

CENTRO DE INVESTIGACIÓN Y ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL



TESIS DOCTORAL

**METODOLOGÍA PARA EL MODELADO Y CONTROL DE
PROCESOS INDUSTRIALES DESCRITOS POR LA NORMA
ISA 5.1, UN ENFOQUE EN REDES DE PETRI**

Tesis sometida como requerimiento para la obtención del grado de
Doctor en Ciencias en Ingeniería Eléctrica

PRESENTA

M. I. Durvvin Alexis Rozo Ibañez

Asesor: Dr. Jose Javier Ruiz León

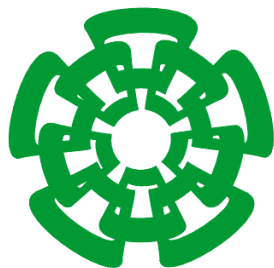
Asesor: Dr. Antonio Ramírez Treviño

Control Automático

Unidad Guadalajara

ZAPOPAN, JALISCO

2022



Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Metodología para el modelado y control de procesos industriales descritos por la norma ISA 5.1, un enfoque en redes de Petri

Tesis que presenta:

Durvvin Alexis Rozo Ibañez

para obtener el grado de:

Doctor en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. José Javier Ruíz León

Dr. Antonio Ramírez Treviño

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, Abril de 2023.

Metodología para el modelado y control de procesos industriales descritos por la norma ISA 5.1, un enfoque en redes de Petri

**Tesis de Doctorado en Ciencias
Ingeniería Eléctrica**

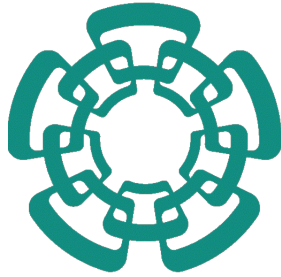
Por:

Durvvín Alexis Rozo Ibañez
Magister en Controles Industriales
Universidad de Pamplona 2009-2011

Becario de CONACYT, expediente no. 613626

Directores de Tesis
Dr. José Javier Ruíz León
Dr. Antonio Ramírez Treviño

CINVESTAV del IPN Unidad Guadalajara, Abril de 2023.



**Centro de Investigación y de Estudios
Avanzados del Instituto Politécnico
Nacional
Unidad Guadalajara**

**Methodology for the modeling and control of
industrial processes described by the ISA 5.1
Standard, a petri nets approach**

A thesis presented by:
Durvvin Alexis Rozo Ibañez

to obtain the degree of:
Doctor in Science

in the subject of:
Electrical Engineering

Thesis Advisors:
**Dr. José Javier Ruiz León
Dr. Antonio Ramírez Treviño**

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, April, 2023.

Methodology for the modeling and control of industrial processes described by the ISA 5.1 Standard, a Petri nets approach

**Doctor of Science Thesis
In Electrical Engineering**

By:
Durvvín Alexis Rozo Ibañez
Master in Industrial Controls
University 2009-2011

Scholarship granted by CONACYT, No. 613626

Thesis Advisors:
Dr. José Javier Ruiz León
Dr. Antonio Ramírez Treviño

CINVESTAV del IPN Unidad Guadalajara, April, 2023.

Resumen

Este trabajo presenta el diseño de una metodología formal basada en Redes de Petri Interpretadas (RdPI) para traducir la información de un proceso industrial presentada mediante un Diagrama de Tuberías e Instrumentos (DTI) y sus respectivas narrativas de control y operación dadas por el Estándar ANSI ISA 5.1 a un lenguaje de programación para PLC usando la Norma IEC 61131-3.

La metodología transfiere toda la información del proceso en tablas. La información del DTI está representada en un lenguaje estructurado y no es ambigua, por lo tanto se traduce a las tablas sin mayor problema. Sin embargo, como la información proveniente de las narrativas de control y operación están dadas en lenguaje natural, su traducción a tablas es más compleja y hay información que no se puede traducir, bien porque no se entiende, o bien porque existen ambigüedades y/o omisiones. La metodología propuesta es capaz de detectar estos errores de omisión, ambigüedad, o inconsistencia. Debido a que no siempre se pueden corregir automáticamente estos errores, en esta parte se puede requerir de la asistencia de un operador.

Toda vez que se compruebe que los errores han sido eliminados, se construye de forma automática el modelo del sistema y de la especificación. Después, usando la técnica de diseño de control por regulación, también de forma automática se diseña el controlador y se programa el PLC.

Con la metodología propuesta se eliminan todos los posibles errores en el diseño del controlador y la programación del PLC, ahorrando tiempo de programación y depuración de código. Además, no es necesario que el operador se re-entrene en teorías de Control Supervisor y/o regulación, así como en los formalismos de Autómatas Finitos y Redes de Petri, ya que el uso de tales técnicas y herramientas formales es transparente para el operador.

Abstract

This work presents the design of a formal methodology based on Interpreted Petri Nets (RdPI) to translate the information of an industrial process presented through a Piping and Instrumentation Diagram (P&ID) and their respective control and operation narratives given by the ANSI Standard. ISA 5.1 to a programming language for PLCs using the IEC 61131-3 Standard. The methodology transfers all the process information into tables. The DTI information is complete and is transferred to the tables without major problems. However, since the information coming from the control and operation narratives is given in natural language, its translation is more complex and there is information that cannot be translated, either because it is not understood, or because there are ambiguities and/or omissions. The proposed methodology is capable of detecting these errors of omission, ambiguity, or inconsistency. Because they cannot always be corrected automatically, the assistance of an operator may be required in this part. Every time it is verified that the errors have been eliminated, the system model and the specification are automatically built. Then, using the control or regulation design technique, the controller is also designed automatically and the PLC is programmed. With the proposed methodology, all possible errors in the controller design and PLC programming are eliminated, saving programming time and code debugging. Furthermore, the operator doesn't need to be re-trained in Supervisory Control and/or regulation theories, as well as in the formalisms of Finite Automata and Petri Nets, since the use of such techniques and tools is transparent to the operator.

Agradecimientos

En primer lugar quiero expresar mi más sinceros agradecimientos a mi esposa Yennifer y mis hijas Amada y celeste, por su compañía y apoyo incondicional en esta etapa importante de mi vida. Sin su apoyo este trabajo nunca se habría escrito y, por eso, este trabajo es también el suyo.

De igual forma, agradezco a mi familia formada por mi madre Olga, mis hermanos Favian y Jair, mis suegros Doraima y Miguel, mis cuñadas Diana, Claudia y Laura, y demás familiares que nos han apoyado desde la distancia para cumplir nuestros sueños. Agradecerle a mi Padre José Rozo (QEPD), Porque estuviste ahí para enseñarme a transmitir valores para la vida, por el esfuerzo y la educación para que nos brindaste para ser mejores personas cada día.

Le agradezco a mis tutores Antonio Ramírez Treviño y José Javier Ruiz León por su acompañamiento, orientación y paciencia durante esta etapa de aprendizaje, sin sus consejos y correcciones no hubiese podido llegar a esta instancia tan anhelada. Los llevaré en la memoria en mi futuro profesional.

Agradecerles a todos mis compañeros de estudio Anuar, Ricardo, Larbi, Daniel, Fredy, Manuel, Gustavo, Jaime, Miguel, los cuales muchos de ellos se han convertido en mis amigos y hermanos. Gracias por las horas compartidas, los trabajos realizados en conjunto y las historias vividas. De la misma manera agradecerle a Harold López por su valiosa ayuda y amistad, la cual ha sido muy importante para culminar mis estudios.

Le agradezco al CINVESTAV por otorgarme un espacio de trabajo, así como los recursos humanos y económicos para la ejecución de las actividades de investigación. Al CONACYT por otorgarme el financiamiento económico para mis estudios de doctorado, CVU 774036.

Índice general

Índice de figuras	XI
Índice de tablas	XV
1. Introducción	1
1.1. Objetivos	3
1.2. Aportaciones y publicaciones	4
1.3. Estructura de la tesis	5
2. Preliminares	9
2.1. Redes de Petri	9
2.2. Redes de petri interpretadas	12
2.3. Proceso industrial	15
3. Metodología de traducción y recuperación	23
3.1. Metodología de traducción y recuperación	24
3.1.1. Estructura de tablas del sistema	26
3.1.2. Estructura de grafo del proceso	37
3.1.3. Modelo de RdPI general del sistema	39
3.2. Metodología de detección y recuperación	47
3.2.1. Detección y clasificación de errores	48
3.2.2. Recuperación de errores	51
4. Control por regulación y traducción a PLC	55
4.1. Control por regulación	55
4.2. Sistema de lazo cerrado	57

4.2.1.	Condiciones de solución para el problema de control de regulación	58
4.2.2.	Condiciones de la solución	60
4.3.	Diseño del controlador	60
4.3.1.	Cálculo de Π y secuencias de disparo controlables	60
4.3.2.	Representación de RdPI en lazo cerrado del sistema	64
4.4.	Traducción a lenguaje escalera	67
4.4.1.	Reglas de traducción de estructura de RdPI a diagrama escalera	68
4.4.2.	Ejemplo de traducción de RdPI a lenguaje escalera	72
5.	Aplicación de la metodología de traducción	77
5.1.	Aplicación de la metodología de traducción	77
5.2.	Aplicación de la metodología de traducción	84
5.3.	Aplicación de la metodología de traducción	88
5.4.	Aplicación de la metodología de traducción	89
5.5.	Aplicación de la metodología de traducción	90
5.6.	Aplicación de la metodología de traducción	92
5.7.	Aplicación de la metodología de traducción	93
6.	Conclusiones y trabajo futuro	101
6.1.	Conclusiones	101
6.2.	Trabajo futuro	102
	Bibliografía	103

Índice de figuras

2.1. Sistema de Red de Petri	10
2.2. Red de Petri con $\Sigma_O = \{A, B, C\}$	13
2.3. a) Grafo de alcanzabilidad de la RdPI para el Ejemplo 2.2.1, b) Vectores de salida correspondientes a la evolución del marcado. . .	14
2.4. Diagrama de flujo para la producción de etilenglicol.	16
2.5. Instrumentos generales o símbolos funcionales	17
2.6. Sistema de llenado de botellas.	19
2.7. DTI Sistema de llenado de botellas.	19
3.1. Metodologías para el diseño de lógica de programación basada en autómatas programables. a) Metodología típica, basada en la experiencia y conocimiento del proceso. b) Metodología propuesta, basada en formalismos matemáticos y análisis de errores. .	25
3.2. Grafo del proceso para el sistema de llenado de botellas.	39
3.3. Construcción de lugares y estado inicial de la válvula LV-104. . .	42
3.4. Módulo de RdP de la válvula de nivel LV-104.	43
3.5. Módulos de RdP del proceso de llenado de botellas.	43
3.6. RdPI del proceso de llenado de botellas.	44
3.7. Módulos RdPI de operación para las tareas de descarga y llenado botellas	46
3.8. Modelo RdPI de la especificación para el proceso de llenado de botellas.	47
3.9. Metodología de detección y recuperación de errores.	48
3.10. Detección y clasificación de errores.	48
3.11. Recuperación de errores.	51

4.1. RdPI que representa una Especificación de un proceso industrial. Es una máquina de estado segura y representa una concurrencia.	56
4.2. Esquema de sistema a lazo cerrado.	58
4.3. Diagrama de conmutación. Cada marcado en la especificación es relacionado a un marcado en la RdPI de la planta, y cada transición de la especificación es equivalente a una secuencia de transiciones en la planta.	59
4.4. Expansión parcial del grafo de alcanzabilidad.	62
4.5. Sistema en lazo cerrado aplicando el Algoritmo 5.	65
4.6. Lazo cerrado resultante del sistema de llenado de botellas aplicando el Algoritmo 5.	66
4.7. Representación de una transición en lenguaje escalera.	69
4.8. Representación de marcado inicial en lenguaje escalera.	70
4.9. Traducción de transición de tiempo cero de RdPI a lenguaje escalera.	71
4.10. Traducción de transición de tiempo cero ($d=0$) con n lugares de entrada y m lugares de salida de RdPI a lenguaje escalera.	71
4.11. Traducción de transiciones de tiempo cero ($d = 0$) con n lugares de entrada y un lugar de salida de RdPI a lenguaje escalera.	72
4.12. lazo cerrado del proceso de evacuación de botellas del Ejemplo 2.3.1.	73
4.13. Diagrama escalera de la condición inicial para el proceso de evacuación de botellas.	74
4.14. Diagrama escalera de la especificación del proceso de evacuación de botellas.	75
4.15. Diagrama escalera de transiciones energizadas para proceso de evacuación de botellas.	76
5.1. Sistema de envasado de bebidas carbonatadas.	78
5.2. DTI Sistema de envasado de bebidas carbonatadas.	79
5.3. Estructura de grafo del sistema de envasado de bebidas carbonatadas.	89
5.4. Estructura de RdPI de la planta del sistema de envasado de bebidas carbonatadas.	90

5.5. Estructura de RdPI de la especificación del sistema de envasado de bebidas carbonatadas. La figura a) muestra la secuencia de operaciones para las tareas llenado tanque T1 y T2; b) RdP de operación para el llenado del reactor; c) secuencia de operación para la tarea calentar reactor; d) secuencia de operación para la tarea descargar latas; e) secuencia de operaciones para la tarea llenar latas.	91
5.6. Lazo cerrado del sistema de envasado de bebidas carbonatadas. . .	92
5.7. Representación de la condición inicial en lenguaje escalera del sistema de envasado de bebidas carbonatadas.	93
5.8. Lenguaje escalera del proceso de llenado de los tanques T1 y T2 del sistema de envasado de bebidas carbonatadas.	94
5.9. Lenguaje escalera del proceso de llenado y calefacción del reactor para el sistema de envasado de bebidas carbonatadas.	95
5.10. Lenguaje escalera del proceso de liberación de banda transportadora para el sistema de envasado de bebidas carbonatadas.	96
5.11. Lenguaje escalera del proceso llenado de latas para el sistema de envasado de bebidas carbonatadas.	97
5.12. Lenguaje escalera de transiciones energizadas para el proceso de llenado de tanque T1 del sistema de envasado de bebidas carbonatadas.	97
5.13. Lenguaje escalera de transiciones energizadas para el proceso de llenado de tanque T2 del sistema de envasado de bebidas carbonatadas.	98
5.14. Lenguaje escalera de transiciones energizadas para el proceso de llenado de reactor del sistema de envasado de bebidas carbonatadas.	99

Índice de tablas

2.1. Nombre típico de etiqueta	17
3.1. Descripción de elementos del proceso.	26
3.2. Comportamiento de elementos del proceso.	27
3.3. Relación permisiva para el ejemplo de llenado de líquidos.	27
3.4. Relación síncrona.	28
3.5. Descripción de elementos para el proceso de llenado de botellas.	29
3.6. Comportamiento de elementos para el ejemplo de llenado de líquidos.	30
3.7. Relación permisiva proceso llenado de botellas.	31
3.8. Relación síncrona proceso llenado de botellas.	32
3.9. Rangos de operación.	33
3.10. Condiciones de operación.	34
3.11. Rangos de operación.	35
3.12. Condiciones de operación proceso de llenado de botellas.	36
3.13. Errores capturados desde la estructura de tablas	50
3.14. Funciones de recuperación de errores de omisión	52
3.15. Funciones de recuperación de errores de inconsistencia	52
3.16. Descripción de elementos.	54
3.17. Descripción de elementos.	54
5.1. Descripción de elementos del sistema de envasado de bebidas carbonatadas.	84
5.2. Tabla de comportamiento de elementos del sistema de envasado de bebidas carbonatadas	85

5.3. Relación permisiva para el sistema de envasado de bebidas carbonatadas.	86
5.4. Relación síncrona para el sistema de envasado de bebidas carbonatadas.	87
5.5. Rangos de operación para el sistema de envasado de bebidas carbonatadas.	87
5.6. Condiciones de operación para el sistema de envasado de bebidas carbonatadas.	88

Capítulo 1

Introducción

Los procesos industriales están formados por secuencias de operaciones unitarias, las cuales se pueden clasificar como físicas (molienda, mezcla, intercambio de calor, secado, etc.), químicas (refinación, pelado químico) y/o bioquímicas (fermentación, esterilización, pasteurización) [1, 2], entre otras. Por lo general, estos procesos son representados de forma gráfica mediante Diagramas de Tuberías e Instrumentación (DTI) con el fin realizar una identificación funcional de los elementos del proceso, además de plasmar algunas de sus características (si es sensor, actuador, lazo del control, etc.). Aunque éste es un mecanismo conveniente y no ambiguo para la comunicación entre ingenieros, se encuentra incompleto debido a que no es posible representar la funcionalidad del proceso y el comportamiento deseado del sistema en lazo cerrado. Adicional a estos inconvenientes, los DTI's al ser un lenguaje estructurado no formal de comunicación en ingeniería, carecen de un formalismo matemático que pueda ser usado con propósitos de análisis y diseño de control de procesos.

En la práctica, a los DTI's se les acompaña con las narrativas de proceso y operación, lo que permite la descripción de la funcionalidad del sistema y del comportamiento requerido en lazo cerrado del proceso. Las narrativas se describen en forma de lenguaje natural [3]. Este hecho puede introducir errores [4, 5, 6, 7] en la información del proceso. Los errores pueden ser del tipo de omisión, cuando se olvida describir un dispositivo o su funcionamiento; ambigüedad, cuando de la descripción en lenguaje natural se pueden interpretar varios hechos diferentes; inconsistencia, cuando la información describe hechos opuestos entre sí. Para remover estos errores, el ingeniero que programa el

controlador, generalmente un Controlador Lógico Programable (PLC, por sus siglas en inglés) usará diferentes técnicas de búsqueda y depuración de errores. Desafortunadamente, estas técnicas resultan incompletas y costosas. Además, durante el proceso de interpretación de las narrativas y codificación, el ingeniero puede introducir nuevos errores, por lo que, usando estas técnicas, no se podrá asegurar que la implementación del controlador está libre de errores.

Para hacer frente a estos problemas, muchos investigadores han propuesto herramientas y metodologías formales para modelar y analizar estos procesos, ya sea con base en Autómatas finitos o redes de Petri [8]. Por ejemplo, [9, 10, 11, 12, 13, 14] introducen metodologías para el modelado y análisis de procesos. Desafortunadamente, todos ellos exigen un conocimiento profundo de la herramienta formal, limitando su aplicabilidad en industrias reales.

El enfoque propuesto en este documento utiliza la información del proceso dada por las características y comportamiento de los elementos (sensores y actuadores), además del funcionamiento y operación del proceso, los cuales siempre están disponibles en la industria y cuyo uso está ampliamente difundido entre los operadores e ingenieros. Esta información es suministrada a través de una estructura gráfica (DTI), y en forma de lenguaje natural por las narrativas de proceso y operación del proceso, las cuales permiten construir el modelo de Red de Petri de la planta y comportamiento requerido del proceso. Con base a estos modelos, un controlador supervisor [15, 16, 17, 18] se puede diseñar e implementar formalmente, evitando errores introducidos en los enfoques heurísticos. La ventaja de las redes de Petri reside en la compacidad del modelo derivado y en que se pueden diseñar diferentes clases de controladores (por ejemplo, control de seguimiento [19, 20, 21]). Incluso si diferentes trabajos en la literatura han propuesto modelos de red de Petri para sistemas industriales, hasta donde sabemos, ninguna de esas metodologías de modelado se basa en el estándar ISA S5.1, por lo tanto, los ingenieros deben comenzar desde cero la descripción y el modelado del proceso, que requieren aprender nuevas herramientas formales y abandonar los estándares bien establecidos. El enfoque de la metodología propuesta sigue una estrategia modular, donde la información del proceso industrial dada por el DTI, la narrativa del proceso ($\mathcal{PN}$) y la narrativa de operación ($\mathcal{ON}$) la cual se traduce a un lenguaje estructurado en forma de tablas, representan las características, comportamiento y la operación requerida de los elementos del proceso. Luego, la información capturada en la estructura de tablas se traduce en un grafo de proceso ($\mathcal{PG}$) que representa la interacción

entre actuadores y variables controladas de un proceso industrial. Finalmente, la información de las tablas y el grafo del proceso son usados en la construcción de modelos de Red de Petri Interpretada RdPI que ilustran el comportamiento del proceso y el funcionamiento requerido del proceso, respectivamente. En cada etapa, se detectan y eliminan errores tales como omisiones, inconsistencias, lo que lleva a obtener modelos de RdPI libres de errores.

1.1. Objetivos

Con base en lo expuesto anteriormente los objetivos planteados en este trabajo de tesis se presentan a continuación.

Objetivo general

Diseñar una metodología de trabajo para diseño de controladores de procesos industriales que contenga las etapas de: a) traducción de la información de un proceso industrial descrito con un DTI y sus narrativas de proceso y operación a modelos de RdPI; b) remoción de errores introducidos durante la etapa de traducción; c) diseño automático de especificaciones y controladores así como programación automática de PLC.

Objetivos específicos

- Establecer una metodología para traducir la información de un proceso industrial a RdPI, donde se verifique la coherencia de los requisitos y se valide el modelo de RdPI derivado.
- Diseñar una estructura de grafos que permita identificar de forma anticipada errores de omisión en la información de un proceso industrial.
- Diseñar una metodología de diseño de especificaciones y controladores que garantice una programación automática de controladores en PLC que deje código libre de errores.

1.2. Aportaciones y publicaciones

Acorde con los objetivos planteados, las aportaciones principales de este trabajo de tesis son:

- Representación de procesos industriales en estructura de tablas que capturan el comportamiento del sistema y su especificación, así como los elementos que pertenecen al sistema y sus funcionalidades.
- Metodología de traducción de un DTI y narrativas de proceso y operación a la estructura de tablas propuesto. Esta metodología incluye una detección y remoción temprana de errores.
- Traducción de los procesos industriales especificados en la estructura de tablas a RdPI. La traducción obtiene una RdPI para el modelo del sistema y otra para la especificación.
- Construcción automática de controladores y programación de PLC. En esta parte se muestra que el diseño de controladores se puede automatizar así como la traducción de controladores diseñados en RdPI pueden ser traducidos a código de PLC, el cual está libre de errores.

De los aportes de este trabajo se obtuvieron las siguientes publicaciones:

- **Petri net modelling of industrial processes from a PID description.**
 - Autores: D Rozo-Ibanez, J Ruiz-León, D Guevara-Lozano, CR Vázquez.
 - Conferencia: 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)
- **Removing Omissions and Inconsistencies from the ISA 5.1 Narratives of Industrial Processes**
 - Autores: D Rozo-Ibañez, J Ruiz-León, D Guevara-Lozano, CR Vázquez, Y Yuliana Rios.
 - Conferencia: 2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)

■ Regulation Control of Discrete Event Systems for Industrial Automation

- Autores: Daniel Guevara-Lozano, Durvvín Rozo-Ibañez, Carlos Renato Vázquez, Javier Ruiz-León and Antonio Ramírez-Treviño.
- Capítulo de libro: Advances in Robotics and Automatic Control: Reviews, Vol. 2.

1.3. Estructura de la tesis

Para la presentación del material de investigación, la tesis se ha estructurado de la siguiente manera.

Capítulo 2: Preliminares

Este capítulo presenta los conceptos básicos relacionados con procesos industriales, Redes de Petri y Redes de Petri Interpretadas. Estas últimas son la herramienta usada para la formalización y análisis de los procesos industriales. Además, de la teoría de regulación basada en RdPI será usada para el diseño del controlador así como la identificación y funcionamiento de las variables que intervienen en un proceso industrial. La sección 2.1 presenta el concepto de Red de Petri como herramienta gráfica para la representación y análisis de los elementos que conforman un proceso industrial. Cada elemento del proceso se presenta como una Red de Petri individual, ilustrando las características y comportamiento del mismo. La sección 2.2 muestra la definición de Red de Petri Interpretada como herramienta de modelado y análisis del comportamiento de los elementos y del proceso industrial. En esta sección, los módulos individuales de Red de Petri se interconectan entre sí dependiendo del comportamiento de las variables del proceso por la acción de los actuadores. La sección 2.3 presenta conceptos asociados a los procesos industriales y la forma de representarlos mediante diagramas DTI, y las narrativas de proceso y operación. Esta información permite conocer las características y comportamiento de los elementos, además de ilustrar el funcionamiento del proceso.

Capítulo 3: Metodología de traducción, detección y recuperación de errores

Este capítulo describe en detalle la metodología de traducción de la información de un proceso industrial a modelos de RdPI con fines de

formalización y análisis de procesos. La sección 3.1 ilustra la forma de representar la información de un proceso industrial (dada en forma de lenguaje natural) a una forma estructural mediante un conjunto de tablas, donde es posible capturar con mayor facilidad errores en la información del proceso. Esta sección también presenta la traducción de la estructura de tablas a una estructura de grafos, donde algunos errores por completitud son capturados previamente en la información del proceso; y finalmente, la traducción de la estructura de grafos a RdPI que ilustra la relación entre módulos independientes de RdP, las cuales representan la relación y comportamientos de los elementos, además del funcionamiento del proceso. La sección 3.2 presenta la metodología de detección y recuperación de los posibles errores de la información de un proceso industrial. La primera parte presenta la detección y clasificación de errores, donde un conjunto de errores potenciales son establecidos de acuerdo a la información de la estructura de tablas, además de presentar el algoritmo de detección de errores con el fin de automatizar la búsqueda y detección de errores en la información presentada en la estructura de tablas. La siguiente sección presenta la recuperación de errores a través de la metodología de traducción. En esta sección se define un conjunto de funciones de recuperación de acuerdo a los errores capturados en la sección anterior, además se muestra el algoritmo de recuperación de errores en donde se ejecuta una función de recuperación dependiendo del error capturado, y como resultado final se obtiene un conjunto de tablas libres de errores de la estructura de tablas.

Capítulo 4: Control por regulación y traducción a PLC

Este capítulo presenta la definición del problema de control por regulación usado para diseñar un controlador que iguale la salida del sistema con la salida de la especificación. La primera parte de esta sección define la planta y la especificación como RdPIs seguras, además de definir el problema de control que consiste en encontrar una función de controlador que proporcione la secuencia de disparo de transiciones en la planta. La siguiente parte corresponde al sistema de lazo cerrado donde se muestra como evolucionan los modelos de RdPI de la planta y la especificación, dada una secuencia de disparo proporcionada por el controlador. Luego, se presenta el diseño del controlador basado en el cálculo de las secuencias de disparo; se presenta un ejemplo donde se muestra el controlador diseñado para el proceso de llenado de botellas. Finalmente, se presenta la teoría

de traducción de RdP a un lenguaje basado en autómatas programables (lenguaje escalera), donde cada transición del lazo cerrado y la secuencia de disparo da como resultado la combinación de condiciones de entrada y bobinas de salida presentes en un brazo de la red.

Capítulo 5: Ejemplo ilustrativo

En el capítulo 5 se implementan a través de un ejemplo ilustrativo la metodología de traducción de la información de un proceso industrial a RdPI y la metodología de captura y recuperación de errores expuestas en los capítulos anteriores. En el ejemplo se implantarán algunos errores de omisión e inconsistencia con el propósito de validar la metodología de captura y recuperación de errores, además de ilustrar la metodología de traducción con un grado mayor de complejidad.

Finalmente, en el capítulo 6 se presentan las conclusiones y trabajos futuros relacionados con este trabajo.

Capítulo 2

Preliminares

En este capítulo se presentan los conceptos básicos relacionados con las Redes de Petri RdP y las Redes de Petri Interpretadas RdPI, las cuales son utilizadas para formalizar y analizar sistemas distribuidos, paralelos o concurrentes; también se introduce el problema de control por regulación utilizado en la construcción de un controlador que permita que el sistema sea igual a la especificación. Finalmente, se presenta la definición de un proceso industrial y los elementos que lo conforman para una adecuada descripción.

2.1. Redes de Petri

Definición 2.1.1 *Una estructura de Red de Petri G es un digrafo bipartido representado por la 4-tupla $G = (P, T, I, O)$ donde:*

- $P = \{p_1, p_2, \dots, p_n\}$ es el conjunto finito de lugares,
- $T = \{t_1, t_2, \dots, t_m\}$ es el conjunto finito de transiciones,
- $P \cup T \neq \emptyset, P \cap T = \emptyset$
- $I : P \times T \rightarrow \mathbb{Z}_{\geq 0}$ es una función que representa el peso de los arcos que van desde lugares a transiciones
- $O : P \times T \rightarrow \mathbb{Z}_{\geq 0}$ es una función que representa el peso de los arcos que van desde transiciones a lugares.

Gráficamente, una estructura de Red de Petri (*RdP*) es ilustrada por cuatro componentes que describen la topología de un sistema distribuido los cuales son dados por círculos que representan los lugares, los rectángulos que ilustran las transiciones y las flechas que representan los arcos de una *RdP* [8, 22]. La Fig. 2.1 ilustra los componentes que forman parte de un sistema Red de Petri.

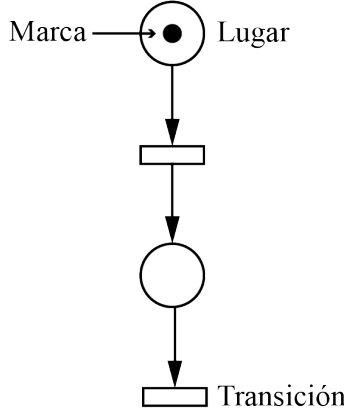


Figura 2.1: Sistema de Red de Petri

La *matriz de incidencia* de una estructura de Red de Petri G es dada por $\mathbf{C} = [c_{ij}]$, donde $\mathbf{C}[ij] = O(p_i, t_j) - I(p_i, t_j)$.

El símbolo $\bullet t_j$ denota el conjunto de todos los lugares p_i tal que $I(p_i, t_j) \neq 0$. El símbolo $\bullet p_i$ denota el conjunto de todas las transiciones t_j tal que $O(p_i, t_j) \neq 0$. Similarmente, $t_j^\bullet$ denota el conjunto de todos los lugares p_i tal que $O(p_i, t_j) \neq 0$, y $p_i^\bullet$ denota el conjunto de todas las transiciones t_j tal que $I(p_i, t_j) \neq 0$. De acuerdo a la estructura de una *RdP*, estas se pueden clasificar como sigue:

- Una máquina de estado (MS) es una *RdP* ordinaria tal que cada transición t tiene exactamente un lugar de entrada y un lugar de salida, es decir,

$$|\bullet t_j| = |t_j^\bullet| = 1, \forall t_j \in T.$$

- Un grafo marcado (GM) es una *RdP* ordinaria tal que cada lugar p tiene exactamente una transición de entrada y una transición de salida, es decir,

$$|\bullet p_i| = |p_i^\bullet| = 1, \forall p_i \in P.$$

- Una red de libre elección (LE) es una RdP ordinaria tal que cada arco desde un lugar p_i es un único arco saliente o un único arco entrante a una transición, es decir,

$$\forall p_i \in P, |p_i \bullet| \leq 1 \text{ o } \bullet(p_i \bullet) = \{p\}.$$

Definición 2.1.2 Dada una estructura de RdP, la distribución de marcado se define como una función $M : P \rightarrow \mathbb{Z}_{\geq 0}$, donde $M(p_i)$ representa el número de marcas o tokens que se encuentran dentro de un lugar p_i (las marcas o tokens se representan como puntos). La distribución de marcado se expresa como un vector columna M de longitud $|P|$, donde $M[i] = M(p_i)$, $\forall p_i \in P$.

