2015
Fecha de aceptado: 26/07/2015

Forma de citar/How to cite: GAVIRIA, Carlos; BUCHELY, Cristhian y RUIZ, Fausto. Aplicación para control supervisor con redes de Petri y traducción automática a código Ladder para el software RSLogix500™. Rev. UIS Ingenierías, 2015, vol. 14, no 2, p.p. 47-55.
DOI: http://dx.doi.org/10.18273/revuin.v14n2-2015005

RESUMEN

Este artículo presenta una aplicación software para el sistema operativo Windows®, que permite el diseño formal y sistemático de supervisores de sistemas dinámicos de eventos discretos bajo el formalismo de red de Petri, y la generación automática de código Ladder para el software RSLogix 500 ™ de Rockwell Automation. La aplicación permite diseñar el supervisor mediante el método de restricciones lineales vectoriales generales, y en el caso de la inadmisibilidad de algunas de las restricciones impuestas, el software provee tres métodos para transformarlas en unas admisibles. También provee dos métodos para hallar los sifones y trampas de la RdP supervisada, lo que permite tomar acciones para prevenir el bloqueo del sistema supervisado. Una vez cumplidos los requerimientos de control, la aplicación permite mapear el supervisor de forma automática, a código Ladder mediante el método extendido Token Passing Logic.

PALABRAS CLAVE: Redes de Petri, Control supervisor, Código Ladder.

ABSTRACT

This paper presents a software tool running on Windows® operating system, intended for formal and systematic design of supervisory controllers for discrete event systems described by Petri nets, and the automatic generation of Ladder code for RsLogix 500™ software. The software is based on general linear vector constraints, and in case of the inadmissibility of some of the imposed constraints, the software has three methods to transform them in admissible ones. The software has also two methods to find siphons and traps on the supervised PN, which is useful to prevent the dead lock of the supervised system. Once the supervisor fulfills the control requirements, the tool can transfer the supervisor, automatically, to Ladder code by the Token Passing Logic extended method.

KEYWORDS: Petri Nets, Supervisory Control, Ladder Code.

1. INTRODUCCIÓN

Numerosos procesos industriales, tales como secuencias de procesamiento o coordinación de recursos compartidos entre estaciones; pueden modelarse como sistemas de eventos discretos (SED). En el ambiente industrial, el control de este tipo de sistemas suele implementarse en controladores lógicos programables (PLC). No obstante, la obtención del algoritmo de control para su posterior implementación se suele realizar con métodos heurísticos donde la solución depende fuertemente del estilo y experiencia del programador, con lo que el mantenimiento de las soluciones se convierte en una tarea difícil de realizar. Por esta razón, es necesario adoptar métodos formales para el diseño del controlador, ya que estos sistemas se vuelven cada vez más complejos y su mantenimiento más complicado (Uzam, 1998).

En este artículo se denomina supervisor a un nuevo SED que es capaz de imponer restricciones de funcionamiento a un SED original denominado de lazo abierto. No existen muchas herramientas de software disponibles que permitan realizar el diseño del supervisor mediante un modelado formal del sistema, y posteriormente traducir el supervisor diseñado en un PLC. Existe una clara brecha entre el desarrollo de herramientas software para modelado y diseño de supervisores bajo enfoques formales tales como máquinas de estados finitos (MEF) o redes de Petri (RdP) en problemas de tipo académico, y el uso de tales posibilidades para la programación de controladores basados en PLC en ambientes de aplicación en la industria. Algunas herramientas existentes para el diseño de supervisores bajo el formalismo de MEF son DESUMA (wiki.eecs.umich.edu, 2014) y SUPREMICA (ÅKesson et al., 2006), donde en esta última se puede generar código Ladder solo en la versión comercial. Bajo el enfoque de RdP, a la fecha de este trabajo se han encontrado herramientas como Petri-LLD (petrilld.sourceforge.net, 2014) y SIPN Editor (Frey et al., 2000), las cuales permiten la generación de código Ladder a partir de RdP. Sin embargo, éstas no proveen facilidades para diseñar un supervisor mediante métodos formales, solo permiten implementar RdPs seguras, y se reportan dificultades en la generación de código Ladder.

En este trabajo se hace un aporte a la reducción de la brecha descrita, mediante la construcción de una aplicación software que permite, en un único entorno: el modelado formal mediante RdP del sistema a controlar, el diseño de un supervisor para el logro de especificaciones deseadas mediante métodos formales para RdP, la verificación de las buenas propiedades del sistema supervisado, y finalmente, la generación automática de código Ladder para la familia de controladores que pueden programarse con el software comercial RSLogix 500, de amplia difusión en el ambiente industrial. La herramienta se ha denominado Control con redes de Petri (CRP), la cual funciona integrada con el software libre PIPE, herramienta para el análisis y simulación de modelos de RdP pipe2.sourceforge.net (2014). CRP se puede descargar junto a su manual en: https://sites.google.com/site/supervisorpetri. CRP es superior a otras herramientas disponibles de forma abierta en cuanto a que permite utilizar métodos de soporte teórico sólido ampliamente aceptados en el ambiente académico, para el diseño del supervisor de un sistema modelado mediante RdP. Además CRP permite realizar la programación de un PLC, dispositivo de control ampliamente utilizado en el ambiente industrial, mediante la opción de traducir el modelo supervisado en RdP a código Ladder. Aunque la generación de código Ladder es específica para un tipo comercial de PLC, el supervisor obtenido podría implementarse de forma manual en otro tipo de plataformas, siguiendo el método descrito en este artículo.

