

Verificación de la Diagnosticabilidad de un Sistema de Eventos Discretos, en función del número de señales de salida



Daniel Felipe Valencia Medina

Directora: PhD. Mariela Muñoz Añasco

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Electrónica, Instrumentación y Control

Ingeniería en Automática Industrial

Popayán, 2018

Verificación de la Diagnosticabilidad de un Sistema de Eventos Discretos, en función del número de señales de salida

Daniel Felipe Valencia Medina

Trabajo de grado presentado a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para la obtención del Título de:
Ingeniero en Automática Industrial

Directora: PhD. Mariela Muñoz Añasco

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería en Automática Industrial
Popayán, 2018

Dedicatoria

A mi madre Oliva

En todos los proyectos en los que he estado y en cada uno de los momentos que ha existido un inconveniente, una dificultad o he necesitado alguien que me escuchara siempre he podido contar con el apoyo incondicional, de la persona que siempre ha estado a mi lado como un faro de guía, una voz de aliento y una gran consejera.

Este trabajo de grado no es la excepción, por estas y muchas otras razones me enorgullece dedicárselo a ella.

Agradecimientos

Este proyecto no se pudo haber culminado sin la ayuda de múltiples personas, a las que les agradezco infinitamente su colaboración en mi trabajo de grado:

- A Dios por estar siempre conmigo, por iluminarme y por regalarme a mi familia y la vida que he tenido.
- A mi familia por siempre darme su apoyo incondicional.
- A la Universidad del Cauca y al PIAI por toda la formación que me han brindado como persona y profesional.
- A la profesora Mariela Muñoz, por confiar en mí, por todo su apoyo a lo largo del proyecto, por compartir su conocimiento e investigación, por orientar y aclarar mis dudas, por todo el tiempo que me ha brindado y por permitirme experimentar y proponer mis ideas a lo largo del trabajo de grado.
- Al profesor Francisco Enriquez, por oír mis planteamientos matemáticos, por decirme cuando estaba equivocado, y por alentarme para mejorarlos y terminarlos.
- Al profesor Ermilson Diaz, por su ayuda en el manejo de la planta didáctica Fischertechnik.
- Al ingeniero Ruben Muñoz, por la inducción que me brindó en el manejo de Graphviz.
- Al doctor Antonio Ramirez Treviño, por todas las observaciones que hizo del trabajo de grado y por compartir su conocimiento.

Índice general

Lista de figuras	V
Lista de tablas	IX
Introducción	X
1. Conceptualización	1
1.1. Autómatas	2
1.2. Redes de Petri	3
1.2.1. Redes de Petri Interpretadas	4
1.3. Modelado por Lenguajes	4
1.3.1. Operaciones sobre Lenguajes	5
1.4. Comportamiento de Sistemas de Eventos Discretos	6
1.5. Diagnosticabilidad	7
1.6. Evaluación de Diagnosticabilidad	8
1.6.1. Diagnosticabilidad Autómatas	8
1.6.2. Diagnosticabilidad en Redes de Petri	8
1.7. Identificación de Sistemas	9
1.7.1. Identificación por señales en Modelo de Caja Negra	9

2. Verificación de Diagnosticabilidad	17
2.1. Verificación de Diagnosticabilidad propuesta	18
2.2. Método de Ubicación Adecuada de Sensores para Diagnosticabilidad	20
2.3. Aplicación de la Verificación de Diagnosticabilidad Propuesta	21
3. Aplicativo Software de Evaluación	28
3.1. Metodologías Ágiles	29
3.2. Dynamic Systems Development Methodology	30
3.2.1. Fases de la DSDM	31
3.3. Desarrollo Software	33
3.3.1. Pre-proyecto	34
3.3.2. Ciclo de vida DSDM	34
Estudio de Viabilidad	34
Estudio de Mercado	34
Análisis funcional y Diseño	35
Implementación	40
3.3.3. Post-proyecto	41
4. Resultados	42
4.1. Sistema de Embotellamiento y Clasificación	43
4.2. Análisis	79
5. Conclusiones y Trabajos Futuros	82
5.1. Conclusiones	82
5.2. Trabajos Futuros	84