Definición 2.1.3 Un sistema de RdP es el par $(G, \mathbf{M}_0)$, donde G es una estructura de RdP y $\mathbf{M}_0$ es una distribución inicial de marcas.

Una transición t_j es habilitada en el marcado $\mathbf{M}_k$ si y solo si $\forall p_i \in \bullet t_j$, $\mathbf{M}_k(p_i) \geq I(p_i, t_j)$, esto es denotado como $\mathbf{M}_k \xrightarrow{t_j}$. Una transición t_j puede dispararse (el evento correspondiente puede ocurrir) si está habilitado, en tal caso, el sistema alcanza un nuevo marcado $\mathbf{M}_{k+1}$ que puede ser calculado como

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{C} \vec{t}_j$$

donde $\vec{t}_j(i) = 0$ para $i \neq j$ y $\vec{t}_j(j) = 1$. Esto es denotado como $\mathbf{M}_k \xrightarrow{t_j}$.

Una secuencia de disparo en el sistema de RdP $(G, \mathbf{M}_0)$ es representada por la secuencia de transiciones $\sigma = t_i t_j, \dots, t_k$ tal que $\mathbf{M}_0 \xrightarrow{t_i} \mathbf{M}_1 \xrightarrow{t_j} \dots, \mathbf{M}_w \xrightarrow{t_k}$.

El vector de Parikh $\vec{\sigma}$ de una secuencia de disparo σ se define como un vector columna de longitud $|T|$ tal que $\vec{\sigma}(j) = k$ si t_j se dispara k veces en la secuencia σ . De esta forma, el marcado $\mathbf{M}'$ se alcanza después de disparar σ desde un marcado $\mathbf{M}$ la cual puede ser calculada con la llamada *ecuación fundamental*

$$\mathbf{M}' = \mathbf{M} + \mathbf{C} \vec{\sigma}$$

Esto se denota como $\mathbf{M} \xrightarrow{\sigma} \mathbf{M}'$, se dice que σ es *disparable* desde $\mathbf{M}$ y $\mathbf{M}'$ se dice que es *alcanzable* desde $\mathbf{M}$. El conjunto de alcanzabilidad de una RdP es el conjunto de todos los marcados alcanzables desde $\mathbf{M}_0$, y se denota como $R(G, \mathbf{M}_0)$. Se dice que un sistema de RdP es segura si $\forall p \in P, \forall M \in R(G, \mathbf{M}_0), M(p) \leq 1$.

2.2. Redes de petri interpretadas

Las Redes de Petri interpretadas $RdPI$ son una extensión a la definición de las RdP , las cuales permiten la representación de señales de entrada y salida para un sistema de eventos discretos [23, 24, 17, 25]. Este tipo de redes permiten establecer propiedades de régimen estacionario o dinámico, lo que significa que el disparo de una transición está condicionada por eventos externos y acciones relacionadas con lugares que afectarán específicamente el cambio de estado del sistema. Por lo tanto, la interpretación en la RdP permite establecer propiedades activas relacionadas con factores externos, estimulando la evolución de la red, que a su vez estimula acciones externas.

Definición 2.2.1 *Un sistema de Red de Petri Interpretada o Red de Petri Interpretada $RdPI$ es la tupla $Q = (G, M_0, \Sigma_I, \Sigma_O, \lambda, \varphi)$ donde:*

- (G, M_0) es un sistema de Red de Petri RdP ;
- $\Sigma_I = \{\alpha_1, \dots, \alpha_s\}$ es el alfabeto de entrada del sistema de RdP , donde cada elemento del conjunto Σ_I es un símbolo de entrada;
- $\lambda : T \rightarrow \Sigma_I \cup \{\epsilon\}$ es la función de etiquetado de la transición con la restricción de no permitir entradas no deterministas, es decir, $\forall t_j, t_k \in T, j \neq k$, si $I(p_i, t_j) = I(p_i, t_k) \neq 0$ y ambos $\lambda(t_j), \lambda(t_k) \neq \epsilon$, entonces $\lambda(t_j) \neq \lambda(t_k)$. Aquí, ϵ representa un evento interno del sistema;
- $\Sigma_O = \{\beta_1, \dots, \beta_q\}$ es el alfabeto de salida del sistema de RdP , donde cada elemento del conjunto Σ_O es un símbolo de salida;
- $\varphi : R(G, M_0) \rightarrow [\mathbb{Z}^+]^q$ es una función de salida, donde q es el número de salidas disponibles.

La función de salida φ asocia los marcados con los vectores de salida. El vector de salida de la $RdPI$ en el marcado M_k está dado por $y_k = \varphi M_k$.

Si $\lambda(t_j) \neq \epsilon$, se dice que la transición t_j es controlable, de otra manera es incontrolable.

Por lo tanto, T se particiona en $T = T_u \dot{\cup} T_c$, donde T_u es el conjunto de las transiciones incontrolables, T_c el conjunto de las transiciones controlables y $\dot{\cup}$ es el símbolo usado para identificar la partición de los conjuntos T_u y T_c . Un símbolo

de entrada $\alpha \in \Sigma_I$ se dice que está *indicado* cuando es activado por un agente externo como un controlador o un usuario.

Si t_i está habilitada y $\lambda(t_i) \neq \epsilon$ se indica, entonces t_i debe ser disparada; si t_i está habilitada y $\lambda(t_i) = \epsilon$ entonces t_i puede ser disparada.

Definición 2.2.2 El conjunto de todas las secuencias de disparo de una RdPI $(Q, \mathbf{M}_0)$ es nombrado el lenguaje de disparo $\mathcal{L}(Q, \mathbf{M}_0) = \{\sigma | \sigma = t_i t_j, \dots, t_k \text{ y } \mathbf{M}_0 \xrightarrow{t_i} \mathbf{M}_1 \xrightarrow{t_j} \dots, \mathbf{M}_w \xrightarrow{t_k} \mathbf{M}_l\} \cup \{\sigma_\epsilon\}$, donde σ_ϵ denota la secuencia vacía (no se dispara ninguna transición).

Definición 2.2.3 Un grafo de alcanzabilidad se define como la representación del cambio en la distribución de marcas en una RdP, la cual se realiza por el disparo de una transición habilitada. En una RdP, un marcado M_i puede ser alcanzable desde un marcado m_0 si existe una secuencia de disparo σ tal que se pueda transformar M_0 en M_i .

A continuación, el Ejemplo 2.2.1 es usado para ilustrar las características de una RdPI como su matriz de incidencia, la matriz de salida, y el vector de salida, las cuales son usados para describir el comportamiento de diversos tipos de sistemas.

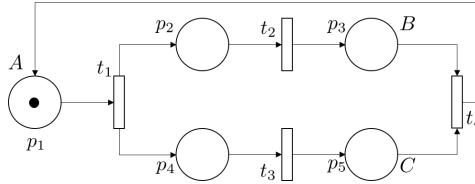


Figura 2.2: Red de Petri con $\Sigma_O = \{A, B, C\}$.

Ejemplo 2.2.1 Considere la RdPI de la Fig. 2.2, la cual está formada por cinco lugares $P = \{p_1, p_2, \dots, p_5\}$ y cuatro transiciones $T = \{t_1, t_2, \dots, t_4\}$. Su matriz de incidencia $\mathbf{C}$ está dada por

$$\mathbf{C} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

Como el alfabeto de salida $\Sigma_O = \{A, B, C\}$, su matriz de salida φ es,

$$\varphi = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 & p_5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} A \\ B \\ C \end{matrix} \quad (2.1)$$

Por ejemplo, para un marcado $\mathbf{M}_2 = [0 \ 1 \ 0 \ 0 \ 1]^T$, el vector de salida resultante es $y_2 = \varphi \mathbf{M}_2 = [0 \ 0 \ 1]^T$.

La Fig. 2.3a ilustra el grafo de alcanzabilidad de la red mostrada en la Fig. 2.2, mientras que la Fig. 2.3b muestra los vectores de salida correspondientes a la evolución del marcado.

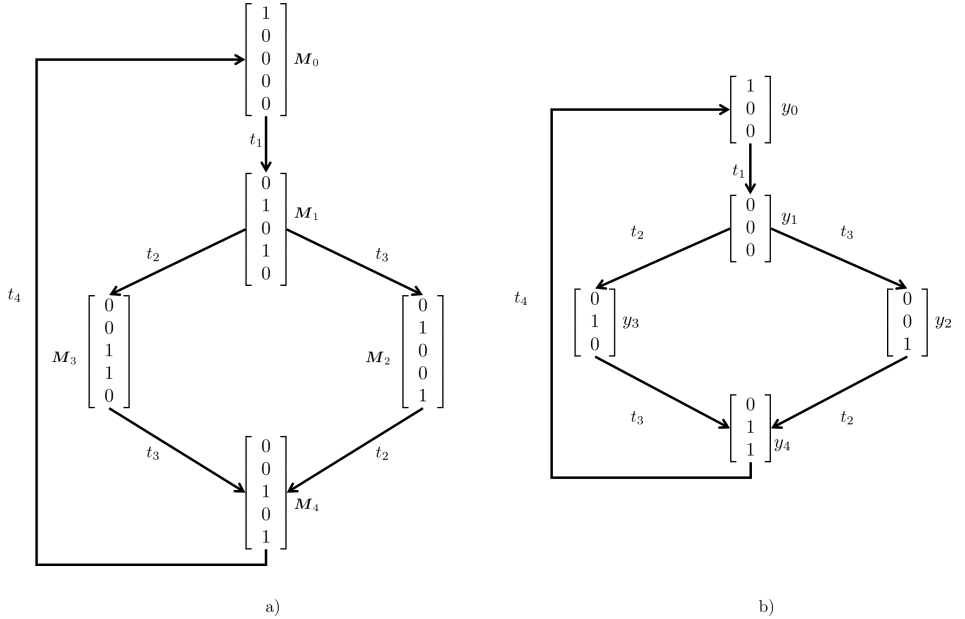


Figura 2.3: a) Grafo de alcanzabilidad de la RdPI para el Ejemplo 2.2.1, b) Vectores de salida correspondientes a la evolución del marcado.

En este trabajo, las RdPI binarias son usadas en la implementación de automatismos debido a que estas redes son las más adecuadas para el modelado

de las características y comportamiento de los elementos que forman parte de un proceso industrial.

2.3. Proceso industrial

Un proceso industrial, también conocido como proceso de fabricación, son todas las formas y pasos que se ponen en marcha para convertir las materias primas en un producto final. En estos pasos o etapas es posible realizar procesos que son totalmente físicos (las especies químicas no cambian) o procesos químicos, donde las especies químicas cambian por medio de una reacción química. En este sentido, los pasos o etapas en los que se producen cambios puramente físicos se denominan operaciones unitarias, y los pasos en los que se produce una reacción química se denominan procesos unitarios. En las plantas químicas, los equipos que forman parte de la operación unitaria son similares entre sí, y debido a esto se analizan operación por operación sin incluir los equipos de cada proceso industrial. Las plantas químicas normalmente son representadas por diagramas de flujo, donde cada operación se presenta por medio de símbolos o rectángulos unidos entre sí por medio de arcos que ilustran las corrientes del proceso que pueden transportar productos al estado sólido, líquido o gaseoso. La Fig. 2.4 muestra un diagrama de flujo sencillo de una planta para la producción de etilenglicol.

Otro diagrama de mayor uso en la industria donde no solo se presentan los equipos y corrientes sino también los elementos de control se denominan Diagramas de Tuberías e Instrumentos **DTI**, también conocida como P&ID por sus siglas en inglés “Piping and Instrumentation Diagram” [26, 27]. Un **DTI** es una representación gráfica que ilustra la interconexión de los equipos, elementos de control (sensores, actuadores, convertidores, temporizadores, etc.) usados en el control de procesos [28, 27, 29], así como las variables que deben ser controladas. Este diagrama utiliza un conjunto de símbolos y nomenclaturas para la identificación de elementos y variables según los estándares emitidos por la Instrumentation Systems and Automation Society (ISA).

La forma correcta de dibujar un DTI se rige por el estándar ANSI ISA S5.1 introducido en 1984 y revisado en 1992 por la Sociedad Internacional de Automatización (ISA), el estándar proporciona un marco de comunicación común para ingenieros y técnicos, operadores, etc. La identificación de un instrumento se realiza teniendo en cuenta la información de la variable que se va a controlar

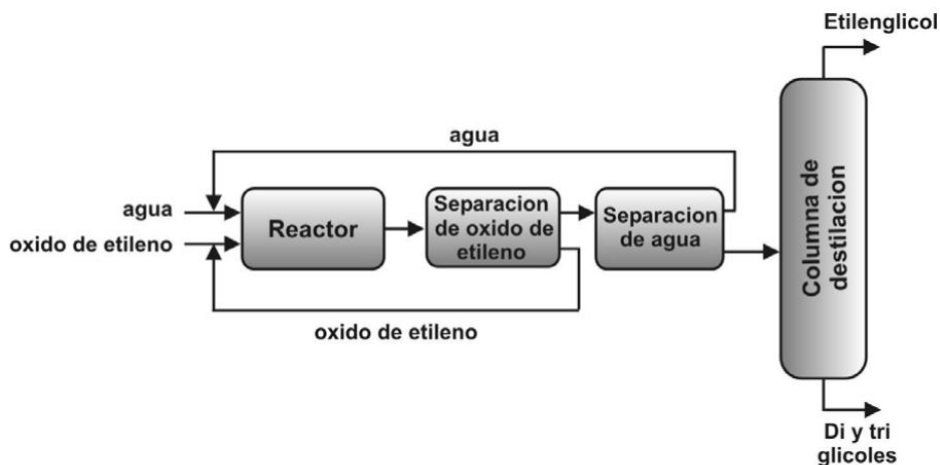


Figura 2.4: Diagrama de flujo para la producción de etilenglicol.

y no con la variable que se va a manipular, es decir, si el nivel de un líquido al interior de un tanque se desea mantener en un valor deseado, se define la variable manipulada como el flujo que entra al tanque, y la variable controlada como el nivel que se desea alcanzar, entonces todos los instrumentos deben pertenecer al lazo de control nivel. Para la representación de los instrumentos de un proceso, la norma proporciona un conjunto de símbolos que representan el tipo de elemento usado en la planta, y algunos de ellos se pueden observar en la Fig. 2.5. Los instrumentos discretos son ilustrados como elementos circulares llamados globos o burbujas (la mayoría de los elementos son representados por esta imagen), los elementos que comparten la visualización y control son ilustrados por elementos encerrados entre un cuadrado, los elementos que realizan una función de cálculo o cómputo son ilustrados por un hexágono, y finalmente los controladores lógicos programables son ilustrados como un rombo dentro de un cuadrado.

Adicionalmente, cada elemento del DTI debe ser etiquetado usando letras y números de acuerdo a la Norma ANSI ISA S5.1, donde la primera letra representa una variable inicial o medida, y las letras sucesivas representan los instrumentos de salida como las válvulas y motores, o instrumentos de función pasiva o de lectura como indicadores, transmisores o registradores.

En la Tabla 2.1, se describe el código alfanumérico utilizado para etiquetar los dispositivos y variables en un DTI.

	Visualización compartido, control compartido		Control lógico programable
	Función de computo		Instrumento discreto
	Motor eléctrico		Válvula

Figura 2.5: Instrumentos generales o símbolos funcionales

Tabla 2.1: Nombre típico de etiqueta

Nombre típico de la etiqueta	
TIC 103	Nombre de la etiqueta
T 103	Identificación del lazo
103	Número del lazo
TIC	Identificación funcional
T	Primera letra
IC	Letras sucesivas

Como ejemplo de identificación de instrumentos, un elemento etiquetado como **LTHL** representa un transmisor de nivel alto y bajo, ya que la primera letra “L” representa la variable inicial o medida **nivel**, la segunda letra “T” representa la función de salida identificada como **transmisor**, y las últimas dos letras “H” y “L” modifican la función de salida definidas como **alto** y **bajo**. La norma ANSI ISA S5.1 se describe a fondo en [26].

En un DTI, los elementos de medición establecen las variables controladas del proceso las cuales son llamadas **variable controladas**, cada variable controlada tiene un valor denominado **estado de la variable**, y el conjunto de los posibles valores es nombrado como el **rango de la variable**. De manera similar, un DTI también está formado por elementos finales de control o actuadores los cuales tienen diferentes condiciones o **estados del actuador**, y el conjunto de los posibles estados del actuador definen el **rango del actuador**. En este trabajo, las variables controladas y los actuadores en adelante serán llamados **elementos**

del proceso, y la relación entre ellos definen el **lazo de control**.

Además del diagrama DTI, una narrativa del proceso y una narrativa de operación son usadas para completar su descripción, las cuales representan el funcionamiento y su comportamiento, respectivamente. La **narrativa del proceso** es una descripción de la funcionalidad del proceso dado en lenguaje natural por un experto conocedor del proceso. Esta narrativa describe la posible evolución de cada variable controlada en función de los estados de las variables y los estados de los actuadores. Las variables y actuadores interactúan de dos formas diferentes y se encuentran directamente implicadas en la narrativa del proceso, la primera hace referencia a la ocurrencia simultánea del cambio de estados en las variables y los actuadores la cual es representada por una relación sincrónica [10, 30]. La segunda es la relación permisiva que representa la habilitación de un cambio de estado en la variable cuando algún actuador está en un estado particular [10]. La **narrativa de operación** es un conjunto de oraciones que describe el comportamiento requerido del sistema. En particular, esta narrativa indica las secuencias ordenadas de valores de los estado que los actuadores y las variables deben alcanzar.

El Ejemplo 2.3.1 ilustra la descripción de un proceso industrial denominado *sistema de llenado de botellas*. Este proceso es descrito usando un DTI, la narrativa de proceso que describe la característica de los elementos y la narrativa de operación que describe el funcionamiento del proceso.

Ejemplo 2.3.1 *Sistema de llenado de botellas. La Fig. 2.6 ilustra un sistema de llenado de botellas el cual está conformado por dos sub-procesos. El primer sub-proceso es el transportador de botellas (C-1) el cual tiene tres sensores que determinan la posición de la botella sobre la banda transportadora, un motor que tiene la función de transportar la botella desde la posición inicial a la posición final, y un robot con su pinza que retira la botella de la banda cuando esta se encuentra en la posición 3. El segundo sub-proceso es el llenado de botellas (T1), este sub-proceso está conformado por un sensor de nivel alto que determina la posición alta del líquido en la botella, y una válvula de llenado que vierte el líquido en la botella. Cada sensor en el proceso define la variable controlada del sistema, así, los sensores de posición 1, 2 y 3 definen la variable controlada **posición de la botella 1, posición de la botella 2, y posición de la botella 3**, respectivamente; de manera similar, el sensor de nivel establece la variable controlada **nivel de botella**.*

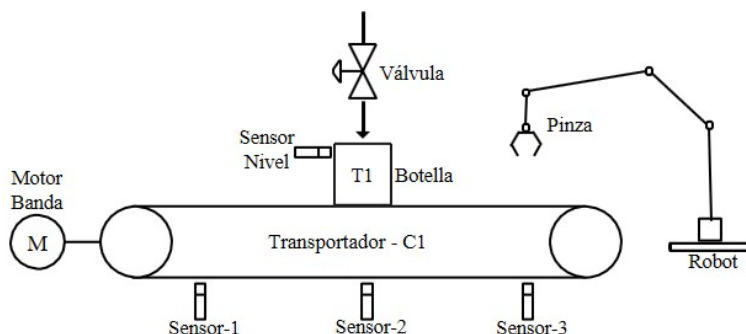


Figura 2.6: Sistema de llenado de botellas.

El diagrama de tuberías e instrumentación (DTI) del proceso de llenado de botellas se ilustra en la Fig. 2.7. En este DTI, cada elemento del sistema es etiquetado basado en la Norma ANSI ISA S5.1. Así, la válvula LV-104 y el transmisor de nivel LT-104 pertenecen al lazo de control 104 definido como **nivel de la botella**; los transmisores de posición ZT-1-103, ZT-2-103 y ZT-3-103, el motor ZZ-103, la pinza ZZC-103 y el robot ZN-103 pertenecen al lazo de control 103 definido como **posición de la botella**. Como se observa en la figura, el autómata programable YIC-1 controla todo el proceso, además de utilizar señales eléctricas para ilustrar la conexión de los elementos.

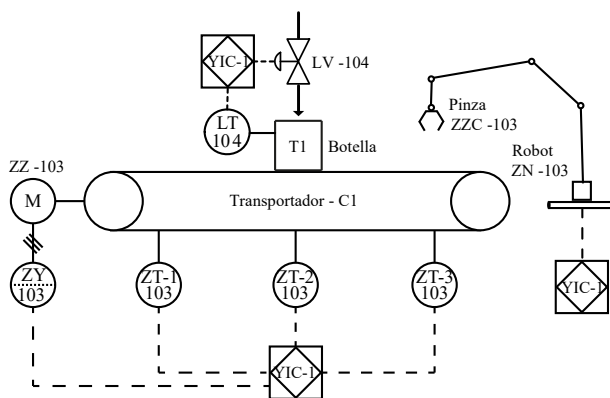


Figura 2.7: DTI Sistema de llenado de botellas.

A continuación se presenta la narrativa de proceso y operación del sistema de llenado de botellas donde se describen las características y relación de los elementos y el funcionamiento del sistema.

Narrativa del proceso. El proceso de llenado de botellas está formada por dos sub-procesos denominados transporte de botellas y llenado de botellas. Según el DTI del sistema, el sub-proceso de transporte de botellas está formado por tres transmisores de posición enumerados como $ZT - 1$, $ZT - 2$, y $ZT - 3$, un motor $ZZ - 103$ que acciona la banda transportadora para el transporte de botellas, un brazo robótico $ZN - 103$ con una pinza $ZZC - 103$ que vacía el transportador de la botella cuando se encuentra en la posición 3. Cada transmisor de posición en la banda transportadora define una variable controlada establecidas como posición de la botella 1 BP_1 , posición de la botella 2 BP_2 y posición de la botella 3 BP_3 , cada una de estas variables controladas pueden estar en dos estados (presencia y ausencia) definiendo así los estados para la posición de la botella 1 como BP_1^p y BP_1^a , para la posición de la botella 2 BP_2^p y BP_2^a , y para la posición de la botella 3 BP_3^p y BP_3^a , donde a y b significan ausencia y presencia, respectivamente. El motor $ZZ - 103$ tiene dos estados definidos como encendido y apagado, cuando el motor $ZZ - 103$ esta en el estado encendido, la banda transportadora transfiere las botellas de la posición 1 a la posición 2 y de la posición 2 a la posición 3, cabe mencionar que al final de la banda transportadora existe una barrera física para que las botellas no caigan del transportador; cuando el motor $ZZ - 103$ esta en el estado apagado, la banda transportadora no transfiere botellas. El brazo robótico está programado para alcanzar las posiciones R_H (robot en posición inicial), RP_1 (posición del robot 1 sobre $ZT - 3$) y RP_2 (posición del robot 2 sobre la salida), y La pinza $ZZC - 103$ puede estar en los estados abierto o cerrado. La función del conjunto brazo robótico y pinza es el de liberar botellas la banda transportadora, cuando alguna de ellas se encuentre en la posición 3 BP_3^p .

El sub-proceso de llenado de botellas está formado por un transmisor de nivel ($LT - 104$) y una válvula ($LV - 104$). El transmisor de nivel $LT - 104$ define la variable controlada **nivel de botella** (BL), esta variable controlada puede estar en dos estados **vacío** (BL^e) o **lleno** (BL^f), donde e y f significan empty y full por sus siglas en ingles. La válvula $LV - 104$ tiene dos estados definidos como abierto y cerrado, si la válvula $LV - 104$ está en el estado abierto, entonces los estados de BL cambian de BL^e a BL^f . Los procesos de transporte y llenado de botellas son controlados por un PLC ubicado en campo como lo representa el

DTI.

Narrativa de operación. La narrativa de operación establece las condiciones iniciales de las variables del proceso, además de establecer la secuencia de operación que determina lo que el proceso debe hacer. Las condiciones iniciales de las variables posición de la botella 1, posición de la botella 2 y posición de la botella 3 son BP_1^a , BP_2^a y BP_3^a respectivamente, indicando que la banda transportadora está vacía y no existe alguna botella en las posiciones 1, 2 y 3. La condición inicial de la variable nivel de botella BL es BL^e , indicando que la botella está en un estado vacío. La narrativa de operación viene dada por tareas que debe realizar el proceso las cuales son definidas por el ingeniero de planta. El sistema de llenado de botellas debe realizar las siguientes tareas:

- **Llenar botella.** Se debe llenar una botella (BL^f) cuando se encuentre en la posición 2 del transportador (BP_2^p). Si no hay una botella en la posición 3 (BP_3^a), la botella llena (BL^f) sale de la posición 2 (BP_2^a) y en ese mismo instante la variable nivel de botella cambia a vacío (BL^e).
- **Descargar botella.** Se desea descargar una botella desde la posición 3 del transportador (BP_3^a). si hay una botella en la posición 3 (BP_3^p) entonces el robot se dirige a la posición 1 con la pinza abierta (RP_1 , G_o), cuando el robot llega a la posición 1 se cierra la pinza (RP_1 , G_c), luego el robot se dirige a la posición 2 con la pinza aún cerrada (RP_2 , G_c), cuando el robot llega a la posición 2 se abre la pinza (RP_2 , G_o) y finalmente el robot se dirige a la posición inicial con la pinza aún abierta (R_H , G_o).

La información dada por el DTI, la narrativa del proceso, y la narrativa de operación del proceso presentada en esta descripción, es usada en el diseño de la metodología de traducción, detección y corrección de errores, las cuales se presentan en capítulos posteriores.

Capítulo 3

Metodología de traducción, detección y recuperación de errores

Este capítulo describe en detalle las etapas de la metodología de traducción, detección y recuperación de errores, donde la información de un proceso industrial dada por un DTI, la narrativa de proceso y operación se traduce a Redes de Petri Interpretadas RdPI que representan el comportamiento de la planta y la especificación del proceso. Además, se presenta una metodología de detección y recuperación de errores para eliminar automáticamente los errores de inconsistencia y omisión introducidos en las narraciones. Como primer paso, la información de un proceso (sistema neumático, proceso industrial, o sistema de manufactura) dada por el DTI, la narrativa del proceso y la narrativa de operación, es estructurada en forma de tablas con el propósito de obtener la información que representa las características de los elementos y el funcionamiento del sistema. Luego, la estructura de tablas es usada para proponer una estructura basada en grafos que representan la cantidad de elementos presentes en el sistema y la relación que existe entre ellos. Después, con la información de las tablas y la estructura de grafos se proponen modelos de RdPI que describen el proceso y el comportamiento requerido del sistema dado por la especificación. Ahora, la metodología de detección y recuperación de errores es usada para analizar la información de la estructura de tablas y detectar errores como omisiones e

inconsistencias en la información. Así, si un error es detectado en la estructura de tablas, la información es actualizada y los errores son removidos.

3.1. Traducción de la información de un proceso industrial a RdPI

Los procesos industriales y sus operaciones unitarias usualmente son controladas siguiendo una metodología típica como se muestra en la Fig. 3.1 (a), esta metodología se realiza teniendo en cuenta la experiencia del operador de planta, además de realizar una validación de la lógica de programación implantada en el PLC, la cual es realizada mediante una técnica de prueba y error. La descripción del sistema como parte inicial de la metodología tradicional, se realiza mediante la recolección de información, donde se encuentra el listado de los dispositivos que intervienen en el proceso, variables a medir, variables a controlar, rangos de operación, función de los dispositivos, etc. Esta actividad se logra a través de entrevistas con operadores y personal de mantenimiento de operaciones, visitas de campo y experiencia de integración. El siguiente paso es la representación gráfica del proceso mediante diagramas de flujo o DTIs, y apoyado con las narrativas de proceso y operación se determina el funcionamiento y descripción de los elementos, lo que permite conocer en detalle el sistema y las funciones de los dispositivos. Luego se realiza la programación del autómata programable donde existen dos formas: el método heurístico o informal, en el cual la experiencia del ingeniero es esencial para lograr un programa adecuado; y el método formal, donde se usa un gráfico de etapas y transiciones denominado GRAFCET [31], donde se indican las secuencias de operaciones y las condiciones lógicas para pasar de una etapa a la otra. Esta metodología resulta lenta y tediosa en la depuración de la información, y finalmente el resultado no garantiza que se encuentre libre de errores debido a que no son visibles en la depuración y estos aparecen en la operación. La metodología de traducción presentada en esta sección muestra una estrategia modular incremental como se ilustra en la Fig. 3.1 (b), donde la información del proceso se traduce en modelos de RdPI que representan el comportamiento de los elementos y funcionamiento del proceso. En comparación con la metodología tradicional, las etapas mencionadas en la metodología de traducción de la Fig. 3.1 (b) se realizan de forma automática, y se adaptan a cualquier proceso industrial, sistema de fabricación o sistema

electro-neumático, además de incluir la detección y eliminación de errores, lo que lleva a modelos libres de errores.

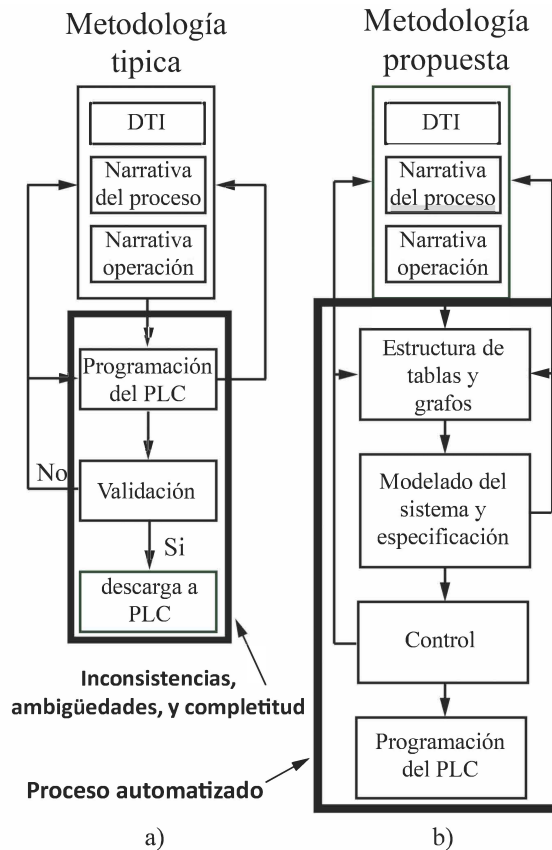


Figura 3.1: Metodologías para el diseño de lógica de programación basada en autómatas programables. a) Metodología típica, basada en la experiencia y conocimiento del proceso. b) Metodología propuesta, basada en formalismos matemáticos y análisis de errores.

A continuación se detalla cada etapa de la metodología de traducción propuesta, las cuales son aplicadas al proceso de llenado de botellas del Ejemplo 2.3.1, donde se realiza una descripción dada por el DTI del proceso y sus narrativas respectivas.