Para el diseño del supervisor, se usa el método de restricciones lineales vectoriales generales (LGVC) (Iordache Antsaklis, 2002), donde en el caso de obtener restricciones no admisibles se recurre a los métodos para obtener restricciones admisibles desde las originales (Iordache Antsaklis, 2002; Moody, 1998; Lima Dórea, 2002). La

herramienta presentada en este artículo permite implementar estos métodos en un entorno de diseño amigable. Otro aspecto a tener en cuenta en el diseño de supervisores para SEDs es el bloqueo. Un método común para prevenir bloqueos es la detección de sifones y trampas (Iordache et al., 1999). Desde este enfoque, la prevención del bloqueo consiste en evitar que los sifones pierdan todas sus marcas. La herramienta cuenta con dos métodos de cálculo de sifones y además un algoritmo para determinar cuándo un sifón puede ser fuente de potenciales bloqueos. Para la implementación del supervisor en PLC, se usa la traducción de la RdP del supervisor obtenido a código Ladder mediante el método Token Passing Logic, el cual se encuentra descrito en (Uzam et al., 1996) y detallado después en (Uzam, 1998). En este trabajo se realizaron algunas modificaciones al método original, con el objetivo de hacerlo factible para casos prácticos.

2. MÉTODOS

A continuación se describen los métodos que soportan las funciones disponibles en CRP.

2.1. Control supervisor basado en LVGC

Las restricciones lineales vectoriales generales (LVGC) son una evolución del control supervisor basado en invariantes de lugar (Moody, 1998; Yamalidou et al., 1996). Este tipo de restricciones tienen la siguiente estructura:

$$L\mu + Hq + Cv \leq b \quad (1)$$

$$L\mu + Hq + Cv \geq b \quad (2)$$

Donde μ es el vector de marcas, q el vector de disparo, v es el vector de Parikh, el cual actúa como un contador de los disparos de las transiciones, b es un vector que define la restricción, y las matrices L, H y C relacionan las transiciones con los lugares afectados por cada vector. El conjunto de ecuaciones lineales con restricciones de las ecuaciones (1) o (2) puede tener solución encontrando, para cada restricción, un nuevo lugar con su correspondiente marcación inicial que satisfaga la inecuación. El resultado es una RdP supervisada, con n lugares de control para n restricciones a imponer por el diseñador. Se invita al lector interesado en conocer en profundidad el método de diseño de supervisores basados en invariantes de lugar, y la extensión LGVC a consultar los trabajos en (Iordache Antsaklis, 2002) y (Iordache, 2004).

2.2. Método Token Passing Logic Extendido

El método utilizado en la aplicación CRP para la generación de código Ladder es el Token Passing Logic puede consultarse en (Uzam, 1998) y (Uzam et al., 1996), ya que sus detalles se omiten en este artículo por razones de espacio. Para facilitar la implementación de RdP en casos de secuencias industriales típicas se introdujeron variantes y extensiones al método en los siguientes aspectos:

- El marcado inicial se establece por medio de una estructura Ladder en el método original.En este trabajo se hace escribiendo directamente en el archivo del código Ladder.
- El método original asume que una transición se dispara cuando, además de estar habilitada, existe un puerto de entrada del PLC asociado con un sensor, en estado uno lógico (denominado evento en este trabajo). En ocasiones, el disparo de una transición pude ser interpretado como la desactivación de una entrada del PLC, esto es muy común en sensores de tipo 1 y 0, por ejemplo supóngase un sensor de proximidad usado para detectar la presencia de un AGV, entonces la desactivación de dicho sensor (estado cero) significaría el disparo de la transición “AGV en marcha”. Para modelar este hecho, es necesario modificar los examinadores de cerrado por examinadores de abierto en la estructura propuesta en la metodología original. Esto se ilustra en la Figura 1, donde X_1 es una entrada asociada a un sensor en el PLC.

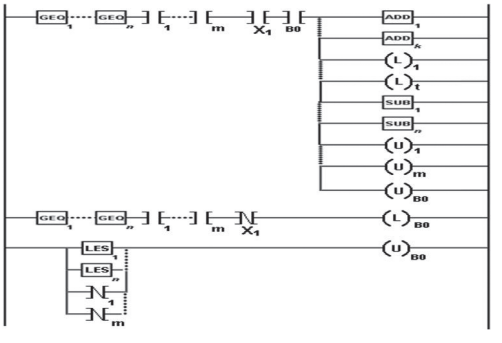

Figura 1. LLD para eventos negados (apagados).