Anexos	85
A. Relación matemática entre Observabilidad y Evento-Detectabilidad	86
B. Guia de Usuario del Aplicativo de Identificación DES	89
C. Sistemas Simulados para Verificación de Diagnosticabilidad	98
C.1. Máquina Dispensadora de Bebidas	98
C.2. Fabricación de Llaves de Tuercas	101
C.3. Cadena de Pintado	103
C.4. Línea de Embalaje	105
C.5. Equipo de Mezclado	107
C.6. Línea de Producción de Yogurt	109
C.7. Planta Clasificadora de Piezas	111
C.8. Fabricación de Lapices	113
C.9. Control de Tráfico Vial	116
Bibliografía	121

Índice de figuras

1.1. Matriz de Observaciones de un Sistema	12
1.2. Construcción de la Matriz de Eventos de un Sistema	12
1.3. Matriz de Eventos de un Sistema	13
1.4. Eventos de un Sistema	13
1.5. Construcción de PN en función de las Transiciones de un Sistema	14
1.6. PN asociada a un Sistema	15
1.7. Representación matricial de la PN de un sistema	16
1.8. Representación de un autociclo en una PN	16
2.1. Proceso de identificación con 6 sensores	22
2.2. Primera verificación de diagnosticabilidad	23
2.3. Interacción de señales In-Out y transiciones con 6 sensores	23
2.4. Proceso de identificación con 10 sensores	24
2.5. Segunda verificación de diagnosticabilidad	25
2.6. Interacción de señales In-Out y transiciones con 10 sensores	25
2.7. Proceso de identificación con 11 sensores	26
2.8. Tercera verificación de diagnosticabilidad	26
2.9. IPN asociada al sistema diagnosticable	27

3.1. Ciclo de vida DSDM (adaptado de [1])	32
4.1. Sistema de Embotellamiento y Clasificación	44
4.2. Sistema de Embotellamiento y Clasificación en Simulink	44
4.3. Controlador del SEC	45
4.4. Cinta transportadora del SEC	45
4.5. Brazo Robot del SEC	46
4.6. Llegada de Botellas del SEC	46
4.7. IPN #1 asociada al SEC	48
4.8. Segmento I de la IPN #1 asociada al SEC	49
4.9. Segmento II de la IPN #1 asociada al SEC	50
4.10. IPN #2 asociada al SEC con 13 sensores	51
4.11. Segmento I de la IPN #2 asociada al SEC	52
4.12. Segmento II de la IPN #2 asociada al SEC	53
4.13. IPN #3 asociada al SEC con 13 sensores - Operación CAC	54
4.14. Segmento I de la IPN #3	55
4.15. Segmento II de la IPN #3	56
4.16. Segmento III de la IPN #3	57
4.17. IPN #4 asociada al SEC con 15 sensores - Operación CAC	58
4.18. Segmento I de la IPN #4	59
4.19. Segmento II de la IPN #4	60
4.20. Segmento III de la IPN #4	61
4.21. IPN #5 asociada al SEC con 15 sensores - Operación ACB	62
4.22. Segmento I de la IPN #5	63

4.23. Segmento II de la IPN #5	64
4.24. Segmento III de la IPN #5	65
4.25. IPN #6 asociada al SEC con 16 sensores - Operación ACB	66
4.26. Segmento I de la IPN #6	67
4.27. Segmento II de la IPN #6	68
4.28. Segmento III de la IPN #6	69
4.29. IPN #7 asociada al SEC con 16 sensores - Operación ACBBCCB	70
4.30. Segmento I de la IPN #7	71
4.31. Segmento II de la IPN #7	72
4.32. Segmento III de la IPN #7	73
4.33. IPN #8 asociada al SEC con 17 sensores - Operación ACBBCCB	74
4.34. Segmento I de la IPN #8	75
4.35. Segmento II de la IPN #8	76
4.36. Segmento III de la IPN #8	77
B.1. Carga de datos	90
B.2. Ejecución del Aplicativo Software	91
B.3. Ingreso de número de señales de salida	91
B.4. Ingreso de número de señales de entrada	92
B.5. Número total de señales del sistema	92
B.6. Ingreso de número de observaciones	93
B.7. Activación de la matriz de observaciones	93
B.8. Activación de la matriz de trazas	94
B.9. Verificación de ciclicidad en la matriz de observaciones	94