3.1.1. Estructura de tablas del sistema

La estructura de tablas del sistema presenta en forma estructurada la información del proceso, y representa las características y funcionamiento del proceso. La estructura de tablas se divide en dos partes, la *estructura de la narrativa del proceso (SNP)* y la *estructura de la narrativa de operación (SNO)*. La SNP captura la información dada por el DTI y la narrativa del proceso, resumiendo las características de los elementos y la relación que existe entre ellos. La *estructura de la narrativa del operación (SNO)* resume en tablas los requerimientos de los diferentes sub-procesos y las tareas que debe cumplir el sistema para su óptimo funcionamiento.

Estructura de la narrativa del proceso (SNP).

La **SNP** describe las características y comportamiento del sistema a través de una estructura de tablas, la cual es alimentada por la información del proceso dada por un diagrama de tuberías e instrumentos DTI y una narrativa del proceso (NP) presentada en forma de lenguaje natural por el ingeniero del proceso.

La SNP está formada por cuatro tablas definidas como: *Tabla descripción de elementos*, *Tabla comportamiento de elementos*, *Tabla relación permisiva*, y la *Tabla relación sincrónica*.

La **tabla descripción de elementos** lista los elementos del proceso (variables controladas y actuadores) y sus características más importantes como el rango del elemento, estados que lo conforman y condición inicial del elemento. La distribución de la tabla descripción se muestra en la Tabla 3.1

Tabla 3.1: Descripción de elementos del proceso.

Elemento del proceso	Rango del elemento	Estados del elemento	Estado inicial del elemento

En esta tabla, la primera columna corresponde al nombre del elemento, el cual es dado por su identificación funcional presentada en el DTI; la segunda columna pertenece al rango del elemento que representa el conjunto de valores reales dado por el número de estados del elemento, así, si un elemento tiene los estados abierto y cerrado, el rango del elemento es 2; la tercera columna corresponde a los estados del elemento que representa las diferentes situaciones

que un actuador o variable controlada puede encontrarse; finalmente, la cuarta columna estado inicial del elemento representa la condición del elemento en su estado inicial. Las características del elemento de las columnas 1, 2, 3 y 4 son capturadas de la información presentada en el DTI y en la narrativa de proceso dada en forma de lenguaje natural por el ingeniero del proceso.

La **tabla comportamiento de elementos** ilustra el funcionamiento de los elementos (actuadores y variables controladas) dado en la narrativa del proceso por el ingeniero de planta. La Tabla 3.2 esta conformada por subtablas que son equivalentes al número de elementos listado en la tabla de descripción de elementos. El comportamiento de un elemento está definido como el cambio desde un estado “A” a un estado “B”, lo cual caracteriza la evolución del elemento. El número de filas y columnas en cada subtabla es equivalente al número de estados del elemento.

Tabla 3.2: Comportamiento de elementos del proceso.

Elemento 1			Elemento 2			Elemento 3		

.....

La **tabla relación permisiva** ilustra la relación entre elementos de un proceso industrial, la cual representa la acción que ejerce un actuador sobre una variable controlada. Esta acción es presentada como el cambio de un estado a otro en una variable controlada cuando el cambio de estado de un actuador permite la evolución de la variable controlada.

Tabla 3.3: Relación permisiva para el ejemplo de llenado de líquidos.

		Variables controladas			
		1	2	3	...
Actuador	Estados del actuador	Eventos			
1					

La Tabla 3.3 está formada por filas y columnas que se relacionan entre si teniendo en cuenta la acción que ejerce el actuador sobre la variable controlada, esta información es dada por el ingeniero del proceso en forma de lenguaje natural en la narrativa del proceso. El número de filas es equivalente al número de actuadores y el número de columnas es equivalente al número de variables

controladas del proceso. Además, cada sub-fila corresponde al número de estados del actuador.

La **tabla relación síncrona** presenta la relación entre eventos (cambio de un estado a otro estado) de variables controladas de un proceso industrial, los cuales se identifican cuando dos eventos se ejecutan en forma simultánea y tienen el mismo significado físico.

Tabla 3.4: Relación síncrona.

Variables y eventos		1	
		EV1	EV2
2	EV1		
	EV2		

La Tabla 3.4 ilustra la estructura de la tabla de relación síncrona, donde tanto las filas como las columnas (1 y 2) representan los variables controladas del proceso, y las sub-filas y sub-columnas los eventos de cada variable (EV1 y EV2).

Metodología de llenado de tablas para la SNP. Como se mencionó en la sección anterior, la SNP está formada por la tabla de descripción de elementos, la tabla comportamiento de elementos, la Tabla relación permisiva, y la Tabla relación síncrona, las cuales presentan en forma estructurada la información del proceso. A continuación se presenta el llenado de cada tabla que representa la SNP.

La *tabla de descripción de elementos* presenta las características de los elementos del proceso. Las filas de la primera columna (**elemento del proceso**) se llenan con la etiqueta de cada elemento del proceso (una fila por cada elemento), la cual se define con la identificación funcional dada por la Norma ANSI ISA S5.1, así, un elemento encargado de realizar el llenado de líquido al interior de un depósito puede identificarse con la etiqueta **LCV-001** que representa a una válvula controladora de nivel del lazo 001; la segunda columna se llena con el rango del elemento que en este documento se describe como el número de estados que tiene el elemento del proceso, así, si la LCV-001 tiene los estados abierto y cerrado, el elemento es de rango **2**; la columna estados del elemento se llena con el conjunto de valores que cada elemento puede tener, es decir, si una válvula de control del tipo discreto permite o restringe el paso de un fluido mediante la acción de abrir o cerrar, este elemento tiene los estados **abierto** y **cerrado**; finalmente, la columna

estado inicial del elemento se llena con la condición o posición inicial que el elemento debe tener al ejecutar el proceso, por lo tanto, si la válvula de control LCV-001 debe iniciar cerrada, entonces su estado inicial es **cerrado**.

La tabla descripción de elementos para el sistema de llenado de botellas del Ejemplo 2.3.1 se muestra en la Tabla 3.5. En esta tabla, las características de cada elemento presentadas en el DTI de la Fig. 2.7 y la narrativa de proceso presentada en la sección 2.3.1 son resumidas con la información más relevante de cada elemento.

Tabla 3.5: Descripción de elementos para el proceso de llenado de botellas.

Elemento del proceso	Rango	Estados	Estado Inicial
ZZ-103(Motor)	2	off, on	off
ZN-103 (Robot)	3	home (R_H), RP_1 , RP_2	R_H
ZZC-103 (Pinza)	2	abierto (G_o), cerrado (G_c)	G_o
LV-104 (Válvula)	2	abierto, cerrado	cerrado
$BP_1 - 103$ (Posición Botella-1)	2	ausencia (BP_1^a), presencia (BP_1^p)	ausencia (BP_1^a)
$BP_2 - 103$ (Posición Botella-2)	2	ausencia (BP_2^a), presencia (BP_2^p)	ausencia (BP_2^a)
$BP_3 - 103$ (Posición Botella-3)	2	ausencia (BP_3^a), presencia (BP_3^p)	ausencia (BP_3^a)
Nivel de la Botella (BL-104)	2	vacío (BL^e), lleno (BL^f)	vacío (BL^e)

Como se muestra en la Tabla 3.5, existen 8 filas en cada columna, ya que 8 elementos son capturados en la información de la narrativa del proceso de llenado de botellas. El elemento etiquetado como ZN-103 es de rango 3 debido a que tiene 3 estados definidos como R_H , RP_1 y RP_2 , los cuales son descritos en la narrativa del proceso como los estados que debe alcanzar el robot ZN-1003. Además, su estado inicial R_H es capturado en la Tabla 3.5, ya que al inicio del proceso el robot debe encontrarse en la posición R_H .

La *tabla comportamiento de elementos* representa el comportamiento que deben tener los elementos del proceso, los cuales son establecidos en la narrativa del proceso. El comportamiento de un elemento está definido como el cambio de un estado “A” a un estado “B”, permitiendo la evolución del elemento. Cada sub-tabla de la SNP es equivalente a un elemento listado en la tabla de descripción de elementos. En cada sub-tabla, las filas y columnas corresponden a los estados del elemento, los cuales son listado en la tabla descripción del elemento. Para cada elemento del proceso, una marca “✓” se dibuja en la posición de la fila estado-1 y la columna estado-2 si el elemento evoluciona o cambia desde el estado-1 al

estado-2 en un solo paso sin visitar otro estado.

La Tabla 3.6 ilustra el comportamiento de los diferentes elementos del proceso del Ejemplo 2.3.1. Como se observa, cada sub-tabla de la Tabla 3.6 es asociada a cada uno de los elementos del proceso, a su vez, las sub-filas y sub-columnas representan los estados de cada elemento, y la “✓” simboliza el cambio de estado del elemento. Para ilustrar el procedimiento de llenado de la Tabla 3.6 el elemento $ZZ - 103$ del Ejemplo 2.3.1 es usado para dicho fin. La posición [2,3] (fila-2 y columna-3) de la Tabla 3.5 establece los estados que conforman el elemento $ZZ - 103$ lo cuales son definidos como *off* y *on*. Teniendo en cuenta esta información una sub-tabla se forma con tres filas y tres columnas, donde la sub-fila 1 y sub-columna 1 se llena con la etiqueta del elemento $ZZ - 103$ y sus respectivos estados. Ahora, la información dada en la narrativa del proceso indica que el motor $ZZ - 103$ debe pasar del estado **off** al estado **on**, y del estado **on** al estado **off** en un solo paso, por lo tanto, una marca “✓” se dibuja en el estado contrario al analizado, así, el cambio de estado de *off* a *on* se representa con una marca “✓” en la sub-fila 2 sub-columna 3. Este cambio de estado en los elementos del proceso son denominados **eventos**. El comportamiento de los demás elementos se pueden observar en la Tabla 3.6, donde cada elemento del proceso de llenado de botellas del Ejemplo 2.3.1 es ilustrado por tablas individuales.

Tabla 3.6: Comportamiento de elementos para el ejemplo de llenado de líquidos.

ZZ-103	off	on
off		✓
on	✓	

ZZC-103	abierto	cerrado
abierto		✓
cerrado	✓	

LV-104	abierto	cerrado
abierto		✓
cerrado	✓	

BP₁ - 103	BP_1^a	BP_1^p
BP_1^a		✓
BP_1^p	✓	

BP₂ - 103	BP_2^a	BP_2^p
BP_2^a		✓
BP_2^p	✓	

BP₃ - 103	BP_3^a	BP_3^p
BP_3^a		✓
BP_3^p	✓	

BL-104	BL^e	BL^f
BL^e		✓
BL^f	✓	

ZN-103	RP_1	R_H	RP_2
RP_1		✓	
R_H	✓		✓
RP_2		✓	

La *tabla relación permisiva* representa la interacción entre los elementos (actuadores y variables controladas) de un proceso industrial. Esta interacción se presenta mediante la acción que ejerce un actuador sobre una variable controlada mediante el cambio de estados (eventos) de variables controladas cuando algún(os) actuador(es) se encuentran en un estado en particular. Al igual

que la tabla comportamiento de elementos, la elaboración de la Tabla *relación permisiva* se basa en la información dada por la tabla descripción de elementos y la narrativa del proceso suministrada por el operario de planta. En la tabla *relación permisiva* se define una fila por cada actuador relacionado en la tabla descripción de elementos, y a su vez, esta fila se divide en sub-filas de acuerdo al número de estados del actuador. Por otro lado, se define una columna por cada variable controlada relacionada en la tabla descripción de elementos, y a su vez, cada columna se divide en sub-columnas de acuerdo al número de eventos declarados para la variable controlada en la tabla comportamiento de elementos. Así, si una variable controlada V_i evoluciona del estado S_1 al estado S_2 , se requiere que un actuador A_j se encuentre en el estado S_i , entonces se debe escribir en la columna asociada a la variable V_i y en la sub-fila del estado S_i del actuador A_j la leyenda “ S_1 a S_2 ”. La Tabla 3.7 ilustra la relación permisiva entre actuadores y variables controladas para el proceso de llenado de líquidos del Ejemplo 2.3.1. En esta tabla se observa que existe una relación entre el motor ZZ-103 y las variables controladas $BP_1 - 103$, $BP_2 - 103$ y $BP_3 - 103$, ya que si un objeto se encuentra sobre la posición $BP_1 - 103$ de la banda transportadora y el motor ZZ-103 cambia su estado de *off* a *on*, un evento se ejecuta en la variable controlada $BP_1 - 103$ identificado como BP_1^p a BP_1^a , de la misma forma en la variable controlada $BP_2 - 103$ se ejecuta un evento cuando un objeto es detectado, el cual cambia su estado desde BP_2^a a BP_2^p y luego desde BP_2^p a BP_2^a , finalmente el objeto llega a la posición $BP_2 - 103$ donde se ejecuta el evento BP_3^a a BP_3^p .

Tabla 3.7: Relación permisiva proceso llenado de botellas.

		Variables controladas			
		$BP_1 - 103$	$BP_2 - 103$	$BP_3 - 103$	$BL - 104$
Actuador	Estado del actuador	Eventos			
ZZ-103	off	-	-	-	-
	on	BP_1^p a BP_1^a	BP_2^a a BP_2^p BP_2^p a BP_2^a	BP_3^a a BP_3^p	-
ZN-103	RP_1	-	-	BP_3^p to BP_3^a	-
	R_H	-	-	-	-
	RP_2	-	-	-	-
ZCC-103	cerrado	-	-	BP_3^p a BP_3^a	-
	abierto	-	-	-	-
LV-104	cerrado	-	-	-	-
	abierto	-	-	-	BL^e a BL^f

Finalmente, la *tabla relación síncrona* presenta la interacción entre dos o más variables controladas cuando cambian simultáneamente sus estados (eventos) representando el mismo significado físico. Esta interacción entre variables controladas se muestra en la Tabla *Relación síncrona* con una marca “X” en la celda que relaciona los dos eventos, es decir, cuando un evento $event_i$ de una sub-fila de una variable controlada V_1 y un evento $event_j$ de una sub-columna de una variable controlada V_2 cambian simultáneamente y dependen una de la otra (eventos sincronizados), se sitúa una marca “X” en la celda donde se relacionan dichas variables. Para ilustrar el llenado de la *tabla relación síncrona* se utiliza el sub-proceso de transporte de latas del Ejemplo 2.3.1, el cual tiene como objetivo transportar latas desde $BP_1 - 103$ hasta $BP_3 - 103$. Como se mencionó en la narrativa del proceso, el sub-proceso de transporte de latas está conformado por tres transmisores de posición representados por ZT-1-103, ZT-2-103 y ZT-3-103 los cuales son los encargados de determinar la posición de la lata sobre la banda transportadora; el motor ZZ-103 permite el cambio de posición de la lata BP_1 , BP_2 , y BP_3 sobre el transportador. En este sub-proceso, si una lata se transporta desde BP_1 a BP_2 en un solo paso, significa que existe un evento en el cambio de estados de la variable controlada BP_1 el cual pasa de BP_{1p} a BP_{1a} ; en ese mismo instante ocurre un evento en la variable controlada BP_2 , la cual pasa de BP_{2a} a BP_{2p} , a este cambio simultaneo entre variables controladas es llamado sincronización. Así, en la *tabla relación síncrona* la sub-fila correspondiente al evento BP_{1p} a BP_{1a} y la sub-columna del evento BP_{2a} a BP_{2p} están sincronizadas, por lo tanto una marca “X” se dibuja en la celda donde estos eventos se cruzan. La *tabla relación síncrona* para el Ejemplo 2.3.1 del proceso de llenado de botellas se muestra en la Tabla 3.8.

Tabla 3.8: Relación síncrona proceso llenado de botellas.

Variables and Events	$BP_1 - 103$		$BP_2 - 103$		$BP_3 - 103$		$BL - 104$	
	BP_1^a a BP_1^p	BP_1^p a BP_1^a	BP_2^a a BP_2^p	BP_2^p a BP_2^a	BP_3^a a BP_3^p	BP_3^p a BP_3^a	BL^c a BL^j	BL^j a BL^c
$BP_1 - 103$	BP_1^a a BP_1^p BP_1^p a BP_1^a		X					
$BP_2 - 103$	BP_2^a a BP_2^p BP_2^p a BP_2^a	X			X			X
$BP_3 - 103$	BP_3^a a BP_3^p BP_3^p a BP_3^a			X				X
$BL - 104$	BL^c a BL^j BL^j a BL^c			X	X			

Estructura de la narrativa de operación (SNO).

La **SNO** describe las características de las variables controladas y la secuencia de operación o tareas que debe cumplir el proceso. Esta información se resume en tablas la cual es dada por el DTI y la narrativa de operación del proceso. La SNO está formada por 2 tablas definidas como: *tabla rango de operación* y la *tabla condiciones de operación*.

La **Tabla rangos de operación** enumera las características de las variables controladas relacionadas a su rango de operación, las cuales corresponden a los diferentes estados de la variable controlada tomados desde la columna estados del elemento en la tabla *descripción de elementos*. La distribución de filas y columnas de la tabla rangos de operación se muestra en la Tabla 3.9.

Tabla 3.9: Rangos de operación.

Elementos	Rangos de operación

En esta tabla, la primera columna corresponde al nombre de la variable controlada usando la Norma ANSI ISA S5.1, y la segunda columna corresponde a los rangos del elemento establecidos por los estados que lo conforman. Esta información se captura desde el DTI y la narrativa de operación dada por el ingeniero del proceso en forma de lenguaje natural.

La **tabla condiciones de operación** describe la secuencia de operaciones que un proceso debe realizar. La primera columna representa las tareas que debe realizar el proceso; la segunda columna muestra los estados de pre-condición del proceso, los cuales son definidos como el conjunto de estados previos al estado(s) que se desea alcanzar; la tercera columna presenta la(s) condición(es) para pasar desde un estado previo a estado(s) que se desea(n) alcanzar; finalmente, la columna estados alcanzados presenta el conjunto de estados que el proceso debe alcanzar para dar cumplimiento a la tarea. La estructura de la tabla *condiciones de operación* se muestra en la Tabla 3.10.

Tabla 3.10: Condiciones de operación.

Tareas	Estados de precondición	Condiciones	Estados alcanzados

En esta tabla, las filas corresponden a las tareas que cada subproceso debe realizar, las subfilas corresponde a los estados y condiciones que se deben cumplir para que una tarea se realice, y las columnas representan a los estados y condiciones que se deben ejecutar para el cumplimiento de las tareas.

Metodología de llenado de tablas para la SNO. La SNO está formada por la tabla rango de operación y la tabla condiciones de operación, las cuales representan las características y la secuencia de operaciones que el proceso debe realizar para cumplir con una tarea específica.

La **tabla rango de operación** presenta en forma estructurada las características de las variables controladas, las cuales son representadas por la etiqueta de las variables controladas y los estados que la variable controlada puede alcanzar. En esta tabla, las filas de la columna *elementos* se llenan con las etiquetas de las variables controladas del proceso, las cuales son capturadas desde la tabla descripción de elementos (una fila por cada variable controlada del proceso). Ahora, las filas de la columna *rangos de operación* se llenan con los estados que cada variable controlada debe alcanzar, los cuales son capturados desde la columna estados del elemento en la tabla descripción de elementos. La *tabla rangos de operación* para el proceso de llenado de botellas del Ejemplo 2.3.1 se muestra en la Tabla 3.11, donde el elemento ZN-103 que representa al brazo robótico y sus rangos de operación R_H , RP_1 , RP_2 son capturados de la información dada por la Tabla 3.5.

La **tabla condiciones de operación** presenta la secuencia de operación que debe realizar un proceso para el cumplimiento de sus tareas. En esta tabla, la columna *tareas* se llena con la lista de actividades que debe realizar cada subproceso, las cuales se capturan desde la narrativa del control, por ejemplo, transportar, llenar, vaciar, calentar, etc. La columna *estados de precondición* se llena con el conjunto de estados previos a los estados alcanzados en cada etapa de evolución de la variable controlada (estados iniciales de variables

Tabla 3.11: Rangos de operación.

Elementos	Rangos de operación
ZN-103	R_H, RP_1, RP_2
ZZC-103	G_o, G_c
$BP_1 - 103$	BP_1^a, BP_1^p
$BP_2 - 103$	BP_2^a, BP_2^p
$BP_3 - 103$	BP_3^a, BP_3^p
$BL - 104$	BL^e, BL^f

controladas). En esta columna, la primera fila se llena con los estados iniciales de variables controladas implicadas en una tarea específica, la segunda fila se llena con el conjunto de estados alcanzados de la primera etapa de evolución de la variable controlada, así, si un ascensor de tres pisos tiene como variables controladas P_1 , P_2 y P_3 , representando la posición del ascensor en el piso 1, 2 y 3 respectivamente, entonces la primera fila se llena con el estado inicial P_1 , la segunda fila con el estado alcanzado en la primera etapa P_2 (cambiando ahora a estado inicial) y la fila 3 se llena con el estado alcanzado P_3 correspondiente al estado alcanzado en la segunda etapa. La tercera columna se llena con las condiciones (guardas) que se deben cumplir para que un estado de precondition evolucione a un estado alcanzado (estado que debe alcanzar la variable del proceso). Siguiendo el ejemplo del ascensor de tres pisos, si este requiere ir de P_1 a P_2 mediante la acción de un pulsador S_1 , en la columna 3 se escribe la leyenda S_1 indicando que es la condición para pasar de P_1 a P_2 . La columna 4 se llena con los estados alcanzados en cada etapa de la evolución de la variable controlada, así, si el ascensor se encuentra en P_1 y desea desplazarse a P_2 , este último se define como el estado alcanzado y se digita en la fila correspondiente de la columna estados alcanzados. Las operaciones para cada tarea se ordenan desde la parte superior de la fila correspondiente a cada tarea, hasta su respectiva fila inferior. Para ilustrar el llenado de la Tabla 3.12, la tarea de descargar la botella del transportador es usada como ejemplo. En esta tarea se desea que el robot retire el envase lleno desde la posición 3 de la banda transportadora y luego se dirija a su destino final. Para que se cumpla esta tarea, se debe llevar el robot desde la posición R_H con la pinza abierta G_o hasta la posición RP_1 , luego cerrar la pinza para tomar el botella, y posteriormente dirigirse a la posición RP_2 y abrir la pinza G_o , finalizando la secuencia con el retorno del robot a R_H con la pinza abierta

G_o . Las condiciones de operación definidas para el proceso de llenado de botellas se muestran en la Tabla 3.12.

Tabla 3.12: Condiciones de operación proceso de llenado de botellas.

Tareas	Estados de precondition	Condiciones	Estados alcanzados
Descargar botella	$\{R_H, G_o\}$	BP_3^p	$\{RP_1, G_o\}$
	$\{RP_1, G_o\}$		$\{RP_1, G_c\}$
	$\{RP_1, G_c\}$		$\{RP_2, G_c\}$
	$\{RP_2, G_c\}$		$\{RP_2, G_o\}$
	$\{RP_2, G_o\}$		$\{R_H, G_o\}$
Llenar botella	$\{BP_2^a, BL^e\}$		$\{BP_2^p, BL^f\}$
	$\{BP_2^p, BL^f\}$	BP_3^a	$\{BP_2^a, BL^e\}$

Como se muestra en la Tabla 3.12, la primera fila corresponde a la tarea de descarga de botellas y la segunda fila a la tarea de llenar botellas. En la primera tarea, la primera subfila de la columna **estados de precondition** se llena con el conjunto de estados iniciales $\{R_H, G_o\}$ de las variables controladas implicadas en la tarea de descarga de botellas; la primera subfila de la columna **condiciones** se llena con el estado BP_3^p (guarda), indicando que existe una botella en la posición 3, que es condición esencial para que los estados iniciales de las variables controladas $\{R_H, G_o\}$ cambien a los estados alcanzados, así, la primera subfila de la columna **estados alcanzados** se llena con los estados $\{RP_1, G_o\}$. Ahora, la segunda subfila de la tarea descargar botella en la columna estados de precondition se llena con el estado alcanzado anterior $\{RP_1, G_o\}$, en esta etapa el brazo robótico se encuentra en la salida del transportador con la pinza cerrada, y en la segunda subfila de la columna estados alcanzados se llena con los estados $\{RP_1, G_c\}$ (agarrando la botella). Las otras subfilas de la tarea *descargar botella* siguen la secuencia de operación hasta que el brazo robótico deposite la botella cuando este se encuentre en la posición $\{RP_2, G_o\}$. Finalmente, el brazo robótico retorna a su estado inicial $\{R_H, G_o\}$, el cual es el estado alcanzado de la secuencia de operación de la tareas descargar botella. Las operaciones se leen de manera similar para ambas tareas.

3.1.2. Estructura de grafo del proceso

La estructura de grafo del proceso G_p presenta en forma gráfica la relación que existe entre actuadores y variables controladas involucradas en un proceso industrial. Además, al usar esta estructura es posible realizar la captura previa de errores como la completitud e inconsistencias presentes en la información del proceso. A continuación se presenta la definición formal de la estructura de grafo del proceso G_p .

Definición 3.1.1 *Un grafo de proceso G_p es una tripleta $G_p = (N, B, D)$, donde $N = \{n_1, \dots, n_i\}$ es un conjunto finito de vértices, n_i representa el i -ésimo elemento de un proceso industrial, $B \cup D \subset N \times N$ representa un conjunto finito de arcos donde B es la relación permisiva entre actuadores y variables controladas y D es la relación síncrona entre variables controladas.*

La estructura de grafo de proceso G_p está ligada fuertemente a la representación dada por el diagrama de tuberías e instrumentos DTI del sistema, donde los vértices en el G_p representan los elementos de un proceso industrial (válvulas, motores, niveles, temperaturas, etc.) y los arcos la relación entre ellos. Así, un vértice n_i corresponde al i -ésimo actuador o elemento final de control del proceso, el vértice n_j a la j -ésima variable controlada de un proceso industrial, los arcos continuos B ilustran las relaciones permisivas entre los vértices n_i y n_j , y los arcos discontinuos D a las relaciones síncronas entre vértices n_j .

Metodología de construcción del G_p . La construcción de la estructura de grafo del proceso G_p representa en forma gráfica los elementos que conforman el proceso industrial y la relación que existe entre ellos. La estructura G_p se forma con la información dada por el diagrama de tuberías e instrumentos DTI y la estructura de la narrativa del proceso SNO. La información del DTI se usa para obtener una aproximación del número de vértices que forman el grafo, el cual se complementa con la estructura de tablas de la Sección 3.1.1. Así, el número de filas de la columna elemento del proceso de la Tabla 3.1 de la estructura de tablas es equivalente al número de vertices n_i y n_j del grafo G_p , donde cada fila de la Tabla 3.1 forma un vértice del grafo G_p . Ahora, los vertices n_i y n_j que representan los actuadores y variables controladas del proceso se relacionan entre sí mediante relaciones permisivas y síncronas, las cuales son ilustradas en las Tablas *Relaciones permisivas* y *Relaciones síncronas*. Así, si en la Tabla

Relaciones permisivas existe una celda no nula en la sub-fila de n_i y la columna n_j que relaciona el “estado” de un actuador con los “eventos” de una variable controlada, entonces un arco continuo $(n_i, n_j) \in B$ se dibuja desde el vértice n_i hasta el(los) evento(s) de la variable controlada n_j . Ahora, si en la Tabla *Relaciones síncronas* existe una celda no nula marcada con “X” en la sub-fila de n_j y la sub-columna n_j que representa un evento de la variable controlada n_j , un arco discontinuo se dibuja entre los vértices de las variables controladas que se relacionan entre sí. En la estructura de grafos, los arcos B son ilustrados con líneas continuas y representan las relaciones permisivas entre un actuador y una variable controlada, y los arcos D son ilustrados con líneas discontinuas y representan las relaciones síncronas entre variables controladas. Para ilustrar la construcción de la estructura de grafo del proceso G_p se usa el sistema de llenado de botellas ilustrado en el Ejemplo 2.3.1. En este ejemplo, cada elemento listado en la primera columna de la Tabla *descripción de elementos* representa un vértice de la estructura de grafo G_p , debido a que en la columna elementos del proceso se listan ocho elementos (ZZ-103, ZN-103, ZCC-103, LV-104, $BP_1 - 103$, $BP_2 - 103$, $BP_3 - 103$ y BL-104), entonces ocho vértices $(n_1, n_2, n_3, \dots, n_8)$ se forman en la estructura del grafo G_p . Luego, como en la Tabla *relación permisiva* del sistema de llenado de botellas existe un evento que relaciona el estado “on” del motor ZZ-103 con los eventos de la variables controladas $BP_1 - 103$, $BP_2 - 103$ y $BP_3 - 103$, entonces se dibujan arcos continuos desde n_1 a n_2 , n_1 a n_3 , y n_1 a n_4 . Otro ejemplo que relacione vértices mediante relaciones permisivas es cuando el estado “abierto” de la válvula LV-104 se relaciona con el evento de la variable controlada BL-104, así un arco dirigido es dibujado desde el vértice n_7 a n_8 , y de la misma forma con las relaciones entre elementos ilustrados en la Tabla *relación permisiva*. La Fig. 3.2 representa el grafo de proceso de llenado de botellas del Ejemplo 2.3.1.

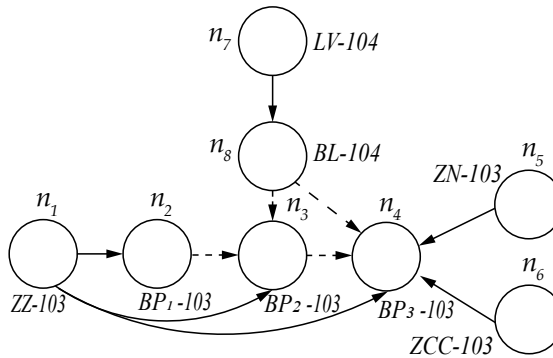


Figura 3.2: Grafo del proceso para el sistema de llenado de botellas.

De la misma forma que los arcos continuos son dibujados en la estructura de grafo G_p , las relaciones síncronas también son dibujadas, así los arcos discontinuos dirigidos desde $(n_2$ a $n_3)$, $(n_3$ a $n_4)$, $(n_8$ a $n_3)$, y $(n_8$ a $n_4)$ representan las relaciones síncronas entre variables controladas dada por la Tabla *relación síncronas*.

3.1.3. Modelo de RdPI general del sistema

El modelo de Red de Petri Interpretada (RdPI) ilustra de forma gráfica las características y comportamiento del proceso y los elementos que lo conforman, además de la relación que existe entre cada uno de ellos y su respectivo funcionamiento. Como se mencionó anteriormente, las RdPI son una extensión de la RdP que permiten la representación de las señales de entrada y salida para sistemas de eventos discretos. En este trabajo se diseñan dos tipos de modelos de RdPI que caracterizan el comportamiento del proceso y las tareas que deben realizar el proceso, las cuales se denominan, modelo en RdPI del proceso y modelo en RdPI de la especificación, respectivamente.

Modelo en RdPI del proceso.