- En muchos casos prácticos el evento asociado al disparo de una transición ocurre cuando la señal del puerto de entrada pasa de cero lógico a uno lógico y no porque el puerto permanezca en uno lógico (por flanco de subida). Estos casos se implementan con el diagrama Ladder de la Figura 2, donde el bit B0 previene el disparo de la transición aunque la señal X_1en el puerto de entrada esté presente.

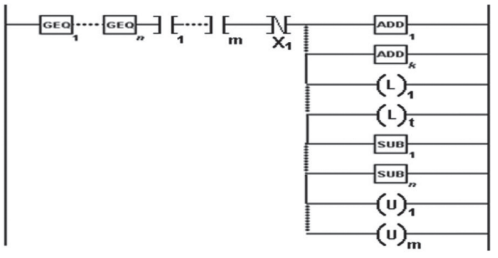

Figura 2. LLD para una transición por flanco.

3. MANUAL DE USUARIO DE CRP

A continuación se describe cómo usar la herramienta de software CRP, la cual ha sido construida en este trabajo en lenguaje C, y consiste en un ambiente integrado basado en pestañas desde donde se pueden lanzar, además del editor de PIPE para el modelado en RdP del sistema, funciones y programas propios para realizar el proceso de: construcción, validación y generación en código Ladder del supervisor del sistema.
Para su funcionamiento CRP necesita los siguientes requisitos:

- Sistema operativo: Windows7/ Vista /WindowsXP/ Windows2003 Server/ Windows 2000/ Windows 98. Otras plataformas Windows podrían funcionar, más no están verificadas.
- Utiliza en promedio 11MB de RAM bajo uso típico (debe sumarse los requerimientos del SO, este valor puede aumentar para redes muy grandes).
- 2.5MB de espacio en disco.
- Si no se tiene instalado, instalar Java Runtime Environment (JRE).

3.1. Edición de redes de Petri

CRP inicia con la apertura del archivo ejecutable "*CRP.exe*" en la carpeta donde se ha copiado el software. El usuario puede crear un nuevo supervisor, construyendo primero el modelo de lazo abierto en RdP, o bien recuperar un trabajo previamente almacenado en un archivo. En ambos casos, CRP lanza automáticamente el software externo PIPE. La versión original de PIPE fue modificada para permitir la edición de transiciones no controlables o no observables y la asignación de entradas y salidas. Además de la edición, PIPE cuenta con algunas herramientas útiles en el proceso de diseño, como son: clasificación, análisis de invariantes, cálculo de la matriz de incidencia y marcado, cálculo de sifones y trampas mínimas y análisis de espacio de estados. La Figura 3 muestra la interfaz de CRP integrada a PIPE.

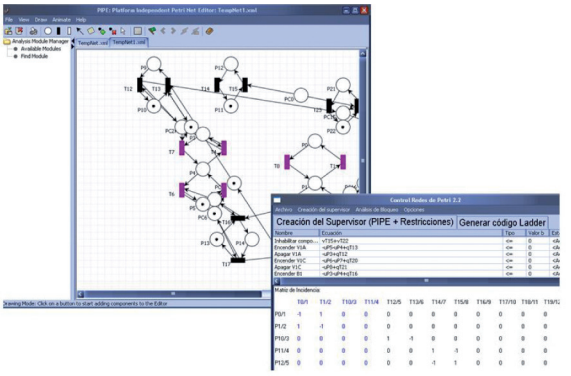

Figura 3. Interfaz de CRP integrada a PIPE.

3.2. Análisis de bloqueo

CRP realiza el análisis de bloqueo mediante el cálculo de sifones de la RdP, utilizando los métodos encontrados propuestos en (Cordone et al., 2003) y (Wegrzyn et al., 2004). Además determina si éstos están o no controlados. Si se encuentran sifones no controlados, éstos se deben controlar ya que son fuente de potenciales bloqueos.

3.3. Adición y edición de restricciones a la RdP

En CRP se pueden agregar restricciones LGVC basadas en las estructuras de las ecuaciones (1) y (2) mediante el menú "Creación del supervisor".

Para agregar una restricción, se usan los caracteres u, v y q para referirse a los vectores de marcado, Parikh y disparo respectivamente, y después el nombre del elemento al cual se hace referencia. El coeficiente por el cual se multiplica el vector debe ser colocado antes del carácter clave, si éste no existe, se supone que es uno. Por ejemplo para imponer la restricción:

$$\mu_0 - \mu_1 + 2\mu_5 + 3q_2 + 2q_5 + 2v_2 - 10v_9 \leq 8,$$

se debe editar en el campo "Ecuación":

$$\text{uP0} - \text{uP1} + 2\text{uP5} + 3\text{qT2} + 2\text{qT5} + 2\text{vT2} - 10\text{vT9},$$

y luego se debe colocar el valor 8 en el campo "Valor b", y seleccionar "Menor igual" en la pestaña "Tipo", como se ilustra en la Figura 4.