B.10.Finalización del proceso de identificación	95
B.11.Despliegue de trazas codificadas	95
B.12.Matriz Pre y Post del sistema identificado	96
B.13.Desactivación de opciones para reinicio	96
B.14.Cierre del aplicativo software	97
C.1. Máquina Dispensadora de Bebidas	99
C.2. Proceso de Fabricación de Llaves de Tuerca	102
C.3. Cadena de Pintado	104
C.4. Linea de Embalaje	106
C.5. Equipo de Mezclado	108
C.6. Linea de Producción de Yogurt	110
C.7. Equipo de Mezclado	112
C.8. Fabricación de Lapices	114
C.9. Control de Trafico Vial	117

Índice de tablas

3.1. Resultados de Desempeño de Implementación DSDM	41
4.1. Señales In/Out de la SEC	47
C.1. Señales In/Out de la Máquina Dispensadora de Bebidas	100
C.2. Señales In/Out de la Fabricación de Llaves de Tuerca	103
C.3. Señales In/Out de la Cadena de Pintado	105
C.4. Señales In/Out de la Línea de Embalaje	107
C.5. Señales In/Out del Equipo de Mezclado	109
C.6. Señales In/Out de la Línea de Producción de Yogurt	111
C.7. Señales In/Out de la Planta Clasificadora de Piezas	112
C.8. Señales In/Out de la Fabricación de Lapices	115
C.9. Señales In/Out del Control de Tráfico Vial	120

Introducción

En la industria, existe la necesidad de llevar la maquinaria a un punto de operación deseado mediante acciones de control, pero este proceso se dificulta porque siempre se presentarán aspectos adversos [2], que desembocan en fallos o averías del sistema [3] [4]. Antes de tomar acciones correctivas o de compensación en el proceso de control de un sistema, es necesario conocer los fallos mediante una aproximación del sistema. Una de las formas de representar el sistema es mediante el modelado por eventos, donde cada representación obtenida del sistema analizado, queda sujeta al funcionamiento y estados que se elijan esquematizar (estados deseados) [5], teniendo entonces como una primera aproximación del sistema el funcionamiento ideal de la planta sin perturbaciones y asumiendo que ninguno de los componentes tendrá un comportamiento inadecuado que pueda desembocar en fallo [6].

Debido a que un fallo es un estado no deseado en cualquier modelo, incluyendo los formalismos de eventos discretos, cuando se detecta un comportamiento no previsto se debe aislar el origen del problema e identificar el fallo como tal y clasificarlo [7]; ya que un fallo puede ser de acción permanente es decir, cuando se presenta afecta el sistema desde ese momento o de acción intermitente, en donde en distintos momentos se observa un funcionamiento adecuado del sistema y en otros una mala operación [8]; todo este proceso se conoce como diagnóstico de fallo, en donde la forma más usual de incorporar esos estados no operativos del sistema al modelo, es colocar dos sistemas en paralelo uno de funcionamiento normal y el otro de acciones de fallo cada uno con sus respectivos conjunto de estados y conjunto de eventos [6]; sin embargo, para ejecutar el diagnóstico como tal, el modelo y el sistema deben cumplir ciertos requisitos. El conjunto de requisitos necesarios para ejecutar un diagnóstico sobre un sistema modelado por medio de eventos se denomina diagnosticabilidad [9], esos requisitos son dos: el primero es que el modelo tenga todos los estados deseados y de fallo del sistema; el segundo es observar todos esos estados de alguna forma física [10]; se puede hacer la analogía con una consulta médica en donde el primer requisito es que el paciente le diga todos los síntomas de la enfermedad y estado

físico actual al médico; el segundo requisito es que el médico tenga todas las herramientas para verificar toda la información que el paciente le ha brindado.

La diagnosticabilidad permite determinar el grado en que un sistema puede detectar, aislar e identificar un comportamiento no deseado cuando se ejecuta un proceso de diagnóstico [9]. La detección del comportamiento no deseado en un sistema depende directamente de las señales que se puedan observar de este. Esta inferencia se ve desde un lazo simple de control retroalimentado, en el cual el controlador ejerce una acción determinada sobre la planta dependiendo del estado actual que presente la variable controlada; para saber dicho estado es necesario observar y medir de una forma directa en el sistema la variable física relacionada [2]. Sin embargo, se debe conocer qué variables medir y qué instrumentos son necesarios para esta acción; ya que, por ejemplo, no es importante medir en un control de flujo la conductividad eléctrica del fluido, siendo más relevantes la temperatura del fluido o la densidad del mismo [3].