El modelo en RdPI del proceso presenta en forma gráfica las características y el comportamiento del proceso mediante Redes de Petri individuales denominadas **módulos**. Cada módulo representa un elemento del proceso (actuador o variable controlada), y la relación entre ellos es dada por arcos

que van desde módulos de RdP que representan un actuador hasta módulos de RdP que representa una variable controlada. Cada lugar de un módulo de RdP representa un estado del elemento del proceso (vacío, lleno, frio, caliente, apagado, encendido, etc.), las transiciones un evento del proceso (llenar, vaciar, enfriar, calentar, abrir, cerrar, etc.) y los arcos entre lugares y transiciones representan la evolución del elemento. En un modelo de RdPI, el i -ésimo módulo de RdP $\mathcal{M}_i$ es dado por $\mathcal{M}_i = \langle G^i, \mathbf{M}_0^i, \Sigma_I^i, \Sigma_O^i, \lambda^i, \varphi^i \rangle$, donde $G^i = \langle P_i, T_i, I_i, O_i \rangle$ corresponde a la RdP independiente, el marcado $\mathbf{M}_0^i$ equivale a la condición inicial del elemento, Σ_I^i y Σ_O^i corresponden al alfabeto de entrada y salida del i -ésimo módulo, y finalmente λ^i y φ^i son la función de etiquetado de las transiciones y la función de salida, respectivamente.

Metodología de construcción del modelo de RdPI del proceso. El modelo de RdPI del proceso ilustra las características y funcionamiento de los elementos del sistema mediante módulos de RdP, además del comportamiento del proceso a través de la unión entre módulos. El modelo de RdPI del proceso está conformado por Redes de Petri independientes denominados módulos, los cuales son el resultado de la traducción de la información presentada en la estructura de tablas y la estructura del grafo del proceso. A continuación, se presenta la metodología usada en la traducción del modelo de RdPI, iniciando con la construcción de módulos independientes de RdP y finalizando con la relación entre módulos mediante relaciones síncronas y permisivas.

Construcción de módulos de RdP del proceso. El módulo de RdP $\mathcal{M}_i$ que representa las características y comportamiento de los elementos se construye con la información dada por la estructura de tablas y el grafo del proceso G_p . Así, cada vértice n_i de G_p se traduce en un módulo de RdP $\mathcal{M}_i$, además, el número de lugares P_i y su marcado inicial $\mathbf{M}_0^i$, depende de las características del elemento enumeradas en la Tabla *descripción de elementos* de la estructura de tablas. Los módulos de RdP resultantes se conectan entre sí, si en las tabla *relación permisiva* y *relación síncrona* existe un vínculo entre los elementos de un proceso industrial. En una RdPI, el número de módulos de RdP se determina por el numero de elementos listados en la columna *elemento del proceso* de la Tabla 3.1; el número de lugares $P_i = \{p_1, \dots, p_r\}$ de un módulo de RdP se fija con el valor del rango del elemento, el cual se establece en la columna *rango del elemento* de la Tabla 3.1. Ahora, una transición t_{jk} y sus respectivos arcos desde el lugar p_j a la transición t_{jk} , y desde la transición t_{jk} al lugar p_k se construyen cuando en la Tabla 3.2 una

marca “X” se sitúa en la celda definida por la fila r_j y la columna c_k . Entonces se puede decir que el número de transiciones en el módulo de RdP depende del número de marcas “X” en la *tabla comportamiento de elementos*. Finalmente, el marcado inicial del módulo de RdP $\mathcal{M}_i$ es dado por $M_0^i(p_k) = 1$, si el lugar del elemento p_k corresponde a la condición inicial del elemento e_i dado por la *Tabla descripción de elementos*, y $M_0^i(p_k) = 0$ en otro caso. Cada lugar del módulo de RdPI se nombra de acuerdo a la etiqueta asignada a cada estado del elemento listado en la tabla de descripción del elemento, es decir, $\varphi^i(p_j) = state_k$, si el lugar p_j del i -ésimo elemento es asociado al estado $state_k$. Además, las distintas etiquetas están asociadas a las transiciones del actuador (todas las transiciones en los actuadores son controlables). Los alfabetos de entrada y salida se definen como la unión de las etiquetas asociadas a todas las transiciones y lugares del módulo, respectivamente.

Interconexión entre módulos de una RdPI. Dos módulos $\mathcal{M}_i$ de una RdPI se conectan entre sí, si el lugar p_h del módulo $\mathcal{M}_i$ es conectado a la transición t_{kl} del módulo $\mathcal{M}_j$ con un arco de ida y vuelta. Esta condición es afirmativa, si existe un evento “ p_k a p_l ”, en la celda definida por la sub-fila r_h del elemento e_i y la columna de e_j en la Tabla 3.3 (Relación permisiva). Adicionalmente a los arcos que representan las relación permisiva, las transiciones t_{ab}, t_{cd} de módulos de RdP $\mathcal{M}_i, \mathcal{M}_j$, respectivamente, se fusionan en una sola transición, si existe una marca “X” en la celda definida por la sub-fila “ a a b ”, y la sub-columna “ c a d ”, como se ilustra en Tabla 3.4 (Relación síncrona). Cabe resaltar que por construcción, el modelo de RdPI es consistente y conservativo.

Ejemplo 3.1.1 : Modelo en RdPI del proceso de llenado de botellas. Para ilustrar la construcción de módulos individuales de RdP y la unión entre cada uno de ellos, se usa el proceso de llenado de botellas ilustrado en la Fig. 2.7. En este ejemplo, cada elemento del sistema es representado por un módulo individual de Red de Petri teniendo en cuenta la información de la estructura de tablas y la estructura de grafos. Posteriormente, los módulos que representan los actuadores son conectados con los que representan variables controladas teniendo en cuenta la tabla de relaciones permisivas y síncronas de la estructura de tablas.

Para ilustrar la construcción de módulos de RdP ($\mathcal{M}_i$) se usa la válvula de nivel LV-104, este elemento tiene la función llenar las botellas que se desplazan sobre la banda transportadora como se muestra en el proceso de llenado de botellas de la Fig. 2.7. Las características de la válvula LV-104 se resumen en

la Tabla 3.5 (descripción del elemento), en esta tabla la columna Rango del elemento especifica que la válvula tiene rango “2”, la columna Estados del elemento establece la etiqueta para cada estado de la válvula como cerrado y abierto, finalmente la columna Estado inicial del elemento determina que el estado inicial de la válvula LV-104 es cerrado. Teniendo en cuenta la información de la Tabla 3.5, el número de lugares del módulo de la válvula LV-104 $\mathcal{M}_7^p$ es definido teniendo en cuenta el rango del elemento, por lo tanto el módulo $\mathcal{M}_7^p$ se forma con dos lugares p_1^7 , y p_2^7 . Cada lugar del módulo de RdP es etiquetado con el nombre asignado a cada estado como se muestra en la Tabla 3.5, así, al lugar p_1^7 se le asigna la etiqueta cerrado y al lugar p_2^7 la etiqueta abierto. Luego, una marca aparece en el lugar etiquetado como cerrado, ya que en la columna estado inicial establece el estado inicial de la válvula LV-104 como cerrado. La Fig. 3.3 ilustra la construcción del módulo de RdP para la Válvula LV-104 basada en la información de la Tabla 3.5.

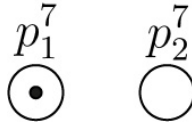


Figura 3.3: Construcción de lugares y estado inicial de la válvula LV-104.

Para ilustrar el comportamiento de la válvula LV-104, la Tabla 3.6 se utiliza para completar el módulo de RdP mediante transiciones y arcos que representan los eventos y el cambio de estado del elemento, respectivamente. Así, las transiciones $t_{1,2}^7$ y $t_{2,1}^7$ y sus arcos dirigidos son incluidos en el módulo $\mathcal{M}_7^p$, debido a que los estados de la válvula LV-104 pueden cambiar en un solo paso desde cerrado (p_1^7) a abierto (p_2^7), y desde abierto (p_2^7) a cerrado (p_1^7) como se muestra en la sub-tabla del elemento LV-104 de la Tabla Comportamiento de elementos. El modelo de RdP que representa las características y comportamiento de la válvula de nivel LV-104 se muestra en la Fig. 3.4.

Al igual que la válvula de nivel LV-104, cada elemento del sistema es traducido a módulos individuales de RdPs. Los módulos de RdP se construyen con la información suministrada por las tablas descripción de elementos y comportamiento de elementos. La Fig. 3.5 muestra los módulos de RdP individuales para cada elemento del sistema de calefacción de líquidos ilustrado

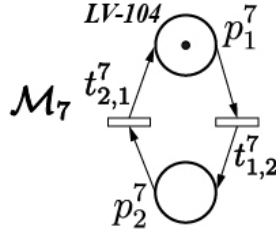


Figura 3.4: Módulo de RdP de la válvula de nivel LV-104.

en la Fig. 2.7. En esta figura, los módulos $\mathcal{M}_1^p, \mathcal{M}_2^p, \mathcal{M}_3^p, \dots, \mathcal{M}_8^p$ representan las RdP independientes de los elementos del sistema listados en la Tabla descripción de elementos.

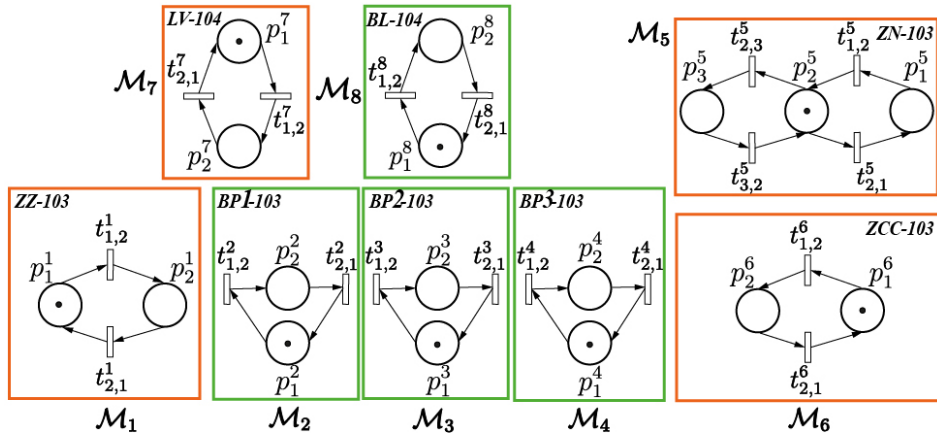


Figura 3.5: Módulos de RdP del proceso de llenado de botellas.

La unión entre módulos independientes de RdP $\mathcal{M}_i$ y $\mathcal{M}_j$ conforman un modelo de RdPI que representa la interacción entre actuadores y variables controladas en un proceso industrial, además de ilustrar el funcionamiento del sistema. La unión entre módulos de RdP se representa mediante relaciones permisivas y síncronas dadas por las Tablas 3.7 y 3.8, respectivamente. Para ilustrar la unión entre módulos de RdP mediante relaciones permisivas se usan los elementos LV-104 y BL-104 del proceso de llenado de botellas debido a que estos elementos están relacionados en la narrativa del proceso y por tanto en

la Tabla relaciones permisivas, así, se incluyen un arco de ida y vuelta desde el lugar p_2^7 del módulo M_7^p a la transición $t_{1,2}^8$ del módulo M_8^p , ya que existe la leyenda BL^e a BL^f que relaciona el estado abierto de la válvula LV-104 con la variable controlada BL-104 como se ilustra en la Tabla 3.7. Ahora, para ilustrar la sincronización entre variables controladas se usan las variables controladas BP_1 y BP_2 , donde la transición $t_{2,1}^2$ del módulo M_2^p y la transición $t_{1,2}^3$ del módulo M_3^p se unen formando la transición $t_{M_{2,3}}$, ya que existe una marca “X” que relaciona el evento BP_1^a a BP_1^a de la variable controlada $BP_1 - 103$ con el evento BP_2^a a BP_2^a de la variable controlada $BP_2 - 103$ como se ilustra en la Tabla 3.8.

El modelo completo de RdPI del proceso de llenado de botellas se ilustra en la Fig. 3.6, donde los módulos que representan los actuadores y variables controladas de un proceso industrial se unen entre si con arcos de ida y vuelta, y la unión de transiciones teniendo en cuenta la información de la tabla Relación permisiva y la Tabla Relaciones síncronas, respectivamente.

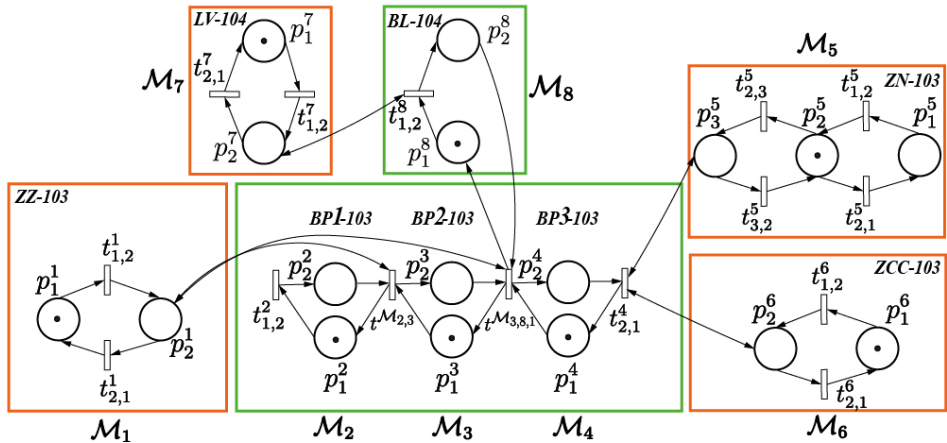


Figura 3.6: RdPI del proceso de llenado de botellas.

Modelo en RdPI de operación del proceso.

El modelo de RdPI de operación comprende un conjunto de módulos individuales M_j^o los cuales representan las tareas que debe realizar un proceso industrial. De forma similar al modelo en RdPI del proceso, el modelo en RdPI

de operación esta dado por $\check{M}_j^o = \langle \check{G}^j, \check{M}_o^j, \check{\Sigma}_I^j, \check{\Sigma}_O^j, \check{\lambda}^j, \check{\varphi}^j \rangle$, donde $\check{G}^j = \langle P_j, T_j, I_j, O_j \rangle$ corresponde al módulo de RdP independiente que representa una tarea del proceso, el marcado $\check{M}_o^j$ equivale a la condición inicial de una tarea del proceso, $\check{\Sigma}_I^j$ y $\check{\Sigma}_O^j$ corresponden al alfabeto de entrada y salida del j-ésimo módulo de la tarea del proceso, y finalmente $\check{\lambda}^j$ y $\check{\varphi}^j$ son la función de etiquetado de las transiciones y la función de salida, respectivamente.

Metodología de construcción del modelo de RdPI de operación. El modelo de RdPI de operación de un proceso industrial presenta en modelos de RdP independientes las tareas que un proceso debe realizar para dar cumplimiento a los requisitos del proceso. Cada modelo de RdP de operación $\check{M}_j^o$ se construye tomando la información presentada en la estructura de la narrativa de operación SNO de la sección 3.1.1, la cual comprende las tablas de rangos de operación y condiciones de operación del proceso. Para determinar el número de módulos de RdP que conforman el modelo de RdPI de operación, la columna 1 de la Tabla 3.10 es usada para dicho fin. Así, si existe una tarea mencionada en la columna **tarea** de la tabla **condiciones de operación**, se construye un módulo de RdP de operación. Cada tarea se define por n lugares y n transiciones, donde n es el número de operaciones relacionada con cada tarea definidas en la Tabla *condiciones de operación*. Cada secuencia de operación se traduce desde las subfilas que compone cada tarea, entonces cada casilla de las columnas **estados de precondition** y **estados alcanzados** se traduce en lugares p_i y p_{i+1} respectivamente, y por cada casilla en la columna **condiciones**, se forma una transición t_i . En una secuencia de operaciones $i \in \{1, \dots, n\}$, se sitúa un arco desde el lugar p_i a la transición t_i , y desde la transición t_i al lugar p_{i+1} , si $i + 1 \leq n$, la secuencia de operaciones de la tarea finaliza situando un arco desde la transición t_n al lugar p_1 . Además, cada lugar p_i y transición t_i se etiqueta con el conjunto de estados de las variables controladas que acompaña a cada casilla.

El marcado inicial $\check{M}_0^j$ del módulo de RdP de operación se define como $\check{M}_0^j(p_1) = 1$ y $\check{M}_0^j(p_i) = 0, \forall i \neq 1$. Además, para cualquier $i \in \{1, \dots, n\}$, se etiqueta p_i como $\check{\varphi}^j(p_i) = pre - state_i$, donde el $pre - state_i$ es el conjunto de estados en la celda de la i-ésima fila y la tercera columna. Los alfabetos de entrada y salida se definen como la unión de las etiquetas asociadas a todas las transiciones y lugares en el modulo, respectivamente.

Para ejemplificar la construcción de módulos de RdP de operación que

conforman el modelo de RdPI de operación de un proceso, el proceso de llenado de botellas es usado. Los módulos resultantes de cada tarea con sus respectivas secuencias ilustradas en la Tabla 3.12 (*condiciones de operación*) se muestran en la Fig. 3.7. En esta figura, cada modelo representa la secuencia de operaciones que se debe realizar para cumplir las tareas de *descargar botella* y *llenar botella*, respectivamente. Así, en la tarea **descargar botella**, existen 5 secuencias de operación y la tarea **llenar botella** se muestran 2 secuencias de operación. Así, el lugar $\{R_H, G_o\}$, la transición etiquetada como BP_3^p , y el lugar $\{RP_1, G_o\}$ corresponde a la primera sub-fila de la tarea *descargar botella* en la Tabla 3.12. Del mismo modo, la segunda sub-fila de la tarea *llenar botella* en la Tabla 3.12 forma dos lugares y una transición etiquetadas como $\{BP_2^p, BL^f\}$, $\{BP_2^a, BL^e\}$, y BP_3^a , respectivamente. Las marcas en los lugares del modelo representan las condiciones iniciales de cada tarea, la cual es dada por la narrativa de operación.

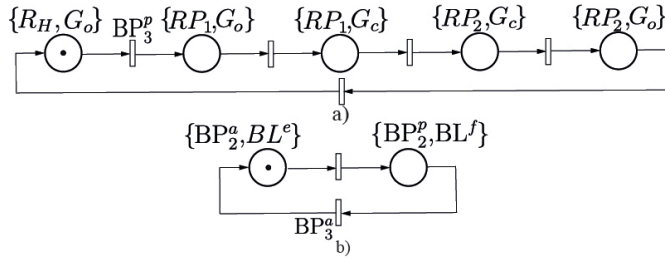


Figura 3.7: Módulos RdPI de operación para las tareas de descarga y llenado botellas

Para obtener una especificación adecuada para el paradigma de control de seguimiento [19, 21], se obtiene el gráfico de alcanzabilidad de los módulos de Operación y se transforma en una RdPI, conservando las etiquetas que pertenecen a los lugares y transiciones. La RdPI resultante representa el modelo de RdPI de la *especificación*, la cual describe el comportamiento requerido por el proceso. El modelo de RdPI que representa la especificación del proceso de llenado de botellas se muestra en la Fig. 3.8.

Los modelos de la planta y la especificación son usados para la construcción del modelo en lazo cerrado del proceso, el cual se presenta en capítulos posteriores y forma parte importante en la construcción de la lógica de programación.

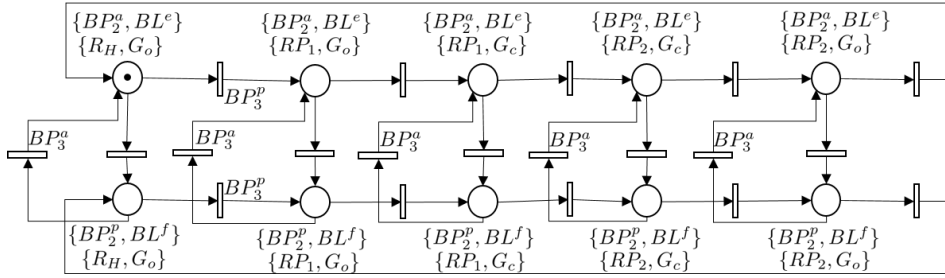


Figura 3.8: Modelo RdPI de la especificación para el proceso de llenado de botellas.

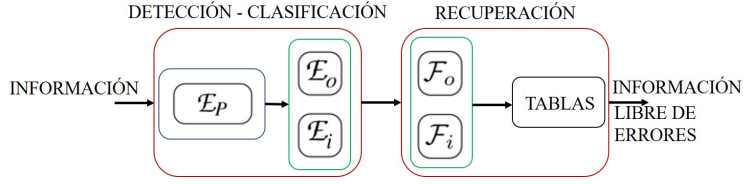
3.2. Metodología de detección y recuperación de errores

En esta sección se presenta la metodología de detección y recuperación de errores de la información de un proceso industrial (sistema neumático, celda de manufactura, etc.), con el fin de garantizar una descripción confiable de los elementos y comportamiento del proceso, y por tanto una lógica de programación libre de errores. Esta metodología propone capturar los errores presentes en la información de la estructura de tablas de la sección 3.1.1, luego estos errores son clasificados como *errores de omisión* y *errores de inconsistencia* según su naturaleza, y finalmente el sistema se recupera de los errores mediante la actualización de la información en la estructura de tablas. Los errores recuperados en la estructura de tablas se propagan a las diferentes estructuras de la metodología de traducción (estructura de grafos, módulos de RdP y modelos de RdPI) de la información de un proceso industrial a RdPI.

Nótese, que la metodología de detección y recuperación de errores propuesta, es de gran ayuda a la hora de estructurar la característica y comportamiento del proceso en la estructura de tablas, partiendo desde la experiencia del operador del proceso, en casos donde no exista una narrativa de proceso y operación.

La Fig. 3.9 ilustra la metodología de detección y recuperación de errores, donde $\mathcal{E}_p$ representa el conjunto de errores potenciales encontrados en la estructura de tablas; $\mathcal{E}_o$ y $\mathcal{E}_i$ son los conjuntos de errores clasificados por omisión e inconsistencia, respectivamente, finalmente, $\mathcal{F}_o$ y $\mathcal{F}_i$ representan las funciones de recuperación de errores por omisión e inconsistencia, respectivamente.

Figura 3.9: Metodología de detección y recuperación de errores.

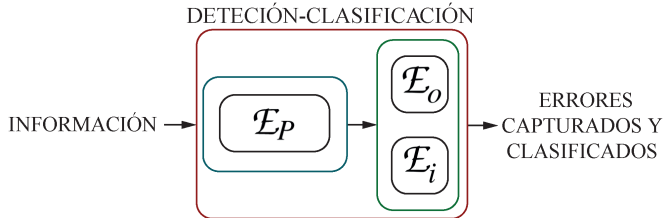


Como se observa en la Fig. 3.9, la información de las características de los elementos del proceso ingresa a un procedimiento de detección y clasificación donde los errores son capturados y según su naturaleza divididos como errores por omisión o errores por inconsistencia. Luego, dependiendo del tipo de error, en la etapa de recuperación, se aplica una función de recuperación de errores, permitiendo obtener una corrección y recuperación de errores en la información del proceso presente en la estructura de tablas, además de propagarse en las etapas de la metodología de traducción. A continuación se presenta en detalle cada una de las fases que forman parte de la metodología de detección y corrección de errores.

3.2.1. Detección y clasificación de errores

La metodología de detección y recuperación de errores comprende dos etapas importantes definidas como detección-clasificación y recuperación. En la etapa de *detección-clasificación*, el conjunto de errores potenciales $\mathcal{E}_P$, donde $\mathcal{E}_P = \mathcal{E}_O \cup \mathcal{E}_I$, representa los errores potenciales capturados desde la información de la estructura de tablas, donde $\mathcal{E}_O = \{\mathcal{E}_O^m\}$ es el conjunto de errores de omisión, y $\mathcal{E}_I = \{\mathcal{E}_I^n\}$ es el conjunto de errores de inconsistencia como se observa en Fig. 3.10.

Figura 3.10: Detección y clasificación de errores.



El listado de errores potenciales que se pueden presentar en la información de un proceso industrial se describen en la Tabla 3.13 (errores capturados desde la estructura de tablas). En esta tabla, un error de omisión $\mathcal{E}_O$ hace referencia a la ausencia de información dentro de la estructura de tablas, esto ocasiona que a través de la metodología de traducción se presenten cambios como la ausencia de uno o mas vértices en la estructura de grafos, o en su defecto la eliminación de arcos que conecten vértices entre si, además, este tipo de error se propaga a los modelos de RdPI, eliminando lugares o arcos en el modelo de RdPI. En esta tabla, la primera columna representa la identificación del error (omisión o inconsistencia) capturado desde la tabla analizada; la segunda columna muestra la descripción del error basado en la característica o comportamiento del elemento; finalmente, la última columna presenta el predicado lógico basado en la descripción del error. Se dice que un error $\mathcal{E}_m^k$ ha ocurrido si su predicado lógico descrito en la Tabla 3.13 llega a ser verdadero. En esta tabla, el error identificado como $\mathcal{T}_1 - \mathcal{E}_{o_r}^k$ corresponde a un error detectado en la Tabla 3.5 (descripción de elementos) etiquetada como $\mathcal{T}_1$; la etiqueta $\mathcal{E}_O$ representa los errores capturados y o indica que los errores son clasificados como errores por omisión; el sub-índice r equivale a la característica del elemento *rango*. Las demás errores basados en las características de los elementos se muestran en la Tabla 3.13.

Para automatizar el proceso de detección-clasificación de los posibles errores potenciales que se encuentran presentes en la estructura de tablas, el Algoritmo 1 es usado para dicho fin, el cual al ser ejecutado, da como resultado un conjunto de errores detectados S_E . Este algoritmo explora cada celda de la Tabla $\mathcal{T}_i$ de la estructura de tablas, y si en la tabla revisada existe un error de los listados en la Tabla 3.13, este error es detectado y clasificado.

Algorithm 1: Algoritmo de detección de errores

Result: Conjunto de errores detectado S_E

Seleccionar tabla $\mathcal{T}_i$

FOR $k=1$ al número de elementos en $\mathcal{T}_i$ **DO**

IF $\mathcal{T}_i - \mathcal{E}_{o_x}^k$ **THEN** se detecta el error de tipo x en la tabla $\mathcal{T}_i$ en la $k - th$ fila. Agregar este error a S_E .

END FOR;

Para ejemplificar el proceso de detección-clasificación mediante el algoritmo de detección de errores, se incluye un error omitiendo información en la

Tabla 3.13: Errores capturados desde la estructura de tablas

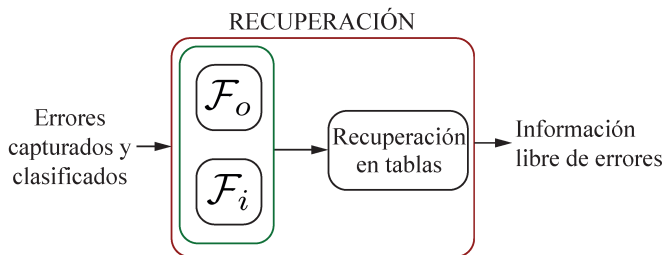
Error desde el sistema de la estructura de tabla $\mathcal{T}$		
Error	Descripción	Predicados lógicos
Errores en la tabla de descripción de elementos (nombre de esta tabla $\mathcal{T}_1$)		
errores de omisión		
$\mathcal{T}_1 - \mathcal{E}o_r^k$	El elemento no tiene un rango declarado r es el rango del elemento.	Hay una celda vacía $[k, 2]$ en la columna rango.
$\mathcal{T}_1 - \mathcal{E}o_s^k$	El elemento no tiene sus estados declarados. s es el estado del elemento.	Hay una celda vacía $[k, 3]$ en la columna estados.
$\mathcal{T}_1 - \mathcal{E}o_{is}^k$	El elemento no tiene su estado inicial declarado. is representa el estado inicial del elemento.	Hay una celda vacía $[k, 4]$ en la columna estado inicial.
Errores de inconsistencia		
$\mathcal{T}_1 - \mathcal{E}i_r^k$	El rango del elemento es 1. r es el rango del elemento.	Hay un rango < 2 en la celda $[k, 2]$.
$\mathcal{T}_1 - \mathcal{E}i_s^k$	El número de estados del elemento y su rango no son iguales. s es el estado del elemento.	Hay un rango del elemento en la celda $[k, 2]$ no igual al numero de estados en la celda $[k, 3]$.
$\mathcal{T}_1 - \mathcal{E}i_{is}^k$	El estado inicial del elemento no pertenece a sus estados declarados. is representa el estado inicial del elemento.	Hay un estado inicial en la celda $[k, 4]$ que no pertenece a los estados de la celda $[k, 3]$.
Errores en la tabla de comportamiento de elementos (nombre para esta tabla $\mathcal{T}_2$)		
Errores de omisión		
$\mathcal{T}_2 - \mathcal{E}o_{cm}^k$	El elemento no tiene una marca X que defina su comportamiento. cm es una marca.	Hay un elemento sin una marca X que relacione sus estados en la celda $[k, :]$.
Errores de inconsistencia		
$\mathcal{T}_2 - \mathcal{E}i_{cm}^{kk}$	El elemento tiene una marca X que relaciona dos estados iguales. cm es una marca.	Hay una marca X que relaciona un mismo estado en la celda $[k, k]$.
$\mathcal{T}_2 - \mathcal{E}i_{cm}^k$	El elemento no tiene una marca X que lo relacione con sus estados anteriores y siguientes. cm es una marca.	Hay un elemento sin una marca X en la celda $[k, k - 1]$ y o $[k, k + 1]$.
Detección de errores en la tabla de relacion permisiva (nombre para esta tabla $\mathcal{T}_3$)		
Errores de omisión		
$\mathcal{T}_3 - \mathcal{E}o_{as}^k$	El actuador no tiene un estado declarado. as es el estado del actuador.	Hay una celda vacía $[k, 2]$ en la columna estado del actuador.
Errores de inconsistencia		
$\mathcal{T}_3 - \mathcal{E}i_{ec}^k$	El estado inicial del actuador modifica los estados de las variables controladas. ec son columnas de eventos de la variable controlada.	Hay una condición inicial sin sus celdas vacías en las columnas de eventos $[k, :]$.
Detección de errores en la tabla de estados de operación (nombre para esta tabla $\mathcal{T}_5$)		
Errores de omisión		
$\mathcal{T}_5 - \mathcal{E}o_{os}^k$	La variable controlada no tiene sus estados de operación declarados. os son los estados de operación de la variable controlada.	Hay una celda vacía $[k, 2]$ en la columna estados de operación
Errores de inconsistencia		
$\mathcal{T}_5 - \mathcal{E}i_{os}^k$	Un estado de operación no pertenece a los estados declarados de la variable controlada. os son los estados de operación de la variable controlada.	Hay un estado de operación en la celda $[k, 2]$ que no pertenece a los estados de la celda $[k, 3]$ de $\mathcal{T}_1$.
Detección de errores en la tabla de condiciones de operación (nombre para esta tabla $\mathcal{T}_6$)		
Errores de omisión		
$\mathcal{T}_6 - \mathcal{E}o_{ps}^k$	El subproceso no tiene sus condiciones de operación anteriores. ps es el estado de pre-condición.	Hay celdas vacías $[:, 2]$ en la columna estados de pre-condición
$\mathcal{T}_6 - \mathcal{E}o_{ts}^k$	El subproceso no tiene su estado objetivo. ts es el estado objetivo.	Hay una celda vacía $[:, 4]$ en la columna estados objetivos.

descripción del proceso. En esta descripción un elemento ($\mathcal{E}_m$) no incluye el valor de su rango, eso quiere decir que en la fila k de la tabla *descripción de elementos* en la celda correspondiente al rango del elemento, se encuentra sin descripción o vacía. Así, verificando el listado de errores potenciales de la Tabla 3.13, se presenta un error definido como $\mathcal{T}_1 - \mathcal{E}o_r^k$, por lo tanto, existe un error por omisión y se hace referencia al predicado lógico correspondiente al error que dice: “Hay una celda vacía $[k, 2]$ en la columna Rango”, lo que indica que en la entrada del rango del elemento se encuentra vacío, por lo tanto este error es detectado y clasificado como un error por omisión.