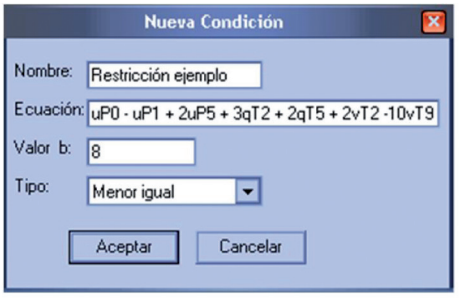

Figura 4. Edición de restricciones.

3.4. Solución de restricciones no admisibles

En caso de existir una restricción inadmisible, se pueden aplicar métodos de solución mediante el comando "Solucionar condición". CRP implementa los métodos: programación lineal de enteros, operaciones por fila de matrices y método de cálculo por el núcleo. La descripción de estos métodos se puede encontrar en los trabajos en (Moody, 1998; Lima Dórea, 2002; Iordache, 2004). Si se encuentra una solución, ésta se puede remplazar por la original.

3.5. Creación del supervisor

Una vez se han editado todas las restricciones del supervisor, este puede ser creado mediante el botón "Generar supervisor". Antes, es necesario que todas las restricciones sean admisibles, si esto es así, en seguida se mostrará en PIPE la red supervisada y en CRP la matriz de incidencia de aquella red.

3.6. Selección del PLC y asociación de entradas y salidas.

Si se va a generar el Ladder para un dispositivo PLC particular por primera vez, antes es necesario obtener la información de ese PLC ya que deben usarse, puertos de entrada o salida válidos para ese dispositivo. Para que el usuario no deba mapear manualmente la información del PLC, se ha construido la aplicación auxiliar "PLC Detector", que se inicia abriendo el archivo ejecutable "*PLC Detector.exe*" en la carpeta donde se ha copiado el software CRP. Dicha aplicación obtiene la información del PLC particular donde se implementará el supervisor. El procedimiento consiste en los siguientes pasos:

- Crear un proyecto nuevo en RsLogix500 con el dispositivo a programar, y configurar su información física y parámetros. El proyecto debe contener al menos una línea de Ladder, cuyo contenido es irrelevante ya que solo interesa la información de configuración generada.
- Guardar el proyecto en formato SLC.
- Cargar el archivo SLC en "PLC Detector", ver Figura 5. Si esta aplicación tiene éxito reconociendo el dispositivo, será posible crear una especificación para este en un archivo con extensión PLC.
- Cargar el archivo PLC a CRP mediante el botón "Cargar PLC".

Este archivo .PLC puede usarse en adelante para cualquier nuevo diseño que use ese dispositivo, a no ser que se agreguen componentes tales como bancos de puertos adicionales, en cuyo caso debe repetirse el anterior procedimiento.

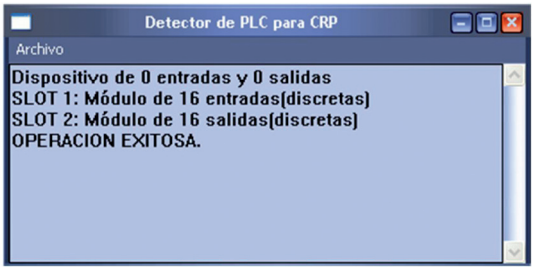

Figura 5. PLC Detector.

3.7. Generación de código Ladder

CRP puede generar Ladder para las familias de PLC Allen-Bradley SLC™ 500 y MicroLogix™, estos PLC son programados mediante RsLogix 500, por lo que es necesario disponer de esta aplicación junto con RsLinx. Antes de generar el Ladder, es posible simular el comportamiento real del supervisor en el PLC, dicha simulación se basa en emular entradas al PLC y observar la respuesta del supervisor, emulando el código Ladder, a diferencia de la simulación en PIPE que simula la RdP. Si CRP ya tiene la información del dispositivo (ver "Cargar PLC"), es posible asignar las entradas y salidas editadas en la RdP a entradas y salidas reales. Realizado lo anterior, se genera el fichero con el Ladder mediante el botón "Generar Ladder".

4. RESULTADOS

Se presenta a continuación la solución de un caso de estudio propuesto en (Moody, 1998) utilizando CRP, para mostrar su capacidad para resolver el problema completo, desde el modelado, y verificación del modelo, hasta la implementación del supervisor en un PLC. Considérese el diagrama de operaciones de la Figura 6.

La máquina toma una pieza desde la cola de entrada. Durante el procesado de la pieza, la maquina puede dañarse y arruinar la pieza. Cuando esto suceda, un vehículo

guiado automáticamente (AGV 2) debe retirar la pieza dañada y ponerla en el sitio de almacenamiento de partes dañadas.