Las señales a medir en un sistema de manufactura, no están relacionadas con las señales de entrada a la planta, ya que estas están gobernadas por un controlador programado para ejecutar dichas señales y el programa de control se diseña de acuerdo a un modelo que se tiene del comportamiento de la planta, sino con las variables de salida de la planta puesto que ésta es la respuesta propia de la planta que está en función de características internas de la misma, y si la planta está en un modo de operación normal su respuesta se asemeja al modelo establecido; pero si se presenta un fallo el comportamiento se sale del esquema planeado y el controlador ejecutará acciones erróneas o quedará completamente inhabilitado para operar [5].

Siguiendo con el ejemplo de la consulta médica, dependiendo de todas las herramientas que el médico tenga y que tan adecuadas sean se podrá hacer un diagnóstico pertinente de la enfermedad que tiene el paciente, esta analogía indica que un sistema donde se desee hacer un diagnóstico de un fallo determinado, debe contar con los instrumentos de medida (sensores) adecuados [11], y debido a que todo sensor tiene un costo asociado no se puede exceder la cantidad que se tenga de ellos, entonces existe un número preciso de sensores que se deben tener sobre la planta [12].

El uso apropiado de sensores y su cantidad adecuada, aunque es un factor relevante en la industria, no ha sido muy estudiado pero se han alcanzado los siguientes avances: Se ha determinado el rango de confiabilidad de las observaciones brindadas por los sensores de forma que los datos de mayor relevancia son los primeros, ya que todos los datos posteriores a ellos son de respaldo, condensando esto en el término de ventana de predicción con lo cual

al conocer una cantidad específica de datos, directamente proporcional a la cantidad de observaciones, se puede prever la misma cantidad de datos posteriores [13], lo cual conduce a que hay un número apropiado de señales de salida con las que se deben contar para realizar una observación determinada sobre el sistema sin que se presente inconveniente, sin embargo no se puede inferir si hay un número de observaciones adecuado para captar un fallo. Por otro lado, se ha inducido la programación óptima de los sensores para la etapa de supervisión de distintos procesos en paralelo [14], esto ha determinado no la cantidad de sensores sino el uso apropiado de cada uno de ellos, para la operación en simultaneo y continuo dentro de cualquier sistema, lo que brinda confiabilidad en las señales que se están midiendo, pero no si estas son el número adecuado . Se han ejercido acciones de diagnóstico sobre sistemas de estados no observables bajo una función objetivo mediante un número fijo de observaciones [12], con lo cual se ha logrado establecer una relación entre el modelo de operación normal de un sistema con un número determinado de características que no se tienen en cuenta en el momento del modelado, pero no se han realizado estudio al tener variación del conjunto de observaciones sobre el sistema. Se ha activado la dinámica de sensores para eventos de diagnóstico sin modificación de la diagnosticabilidad [15], lo cual indica que hay un nivel de interacción entre las órdenes de medición y el modelo que se obtenga, ya que como se configure la respuesta de los sensores pueden llegar a perturbar la medición que se realice hasta el punto de generar un cambio de estado dentro del sistema.

Sin embargo, el problema de definir el número apropiado de sensores, para la existencia de diagnosticabilidad, no ha sido resuelto. Por lo que la investigación que se desarrollará se centra no en la calidad de los sensores, sino en la cantidad adecuada que se debe tener para que exista diagnosticabilidad en un sistema con el fin de detectar, aislar, identificar y clasificar los fallos asociados al sistema. Buscando dar respuesta a la pregunta de investigación ¿Cuál es la incidencia del número de señales de salida observables en un sistema de manufactura, en la medición de su capacidad para detectar, aislar e identificar comportamientos no deseables en él?. Para alcanzar la respuesta a la pregunta de investigación se han planteado los siguientes objetivos.

Objetivo general

Analizar el comportamiento de fallo de un sistema de manufactura modelado a partir de eventos discretos, con base en la cantidad de señales de salida medibles.

Objetivos específicos

- ✓ Establecer una prueba de diagnosticabilidad basada en señales de salida para un sistema de manufactura.
- ✓ Desarrollar un aplicativo software para la evaluación de sistemas de eventos discretos.
- ✓ Relacionar la incidencia de la cantidad de las señales en la diagnosticabilidad de un sistema de eventos discretos.