3.2.2. Recuperación de errores

Esta sección presenta la metodología de corrección y recuperación de errores de la información de un proceso industrial presentes en la estructura de tablas. Para corregir un error $\mathcal{E}_x^k$ capturado y clasificado en la estructura de tablas, se debe ejecutar una función de recuperación $\mathcal{F}_x^k$ que consiste en solicitar la información requerida (en el caso de errores de omisiones) o eliminar y corregir la inconsistencia (en el caso de errores de inconsistencias) al ingeniero del proceso. Para realizar la corrección de los errores potenciales listados en la Tabla 3.13, se presenta la función de recuperación de errores $\mathcal{T}_i - \mathcal{F}o_x^k$, donde $\mathcal{T}_i$ corresponde a la tabla analizada desde la estructura de tablas, y $\mathcal{F}o_x^k$ la función de recuperación propia de cada error potencial listado en la Tabla 3.13. Las acciones correctivas que deben ser realizadas para la eliminación y recuperación de los errores son descritas en las Tablas 3.14 y 3.15, las cuales representan las funciones de recuperación de los errores de omisión e inconsistencia respectivamente como se muestra en la Fig. 3.11.

Figura 3.11: Recuperación de errores.



52 CAPÍTULO 3. METODOLOGÍA DE TRADUCCIÓN Y RECUPERACIÓN

Tanto en la Tabla 3.14 como en la Tabla 3.15, la primera columna muestra el listado de funciones de recuperación (una por cada error potencial de la Tabla 3.13), y la segunda columna presenta las acciones correctivas que se deben realizar para la recuperación de cada error.

Tabla 3.14: Funciones de recuperación de errores de omisión

$\mathcal{F}_o$	Acción correctiva
$\mathcal{T}_1 - \mathcal{F}o_r^k$	El ingeniero debe proporcionar el rango en la celda [k,2]
$\mathcal{T}_1 - \mathcal{F}o_s^k$	El ingeniero debe proporcionar el estado en la celda [k,3]
$\mathcal{T}_1 - \mathcal{F}o_{ts}^k$	El ingeniero debe proporcionar el estado inicial en la celda [k,4]
$\mathcal{T}_2 - \mathcal{F}o_{cm}^k$	El ingeniero debe proporcionar el comportamiento del elemento en la celda. [k,:]
$\mathcal{T}_3 - \mathcal{F}o_{as}^k$	El ingeniero debe proporcionar los estados del actuador en la celda [k,2]
$\mathcal{T}_5 - \mathcal{F}o_{os}^k$	El ingeniero debe proporcionar los estados de operación en la celda [k,2]
$\mathcal{T}_6 - \mathcal{F}o_{ps}^k$	El ingeniero debe proporcionar el estado de la condición previa en las celdas [.,2]
$\mathcal{T}_6 - \mathcal{F}o_{ts}^k$	El ingeniero debe proporcionar el estado de destino en la celda [.,4]

Tabla 3.15: Funciones de recuperación de errores de inconsistencia

$\mathcal{F}_i$	Acción correctiva
$\mathcal{T}_1 - \mathcal{F}i_r^k$	El ingeniero debe proporcionar el rango en la celda [k,2].
$\mathcal{T}_1 - \mathcal{F}i_s^k$	El ingeniero debe proporcionar el estado en la celda [k,3].
$\mathcal{T}_1 - \mathcal{F}i_{ts}^k$	El ingeniero debe proporcionar el estado inicial en la celda [k,4].
$\mathcal{T}_2 - \mathcal{F}i_{cm}^k$	El ingeniero debe proporcionar el comportamiento del elemento en la celda. [k,k].
$\mathcal{T}_2 - \mathcal{F}i_{cm}^k$	El ingeniero debe proporcionar el estado del actuador en las celdas [k,k-1] y/o [k,k+1].
$\mathcal{T}_3 - \mathcal{F}i_{ec}^k$	El ingeniero debe proporcionar los estados de los elementos en las celdas. [k,:].
$\mathcal{T}_5 - \mathcal{F}i_{os}^k$	El ingeniero debe proporcionar los estados de operación en la celda [k,2].

De forma general, para eliminar y recuperarse de los errores detectados y clasificados, se deben aplicar las funciones correspondientes, es decir:

“SI el error $\mathcal{T}_i - \mathcal{E}o_x^k$ es detectado ENTONCES aplicar la función de eliminación de errores $\mathcal{T}_i - \mathcal{F}o_x^k$ ”.

Por lo tanto, la metodología propuesta se compone de dos etapas, la primera se enfoca en detectar y aislar errores (la cláusula de premisa en la oración SI-ENTONCES), y la segunda se enfoca en eliminar los errores (la cláusula de conclusión en la frase SI-ENTONCES). Es importante señalar que se pueden considerar nuevos tipos de errores simplemente agregando sus descripciones en la Tabla 3.13 y sus funciones correctivas en las Tablas 3.14 o 3.15, ya sean errores por omisión o inconsistencia, respectivamente.

Para eliminar y recuperarse de los errores detectados en la Tabla 3.13, el Algoritmo 2 se utiliza para dicho fin. En este algoritmo, S_E equivale al conjunto de errores calculado en el Algoritmo 1, el cual contiene todos los errores potenciales que se deben recuperar.

Algorithm 2: Algoritmo de recuperación de errores

Result: Conjunto de tablas libre de errores

WHILE $S_E \neq \emptyset$ **DO**

IF el error de tipo x detectado en la tabla $\mathcal{T}_i$ en la $k - th$ fila esta en S_E

THEN

Eliminar este error desde S_E

Ejecutar la función de recuperación $\mathcal{T}_i - \mathcal{F}o_x^k$

END IF

END WHILE;

Para ejemplificar las metodologías de detección y recuperación de errores de la información de un proceso industrial se usa el sistema de la Fig. 2.6 que representa el transporte y llenado de botellas sobre una banda transportadora. En este ejemplo, si un error es introducido a la estructura de tablas debido a una traducción errónea, la propagación del error ocasiona alteraciones en la estructura de tablas, la estructura de grafo y los modelos de RdPI. En el sistema de transporte de botellas, si al describir las características de los elementos en la narrativa del proceso se especifica que la válvula LV-104 es de rango *dos*, el estado de la válvula es *cerrado*, y su estado inicial es *cerrado*, la tabla de descripción de elementos de la estructura de tablas queda de la siguiente forma:

Como lo indican las metodologías de detección y recuperación de errores, la aplicación de los Algoritmos 1 y 2 conducen a los siguientes pasos:

1. **Detección y clasificación del error.** AL realizar una inspección en la tabla de descripción de elementos, un error $\mathcal{T}_1 - \mathcal{E}i_s^4$ es detectado, ya que en la fila 4 correspondiente al actuador LV-104 existe un error de inconsistencia entre el número de estados (cerrado) y el rango del elemento (2) identificado con el color rojo en la Tabla 3.16. Formalmente, se puede decir que $\mathcal{T}_1 - \mathcal{E}i_s^4$ es verdadero, porque $\mathcal{T}_1[4, 2] \neq |\mathcal{T}_1[4, 3]|$.
2. **Recuperación del error.** De acuerdo a la instrucción dada en la Tabla 3.15, el error $\mathcal{T}_1 - \mathcal{E}i_s^4$ se corrige solicitando al ingeniero del proceso los estados

Tabla 3.16: Descripción de elementos.

Variable controlada & Actuador	Rango	Estados	Estado Inicial
ZZ-103(Motor)	2	off, on	off
ZN-103 (Robot)	3	home (R_H), RP_1 , RP_2	R_H
ZZC-103 (Pinza)	2	abierto (G_o), cerrado (G_c)	G_o
LV-104 (Válvula)	2	cerrado	cerrado
Posición Botella-1 ($BP_1 - 103$)	2	ausencia (BP_1^a), presencia (BP_1^p)	ausencia (BP_1^a)
Posición Botella-2 ($BP_2 - 103$)	2	ausencia (BP_2^a), presencia (BP_2^p)	ausencia (BP_2^a)
Posición Botella-3 ($BP_3 - 103$)	2	ausencia (BP_3^a), presencia (BP_3^p)	ausencia (BP_3^a)
BL-104(Nivel de Botella)	2	vacío (BL^e), lleno (BL^f)	vacío (BL^e)

asociados al actuador LV-104 ejecutando la función de recuperación $\mathcal{T}_1 - \mathcal{F}i_s^4$ se ejecuta para la celda $\mathcal{T}_1[4, 3]$.

En este caso, el ingeniero del proceso completa la tabla con el estado “abierto” para la válvula LV-104 y el error por inconsistencia es corregido, el cual es identificado en color verde en la Tabla 3.17.

3. **Propagación de la recuperación del error.** Una vez que la tabla es corregida en la celda correspondiente, la recuperación del error se propaga a través de la estructura de metodología de traducción, iniciando con la estructura de tablas, pasando por la estructura de grafo y finalizando con el modelo en RdPI del sistema. Por tanto, los errores se corrigen haciendo que la información de las narrativas, la estructura de grafos y el modelo de RdPI sean coherentes con la estructura de tablas del sistema.

Al finalizar, la corrección y recuperación de la información errónea se realiza en la tabla donde se detectó el error, en este caso en la tabla **característica de elementos**, en la celda correspondiente a los estados del elemento LV-104 como se muestra en la Tabla 3.17.

Tabla 3.17: Descripción de elementos.

Variable controlada & Actuador	Rango	Estados	Estado Inicial
LV-104 (Válvula)	2	abierto, cerrado	cerrado

Capítulo 4

Control por regulación y traducción a PLC

Este capítulo presenta la definición del problema de control por regulación que tiene como objetivo diseñar un controlador tal que la salida del sistema sea igual a la salida de la especificación del proceso. En este caso, los dos modelos, el modelo del sistema y el modelo de la especificación son definidos como RdPI. Además de la definición del problema de control, se presentan las condiciones para la solución del problema de control por regulación que se basa en la existencia de una función que relaciona el conjunto de estados del sistema con el conjunto de estados de la especificación. Finalmente, se presenta la metodología para traducir la información dada por el sistema en lazo cerrado dado en forma de RdPI, a un lenguaje de programación basado en la Norma IEC 1131-3 para el control de procesos.

4.1. Control por regulación

El estudio del problema de control de regulación incluye la definición de dos modelos de RdPI denominados modelo de la planta y modelo de la especificación. La construcción de estos modelos se presentaron en la sección 3.1.3, los cuales representan las características de los elementos y comportamiento del proceso, además de las tareas que se deben cumplir en la ejecución del proceso, respectivamente.

En esta sección, la **planta de un proceso industrial** corresponde al sistema a ser controlado (proceso industrial, sistema neumático, celda de manufactura, etc), y se modela como una *RdPI* segura $Q_p = (G_p, \mathbf{M}_0, \Sigma_I, \Sigma_O, \lambda, \psi, \varphi)$. Los símbolos de salida se asignan de manera que cada lugar tenga como máximo un símbolo de salida y cada símbolo se asigna a un solo lugar. La ecuación fundamental de la red de Petri de la planta se escribe como:

$$\begin{aligned} \mathbf{M}_{k+1} &= \mathbf{M}_k + \mathbf{C}\vec{\sigma}_{k+1} \\ y_k &= \varphi M_k. \end{aligned} \quad (4.1)$$

La **especificación de un proceso industrial** indica el comportamiento que debe tener la planta, el cual es representado como una máquina de estado de *RdPI* $Q_s = (G_s, M_{r_0}, \check{\Sigma}_I, \check{\Sigma}_O, \check{\lambda}, \check{\psi}, \check{\varphi})$, donde todas las transiciones son controlables. Es importante señalar que el alfabeto de salida de la especificación y de la planta deben ser iguales, es decir, $\check{\Sigma}_O = \Sigma_O$. La ecuación fundamental que representa la especificación de un proceso industrial se escribe como:

$$\begin{aligned} \check{\mathbf{M}}_{k+1} &= \check{\mathbf{M}}_k + \check{\mathbf{C}}\vec{t}_{k+1} \\ \check{y}_k &= \check{\varphi} M r_k. \end{aligned} \quad (4.2)$$

El modelo de *RdPI* de la Fig. 4.1 ilustra la especificación de un proceso industrial, la cual es representada como una máquina de estado que muestra la concurrencia mediante la asignación de más de un símbolo de salida en cada lugar de la *RdPI*. Por lo tanto, para expresar la concurrencia de especificaciones de procesos concurrentes, en este trabajo se considera que los lugares en las especificaciones se pueden etiquetar con más de un símbolo.

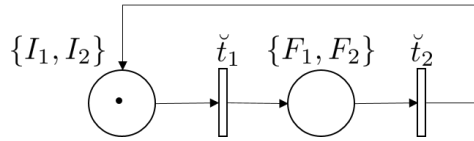


Figura 4.1: *RdPI* que representa una Especificación de un proceso industrial. Es una máquina de estado segura y representa una concurrencia.

Definición 4.1.1 (*Problema de control de regulación*). Sea $(Q_p, \mathbf{M}_0)$ y $(Q_s, \check{\mathbf{M}}_0)$ los modelos de *RdPI* que representan la planta y la especificación de un proceso industrial, las cuales son representadas por las ecuaciones fundamentales de *RdP*

(4.3) y (4.4) respectivamente como sigue:

$$\begin{aligned} \mathbf{M}_{k+1} &= \mathbf{M}_k + \mathbf{C}\vec{\sigma}_{k+1} \\ y_k &= \varphi M_k \end{aligned} \quad (4.3)$$

$$\begin{aligned} \check{\mathbf{M}}_{k+1} &= \check{\mathbf{M}}_k + \check{\mathbf{C}}\check{\vec{t}}_{k+1} \\ \check{y}_k &= \check{\varphi}\check{M}_k. \end{aligned} \quad (4.4)$$

El *problema de control de regulación* (PCR) consiste en encontrar una función de controlador tal que $M_k \xrightarrow{\check{t}_k} M_{k+1}$, proporciona las etiquetas que suministran una secuencia de disparo σ_k en la planta, donde $M_r \xrightarrow{\sigma_k} M_s$, $\varphi M_r = \check{\varphi}\check{M}_k$ y $\varphi M_s = \check{\varphi}\check{M}_{k+1}$.

4.2. Sistema de lazo cerrado

En ingeniería, un sistema de lazo cerrado o realimentado es aquel donde la acción de control está en función de la señal de salida; es decir, la señal de salida a controlar se compara con la señal de entrada y así generar una señal de error que es recibida por el controlador y tomar una acción en el proceso para finalmente llevar la salida del sistema al valor deseado. Teniendo en cuenta el concepto anterior y aplicado a sistemas de eventos discretos, el sistema de lazo cerrado de la Fig. 4.2 muestra como el modelo en RdPI de la planta y la especificación evolucionan simultáneamente cuando el controlador especifica la secuencia de transiciones que deben activarse con el fin de obtener el mismo resultado en el modelo de la planta y el modelo de la especificación. La Fig. 4.2 muestra la estructura del circuito cerrado del enfoque de control de regulación el cual se denominará en adelante *sistema de lazo cerrado* [21], [32].

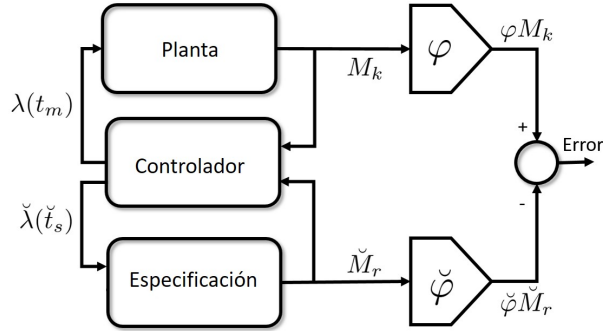


Figura 4.2: Esquema de sistema a lazo cerrado.

4.2.1. Condiciones de solución para el problema de control de regulación

Para dar solución al problema de control de regulación (PCR), primero se define la relación de alcanzabilidad y la relación de alcanzabilidad extendida, las cuales involucran los marcados de una RdPI como se presenta en [33, 34].

Definición 4.2.1 *Relación de alcanzabilidad (R_R). Sea (Q, M_0) una RdPI y $R(Q, M_0)$ su conjunto de alcanzabilidad. $(M_j, M_i) \in R_R$ si $\exists t_a$ habilitada en M_i tal que $M_i \xrightarrow{t_a} M_j$.*

Definición 4.2.2 *Relación de alcanzabilidad extendida (ER_R). Sea (Q, M_0) una RdPI y $R(Q, M_0)$ su conjunto de alcanzabilidad.*

Sea (Q, M_0) una RdPI y $R(Q, M_0)$ su conjunto de alcanzabilidad. $(M_j, M_i) \in ER_R$ si $\exists t_a t_b \dots t_q$ habilitado en M_i tal que $M_i \xrightarrow{t_a} M' \dots \xrightarrow{t_q} M_j$.

Con base en las definiciones anteriores, una noción de simulación se define como sigue.

Definición 4.2.3 *Sea $(Q_p, \mathbf{M}_0)$ y $(Q_s, \check{\mathbf{M}}_0)$ el modelo de RdPI de la planta y la especificación, con un conjunto de alcanzabilidad denotado como $R(Q_p, \mathbf{M}_0)$ y $R(Q_s, \check{\mathbf{M}}_0)$, respectivamente, donde $R_R \subset R(Q_s, M_{r0}) \times R(Q_s, M_{r0})$ y $ER_R \subset R(Q_p, M_0) \times R(Q_p, M_0)$. $(Q_s, \check{\mathbf{M}}_0)$ se dice que es simulado para $(Q_p, \mathbf{M}_0)$ si $\exists \Pi : R(Q_s, \check{\mathbf{M}}_0) \rightarrow R(Q_p, \mathbf{M}_0)$ tal que*

$$\Pi \circ R_R = ER_R \circ \Pi \quad (4.5)$$

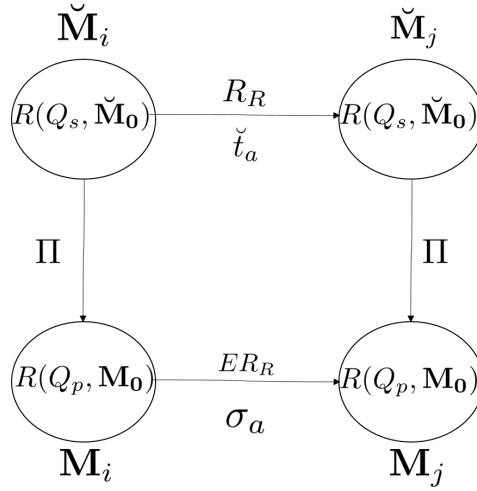


Figura 4.3: Diagrama de conmutación. Cada marcado en la especificación es relacionado a un marcado en la RdPI de la planta, y cada transición de la especificación es equivalente a una secuencia de transiciones en la planta.

La Fig. 4.3 muestra los conjuntos de alcanzabilidad de la planta y la especificación, los cuales son relacionados por Π . Si se cumple la ecuación (4.5), el diagrama de la Fig. 4.3 conmuta. La relación de simulación significa que el disparo de una transición $\check{t}_x$ en la RdPI de la especificación $\check{\mathbf{M}}_i \xrightarrow{\check{t}_x} \check{\mathbf{M}}_{i+1}$ puede ser reproducido por la planta disparando una secuencia de transición σ_x , es decir, $\mathbf{M}_i \xrightarrow{\sigma_x} \mathbf{M}_{i+1}$, donde $(\check{\mathbf{M}}_i, \mathbf{M}_i), (\check{\mathbf{M}}_{i+1}, \mathbf{M}_{i+1}) \in \Pi$. En detalle, el lado izquierdo de (4.5) es $\Pi \circ R_R(\check{\mathbf{M}}_i)$, y es equivalente a $\Pi(\check{\mathbf{M}}_{i+1}) = \mathbf{M}_{i+1}$, el lado izquierdo es $ER_R \circ \Pi(\mathbf{M}_i)$ y es equivalente a $ER_R(\mathbf{M}_i) = \mathbf{M}_{i+1}$, es decir que la planta simula la especificación mediante Π . La siguiente proposición da algunas condiciones estructurales necesarias para $(Q_s, \check{\mathbf{M}}_0)$ para ser simulado por $(Q_p, \mathbf{M}_0)$.

Proposición 4.2.1 *Sea $(Q_p, \mathbf{M}_0)$ y $(Q_s, \check{\mathbf{M}}_0)$ los modelos de RdPI de la planta y la especificación, con el conjunto de alcanzabilidad denotado por $R(Q_p, \mathbf{M}_0)$ y $R(Q_s, \check{\mathbf{M}}_0)$, respectivamente. $(Q_s, \check{\mathbf{M}}_0)$ es simulado por $(Q_p, \mathbf{M}_0)$ si existe una matrix Π tal que:*

1. $\mathbf{M}_0 = \Pi \check{\mathbf{M}}_0$

$$2. \forall \check{t}_a \in \check{T} \quad \exists \sigma_a = t_q t_r \dots t_s \in \mathcal{L}(Q_p, M_0) \text{ tal que } \mathbf{C}\vec{\sigma}_a = \mathbf{\Pi}\check{\mathbf{C}}\check{t}_a$$

donde $\mathbf{\Pi}$ es la matriz que implementa la función Π .

4.2.2. Condiciones de la solución

La siguiente proposición presenta condiciones suficientes para resolver el problema del control de la regulación como se presenta en [33, 34].

Proposición 4.2.2 *Sea $(Q_p, \mathbf{M}_0)$ y $(Q_s, \check{\mathbf{M}}_0)$ los modelos RdPI de la planta y la especificación, con conjuntos de alcanzabilidad denotados como $R(Q_p, \mathbf{M}_0)$ y $R(Q_s, \check{\mathbf{M}}_0)$, respectivamente. El problema del control de la regulación tiene solución si*

1. $(Q_s, \check{\mathbf{M}}_0)$ es simulado por $(Q_p, \mathbf{M}_0)$.
2. Para cada $\check{t}_a \in \check{T}$ la secuencia calculada para la simulación $\sigma_a \in \mathcal{L}(Q, M_0)$ es controlable.
3. $\check{\varphi} = \varphi\mathbf{\Pi}$.

4.3. Diseño del controlador

Esta sección presenta el diseño de un controlador basado en el cálculo de $\mathbf{\Pi}$ y secuencias σ_k para cada $\check{t}_k \in \check{T}$. Para el diseño del controlador el problema se divide en dos partes, una calculando $\mathbf{\Pi}$ y el vector de Parikh $\vec{\sigma}_k$ cumpliendo las condiciones algebraicas de las Proposiciones 4.2.1 y 4.2.2, y el otro para calcular las secuencias controlables σ_k .

4.3.1. Cálculo de $\mathbf{\Pi}$ y secuencias de disparo controlables

La función $\mathbf{\Pi}$ en el diseño del controlador tiene como propósito asociar cada marcado del modelo de la especificación con un marcado equivalente en la salida particular de la planta. La función $\mathbf{\Pi}$ y los vectores de Parikh $\vec{\sigma}_k$ para la secuencia de transiciones de la planta, son calculadas por el **Algoritmo 3** como un problema de programación lineal (LPP) [35] de la misma forma como se presenta en [36]. Luego, los vectores de Parikh encontrados son usados para calcular las secuencias de transiciones σ_k requeridas en el **Algoritmo 4**.

Algorithm 3: Cálculo de Π .

-
- 1: **Input** Modelo de RdPI de la planta $(Q_p, \mathbf{M}_0)$ y modelo de la especificación $(Q_s, \check{\mathbf{M}}_0)$.
 - 2: **Output** Función Π , vectores de Parikh $\vec{\sigma}_k$.
 - 3: **Solve** el siguiente LPP
 - 4: $\min \sum_{i,j} \Pi(i, j) + \sum_{k=1}^{|\check{T}|} \sum_{n=1}^{|\check{T}|} \vec{\sigma}_k(n)$ subject to
 - 5: $\Pi \check{\mathbf{M}}_0 = \mathbf{M}_0$
 - 6: $\forall \check{t}_k \in \check{T}, \Pi \check{\mathbf{C}}(\bullet, k) = \mathbf{C} \vec{\sigma}_k$
 - 7: $\varphi \Pi = \check{\varphi}$
 - 8: $\vec{\sigma}_k \geq \mathbf{0}$
 - 9: Si no hay solución, entonces el algoritmo termina con un error: =0
-

Observe que, si $\mathbf{C}\chi = 0$, es decir, si χ es un T-semiflujo de la planta, entonces $\mathbf{C}(\vec{\sigma}_k + \chi) = \mathbf{C}\vec{\sigma}_k$ por tanto $(\vec{\sigma}_k + \chi)$ también es una solución en la línea 6 del Algoritmo 3. Sin embargo, $\vec{\sigma}_k + \chi$ también implica la activación de las transiciones en χ .

Luego, cada vector de Parikh es transformado en una secuencia deseable σ_k expandiendo parcialmente el grafo de alcanzabilidad. Las secuencias σ_k pueden incluir el disparo de T-semiflujos y deben ser controlables. En la Fig. 4.4, se muestra el grafo de alcanzabilidad de expansión parcial de la planta $R(Q_p, \mathbf{M}_0)$, en este grafo se describe una secuencia de marcado de $\Pi \check{\mathbf{M}}_i$ a $\Pi \check{\mathbf{M}}_j$, es decir, $\Pi \check{\mathbf{M}}_i \xrightarrow{t_k} \mathbf{M}_3 \xrightarrow{t_q} \mathbf{M}_4 \xrightarrow{t_r} \mathbf{M}_5 \xrightarrow{t_s} \Pi \check{\mathbf{M}}_j$, donde la secuencia de transiciones $\sigma_k = t_k t_q t_r t_s$ es descrita por los arcos de la expansión parcial.

Para la controlabilidad de la secuencia de disparo σ_k , se requiere que si una transición incontrolable se habilita en algún estado de la secuencia de marcado, su disparo debe coincidir con la secuencia de marcado. Por ejemplo, si hay una transición incontrolable t_u habilitada al marcar $\mathbf{M}_4$ diferente de t_r , entonces la secuencia es incontrolable porque $t_k t_q t_u$ no es un prefijo de la secuencia de disparo σ_k .

Para el cálculo de secuencias controlables es posible aplicar diferentes algoritmos de búsqueda, los cuales realizan el cálculo expandiendo parcialmente el grafo de alcanzabilidad. El Algoritmo 4 implementa un algoritmo de búsqueda A^* para calcular una secuencia σ la cual minimiza un costo f , teniendo en cuenta los nodos inicial $\Pi \check{\mathbf{M}}_i$ y final $\Pi \check{\mathbf{M}}_j$, y las restricciones adicionales para

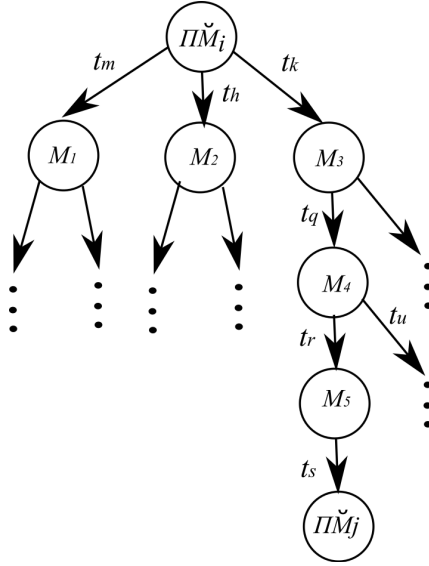


Figura 4.4: Expansión parcial del grafo de alcanzabilidad.

asegurar la capacidad de control. Las líneas 8-20 del Algoritmo 4 garantizan la controlabilidad de la secuencia calculada denominada σ_k . Si en el modelo hay dos o más transiciones incontrolables habilitadas desde $\mathbf{M}$, entonces cualquier secuencia que visite $\mathbf{M}$ es incontrolable, por lo que $T_M = \emptyset$ (línea 9), lo que significa que los sucesores de $\mathbf{M}$ no serán explorados; ahora, si solo hay una transición incontrolable habilitada desde $\mathbf{M}$ entonces cualquier secuencia que visite $\mathbf{M}$ debe incluir dicha transición, por lo tanto $T_M = T_M \cap T_{un}$ (línea 11). En las líneas 19-20, el algoritmo se detiene si hay una transición incontrolable habilitada en la marca de destino. La condición en la línea 14 es necesaria para evitar conflictos no deterministas; si se elimina dicha condición, entonces t o t' podrían activarse cuando el controlador indique $\lambda(t)$ que impondrá σ_k . En la línea 16 el algoritmo se detiene con éxito cuando se alcanza el nodo final y el vector Parikh de la secuencia calculada es igual o mayor que el $\vec{\sigma}_k$ dado. En las líneas 22 y 23 se define la función de costo de un estado alcanzado f' , como la suma del número de disparos para llegar a $\mathbf{M}'$ (costo de g') más el vector Parikh diferencia con respecto a $\vec{\sigma}_k$.

Algorithm 4: Cálculo de una secuencia controlable σ_k .

-
- 1: **Input** Estructura de RdPI Q_p , Nodo inicial $\Pi\check{M}_i$, Nodo final $\Pi\check{M}_j$, Vector de Parikh $\vec{\sigma}_k$.
 - 2: **Output** Secuencia controlable σ_k .
 - 3: Inicializar $Open = \{\Pi\check{M}_i\}$. Set $g(\Pi\check{M}_i) = 0$ and $f(\Pi\check{M}_i) = \sum_{i=1}^{|T|} |\vec{\sigma}_k(i)|$. Set $\sigma(\Pi\check{M}_i) = \sigma_\epsilon$.
 - 4: Inicializar $Closed = \emptyset$
 - 5: **while** $Open \neq \emptyset$ **do**
 - 6: Encuentre el estado $M \in Open$ con el costo mas bajo $f(M)$
 - 7: Calcular el conjunto $T_M = \{t \in T | M \xrightarrow{t}\}$
 - 8: **if** $|T_M \cap T_u| \geq 2$ **then**
 - 9: Actualizar $T_M = \emptyset$
 - 10: **else if** $|T_M \cap T_u| = 1$ **then**
 - 11: Actualizar $T_M = T_M \cap T_u$
 - 12: **end if**
 - 13: **for** cada $t \in T_M$ **do**
 - 14: **if** $\nexists t' \in T_M$ tal que $\lambda(t) = \lambda(t')$ **then**
 - 15: Denotar como M' el estado alcanzado, es decir, $M \xrightarrow{t} M'$. Definir la secuencia $\sigma' = \sigma(M)t$
 - 16: **if** $M' = \Pi\check{M}_j$ y $\vec{\sigma}' \geq \vec{\sigma}_k$ y M' no habilitar transiciones incontrolables **then**
 - 17: Set $\sigma_k = \sigma'$
 - 18: **STOP** el algoritmo.
 - 19: **else if** $M' = \Pi\check{M}_j$ y M' habilite una transición incontrolable **then**
 - 20: No hay una secuencia controlable. **STOP** el algoritmo.
 - 21: **else**
 - 22: Set $g' = g(M) + 1$
 - 23: Set $f' = g' + \sum_{i=1}^{|T|} |\vec{\sigma}'(i) - \vec{\sigma}_k(i)|$
 - 24: **if** $M' \notin Closed \cup Open$ **then**
 - 25: Set $Open = Open \cup \{M'\}$
 - 26: Set $g(M') = g'$, $f(M') = f'$ y $\sigma(M') = \sigma'$.
 - 27: **else if** $M' \in Open$ y $f' < f(M')$ **then**
 - 28: Actualizar $g(M') = g'$, $f(M') = f'$ y $\sigma(M') = \sigma'$.
 - 29: **end if**
 - 30: **end if**
 - 31: **end for**
 - 32: **end while**
 - 33: Actualizar $Closed = Closed \cup \{M\}$ y $Open = Open \setminus \{M\}$
 - 34: **end while=0**
-

4.3.2. Representación de RdPI en lazo cerrado del sistema

Algorithm 5: RdPI del sistema general en lazo cerrado.