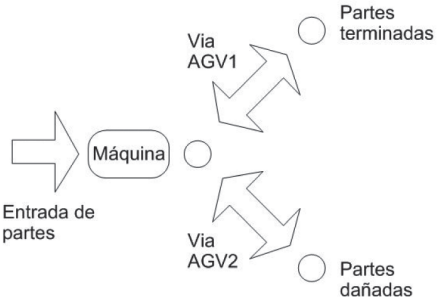

Figura 6. Configuración física del caso de estudio.

Las piezas que son procesadas satisfactoriamente son retiradas por el AGV 1 y puestas en el sitio de almacenaje de piezas completadas. Este caso de estudio se modela mediante la RdP de la Figura 7, cuya descripción de lugares y transiciones se muestra en el Cuadro 1. Las transiciones T2 y T6 se definen como no controlables por cuanto no le es posible a un controlador decidir si ocurren o no. En CRP este tipo de transiciones se distinguen con color púrpura.

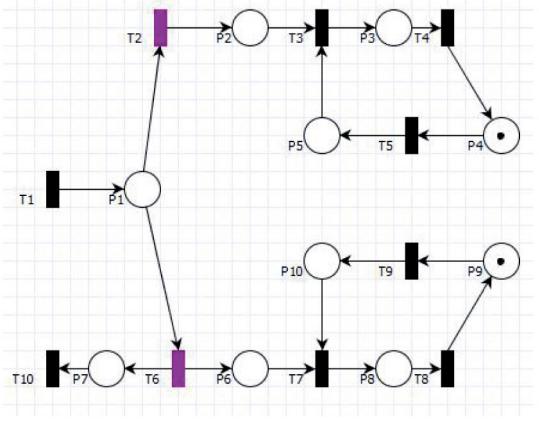

Figura 7. Modelo en red de Petri del sistema caso de estudio.

4.1. Restricciones del proceso

En el caso de estudio, ciertas restricciones se derivan del comportamiento físico del sistema. A continuación se describen tales restricciones y se representan como restricciones LVGC.

1. La pieza producida no puede, a la vez, ser correcta y estar dañada.

$$\mu_2 + \mu_6 \leq 1$$

Cuadro 1. Descripción de lugares y transiciones.

P1	Máquina procesando una pieza
P2	Pieza esperando su traslado a piezas terminadas
P3	AGV 1 trasladando pieza terminada
P4	AGV 1 pone pieza en piezas terminadas
P5	AGV 1 esperando que la pieza sea procesada
P6	Pieza esperando su traslado a piezas dañadas
P7	Máquina en espera de ser reparada
P8	AGV 2 trasladando pieza dañada
P9	AGV 2 pone pieza en piezas dañadas
P10	AGV 2 esperando pieza dañada
T1	Parte removida desde la cola de entrada
UC:T2	Parte procesada satisfactoriamente (no controlable)
T3	Parte tomada por AGV 1 para su traslado
T4	Parte depositada en piezas terminadas
T5	AGV 1 se posiciona para esperar pieza procesada
UC:T6	Fallo de máquina, pieza dañada (no controlable)
T7	Parte tomada por AGV 2 para su traslado
T8	Parte depositada en piezas dañadas
T9	AGV 1 se posiciona para esperar pieza dañada
T10	Máquina reparada

2. Solo un AGV puede acceder a extraer la pieza del buffer.

$$\mu_5 + \mu_1 0 \leq 1$$

3. La máquina no puede operar si está dañada:

$$\mu_1 + \mu_7 \leq 1$$

Si se genera el supervisor sin tener en cuenta la no controlabilidad de T2 y T6, se obtiene la solución mostrada en la Figura 8. Puede verse que el lugar de control PC0 podría inhibir T2 o T6 para cumplir la restricción 1, pero esto no es posible físicamente.

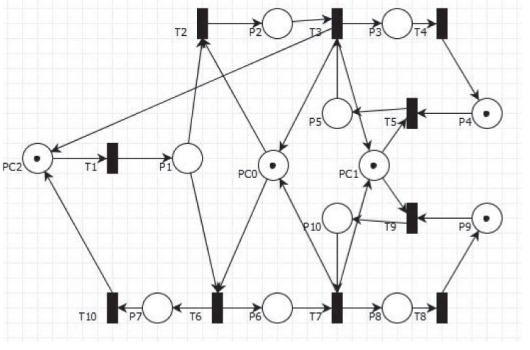

Figura 8. Supervisor con T2 y T6 controlables.

CRP calcula las condiciones de admisibilidad de las restricciones. Teniendo en cuenta la no controlabilidad de T2 y T6, el resultado es el de la Figura 9, que muestra que efectivamente la regla LVGC establecida para la restricción 1, no es admisible. CRP también calcula una

solución admisible para esta restricción. El resultado calculado por CRP es:

$$\mu_1 + \mu_2 + \mu_6 \leq 1$$

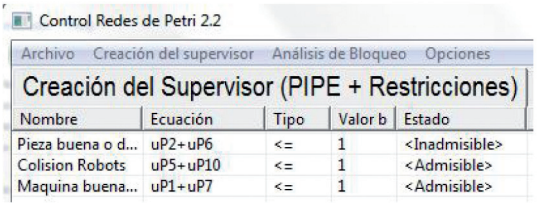

Figura 9. No admisibilidad de la regla para la restricción 1.