La presente investigación esta descrita por capítulos:

1. Conceptualización: donde se presentan los conceptos básicos y base teórica, necesarios para entender la pregunta de investigación.
2. Verificación de diagnosticabilidad: Se desarrolla para cumplir el primer objetivo específico, describiendo las características generales de diagnosticabilidad y su relación con las señales de un sistema.
3. Aplicativo software de evaluación: Se expone la información necesaria para alcanzar el segundo objetivo, al igual de como fue construido el aplicativo software mediante una metodología ágil de desarrollo.
4. Resultados: En este capítulo se presenta la aplicación de la prueba de diagnosticabilidad desarrollado, mediante un aplicativo software a distintas plantas de manufactura; explicando el comportamiento de la diagnosticabilidad en función de la cantidad de las señales que se tienen de la planta.
5. Conclusiones: Es el cierre de la investigación, donde se presentan todos los logros alcanzados por la misma para dar respuesta a la pregunta de investigación.

Los anexos de la monografía son: Los conceptos matemáticos utilizados para desarrollar la prueba de diagnosticabilidad. La guía de usuario del módulo en Labview del aplicativo software. Por último, esta el anexo donde en papel se describen las plantas simuladas que no están en el capítulo 4, y de forma virtual las imágenes relacionadas a estas. [16]

Capítulo 1

Conceptualización

Desde comienzos de la industria, se evidenció la necesidad de llevar la maquinaria a un estado de operación determinado; de dicha necesidad se originó la teoría de sistemas de control, la cual busca lograr un dominio adecuado en un rango de funcionamiento de una planta determinada; sin embargo y por más robusto que se desarrolle un sistema de control siempre se presentarán aspectos no deseados [2], como lo son los errores, disturbios y perturbaciones, que en el mejor de los casos pueden ser compensados, pero en otros generan una modificación completa de la planta hasta un nivel destructivo [3]. Razón por la cual es de vital importancia detectar y aislar todos estos aspectos negativos que se pueden presentar en el sistema generando actos correctivos o preventivos antes que sus efectos se propaguen de forma irreversible [4].

Para detectar y aislar los aspectos negativos en un sistema (conjunto de partes que cumplen una funcionalidad) es imprescindible el tener un modelo de este [17], de los cuales se puede tener una perspectiva probabilística (estocástica), donde la ocurrencia de sucesos se dan en un margen de posibilidad; o determinística donde los sucesos siguen la ley de causa y efecto, indicando que en diferentes repeticiones de aplicación de una misma entrada al sistema se obtiene la misma respuesta. De forma general existen dos maneras de parametrizar un sistema determinista [18]:

- ✓ Ecuaciones diferenciales, donde hay una sucesión infinita de intervalos de tiempo entre dos puntos cualesquiera.
- ✓ Ecuaciones en diferencia, que propone la existencia de un número finito de intervalos de tiempo para llevar un sistema de un punto A a un punto B.

La segunda manera es analizar la evolución del sistema no en función del tiempo, o tomar el tiempo como variable secundaria, sino por transiciones de estados (conjunto de situaciones del sistema sujeta a cambios entre sí) relacionados a diferentes eventos (acontecimiento que genera cambios sobre el sistema) en los que se obtienen representaciones tales como:

- ✓ Autómatas [19].
- ✓ Redes de Petri [20].
- ✓ GRAFCET [7].
- ✓ DEVS [8].

Las dos últimas representaciones son desarrollos técnicos, cumpliendo con normatividad de operación y buenas prácticas respecto al entendimiento y aplicación del modelo entregado por los autómatas y las redes de Petri (formalismos de sistemas de eventos discretos, DES), que sirven para la implementación en elementos de control industriales (Stand Alone, IPC, PLC, etc.).

1.1. Autómatas

Un autómata es la representación matemática de un sistema discreto, donde se evalúa la pertenencia de un conjunto de símbolos a un lenguaje definido por un alfabeto determinado. Un autómata determinista G queda representado por:

$$G = (X, E, f, x_o, X_m) \tag{1.1}$$

Donde:

X : Es un conjunto de estados.

E : Es un conjunto finito de eventos.

f : Es la función de transición de estados.

x_o : Es el estado inicial.

X_m : Es el conjunto de estados marcados ($X_m \subseteq X$).