-
- 1: **Entrada** Modelo RdPI de la planta ($Q_p, \mathbf{M}_0$) y Modelo RdPI de la especificación ($Q_s, \check{\mathbf{M}}_0$).
Controlador h y secuencias equivalentes $\sigma_a, \forall \check{t}_a \in \check{T}$.
 - 2: **Salida** Modelo de RdPI en lazo cerrado ($Q_{cl}, \mathbf{M}_0^{cl}$).
 - 3: Definir una estructura $PdPI$ preliminar del sistema del lazo cerrado como $Q_{cl} = Q_p \cup Q_s$.
 - 4: Definir el marcado inicial del sistema de RdPI del lazo cerrado como $\mathbf{M}_0^{cl} = [\mathbf{M}_0^T, \check{\mathbf{M}}_0^T]^T$.
 - 5: Inicializar un conjunto de transiciones de plantas visitadas $T_V = \emptyset$.
 - 6: Inicializar un conjunto de lugares $P_{sub} = \emptyset$.
 - 7: Agregue nodos y arcos adicionales de la siguiente manera:
 - 8: **for** cada $\check{t}_a \in \check{T}$ **do**
 - 9: Sea $\check{\mathbf{M}}_i, \check{\mathbf{M}}_j$ tal que $\check{\mathbf{M}}_i \xrightarrow{\check{t}_a} \check{\mathbf{M}}_j$. Sea $\sigma_a = t_1^a \dots t_m^a$ la secuencia de disparo equivalente descrita como $\Pi \check{\mathbf{M}}_i = \mathbf{M}_1^a \xrightarrow{t_1^a} \mathbf{M}_2^a \xrightarrow{t_2^a} \dots \xrightarrow{t_m^a} \mathbf{M}_{m+1}^a = \Pi \check{\mathbf{M}}_j$
 - 10: Agregue la sub-estructura de RdP la cual supervisará la aparición de σ_a de la siguiente manera:
 - 11: Agregar un nuevo conjunto de lugares $P_a = \{p_1^a, \dots, p_{m+1}^a\} \in P_{sub}$ y un nuevo conjunto de transiciones $\{t_1^{a'}, \dots, t_m^{a'}\}$. La marca inicial en estos nuevos lugares es cero, sus símbolos de salida son ε y los símbolos de las nuevas transiciones son ϵ . Además, $\forall j \in \{1, \dots, m\}$, agregar el arco de entrada ($p_j^a, t_j^{a'}$) y el arco de salida ($t_j^{a'}, p_{j+1}^a$).
 - 12: A continuación, agregue autolazos entre la sub-estructura y la planta para establecer la secuencia σ_a de la siguiente manera:
 - 13: **for** $j \in \{1, \dots, m\}$ **do**
 - 14: **if** $t_j^{a'}$ es controlable **then**
 - 15: **if** $t_j^{a'} \notin T_V$ **then**
 - 16: Agregue un autolazo entre p_j^a y t_j^a y actualice el conjunto de transiciones visitadas:
 $T_V = T_V \cup \{t_j^a\}$.
 - 17: **else**
 - 18: Haga una copia de t_j^a con sus arcos de entrada y salida de la planta, denotemos la copia como $t_j^{a,d}$. Agregue un autolazo entre p_j^a y $t_j^{a,d}$. El símbolo de $t_j^{a,d}$ es ϵ .
 - 19: **end if**
 - 20: **end if**
 - 21: Agregue un autolazo entre $t_j^{a'}$ a cada uno de los lugares en $(t_j^a)^\bullet$.
 - 22: **end for**
 - 23: Agregue un autolazo desde $\check{t}_a$ a p_1^a , por lo que el supervisor de secuencia (la sub-estructura) puede evolucionar después de que se dispare $\check{t}_a$.
 - 24: Agregue arcos desde p_{m+1}^a a cada una de las transiciones de la especificación en $\check{t}_a^\bullet$, para detener la evolución de la especificación hasta que se haya producido la secuencia σ_a .
 - 25: **if** $\exists p_l \parallel p_{m+1}^a \bullet \cap \check{T} = p_l^\bullet \cap \check{T}, p_l \in P_{sub}$ **then**
 - 26: Definir un nuevo lugar $p^{a'}$ tal que $p^{a'} = p_{m+1}^a \bullet \cup p_l^\bullet$ y $p^{a'} = p_{m+1}^a \bullet \cup p_l^\bullet$.
 - 27: $P_{sub} = P_{sub} \cup p^{a'}$.
 - 28: Eliminar p_{m+1}^a y p_l junto con todos sus arcos de entrada y salida.
 - 29: **end if**
 - 30: **end for**
 - 31: Sea $\check{t}_1$ tal que $\check{\mathbf{M}}_0 \xrightarrow{\check{t}_1}$ en el modelo de la especificación.
Agregue una marca que corresponda al marcado inicial del lugar en $\bullet \check{t}_1 \cap (P_{cl} \setminus \check{P})$, así, $\check{t}_1$ está habilitado por las sub-estructuras RdP añadidas en la marca inicial. =0
-

El lazo cerrado del sistema (Q_{cl}, M_0^{cl}) está conformado por los modelos en RdPI de la planta y la especificación, además de incluir para cada secuencia σ_a una sub-estructura de RdP encargada de supervisar o hacer seguimiento de la ocurrencia de la secuencia σ_a , lo que se realiza habilitando aquellas transiciones cuyos símbolos son provistos por h en cada paso. El lazo cerrado del sistema se obtiene como una RdPI general haciendo uso del Algoritmo 5. En la Fig. 4.5 se observa como una sub-estructura se incorpora a los modelos en RdPI de la planta y la especificación con el propósito de supervisar las secuencias $\sigma_i = t_1^i t_2^i$ equivalente a $\check{t}_i$, y $\sigma_j = t_1^j t_2^j$ relacionado a $\check{t}_j$. Como se observa en la Fig. 4.5, cuando se dispara la transición $\check{t}_i$ en el modelo RdPI de la especificación, una marca llega al lugar p_1^i de la sub-estructura, además, ninguna transición de la especificación será habilitada hasta que la marca en la sub-estructura llegue a p' , es decir, hasta que las transiciones $t_1^{i'}$ y $t_2^{i'}$ se habiliten, para que esto ocurra, la secuencia $\sigma_i = t_1^i t_2^i$ debe dispararse en la planta. La sub-estructura restringe la planta de tal manera que σ_i es la única secuencia que puede dispararse, un comportamiento similar ocurre cuando se dispara $\check{t}_j$.

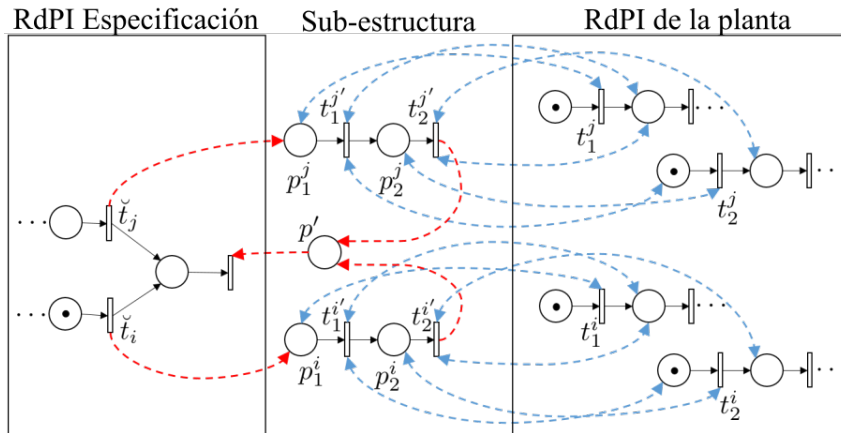


Figura 4.5: Sistema en lazo cerrado aplicando el Algoritmo 5.

Para ilustrar la interacción entre el modelo de RdPI de la planta y el modelo de RdPI de la especificación se usa el ejemplo del proceso de transporte y llenado de botellas de la Fig. 2.6, en este ejemplo el Algoritmo 5 se usa para determinar el lazo cerrado del sistema. En este sistema, los modelos en RdPI que

representan la planta y la especificación son relacionados mediante una secuencia de operaciones que debe realizar el proceso para garantizar el cumplimiento de las tareas.

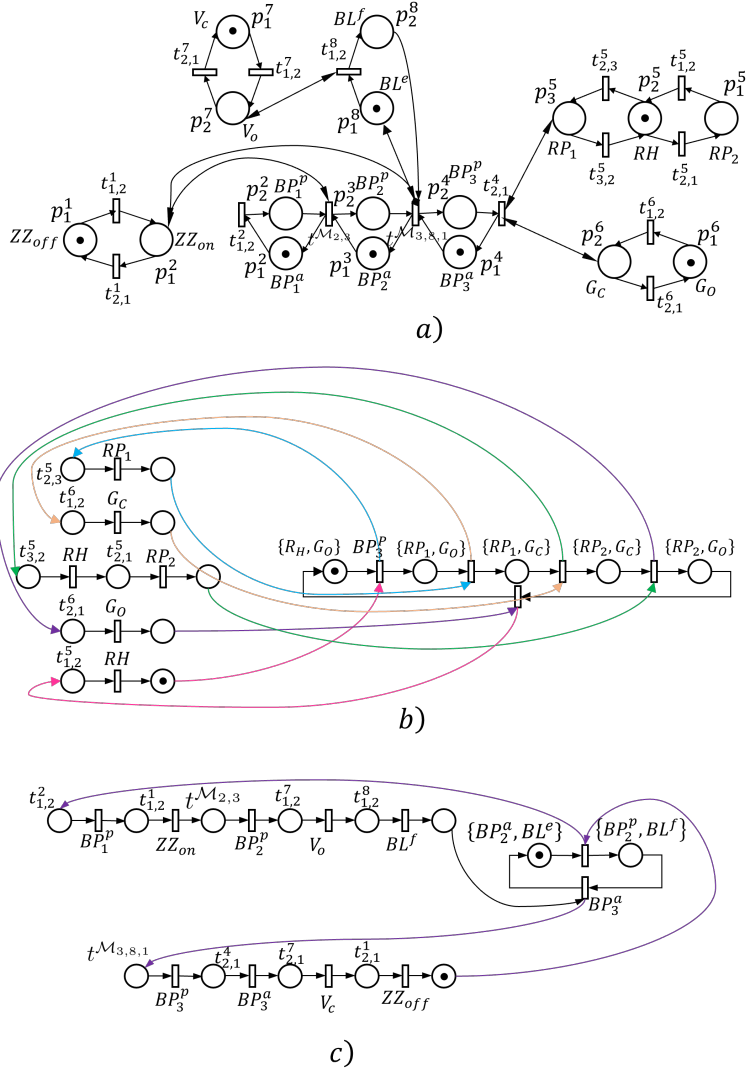


Figura 4.6: Lazo cerrado resultante del sistema de llenado de botellas aplicando el Algoritmo 5.

Al aplicar el Algoritmo 5 a los modelos de RdPI que representan la planta y la especificación del sistema de llenado y transporte de botellas, el conjunto de sub-estructuras se forma por cada transición encontrada en los modelos de RdPI de las especificaciones, las cuales representan las tareas que deben cumplir el proceso.

La Fig. 4.6 ilustra el lazo cerrado completo del sistema de transporte y llenado de botellas, donde la Fig. 4.6 b) y Fig. 4.6 c) representan las especificaciones y subestructuras para los procesos definidos como *evacuar botella* y *llenar botella* respectivamente. En las sub-estructuras de cada tarea, los lugares son nombrados con las etiquetas de las transiciones que deben ser disparadas en la RdPI que representa la planta (Fig. 4.6 a)), reemplazando arcos de ida y vuelta desde la sub-estructura a la planta (con fines de estética). Para ejemplificar la duplicidad de la etiqueta y la eliminación del arco, la etiqueta $t_{2,3}^5$ en la sub-estructura de evacuación de botellas, representa un arco de ida y vuelta hasta la transición etiquetada como $t_{2,3}^5$ en el modelo de la planta. Así, si una marca llega al lugar etiquetado como $t_{2,3}^5$ en la sub-estructura, la transición etiquetada como $t_{2,3}^5$ en el elemento ZN-103 del modelo de la planta se habilita, y al dispararse la marca pasa al lugar p_3^5 , lo que significa que el brazo del robot pasa de la posición RH a RP_1 .

4.4. Traducción del sistema en lazo cerrado a lógica de programación

Esta sección presenta la metodología de traducción de una estructura de Red de Petri interpretada que representa el lazo cerrado de un sistema (proceso industrial, sistema neumático, celda de manufactura, etc.) a un lenguaje de programación basado en lenguaje escalera bajo la Norma IEC61131-3. Esta metodología se basa en que cada transición de lazo cerrado del sistema sumado con la secuencia de disparo dada por el controlador, da como resultado la combinación de condiciones de los contactos de entrada y bobinas de salida en un brazo del diagrama escalera de la lógica de programación.

4.4.1. Reglas de traducción de estructura de RdPI a diagrama escalera

El lenguaje escalera pertenece a uno de los cinco lenguajes de programación usados en el diseño de la lógica de programación para autómatas programables los cuales se encuentran presentes en el estándar IEC-61131-3 [37, 38, 39]. El uso del lenguaje escalera se debe a la similitud que se tiene con los circuitos eléctricos y el comportamiento de los relé electromecánicos. Dentro de la lógica de programación de autómatas programables se encuentran características que se pueden transformar en estructuras de RdP como son: las lógicas AND, OR, AND-OR, autolazos y enclavamiento, los cuales tienen entradas y salidas discretas; y los bloques lógicos como los temporizadores, contadores y comparadores, los cuales tienen entradas analógicas pero su salida de control es del tipo discreto. Los elementos de una RdP definen la condiciones y acciones asociados a los elementos de un lenguaje escalera, así, una transición define las conexión entre variables de entradas y salidas. Para representar una lógica de programación basada en lenguaje escalera desde una estructura de RdPI se usa la metodología de traducción propuesta por [40], [41] y [42]. Estas reglas definen la interpretación de la definición de la RdP en la sintaxis del lenguaje escalera como sigue:

- Cada lugar en el modelo de RdPI tiene una variable interna o externa de un bit (booleano) en la representación del lenguaje escalera. Un lugar de RdPI que representa una señal de control para el actuador se traduce en una variable externa de un bit. Las variables externas en el autómata programable físico se asignan a la salida o entrada física. Otros lugares de la RdPI se representan en el lenguaje escalera como variables internas.
- Cada transición de la RdPI es equivalente a uno o dos brazos en el diagrama de escalera. La transición de tiempo distinto de cero (el tiempo d es mayor que cero) tiene dos renglones en LD. El primer peldaño contiene una traducción de las condiciones de habilitación y un temporizador con el tiempo de conmutación d . El segundo peldaño comprende la variable de salida del temporizador y la traducción del disparo, es decir, la ejecución de la transición. La traducción de una transición con disparo instantáneo es un caso especial de la traducción de una transición de tiempo distinto de cero donde se elimina el temporizador y ambos brazos se fusionan en uno

solo.

- Cada lugar de entrada o salida de una transición en la RdPI se traduce en una entrada y una entrada negada en el primer brazo del lenguaje escalera, respectivamente. Cada lugar de entrada de una transición, conectado a la transición con conexión negada, se traduce en entrada negada en el primer brazo del lenguaje escalera.
- El disparo de una transición en una RdPI, se traduce al lenguaje escalera como el restablecimiento de una variable de un bit (establecida en 1 o 0 respectivamente) en el segundo renglón.
- Una secuencia de brazos en el lenguaje escalera define la prioridad de activación de las transiciones en la RdPI. El brazo con la traducción del disparo de la transición con mayor prioridad debe escribirse antes que los brazos de otras transiciones concurrentes. Los brazos de transición traducida de tiempo distinto de cero deben escribirse juntos en un orden establecido.
- El marcado inicial M_0 del modelo de RdPI se traduce en una red del lenguaje escalera que se ejecuta solo una vez en el primer ciclo del programa y antes de otras sentencias.

Como se mencionó anteriormente, la característica principal de la traducción de una estructura de RdPI a lenguaje escalera radica en que cada transición se traduce en un brazo del lenguaje escalera como se muestra en la Fig. 4.7.

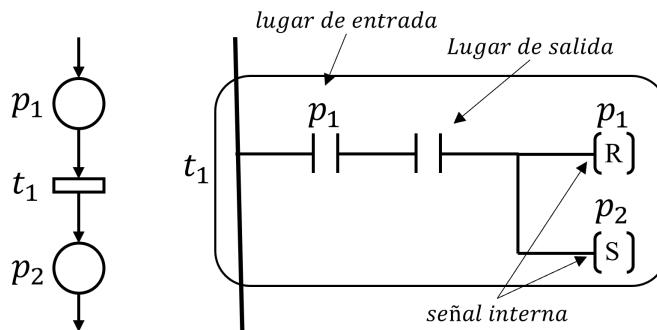


Figura 4.7: Representación de una transición en lenguaje escalera.

En esta figura, los lugares p_1 y p_2 se traducen como contactos de entrada y bobinas de salida en el lenguaje escalera, y la transición t_1 como un brazo donde se realiza la lógica de programación.

El marcado inicial M_0 de una RdPI se representa en lenguaje escalera como un brazo con un contacto cerrado tal que p_i sea un lugar binario. Además, para cada lugar inicial marcado en la RdPI debe existir una bobina de salida de ajuste y debe existir una bobina de salida de restablecimiento como se muestra en la Fig. 4.8.

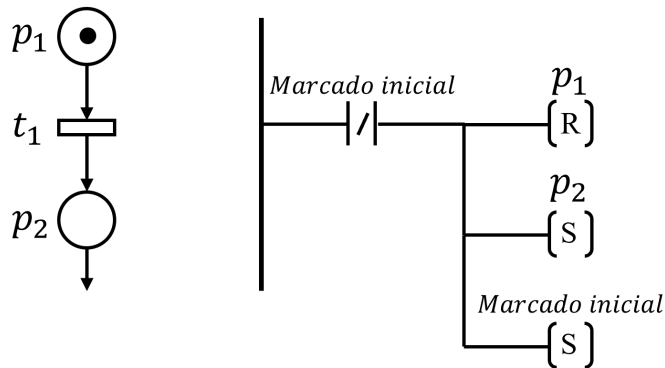


Figura 4.8: Representación de marcado inicial en lenguaje escalera.

Ahora, cada transición de tiempo distinto de cero en RdPI (**d**) tiene dos brazos en el lenguaje escalera, el primero para verificar las condiciones de habilitación y el segundo para disparar la ejecución. Ambas declaraciones (brazos en lenguaje escalera) con el temporizador en el centro del brazo, pero no todos los autómatas admiten este tipo de programación. En el caso de una transición de tiempo cero ($d=0$) que se dispara instantáneamente y no necesita un elemento temporizador. Sin un temporizador, ambas declaraciones se escriben en un escalón, como se muestra en la Fig. 4.9.

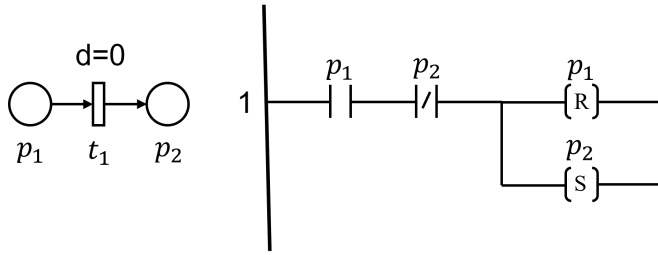


Figura 4.9: Traducción de transición de tiempo cero de RdPI a lenguaje escalera.

Lo mismo se aplica a la traducción de una transición con n lugares de entrada y m lugares de salida en el lenguaje escalera como se observa en la Fig. 4.10.

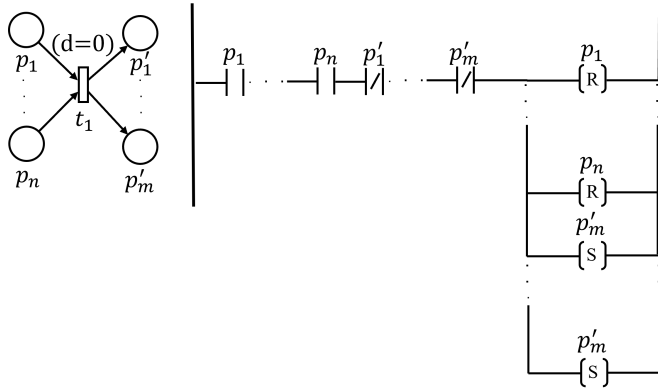


Figura 4.10: Traducción de transición de tiempo cero ($d=0$) con n lugares de entrada y m lugares de salida de RdPI a lenguaje escalera.

Una secuencia de condiciones definen la prioridad de las transiciones que compiten entre sí. La secuencia correcta es significativa en el caso de transiciones con al menos un lugar común de entrada como se observa en la Fig. 4.11, debido a que la declaración escrita en el lenguaje escalera se ejecuta antes que otras.

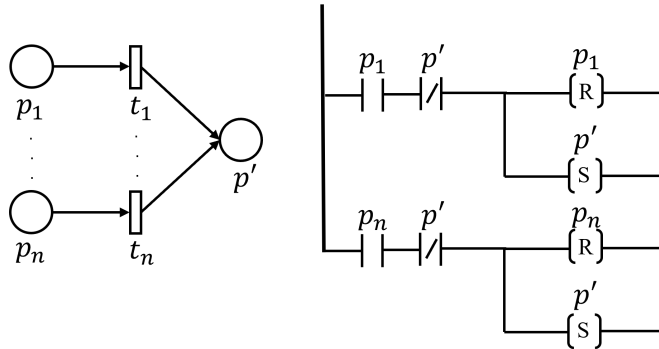


Figura 4.11: Traducción de transiciones de tiempo cero ($d = 0$) con n lugares de entrada y un lugar de salida de RdPI a lenguaje escalera.

4.4.2. Ejemplo de traducción de RdPI a lenguaje escalera

Para ejemplificar el proceso de traducción de una estructura de RdPI a una lógica de programación basada en lenguaje escalera para autómatas programables, se usa el proceso de descarga de botellas del sistema de llenado de botellas del Ejemplo 2.3.1. Como se muestra en la Fig. 4.12, el lazo cerrado del proceso de evacuación de botellas comprende de la RdPI de la planta, la RdP de la especificación y el conjunto de subestructuras, las cuales suministran la información para el diseño de la lógica de programación.

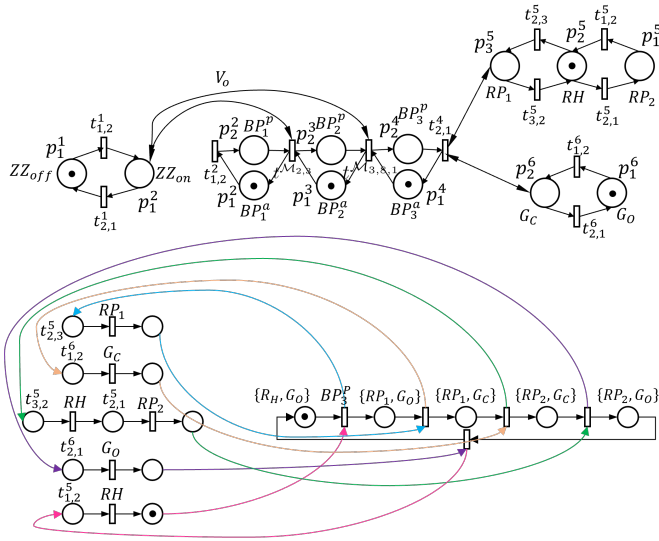


Figura 4.12: lazo cerrado del proceso de evacuación de botellas del Ejemplo 2.3.1.

Para traducir el lazo cerrado del proceso de evacuación de botellas se usa la metodología de traducción de RdPI a lenguaje escalera de la Sección 4.4.1. La primera parte corresponde a la representación de la condición inicial para el proceso de evacuación de botellas el cual se calcula a partir de los lugares y marcado inicial de la especificación. Así, el lugar marcado que corresponde al conjunto de etiquetas $\{R_H, G_O\}$ se representa como una bobina de salida en estado SET en el lenguaje escalera, y el conjunto de etiquetas de cada lugar no marcado con bobinas de salida en estado RESET. La Fig. 4.13 ilustra la condición inicial para el proceso de evacuación de botellas del Ejemplo 2.3.1.

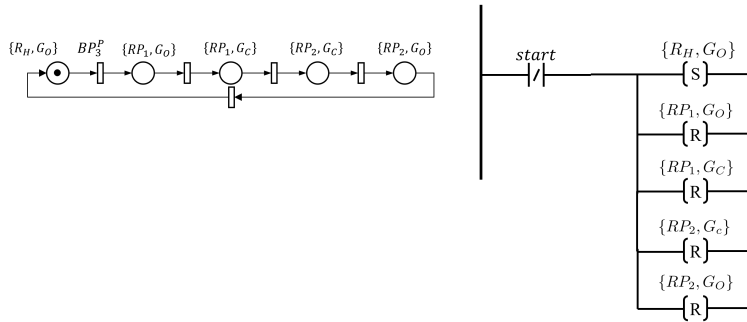


Figura 4.13: Diagrama escalera de la condición inicial para el proceso de evacuación de botellas.

Ahora, por cada transición de la especificación un brazo de lenguaje escalera se crea en la lógica, además por cada lugar de entrada a la transición se traducen como un contacto abierto y una bobina de salida en “Reset”, y los lugares de salida de la transición se traducen como una bobina en estado “Set”. Así, para el proceso de evacuación de botellas, la lógica de programación en lenguaje escalera se muestra en la Fig. 4.14.

Ahora, las transiciones del modelo de la planta son energizadas como se muestra en la Fig. 4.15, las cuales corresponden a las salidas físicas de la planta haciendo que evolucione el marcado en la planta y un seguimiento en la subestructura.

Como se observa en la Fig. 4.15, en el brazo número 1 del lenguaje escalera, la habilitación de la entrada p_1 hace que la bobina de la transición $t_{2,3}^5$ se energice y en el modelo de RdPI de la planta evolucione el marcado a RP_1 , entonces en el brazo 2 del lenguaje escalera la entrada RP_1 se activa haciendo que en la estructura evolucione el marcado desde p_1 a p_2 .

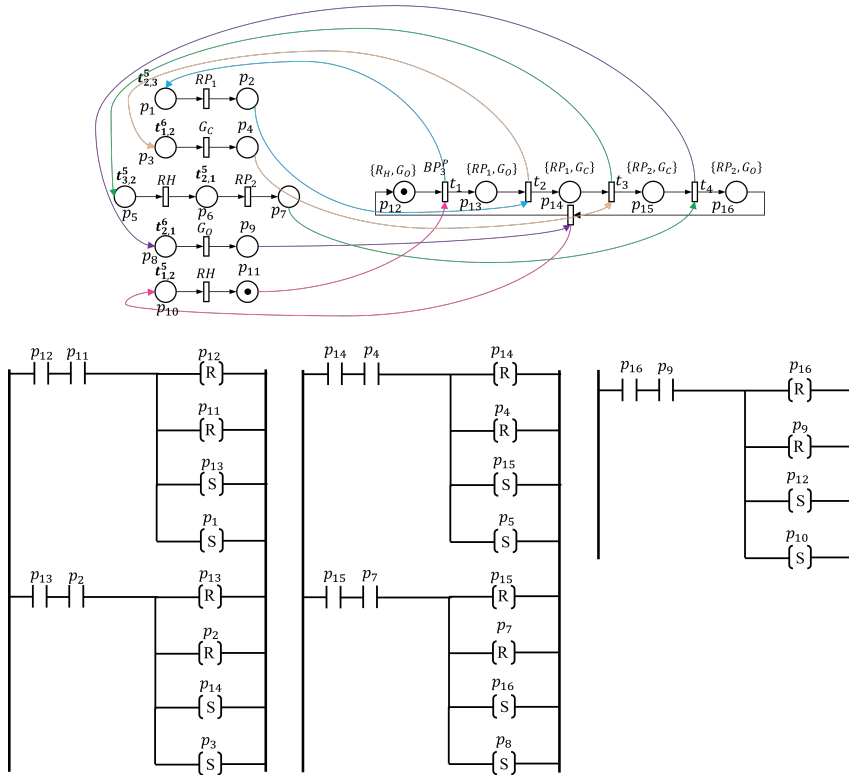


Figura 4.14: Diagrama escalera de la especificación del proceso de evacuación de botellas.

Capítulo 5

Aplicación de la metodología de traducción y recuperación de errores a sistema de envasado de bebidas carbonatadas

En este capítulo se presenta la implementación de la metodología de traducción de la información de un proceso industrial basado en la Norma ANSI ISA S5.1 a un modelo de RdPI como se presenta en [43, 44] y como producto final la obtención de la lógica de programación basada en lenguaje escalera para autómatas programables. La aplicación de las diferentes metodologías se aplican al proceso denominado *sistema de envasado para bebidas carbonatadas*, el cual se expone en forma detallada en las siguientes secciones.

5.1. Descripción del sistema de envasado de bebidas carbonatadas

El sistema de envasado de bebidas carbonatadas se describe mediante las narrativas de proceso y operación, las cuales presentan en forma de lenguaje natural las características y funcionamiento de los elementos que conforman el proceso, además del diagrama DTI que ilustra en forma gráfica la posición y descripción funcional de los elementos del sistema. La Fig. 5.1 ilustra el sistema

de envasado de bebidas carbonatadas, el cual está conformado por diferentes sub-procesos destinados al ingreso de la materia prima, mezclado de productos, preparación, y distribución del producto.