El supervisor obtenido luego de resolver la admisibilidad de las restricciones, es el de la Figura 10, donde es fácil comprobar que la restricción 1 se puede lograr sin que PC0 llegue a inhibir transiciones no controlables.

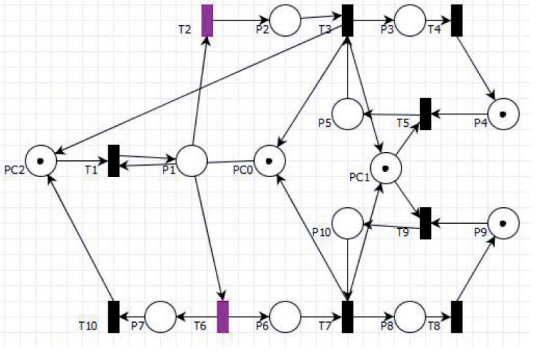

Figura 10. Supervisor admisible.

El análisis de propiedades del supervisor obtenido puede realizarse desde CRP usando las herramientas de PIPE y la prueba de sifones creada en este trabajo. El análisis de espacio de estados provisto por PIPE arroja el resultado de la Figura 11, donde puede verse que la red obtenida es acotada (la red original no es acotada), pero existe bloqueo por punto muerto. En efecto es fácil ver que debido a la restricción 2, si el AGV1 se dirige hacia la máquina, pero la pieza producida está dañada, hay un punto muerto porque el AGV2 ya no puede ir por esa pieza, el AGV1 no puede moverse y la máquina no puede procesar más piezas. Algo similar ocurre si el AGV2 va hacia la máquina y la pieza producida es correcta.

La herramienta de cálculo de sifones de CRP también muestra la condición de bloqueo, encontrando dos sifones mínimos no controlados en la RdP obtenida para el supervisor. La Figura 12 muestra uno de ellos, conformado por los lugares {P1, P5, P6, PC0, PC1}, que indica que si esos lugares quedan sin marcas, seguirán sin marcas en adelante. Ya que el sifón no está controlado, no es posible evitar que esa situación ocurra, como en la situación de bloqueo descrita arriba.

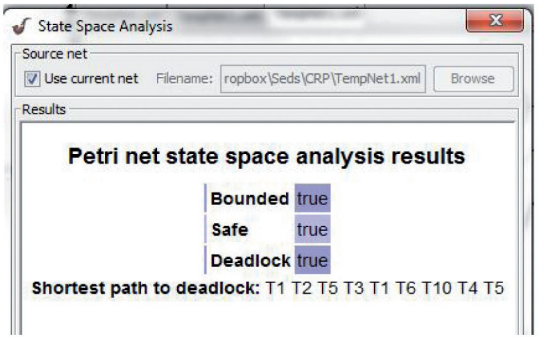

Figura 11. Condición de bloqueo detectada por PIE.

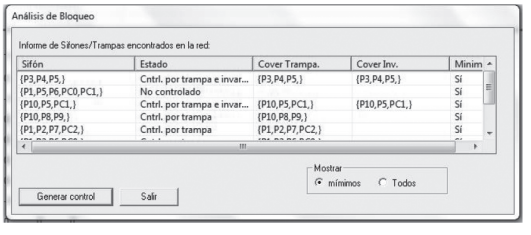

Figura 12. Existencia de sifones mínimos no controlados detectada por CRP.

En este problema, la solución del bloqueo puede hacerse por simple inspección, una vez identificado el problema. Si se corrige el modelo agregando el chequeo de la existencia de piezas dañadas para permitir el desplazamiento del AGV2 y el chequeo de la existencia de piezas correctas antes de permitir el desplazamiento del AGV1, todos los sifones de la red quedan controlados. La Figura 13 muestra la RdP corregida.

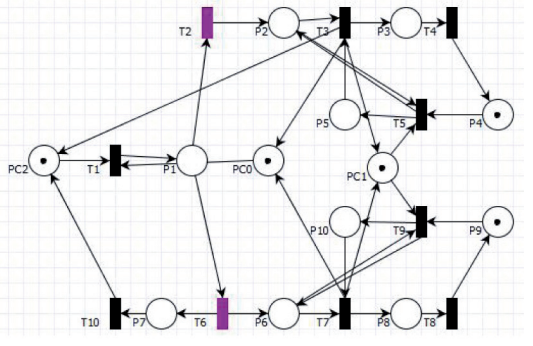

Figura 13. Supervisor sin bloqueos.

El análisis de espacio de estados con PIPE comprueba

que la red no tiene bloqueos, como se observa en la Figura 14.