1.2. Redes de Petri

Las redes de Petri (Petri Net, PN) son una representación de un sistema de eventos discretos, mediante un formalismo matemático que tiene un comportamiento gráfico respectivo, evaluando sus cambios de estados. Desde el punto de vista gráfico, una PN es un grafo bipartito con 3 componentes: Lugares(Estados), Transiciones(Eventos) y Arcos dirigidos; los cuales interactúan entre sí respecto al marcado (condiciones externas al sistema), siendo el conjunto de marcas o puntos dentro de un lugar, y por las condiciones existentes en transiciones y arcos.

Sea N una PN, representada como:

$$N = (P, TR, Pre, Post, M_o) \quad (1.2)$$

Donde:

P : Es el conjunto de lugares de cardinalidad n_p .

TR : Es el conjunto de transiciones de cardinalidad n_{tr} .

Pre : Es la matriz de incidencia previa de arcos que conectan lugares con transiciones.
($P \times TR$).

Post : Es matriz de incidencia posterior de arcos que conectan transiciones con lugares.
($TR \times P$).

M : Es la función de marcado ($M:P \rightarrow N$) indica el número de marcas en cada lugar.
(M_o Marcado inicial).

Cada transición tiene un número determinado de lugares de entrada y salida, representados por las matrices Pre y Post, a su vez la matriz de incidencia C ($C=Pre - Post$) indica las condiciones actuales del sistema para un marcado dado M_i .

Los conjuntos de lugares previos y posteriores a una transición se denotarán respectivamente como:

- $\bullet tr = \{p \in P : Pre(p, tr) > 0\}$
- $tr \bullet = \{p \in P : Post(p, tr) > 0\}$

Respecto a la activación y la secuencia de las transiciones, se obtiene la secuencia de disparo (vector característico) $\sigma = t_1 t_2 \dots t_k$. Donde la ecuación de estado para un marcado relacionado a una secuencia de disparo es $M_i = M_{i-1} + C \cdot \sigma$.

1.2.1. Redes de Petri Interpretadas

Una red de Petri Interpretada (IPN) [21] asocia en el modelo los lugares, las transiciones y las marcas con un significado específico y propio al contexto del sistema que se está analizando. Las secuencias de disparo se efectúan con la habilitación de las transiciones y los eventos asociados que deben cumplirse, con un conjunto global de n-lugares sobre el cual se define q-lugares observables ($q < n$).

Una IPN Q se define como:

$$Q = (N, E, \lambda, \varphi) \quad (1.3)$$

Donde:

N : Es la red de Petri asociada al sistema.

E : Es el conjunto de eventos.

λ : Es la función de etiquetado de transiciones. ($\lambda: TR \rightarrow E$)

φ : Es la matriz de mapeo de una salida observable (y_k) para un marcado determinado (M_k). ($y_k = \varphi M_k : \varphi(q \times n)$)

1.3. Modelado por Lenguajes

Cada uno de los modelados DES poseen una característica en común, el tener un conjunto de eventos E asociados a su representación. De forma general ese conjunto de eventos es el alfabeto de un lenguaje y las posibles secuencias que se generan en el sistema conforman las palabras o cadenas de eventos del lenguaje [22].

El lenguaje L asociado al conjunto de eventos E, se define como el conjunto de cadenas de longitud finita formadas por el alfabeto de E, una cadena sin eventos queda representada por ε denominada cadena vacía; dado un alfabeto E con más elementos que ε es un

subconjunto de E^* ($E \subseteq E^*$), el cual son todas las posibles combinaciones de los eventos en E sin restricción de repetición u orden generando infinitas cadenas ($\therefore L \subseteq E^*$).

1.3.1. Operaciones sobre Lenguajes

Existen múltiples operaciones que se pueden realizar en los formalismos de modelado DES, de los cuales algunos son adaptaciones de las operaciones sobre lenguajes [22], por lo cual se definirán las operaciones más utilizadas en lenguajes, para un determinado lenguaje L , con un conjunto de eventos E y una palabra del lenguaje s ; dichas operaciones son:

✓ Concatenación

Sea $L_a, L_b \subseteq E^* \Rightarrow L_a L_b := \{s \in E^* : (s = s_a s_b) \wedge (s_a \in L_a) \wedge (s_b \in L_b)\}$.

Una cadena pertenece a $L_a L_b$, si se expresa como la concatenación de una cadena de L_a con un cadena de L_b .

✓ Prefijo de Clausura

Sea $L \subseteq E^* \Rightarrow \bar{L} := \{s \in E^* : (\exists t \in E^*) [st \in L]\}$.