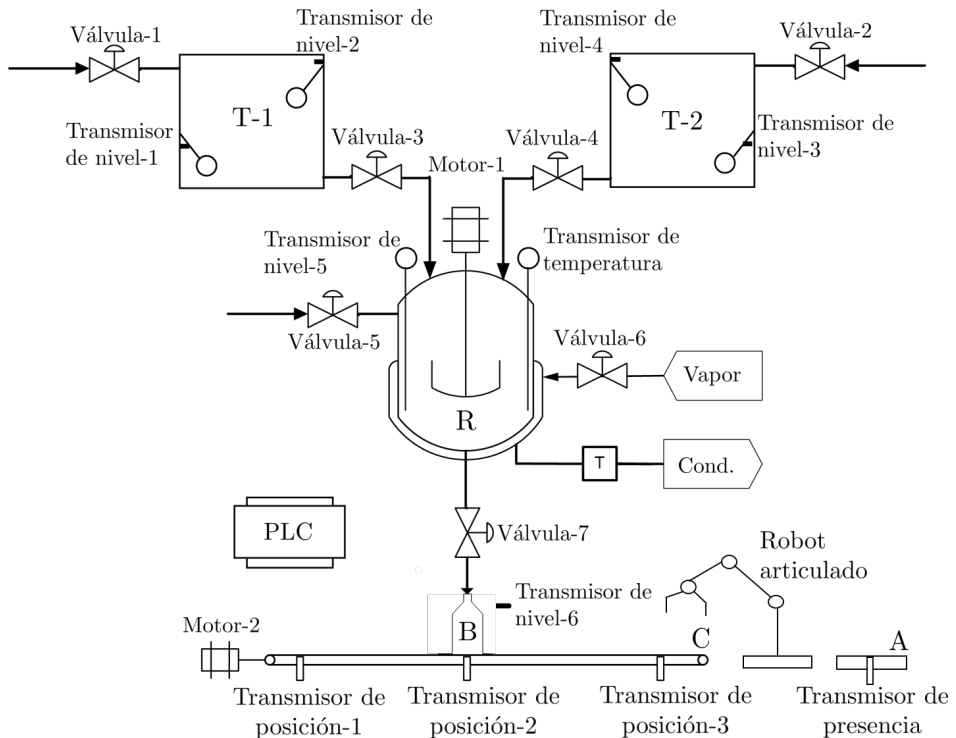


Figura 5.1: Sistema de envasado de bebidas carbonatadas.

La descripción del sistema de envasado de bebidas carbonatadas se presenta a continuación haciendo uso de la Norma ANSI ISA-S5.1. Esta descripción se realiza mediante el DTI del sistema que presenta cada uno de los elementos que intervienen en el proceso, además de los lazos de control a que pertenece cada elemento. La narrativa del proceso y operación describen en detalle las características, funcionamiento y operación del sistema. Para el proceso de envasado de bebidas carbonatadas el DTI se ilustra en la Fig. 5.2.

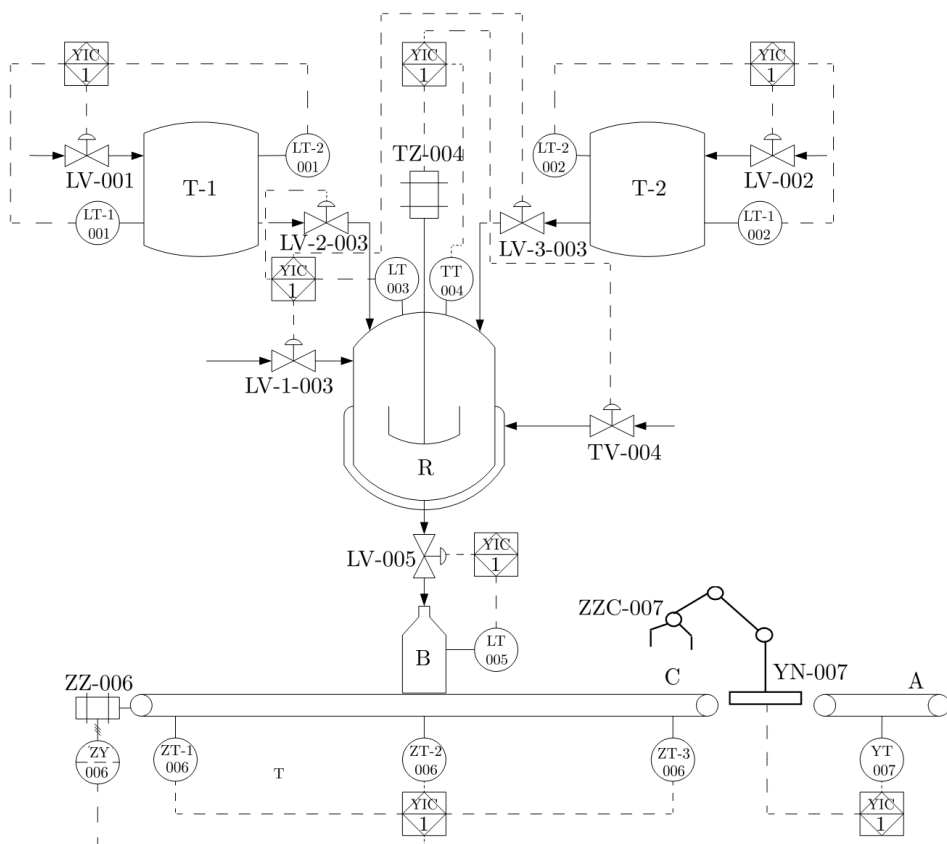


Figura 5.2: DTI Sistema de envasado de bebidas carbonatadas.

Narrativa del proceso para el sistema de envasado de bebidas carbonatadas. El sistema de envasado de bebidas carbonatadas está formado por 7 subprocesos, los cuales son establecidos por los lazos de control en el diagrama DTI. Así, el lazo 001 corresponde al proceso de llenado de jarabe, el proceso 002 al proceso de llenado de ácido cítrico, el proceso 003 es el llenado de producto en el reactor, el cuarto proceso corresponde a la cocción de la mezcla, el quinto proceso es el llenado de latas, el sexto proceso es el transporte de latas, y finalmente el séptimo proceso es la distribución de latas.

El primer proceso del sistema de envasado de bebidas carbonatadas corresponde al llenado de jarabe en el tanque *T1*, este sub-proceso está formado por dos

transmisores de nivel (LT-1-001, y LT-2-001) y una válvula de llenado (LV-001). Los transmisores de nivel LT-1-001, y LT-2-001 son usados con el fin de determinar el nivel bajo y alto del jarabe al interior del tanque respectivamente, y la válvula de llenado LV-001 tiene la función de permitir el paso del jarabe al interior del tanque *T1*.

El segundo proceso es el llenado de ácido cítrico en el tanque *T2*, este está conformado por dos transmisores de nivel (LT-1-002, y LT-2-002), y una válvula de llenado (LV-002). Los transmisores de nivel LT-1-002, y LT-2-002 determinan el nivel alto y bajo del ácido cítrico al interior del tanque *T2*, y la válvula LV-002 permite la entrada del ácido cítrico al interior del tanque *T2*.

El tercer proceso es el llenado de productos en el reactor *R*, este sub-proceso está compuesto de un transmisor de nivel (LT-003), una válvula de entrada de agua (LV-1-103), una válvula de entrada de jarabe (LV-2-003), y una válvula de entrada de ácido cítrico (LV-3-003). El transmisor de nivel LT-003 detecta el nivel de la mezcla de producto al interior del tanque, las válvulas LV-1-103, LV-2-103 y LV-3-103 permiten la entrada del jarabe, el ácido cítrico y el agua, respectivamente.

El cuarto proceso es el de cocción o tratamiento del producto en el reactor *R*, este proceso está compuesto por un transmisor de temperatura (TT-004), una válvula de entrada de vapor (TV-004) y un motor-agitador (TZ-004). El transmisor de temperatura TT-004 detecta la temperatura de la mezcla al interior del reactor, la válvula TV-004 permite la entrada de vapor a la camisa del reactor, y el motor-agitador TZ-004 homogeneiza la temperatura del producto al interior del reactor.

El quinto proceso es el llenado de producto en las latas *B*, este proceso está formado por un transmisor de nivel (LT-005) y una válvula de llenado de latas (LV-005). El transmisor de nivel LT-005 determina el nivel del producto al interior de la lata, y la válvula LV-005 permite el llenado de producto en las latas.

El sexto proceso es el transporte de latas sobre la banda transportadora *C*, el cual está formado por tres transmisores de posición (ZT-1-006, ZT-2-006, y ZT-3-006) y un motor (ZZ-006). Los transmisores de posición ZT-1-006, ZT-2-006, y ZT-3-006 determinan la posición del envase sobre la banda transportadora, y el motor ZZ-006 transporta latas desde la posición 1 a la posición 3.

El séptimo proceso es la distribución de latas, el cual está formado por un transmisor de presencia (YT-007), un brazo robótico (YN-007) y una pinza

(ZZC-007). El transmisor de presencia YT-007 determina la presencia de una lata en el almacén A y el conjunto brazo robótico YN-007 y pinza ZZC-007 lleva latas desde la posición 3 de la banda transportadora C al almacén A .

los transmisores de cada proceso definen variables controladas que establecen los lazos de control del sistema. Así, para el lazo 001 que corresponde al llenado del tanque $T1$, los transmisores LT-1-001 y LT-2-001 definen la variable controlada **NT1** (nivel del tanque 1) que tiene los estados $NT1_v$ (nivel del tanque 1 vacío), $NT1_b$ (nivel del tanque 1 bajo), $NT1_m$ (nivel del tanque 1 medio), y $NT1_a$ (nivel del tanque 1 alto); además, el elemento final LV-001 tiene los estados **cerrado** y **abierto**. Los transmisores de nivel del tanque $T2$ LT-1-002 y LT-2-002 definen la variable controlada **NT2** (nivel del tanque 2), esta variable tiene los estados $NT2_v$ (Nivel del tanque 2 vacío), $NT2_b$ (Nivel del tanque 2 bajo), $NT2_m$ (Nivel del tanque 2 medio), y $NT2_a$ (Nivel del tanque 2 alto), la válvula LV-002 tiene los estados **cerrado** y **abierto**. El transmisor LT-003 define la variable controlada **NR** (nivel del reactor), la cual tiene los estados NR_v (nivel de reactor vacío), NR_{bb} (nivel de reactor bajo-bajo), NR_b (nivel de reactor bajo), NR_m (nivel de reactor medio), NR_a (nivel de reactor alto) y NR_{aa} (nivel de reactor alto-alto), las válvulas LV-1-103, LV-2-103, y LV-3-103 tiene los estados **cerrado** y **abierto**. El transmisor de temperatura TT-004 define la variable controlada **TR** (temperatura del reactor), la cual tiene los estados TR_{bb} (temperatura del reactor bajo-bajo), TR_b (temperatura del reactor bajo), TR_m (temperatura del reactor media), TR_a (temperatura del reactor alta), y TR_{aa} (temperatura del reactor alta-alta); la válvula TV-004 tiene los estados **cerrado** y **abierto**. El transmisor de nivel LT-005 establece la variable controlada **NL** (Nivel de la lata), esta variable tiene los estados NL_v (nivel de lata vacío) y NL_l (nivel de lata lleno); la válvula LV-005 tiene los estados **cerrado** y **abierto**. Los transmisores de posición en la banda transportadora ZT-1-006, ZT-2-006, y ZT-3-006 definen las variables controladas **PL1** (posición de la lata 1), **PL2** (posición de la lata 2), y **PL3** (posición de la lata 3), respectivamente, los estados de PL1 son **ausencia** ($PL1_a$) y **presencia** ($PL1_p$), los estados de PL2 son **ausencia** ($PL2_a$) y **presencia** ($PL2_p$), y los estados de PL3 son **ausencia** ($PL3_a$) y **presencia** ($PL3_p$); el motor ZZ-006 tiene los estados **encendido** y **apagado**. Finalmente, el transmisor de presencia define la variable controlada **EL** (existencia de lata), los estados de la variable controlada son **ausencia** (EL_a) y **presencia** (EL_p); el brazo robótico YN-007 tiene los estados BR_h (brazo robótico home), BR_i (brazo robótico izquierda), y BR_d (brazo robótico derecha), la pinza ZZC-007 tiene los

estados P_a (pinza abierta) y P_c (pinza cerrada).

Cada proceso del sistema de envasado de bebidas carbonatadas tiene un funcionamiento propio como se describe a continuación:

- Si la válvula LV-001 cambia del estado cerrado a abierto, entonces el tanque se llena y la variable controlada NT1 cambia desde $NT1_v$ a $NT1_a$ pasando por $NT1_b$ y $NT1_m$.
- Si la válvula LV-002 cambia de la posición cerrado a la posición abierto, entonces el ácido cítrico ingresa al tanque y la variable controlada NT2 cambia desde $NT2_v$ a $NT2_a$ pasando por $NT2_b$ y $NT2_m$.
- Si la válvula LV-1-103 cambia su estado a abierto el reactor se llena con agua, si la válvula LV-2-103 cambia su estado a abierto, el reactor se llena con jarabe y el tanque T-1 se vacía, y si la válvula LV-3-103 cambia su estado a abierto, el reactor se llena con ácido cítrico y el tanque T-2 se vacía.
- Si la válvula TV-004 cambia del estado cerrado a abierto, entonces el vapor ingresa a la camisa térmica del reactor y la temperatura incrementa desde TR_{bb} a TR_{aa} , pasando por TR_b , TR_m , y TR_a .
- Si la válvula LV-005 cambia su estado de cerrado a abierto, la lata se llena de producto y la variable controlada NL cambia desde NL_v a NL_l .
- Si el motor cambia del estado apagado al estado encendido, las latas son transferidas desde la posición 1 a la posición 2 y de la posición 2 a la posición 3. Cada estado de las variables $PL1$, $PL2$ y $PL3$ cambian desde ausencia a presencia. Una barrera física se encuentra al final de la banda.
- Si el brazo robótico cambia su estado de BR_h a BR_i y la pinza cambia su estado desde P_a a P_c , y luego el brazo robótico cambia de BR_i a BR_d y la pinza de P_c a P_a , libera la banda transportadora de latas llenas de producto y el almacén cambia su estado de EL_a a EL_p .

Narrativa de operación para el sistema de envasado de bebidas carbonatadas. La narrativa de operación describe la secuencia de operaciones que debe realizar el proceso para el cumplimiento de las tareas propuestas por el ingeniero de planta.

El sistema de envasado de bebidas carbonatadas tiene las siguientes condiciones iniciales: los tanques T1, T2, el reactor y la lata inician en estado vacío, por lo tanto las variables controladas $NT1$, $NT2$, NR , y NL se encuentran inicialmente en los estados $NT1_v$, $NT2_v$, NR_v , y NL_v respectivamente. La temperatura inicial dentro del reactor es baja-baja, entonces la variable controlada TR tiene el estado inicial TR_{bb} . La banda transportadora inicia sin latas, por tanto el estado de cada variable controlada $PL1$, $PL2$ y $PL3$ es $PL1_a$, $PL2_a$, y $PL3_a$. Finalmente, el almacén inicia vacío, por lo tanto el estado inicial de la variable controlada EL es EL_a .

A continuación se presentan las tareas que debe realizar el sistema de envasado de bebidas carbonatadas para cada uno de los procesos que lo integran.

- **Llenar el reactor con jarabe, agua y ácido cítrico.** Esta tarea se realiza siempre y cuando el nivel del reactor se encuentre en el estado NR_v , el nivel del jarabe dentro del tanque uno este en el estado $NT1_a$, y el nivel del ácido cítrico este en el estado $NT2_a$.
- **Calentar la mezcla de producto.** Esta tarea se realiza cuando la temperatura del reactor se encuentre en el estado TR_{bb} y el nivel del reactor en el estado NR_a .
- **Llenado de latas.** Esta tarea se realiza si el nivel del reactor se encuentra en el estado NR_a y la temperatura del producto al interior del reactor está en el estado TR_a , además que la posición de la lata sobre la banda transportadora se encuentre en el estado $PL2_p$ y el nivel de la lata en el estado NL_v .
- **Transporte de latas para llenado.** Esta tarea se realiza si en la posición 2 de la banda transportadora está en el estado $PL2_a$.
- **Trasladar latas a la posición 3.** Esta tarea se realiza si en la posición 2 de la banda transportadora el estado es $PL2_p$, además que el nivel de la lata se encuentre en el estado NL_l , y la posición 3 esté en el estado $PL3_a$.
- **Evacuación de latas.** Esta tarea se realiza si existe una lata en la posición 3 $PL3_p$ y si el almacén no hay lata EL_a .

5.2. Estructura de la narrativa del proceso del sistema de envasado de bebidas carbonatadas.

La estructura de la narrativa del proceso resume en tablas las características y funcionamiento de los elementos del sistema. A continuación se presenta la estructura de tablas para el sistema de envasado de bebidas carbonatadas iniciando por la **descripción de elementos del sistema de envasado de bebidas carbonatadas** como se muestra en la la Tabla 5.1.

Tabla 5.1: Descripción de elementos del sistema de envasado de bebidas carbonatadas.

VARIABLES CONTROLADAS & ACTUADORES	RANGO	ESTADOS	ESTADO INICIAL
LV-001 (Válvula1)	2	cerrado, abierto	cerrado
LV-002 (Válvula2)	2	cerrado, abierto	cerrado
LV-1-003 (Válvula5)	2	cerrado, abierto	cerrado
LV-2-003 (Válvula3)	2	cerrado, abierto	cerrado
LV-3-003 (Válvula4)	2	cerrado, abierto	cerrado
TV-004 (Válvula6)	2	cerrado, abierto	cerrado
TZ-004 (Motor1)	2	apagado, encendido	apagado
LV-005 (Válvula7)	2	cerrado, abierto	cerrado
ZZ-006 (Motor2)	2	apagado, encendido	apagado
YN-007 (Brazo Robótico)	3	BR_h, BR_i, BR_d	BR_h
ZZC-007 (Pinza)	2	P_a, P_c	P_a
Nivel del tanque 1 ($NT1$)	4	$NT1_v, NT1_b, NT1_m, NT1_a$	$NT1_v$
Nivel del tanque 2 ($NT2$)	4	$NT2_v, NT2_b, NT2_m, NT2_a$	$NT2_v$
Nivel del reactor (NR)	6	$NR_v, NR_{bb}, NR_b, NR_m, NR_a, NR_{aa}$	NR_v
Temperatura del reactor (TR)	5	$TR_{bb}, TR_b, TR_m, TR_a, TR_{aa}$	TR_{bb}
Posición de la lata 1 ($PL1$)	2	$PL1_a, PL1_p$	$PL1_a$
Posición de la lata 2 ($PL2$)	2	$PL2_a, PL2_p$	$PL2_a$
Posición de la lata 3 ($PL3$)	2	$PL3_a, PL3_p$	$PL3_a$
Nivel de la lata (NL)	2	NL_v, NL_l	NL_v
Existencia de lata (EL)	2	EL_a, EL_p	EL_a

Ahora, la tabla **comportamiento de elementos del sistema de envasado de bebidas carbonatadas** describe el comportamiento de los elementos (actuador o variable controlada) para sistema de envasado de bebidas carbonatadas, la cual se presenta en la Tabla 5.2.

Tabla 5.2: Tabla de comportamiento de elementos del sistema de envasado de bebidas carbonatadas

LV-001	cerrado	abierto
cerrado		x
abierto	x	

LV-002	cerrado	abierto
cerrado		x
abierto	x	

LV-1-003	cerrado	abierto
cerrado		x
abierto	x	

LV-2-003	cerrado	abierto
cerrado		x
abierto	x	

LV-3-003	cerrado	abierto			
cerrado		x			
abierto	x				
TV-004	cerrado	abierto			
cerrado		x			
abierto	x				
TZ-004	apagado	encendido			
apagado		x			
encendido	x				
LV-005	cerrado	abierto			
cerrado		x			
abierto	x				
ZZ-006	apagado	encendido			
apagado		x			
encendido	x				
ZZC-007	P_a	P_c			
P_a		x			
P_c	x				
PL1	$PL1_a$	$PL1_p$			
$PL1_a$		x			
$PL1_p$	x				
PL2	$PL2_a$	$PL2_p$			
$PL2_a$		x			
$PL2_p$	x				
PL3	$PL3_a$	$PL3_p$			
$PL3_a$		x			
$PL3_p$	x				
NL	NL_v	NL_l			
NL_v		x			
NL_l	x				
EL	EL_a	EL_p			
EL_a		x			
EL_p	x				
NT1	$NT1_v$	$NT1_b$	$NT1_m$	$NT1_a$	
$NT1_v$		x			
$NT1_b$	x		x		
$NT1_m$		x		x	
$NT1_a$			x		
NT2	$NT2_v$	$NT2_b$	$NT2_m$	$NT2_a$	
$NT2_v$		x			
$NT2_b$	x		x		
$NT2_m$		x		x	
$NT2_a$			x		
NR	NR_v	NR_b	NR_m	NR_a	NR_aa
NR_v		x			
NR_b	x		x		
NR_m		x		x	
NR_a			x		x
NR_aa				x	
TR	TR_b	TR_m	TR_a	TR_aa	
TR_b		x			
TR_m	x		x		
TR_a		x		x	
TR_aa			x		
YN-007	BR_i	BR_b	BR_d		
BR_i		x			
BR_b	x		x		
BR_d		x			

Luego, la interacción entre las variables controladas y los actuadores es capturada en las tablas de permisiva y relación síncrona. La tabla **relación permisiva** ilustra la evolución de los estados de variables controladas cuando los actuadores se encuentran en algún estado en particular. La Tabla 5.3 ilustra la relación permisiva entre actuadores y variables controladas para el sistema de envasado de bebidas carbonatadas.

86CAPÍTULO 5. APLICACIÓN DE LA METODOLOGÍA DE TRADUCCIÓN

Tabla 5.3: Relación permisiva para el sistema de envasado de bebidas carbonatadas.

		Variables controladas								
		NT1	NT2	NR	TR	PL1	PL2	PL3	NL	EL
Actuador	Estado	Eventos								
LV-001	cerrado									
	abierto	NT1 _v a NT1 _b NT1 _b a NT1 _m NT1 _m a NT1 _a								
LV-002	cerrado									
	abierto		NT2 _v a NT2 _b NT2 _b a NT2 _m NT2 _m a NT2 _a							
LV-1-003	cerrado									
	abierto			NR _v a NR _{bb} NR _{bb} a NR _b NR _b a NR _m NR _m a NR _a NR _a a NR _{aa}						
LV-2-003	cerrado									
	abierto	NT1 _a a NT1 _m NT1 _m a NT1 _b NT1 _b a NT1 _v		NR _v a NR _{bb} NR _{bb} a NR _b NR _b a NR _m NR _m a NR _a NR _a a NR _{aa}						
LV-3-003	cerrado									
	abierto		NT2 _a a NT2 _m NT2 _m a NT2 _b NT2 _b a NT2 _v	NR _v a NR _{bb} NR _{bb} a NR _b NR _b a NR _m NR _m a NR _a NR _a a NR _{aa}						
TV-004	cerrado									
	abierto				TR _{bb} a TR _b TR _b a TR _m TR _m a TR _a TR _a a TR _{aa}					
TZ-004	apagado									
	encendido				TR _{bb} a TR _b TR _b a TR _m TR _m a TR _a TR _a a TR _{aa}					
LV-005	cerrado									
	abierto			NR _{aa} a NR _a NR _a a NR _m NR _m a NR _b NR _b a NR _{bb} NR _{bb} a NR _v					NL _v a NL _l	
ZZ-006	apagado									
	encendido					PL1 _a a PL1 _p	PL2 _a a PL2 _p	PL3 _a a PL3 _p		
YN-007	izquierda							PL3 _p a PL3 _a		
	inicio									EL _a a EL _p
ZZC-007	derecha									EL _a a EL _p
	abierto									EL _a a EL _p
ZZC-007	cerrada							PL3 _p a PL3 _a		

La tabla **Relación síncrona** en la metodología de traducción indica cuándo dos o más variables controladas evolucionan simultáneamente y representan el mismo significado físico. La tabla de relaciones síncronas para el sistema de envasado de bebidas carbonatadas se muestra en la Tabla 5.4.

Tabla 5.4: Relación síncrona para el sistema de envasado de bebidas carbonatadas.

Variables y Eventos	PL1		PL2		PL3	
	$PL1_a$ to $PL1_p$	$PL1_p$ to $PL1_a$	$PL2_a$ to $PL2_p$	$PL2_p$ to $PL2_a$	$PL3_a$ to $PL3_p$	$PL3_p$ to $PL3_a$
PL1	$PL1_a$ to $PL1_p$					
	$PL1_p$ to $PL1_a$		X			
PL2	$PL2_a$ to $PL2_p$	X				
	$PL2_p$ to $PL2_a$				X	
PL3	$PL3_a$ to $PL3_p$			X		
	$PL3_p$ to $PL3_a$					

Estructura de la narrativa de operación (SNO).

La estructura de la narrativa de operación resume en tablas las características de las variables controladas y las tareas que debe cumplir el sistema.

La tabla **Rango de operación para el sistema de envasado de bebidas carbonatadas** lista las variables controladas involucradas en el proceso y sus rangos de operación dadas por el DTI y la narrativa de operación del proceso como se muestra en la Tabla 5.5.

Tabla 5.5: Rangos de operación para el sistema de envasado de bebidas carbonatadas.

Elemento	Rango de operación
NT1	$NT1_v, NT1_b, NT1_m, NT1_a$
NT2	$NT2_v, NT2_b, NT2_m, NT2_a$
NR	$NR_v, NR_{bb}, NR_b, NR_m, NR_a, NR_{aa}$
TR	$TR_{bb}, TR_b, TR_m, TR_a, TR_{aa}$
ZZC-007	P_a, P_c
YN-007	BR_i, BR_h, BR_d
PL1	$PL1_a, PL1_p$
PL2	$PL2_a, PL2_p$
PL3	$PL3_a, PL3_p$
NL	NL_v, NL_l
EL	EL_a, EL_p

La tabla **Condiciones de operación** describe las tareas que debe cumplir los procesos del sistema definidos por los lazos de control en el DTI y la narrativa de operación del sistema. La Tabla 5.6 describe las tareas que deben realizar los procesos del sistema de envasado de bebidas carbonatadas.

Tabla 5.6: Condiciones de operación para el sistema de envasado de bebidas carbonatadas.

	Estados de precondición	Condiciones	Estados alcanzados
Llenado tanque T1	$\{NT1_v\}$		$\{NT1_a\}$
	$\{NT1_a\}$	$\sim NR_l$	$\{NT1_v\}$
Llenar tanque T2	$\{NT2_v\}$		$\{NT2_a\}$
	$\{NT2_a\}$	$\sim NR_l$	$\{NT2_v\}$
Llenar reactor	$\{NR_v\}$	$\{\sim NT1_v, \sim NT2_v\}$	$\{NR_a\}$
	$\{NR_a\}$	$\{NL_v, PL2_p, TR_a\}$	$\{NR_v\}$
Calentar reactor	$\{TR_{bb}\}$	NR_a	$\{TR_a\}$
	$\{TR_a\}$	$\{NL_v, PL2_p\}$	$\{TR_{bb}\}$
Descargar latas	$\{BR_H, P_a\}$	$PL3_p$	$\{BR_i, P_a\}$
	$\{BR_i, P_a\}$		$\{BR_i, P_c\}$
	$\{BR_i, P_c\}$		$\{BR_d, P_c\}$
	$\{BR_d, P_c\}$	EL_a	$\{BR_d, P_a\}$
	$\{BR_d, P_a\}$		$\{BR_h, P_a\}$
Llenar latas	$\{PL2_a, NL_v\}$		$\{PL2_p, NL_l\}$
	$\{PL2_p, NL_l\}$	$PL3_a$	$\{PL2_a, NL_v\}$

5.3. Estructura del grafo del proceso del sistema de envasado de bebidas carbonatadas

La estructura del grafo del proceso para el sistema de bebidas carbonatadas se presenta en la Fig. 5.3, donde cada elemento del sistema listado en la Tabla 5.1 (descripción de elementos del sistema de envasado de bebidas carbonatadas) es representado por un vértice n_i de la estructura del grafo, y la relación entre cada elemento se presenta como un arco continuo o discontinuo según la información dada por las Tablas 5.3 (tabla de relaciones permisivas) y 5.4 (tabla de relaciones síncronas) del sistema de envasado de bebidas carbonatadas.

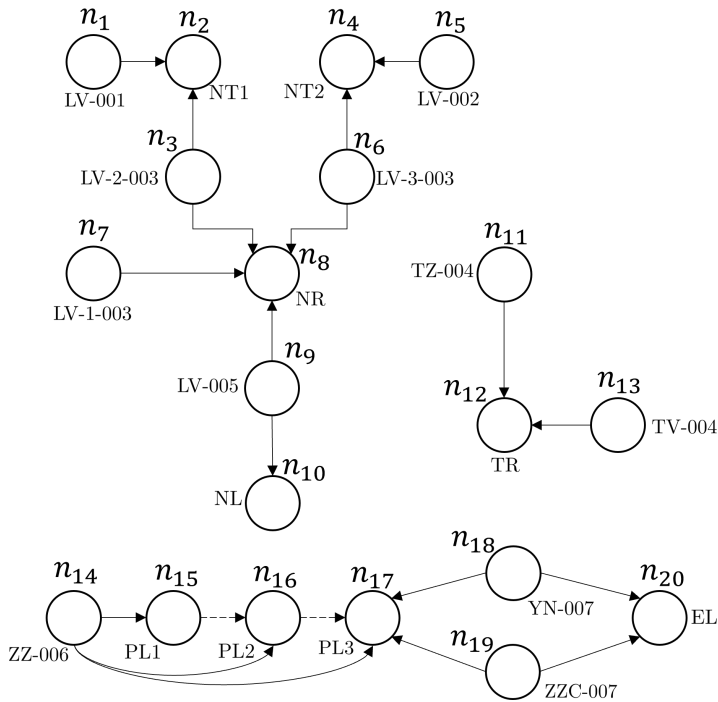


Figura 5.3: Estructura de grafo del sistema de envasado de bebidas carbonatadas.

5.4. Modelo en RdPI del proceso del sistema de envasado de bebidas carbonatadas.

La construcción del modelo en RdPI del sistema de envasado de bebidas carbonatadas obedece a la información presente en la Tabla 5.1 y a la estructura de grafo de la Fig. 5.3. Así, cada vértice de la estructura de grafo y las características de cada elemento dada en la tabla descripción de elementos para el sistema de envasado de bebidas carbonatadas, se traduce en módulos de RdP como se muestra en la Fig. 5.5.