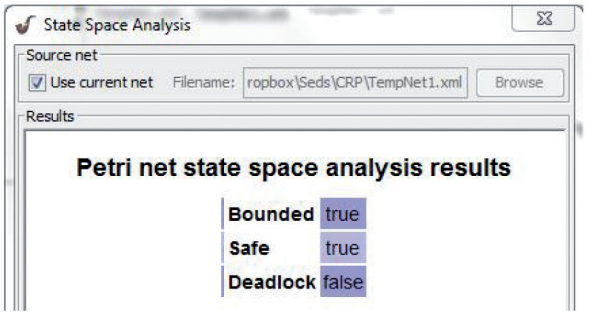

Figura 14. Análisis de propiedades de la RdP de la Figura 13.

CRP permite aumentar restricciones de invariantes para garantizar la no pérdida de marcas de los lugares en un sifón no controlado, de forma amigable. No obstante, en este caso de estudio particular, al hacer uso de esta opción aparecen nuevos sifones no controlados. Luego, la alternativa de control de sifones mediante invariantes de lugar debe usarse con cuidado porque puede resultar mejor opción analizar la causa de pérdida de marcas del sifón y corregir la estructura de la RdP en consecuencia.

4.2. Generación de código Ladder

Si el supervisor cumple los requerimientos de control, CRP permite mapear la RdP a código Ladder para cualquiera de los PLC's que permite programar el software RsLogix 500 de Allen Bradley. Antes de iniciar el mapeo de la RdP se debe realizar la asignación de salidas y entradas en el PLC) a cada lugar y transición que lo requieran, respectivamente. La Figura 15 muestra la interfaz de CRP para la generación de Ladder.

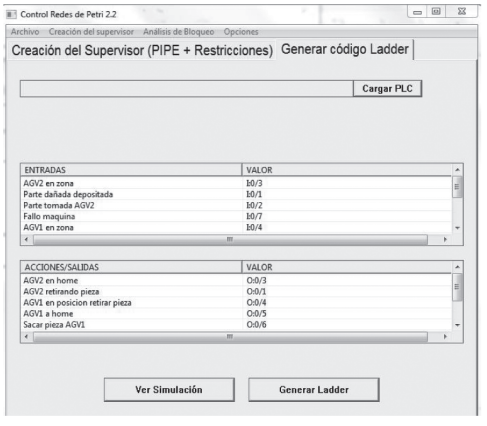

Figura 15. Interfaz para generar código Ladder.

Al usar la opción de Generar Ladder se produce un archivo de extensión SLC compatible con RsLogicx 500. Una fracción del archivo SLC generado para el caso de estudio es la que se muestra en la Figura 16. El archivo Ladder cargado en RsLogicx 500 correspondiente al cuarto peldaño en esa fracción de código, se muestra en la Figura 17, donde se ha superpuesto un recorte con el resultado del verificador del proyecto donde se muestra que no se encontraron errores en el archivo Ladder. Para facilitar la identificación del mapeo de la RdP en la memoria usada en el PLC, CRP genera un reporte en un archivo de texto donde se muestra la asignación de las variables y memoria en el PLC para los correspondientes lugares en la RdP.

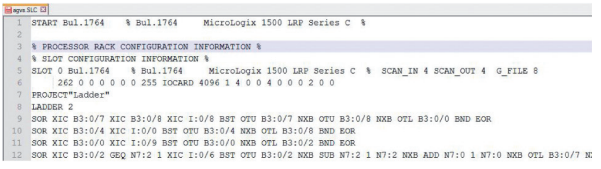

Figura 16. Fracción del código Ladder generado en formato SLC.

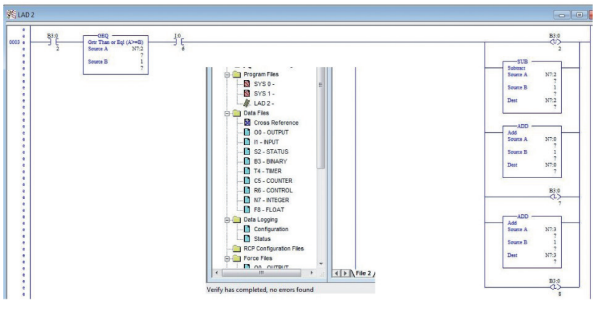

Figura 17. Fracción del código Ladder generado en RsLogix 500.

Para el Ladder de la Figura 17 correspondiente a la implementación del supervisor de la Figura 13, se usó el bit B3:0/2 para representar el lugar P2, B3:0/7 para el lugar PC0, B3:0/8 para el lugar PC2, N7:3 para el lugar PC1, N7:0 para el lugar P3, y N7:2 para el lugar P5. La entrada I:0/6 está asignada al evento de la transición T3. Puede comprobarse que el código generado implementa correctamente los lugares precedentes y posteriores asociados a la transición T3, usando el método Token Passing Logic. Para facilitar la validación del código Ladder generado, CRP cuenta también con un simulador propio, que permite ver la evolución de la RdP desde la interpretación del código Ladder, simulando la ocurrencia de eventos en los puertos de entrada del PLC y registrando las acciones que se realizarían en los puertos de salida. Por razones de extensión no se presenta una figura de este simulador.

Se han realizado prácticas de laboratorio con plantas reales y se ha verificado que el PLC realiza las mismas accio-

nes simuladas con la RdP, pero por el tamaño de las RdP construidas no se presentan en este artículo.

5. CONCLUSIONES

Se ha presentado una herramienta de software disponible de forma libre, CRP, en la cual convergen el potencial del modelado formal de las RdP con los métodos adoptados para el diseño del supervisor, y el método para el mapeo de la RdP a Ladder, para permitir el diseño de supervisores de manera formal y sistemática; disminuyendo el tiempo de diseño, evitando soluciones que dependen de estilos personales de programación, garantizando que el supervisor generado esté libre de errores, y facilitando el mantenimiento del algoritmo de control dado que el programador ya no deberá modificar directamente código Ladder.

CRP permite imponer restricciones sobre el modelo de la planta en lazo abierto de manera sencilla, imponer restricciones sobre los sifones no controlados que puedan aparecer en el diseño para prevenir bloqueos, aunque en ocasiones el tamaño y/o topología de la RdP hace que el número de sifones aumente, lo que dificulta encontrarlos y determinar su control. Cuando el supervisor cumple con los requerimientos de especificaciones y de seguridad, es posible asociar a cada lugar y transición, las salidas y entradas, respectivamente, para mapear la RdP del supervisor de forma automática a código Ladder para un PLC de la familia soportada por CRP. Además se puede verificar el desempeño del código generado mediante la simulación de las entradas que pueden ocurrir en la realidad.

6. REFERENCIAS

ÅKesson, K.; Fabian, M.; Flordal, H.; Malik, R. Supremica—an integrated environment for verification, synthesis and simulation of discrete event systems. En *Proceedings of the 8th International Workshop on Discrete Event Systems* (pp. 384–5), 2006.

Cordone, R.; Ferrarini, L.; Piroddi, L. Some results on the computation of minimal siphons in Petri nets. En *Conference on Decision and Control*. volumen 4, 2003. doi:10.1109/CDC.2003.1271733.

Frey, G.; Kaiserslautern, U.; Minas, M. Editing, visualizing, and implementing signal interpreted petri nets. En *Proceedings of the AWPN 2000* (pp. 57–62), 2000.

Iordache, M. *Methods for the Supervisory Control of Concurrent Systems Based on Petri Net Abstractions*. Ph.D. thesis Notre Dame, IN, USA, 2004.

Iordache, M.; Antsaklis, P. Synthesis of supervisors enforcing general linear constraints in petri nets. The 2002 American Control Conference, 2002.

Iordache, M.; Moody, O.; Antsaklis, P. *A method for deadlock prevention in discrete event systems using Petri nets*. Technical Report University of Notre Dame, 1999. Technical Report of the ISIS Group.

Lima, E.; Dórea, C. An algorithm for supervisory control of discrete-event systems via place invariants. 15th Triennial World Congress of the International Federation of Automatic Control, 2002.

Moody, J. *Petri Net Supervisors for Discrete Event Systems*. University of Notre Dame, 1998. URL: https://books.google.com.co/books?id=6gD6NwAACAAJ.

petrilld.sourceforge.net Petrilld:develop industrial automation with petri nets. http://petrilld.sourceforge.net/mwiki/index.php/Main_Page, 2014. Consulta: 10-01-2014.

pipe2.sourceforge.net Platform independent petri net editor 2. http://pipe2.sourceforge.net, 2014. Consulta: 10-01-2014.

Uzam, M. *Petri-net-based supervisory control of discrete event systems and their ladder logic diagram implementations*. Ph.D. thesis University of Salford, 1998.

Uzam, M.; Jones, A.; Ajlouni, N. Conversion of Petri net controllers for manufacturing systems into ladder logic diagrams. En *Emerging Technologies and Factory Automation*, 1996. doi:10.1109/ETFA.1996.573978.

Wegrzyn, A.; Karatkevich, A.; Bieganowski, J. Detection of deadlocks and traps in Petri nets by means of Thelen's prime implicant method. *International Journal of Applied Mathematics and Computer Science*, *14*, 113–121, 2004.

wiki.eecs.umich.edu Desuma. https://wiki.eecs.umich.edu/desuma, 2014. Consulta: 10-01-2014.

Yamalidou, K.; Moody, J.; Lemmon, M.; Antsaklis, P. Feedback control of petri nets based on place invariants. *Automatica*, *32*, 15–28, 1996. doi:10.1016/0005-1098(95)00103-4.

7. AGRADECIMIENTOS

Los autores agradecen a la Universidad del Cauca por el soporte brindado al desarrollo de este trabajo mediante la prestación de los recursos de personal y equipos necesarios para la verificación de los resultados obtenidos.