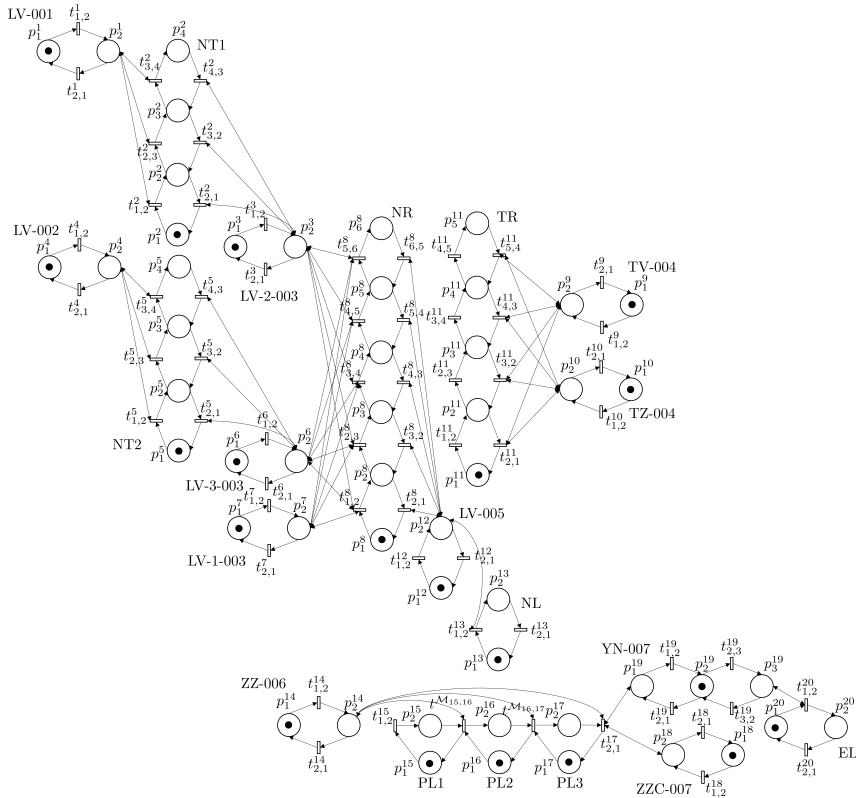


Figura 5.4: Estructura de RdPI de la planta del sistema de envasado de bebidas carbonatadas.

5.5. Modelo en RdP de la especificación para el sistema de envasado de bebidas carbonatadas

El modelo en RdP de la especificación para el sistema de envasado de bebidas carbonatadas se construye con la información suministrada por las Tablas 5.5 (rangos de operación) y 5.6 (condiciones de operación), donde cada tarea definida por el ingeniero del proceso se traduce en una RdP, la cual establece la secuencia de operaciones que debe seguir su ejecución como se observa en la Fig. 5.5.

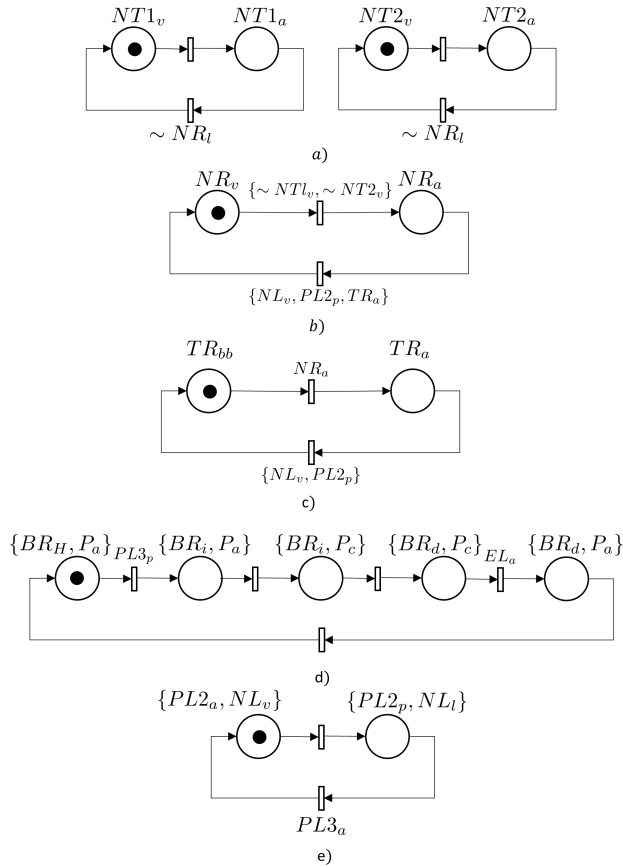


Figura 5.5: Estructura de RdPI de la especificación del sistema de envasado de bebidas carbonatadas. La figura a) muestra la secuencia de operaciones para las tareas llenado tanque T1 y T2; b) RdP de operación para el llenado del reactor; c) secuencia de operación para la tarea calentar reactor; d) secuencia de operación para la tarea descargar latas; e) secuencia de operaciones para la tarea llenar latas.

5.6. Lazo de control del sistema de envasado de bebidas carbonatadas

La construcción del lazo cerrado del sistema de bebidas carbonatadas se construye a partir de la información dada por los modelos de RdPI de la planta y la especificación. Al aplicar el Algoritmo 5 a los modelos, el conjunto de subestructuras y lazos dirigidos se establecen como se ilustra en la Fig. 5.6.

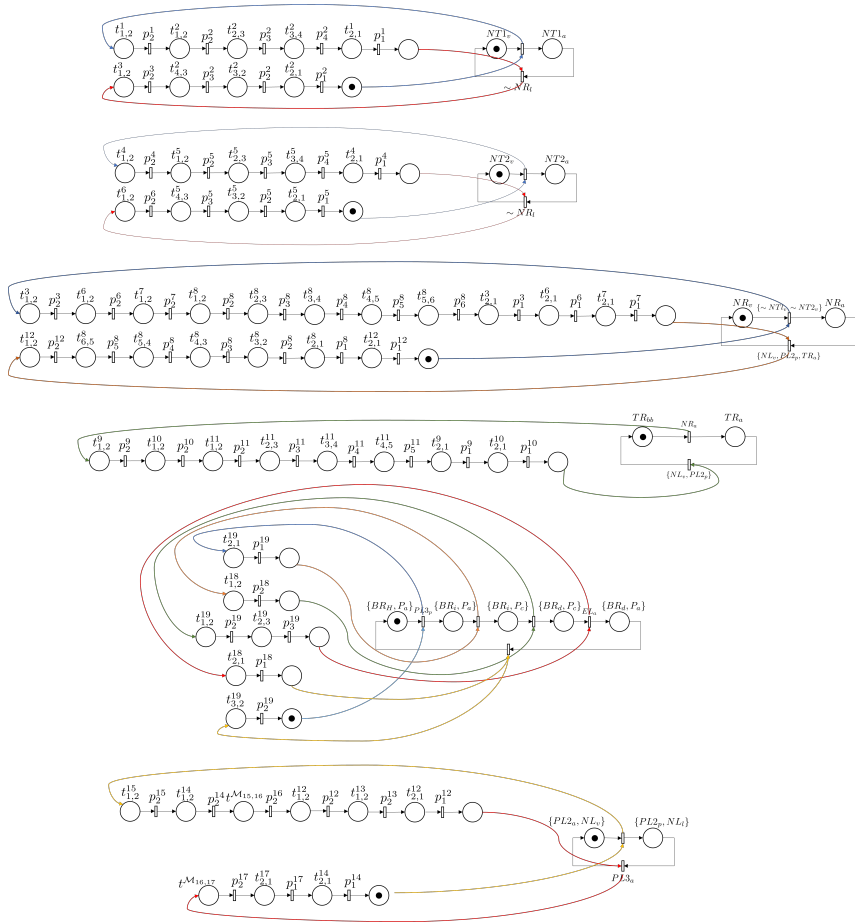


Figura 5.6: Lazo cerrado del sistema de envasado de bebidas carbonatadas.

5.7. Traducción de estructura de RdPI a lenguaje escalera para el sistema de envasado de bebidas carbonatas

La traducción del lazo cerrado del sistema de envasado de bebidas carbonatadas a lenguaje escalera responde a la metodología de traducción de RdPI a lenguaje escalera de la Sección 4.4.1. La primera parte de la traducción corresponde a la representación de la condición inicial la cual se calcula a partir de los lugares y el marcado inicial de la especificación como se muestra en la Fig. 5.7.

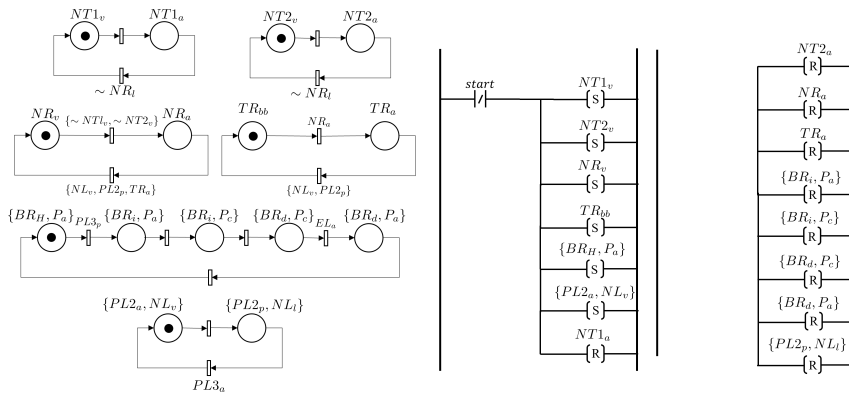


Figura 5.7: Representación de la condición inicial en lenguaje escalera del sistema de envasado de bebidas carbonatadas.

Luego, la representación de la lógica de programación en lenguaje escalera para el sistema de envasado de bebidas carbonatadas se muestra en las Figs. 5.8, 5.9, 5.10 y 5.11, donde por cada transición de la especificación se establece un brazo del lenguaje escalera, cada lugar de entrada a la transición se traducen como un contacto abierto y una bobina de salida en "Reset", y los lugares de salida de la transición se traducen como una bobina en estado "Set".

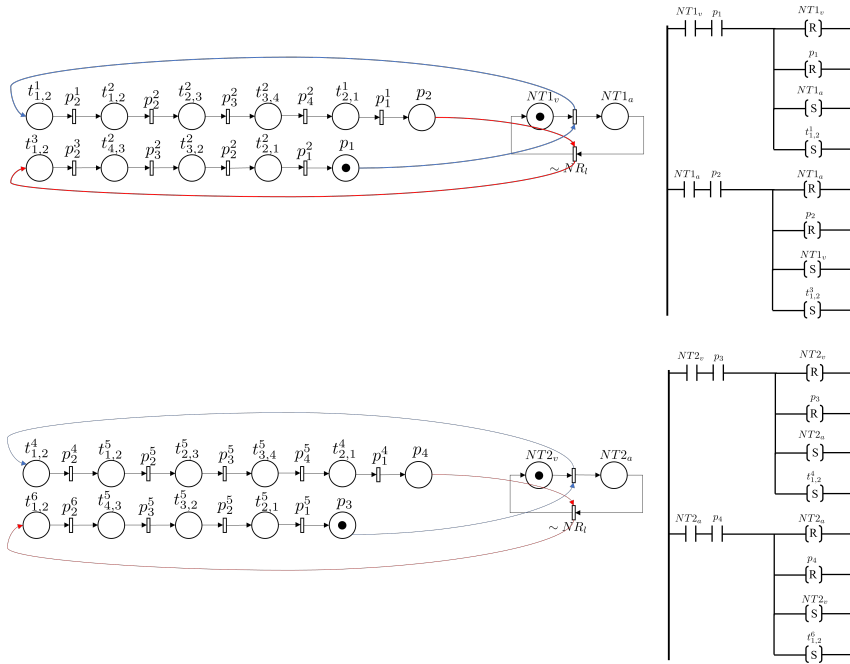
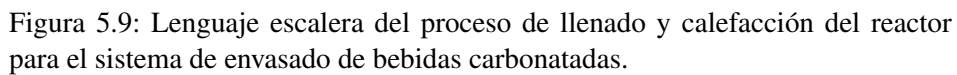


Figura 5.8: Lenguaje escalar del proceso de llenado de los tanques T1 y T2 del sistema de envasado de bebidas carbonatadas.



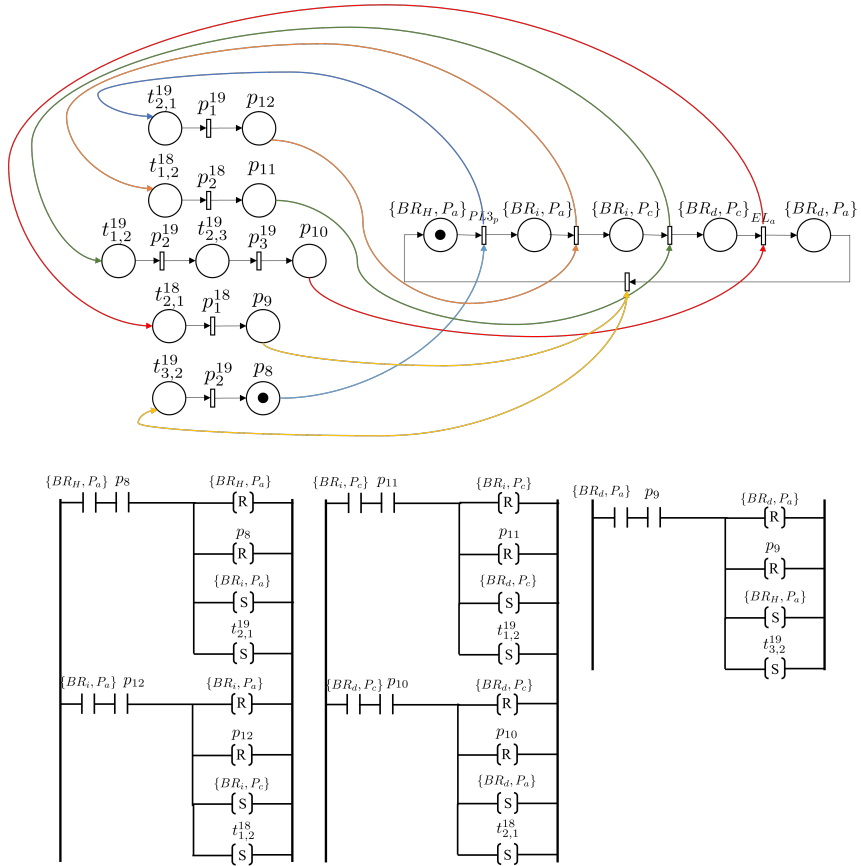


Figura 5.10: Lenguaje escalera del proceso de liberación de banda transportadora para el sistema de envasado de bebidas carbonatadas.

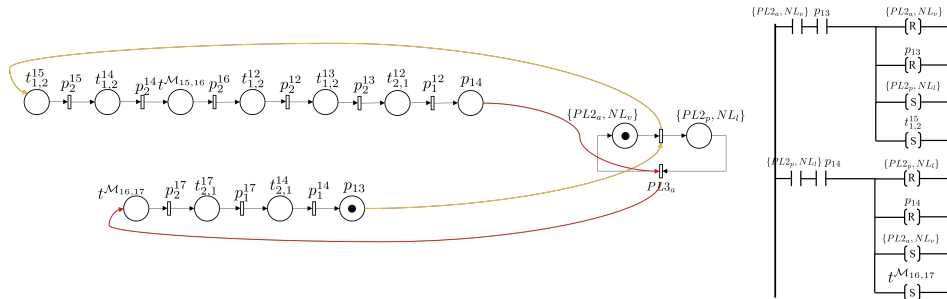


Figura 5.11: Lenguaje escalera del proceso llenado de latas para el sistema de envasado de bebidas carbonatadas.

Finalmente, las transiciones del modelo de RdPI de la planta del sistema de bebidas carbonatadas son energizadas como se muestra en la Fig. 5.12.

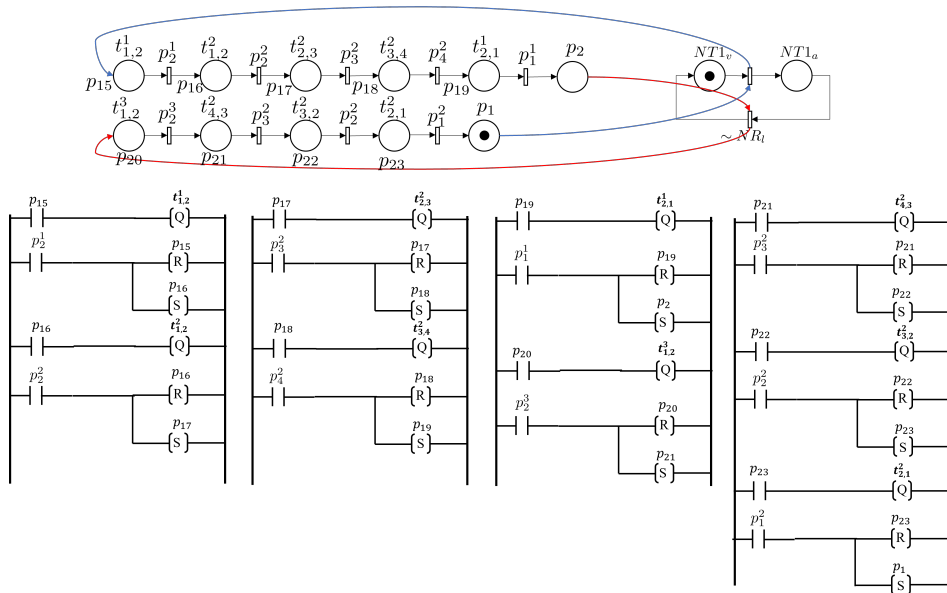


Figura 5.12: Lenguaje escalera de transiciones energizadas para el proceso de llenado de tanque T1 del sistema de envasado de bebidas carbonatadas.

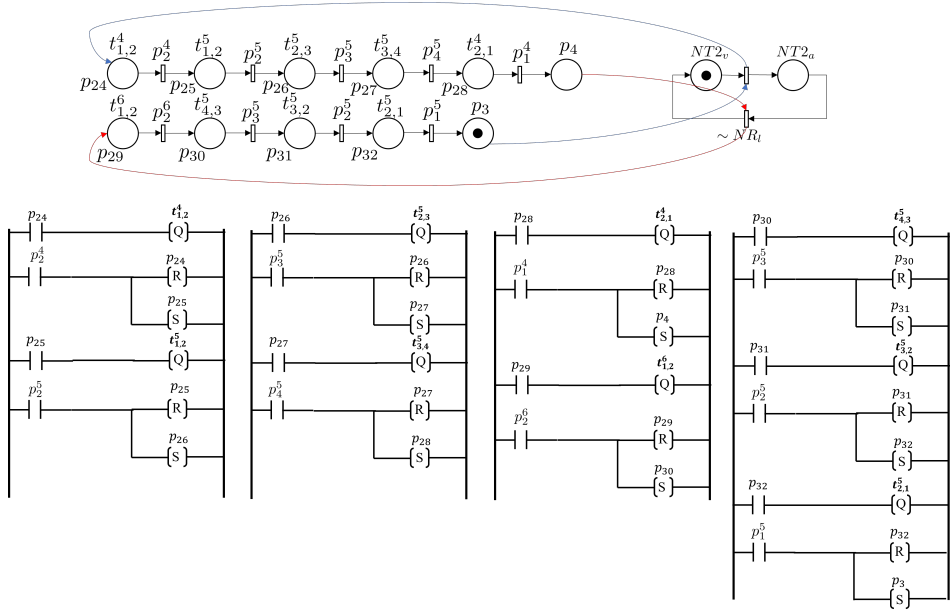


Figura 5.13: Lenguaje escalera de transiciones energizadas para el proceso de llenado de tanque T2 del sistema de envasado de bebidas carbonatadas.

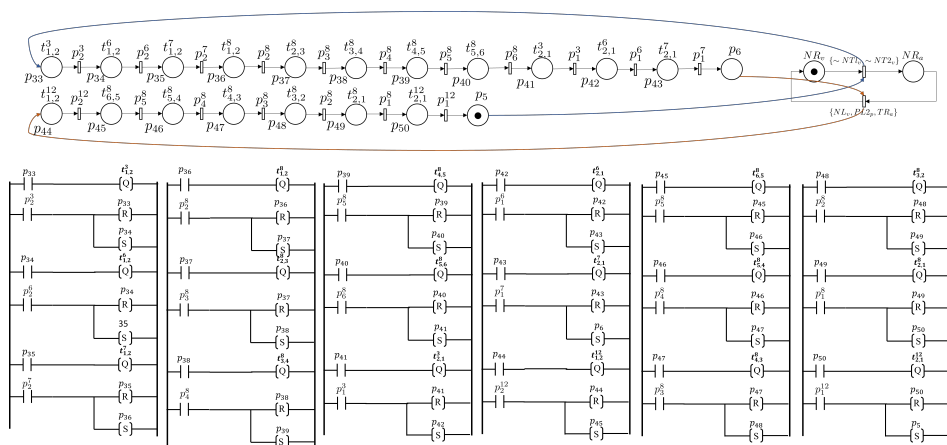


Figura 5.14: Lenguaje escalera de transiciones energizadas para el proceso de llenado de reactor del sistema de envasado de bebidas carbonatadas.

Para ilustrar la efectividad y aplicabilidad de la metodología de traducción propuesta, se presenta el sistema de envasado de bebidas carbonatadas como caso de estudio. A través del ejemplo propuesto, es posible ilustrar la aplicabilidad de los principales resultados como la caracterización de los elementos y el funcionamiento del sistema mediante una estructura de tablas, es posible realizar la captura temprana de errores de omisión o completitud a través de la estructura de grafos, se realiza la representación del modelo completo en RdPI que permita la traducción confiable y libre de errores de la lógica de programación en lenguaje escalera.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este trabajo se presentó una metodología para traducir la información de un proceso industrial dada por un diagrama de tuberías e instrumentación DTI y una narrativa de proceso y operación basado en la Norma ANSI ISA S5.1, a una estructura basada en Redes de Petri Interpretadas con propósitos de análisis y control. Además de la metodología de traducción, se propone una metodología para capturar y recuperar errores por omisión e inconsistencia en la información del proceso. Las contribuciones del trabajo se enumeran a continuación:

- Mediante el uso de la herramienta formal propuesta, los algoritmos de síntesis de control existentes se pueden usar para diseñar e implementar automáticamente controladores para hacer cumplir el comportamiento requerido, reduciendo posibles inconsistencias y otros errores que se introducen con frecuencia en los procedimientos manuales.
- La captura y recuperación de los errores se actualiza en las narrativas, tablas y gráficos, por lo que la información del proceso y los modelos de RdPI resultantes son consistentes entre ellos.
- Los modelos de RdPI calculados (planta y especificación) son usados para diseñar e implementar automáticamente un controlador, el cual se emplea para generar el diagrama escalera correspondiente de forma automática.

- La elección del PLC para implementar la lógica de programación en lenguaje escalera se debe a que es un dispositivo potente, flexible y confiable, además que su uso se extiende ampliamente en la industria para la automatización de procesos y la automatización de la fabricación.
- Se presentó la traducción de la información de un proceso industrial a una estructura de grafos, la cual permite realizar la detección temprana de errores por omisión y la verificación de la completitud de la información del sistema.
- La selección de la Norma ANSI ISA S5.1 como herramienta fundamental para la metodología de traducción, permite establecer las características y funcionamiento del sistema, información clave para la construcción de modelos de RdPI, el lazo cerrado del sistema, y por consiguiente una programación en lenguaje escalera confiable y libre de errores.

6.2. Trabajo futuro

Algunos de los temas que quedaron en este trabajo para futuras investigaciones son los siguientes:

- Desarrollo de un software que permita la automatización del proceso de traducción propuesto en este trabajo, con el propósito de reducir los tiempos de programación por parte del ingeniero del proceso.
- Validación de la metodología de traducción y recuperación de errores en entornos industriales de fabricación.

Bibliografía

- [1] A. Ibarz and G. V. Barbosa-Cánovas, *Unit operations in food engineering*. CRC press, 2002.
- [2] B. R. Mehta and Y. J. Reddy, *Industrial process automation systems: design and implementation*. Butterworth-Heinemann, 2014.
- [3] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, “Natural language processing: an introduction,” *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.
- [4] D. Popescu, S. Rugaber, N. Medvidovic, and D. M. Berry, “Reducing ambiguities in requirements specifications via automatically created object-oriented models,” in *Monterey Workshop* (P. B. and M. C., eds.), vol. 5320 of *Lecture Notes in Computer Science*, pp. 103–124, Springer, 2007.
- [5] G. Lami, S. Gnesi, F. Fabbrini, M. Fusani, and G. Trentanni, “An automatic tool for the analysis of natural language requirements,” *Informe técnico, CNR Information Science and Technology Institute, Pisa, Italia, Setiembre*, 2004.
- [6] C. Rupp and R. Goetz, “Linguistic methods of requirements-engineering (nlp),” in *Proceedings of the European Software Process Improvement Conference (EuroSPI)*, pp. 7–11, 2000.
- [7] F. Pittke, H. Leopold, and J. Mendling, “Automatic detection and resolution of lexical ambiguity in process models,” *IEEE Transactions on Software Engineering*, vol. 41, no. 6, pp. 526–544, 2015.

- [8] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
- [9] I. Rivera-Rangel, A. Ramírez-Treviño, L. I. Aguirre-Salas, and J. Ruiz-Leon, "Geometrical characterization of observability in interpreted petri nets," *Kybernetika*, vol. 41, no. 5, pp. 553–574, 2005.
- [10] A. Ramirez-Trevino, E. Ruiz-Beltran, I. Rivera-Rangel, and E. Lopez-Mellado, "Online fault diagnosis of discrete event systems. a petri net-based approach," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 31–39, 2007.
- [11] M. Skoldstam, K. Akesson, and M. Fabian, "Modeling of discrete event systems using finite automata with variables," in *Decision and Control, 2007 46th IEEE Conference on*, pp. 3387–3392, IEEE, 2007.
- [12] N. Kim, D. Shin, R. Wysk, and L. Rothrock, "Using finite state automata (fsa) for formal modelling of affordances in human-machine cooperative manufacturing systems," *International Journal of Production Research*, vol. 48, no. 5, pp. 1303–1320, 2010.
- [13] H. A. Awad, "Modeling of industrial productivity processes," *J. Eng. Sci. JES*, vol. 10, pp. 763–781, 2010.
- [14] L.-P. Chung and C.-T. Chang, "Petri-net models for comprehensive hazard analysis of mocvd processes," *Computers & chemical engineering*, vol. 35, no. 2, pp. 356–371, 2011.
- [15] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.
- [16] Y. Chen, Z. Li, and M. Zhou, "Optimal supervisory control of flexible manufacturing systems by petri nets: A set classification approach," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 549–563, 2013.
- [17] A. Giua and M. Silva, "Modeling, analysis and control of discrete event systems: a petri net perspective," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1772–1783, 2017.

- [18] A. Dideban and H. Zeraatkar, “Petri net controller synthesis based on decomposed manufacturing models,” *ISA transactions*, vol. 77, pp. 90–99, 2018.
- [19] C. R. Vázquez, J. A. Gómez-Castellanos, and A. Ramírez-Treviño, “Petri nets tracking control for electro-pneumatic systems automation,” in *Informatics in Control, Automation and Robotics (ICINCO) 2018* (O. Gusikhin and K. Madani, eds.), vol. 613 of *Lecture Notes in Electrical Engineering*, pp. 503–525, Springer, 2019.
- [20] A. Santoyo, I. Jimenez-Ochoa, and A. Ramírez-Treviño, “A complete cycle for controller design in discrete event systems,” in *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236)*, vol. 4, pp. 2688–2693, IEEE, 2001.
- [21] D. Guevara-Lozano, C. R. Vázquez, and A. Ramírez-Treviño, “Towards decentralized control for Petri nets,” in *24th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 428–435, 2019.
- [22] Z. Li, N. Wu, and M. Zhou, “Deadlock control of automated manufacturing systems based on petri nets—a literature review,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 437–462, 2011.
- [23] A. Ramírez-Treviño, I. Rivera-Rangel, and E. López-Mellado, “Observability of discrete event systems modeled by interpreted petri nets,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 557–565, 2003.
- [24] A. Giua and M. Silva, “Petri nets and automatic control: A historical perspective,” *Annual Reviews in Control*, vol. 45, pp. 223–239, 2018.
- [25] M. Adamski and J. Monteiro, “From interpreted petri net specification to reprogrammable logic controller design,” in *ISIE’2000. Proceedings of the 2000 IEEE International Symposium on Industrial Electronics (Cat. No. 00TH8543)*, vol. 1, pp. 13–19, IEEE, 2000.

- [26] A. N. Standard, “Instrumentation symbols and identification,” 2009.
- [27] S.-O. Kang, E.-B. Lee, and H.-K. Baek, “A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams (p&id),” *Energies*, vol. 12, no. 13, p. 2593, 2019.
- [28] M. Toghraei, “Principles of p&id development: the tips provided here will streamline efforts to develop piping & instrumentation diagrams,” *Chemical Engineering*, vol. 121, no. 4, pp. 62–72, 2014.
- [29] A. C. Solé, *Instrumentación industrial*. Marcombo, 2005.
- [30] L. Feng and W. M. Wonham, “Supervisory control architecture for discrete-event systems,” *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1449–1461, 2008.
- [31] F. Schumacher and A. Fay, “Formal representation of grafcet to automatically generate control code,” *Control Engineering Practice*, vol. 33, pp. 84–93, 2014.
- [32] A. Santoyo-Sanchez, A. Ramírez-Treviño, C. De Jesus-Velasquez, and L. I. Aguirre-Salas, “Step state-feedback supervisory control of discrete event systems using interpreted petri nets,” in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 926–933, IEEE, 2008.
- [33] D. Guevara-Lozano, C. R. Vázquez, and A. Ramírez-Treviño, “Automatic specification generation for tracking control in interpreted petri nets,” in *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, pp. 341–346, IEEE, 2020.
- [34] I. Jiménez-Ochoa, D. Guevara-Lozano, C. R. Vázquez, and A. Ramírez-Treviño, “Regulation control in interpreted petri nets under partial observation,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2101–2107, 2020.
- [35] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. Wiley-Interscience, 1977.

- [36] J. F. Sánchez-Blanco, A. Ramírez-Trevino, and A. Santoyo, "Multiple specification regulation control in interpreted petri nets," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 5, pp. 4989–4994, IEEE, 2004.
- [37] I. IEC, "61131-3: Programmable controllers—part 3: Programming languages," *International Standard, Second Edition, International Electrotechnical Commission, Geneva*, vol. 1, p. 2003, 2003.
- [38] M. Tiegelskamp and K.-H. John, *IEC 61131-3: Programming industrial automation systems*, vol. 166. Springer, 2010.
- [39] R. Ramanathan, "The iec 61131-3 programming languages features for industrial control systems," in *world Automation Congress (WAC)*, pp. 598–603, IEEE, 2014.
- [40] I. Jimenez, E. Lopez, and A. Ramirez, "Synthesis of ladder diagrams from petri nets controller models," in *Intelligent Control, 2001.(ISIC'01). Proceedings of the 2001 IEEE International Symposium on*, pp. 225–230, IEEE, 2001.
- [41] T. Perme, "Translation of extended petri net model into ladder diagram and simulation with plc," 2009.
- [42] R. Feio, J. Rosas, and L. Gomes, "Translating iopt petri net models into plc ladder diagrams," in *2017 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1211–1216, IEEE, 2017.
- [43] D. Rozo-Ibañez, J. Ruiz-León, D. Guevara-Lozano, and C. R. Vázquez, "Petri net modelling of industrial processes from a p&id description," in *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, pp. 715–720, IEEE, 2020.
- [44] D. Rozo-Ibañez, J. Ruiz-León, D. Guevara-Lozano, C. R. Vázquez, and Y. Y. Rios, "Removing omissions and inconsistencies from the isa 5.1 narratives of industrial processes," in *2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 1–6, IEEE, 2020.