El prefijo de clausura \bar{L} indica todos los prefijos de las cadenas existentes en L , en general se cumple que $L \subseteq \bar{L}$; se denomina un lenguaje L prefijo cerrado si toda cadena de L se encuentra en \bar{L} ($L = \bar{L}$).

✓ Clausura de Kleene

Sea $L \subseteq E^* \Rightarrow L^* := \{\varepsilon\} \cup L \cup LL \cup LLL \dots$

La clausura de Kleene se aplica en conjuntos de eventos, lenguajes y palabras; esta operación indica que un elemento de L^* está formado por la concatenación de un número finito de elementos de L , incluyendo la cadena vacía ε .

✓ Post-Lenguaje

Sea $L \subseteq E^* \Rightarrow L/s := \{t \in E : st \in L\}$.

El post-lenguaje L/s indica todas las cadenas después de eliminar el prefijo s del lenguaje L .

✓ Proyección

Es una operación que se aplica en conjuntos de eventos y lenguajes, teniendo dos grupos de igual naturaleza entre los cuales uno es mayor que el otro. La proyección (P) indica

todos los posibles elementos que se forman a partir del grupo menor en el grupo mayor, eliminando todos los elementos que no cumplan dicha condición. Sea E_l el conjunto de eventos mayor, E_s el conjunto de eventos menor, y un lenguaje $L \subseteq E_l^*$.

$$\Rightarrow P(L) = \{t \in E_s^* : (\exists s \in L)[P(s) = t]\}$$

✓ **Proyección Inversa**

La proyección inversa indica todas las posibles cadenas que se forman en el grupo menor sobre el grupo mayor respecto a una cadena determinada, esta operación se aplica sobre los mismos elementos que la proyección.

$$P^{-1}(L_s) = \{s \in E_l^* : (\exists t \in L_s)[P(s) = t]\}.$$

1.4. Comportamiento de Sistemas de Eventos Discretos

De los formalismos DES, se debe resaltar que cada representación obtenida del sistema analizado, queda sujeta al funcionamiento y estados que se elijan esquematizar (estados deseados) por lo cual, una primera aproximación del sistema es el funcionamiento ideal de la planta sin perturbaciones y asumiendo que ninguno de los componentes presentará una acción inadecuada [5]. Este funcionamiento ideal se conoce como comportamiento normal, donde ningún aspecto no deseado del sistema analizado se ha considerado para la construcción del modelo DES [23]. Siendo esto ajeno a la realidad, una aproximación más adecuada es plantear posibles acciones erróneas del sistema y características de comportamiento anormal, generando un modelo que abarque la operación normal y no deseada del sistema [24], este último componente se conoce como comportamiento de fallo.

De forma general el fallo se define como: Desviación del sistema de su comportamiento normal, el cual es inevitable y en algunos casos genera daños irreversibles sobre el sistema. Por lo que se hace necesario crear modelos de los sistemas que permitan detectar, aislar y corregir los fallos [6].

Un fallo por ser una desviación no deseada del sistema, se asume como un evento no observable del modelo asociado; los modelos DES que abarcan los posibles fallos, o parte de ellos, tienen dos componentes relacionados entre sí: un comportamiento normal encargado de describir las acciones deseadas del sistema y un comportamiento de fallo que simula todas las acciones atípicas o negativas que se han planteado [25].

En particular, en las IPN los componentes del modelo se dividen en [26]: Comportamiento observable, que simula la operación normal del sistema analizado, y Comportamiento no observable, donde se encuentran todo el conjunto de lugares no observables y transiciones relacionadas que no generan cambios apreciables en el sistema real ni provocan efectos negativos, añadiendo al comportamiento no observable el conjunto de lugares y transiciones de comportamiento de fallo.

1.5. Diagnosticabilidad

Para los modelos DES con un comportamiento de fallo asociado, el proceso de detectar y aislar las causas de las posibles anomalías es decir, identificar si ha ocurrido un fallo en el sistema se denomina diagnóstico [27]. El diagnóstico en modelos DES es el determinar la ocurrencia de un fallo, sin ser estrictamente de forma inmediata; existen distintas formas de hacer diagnóstico de forma general se encuentran: Offline, después de registrar datos del sistema en un tiempo de operación determinado, se compila toda la información del sistema y se compara con un modelo exhaustivo de fallo, en donde se encuentra similitudes, se asegura que ha ocurrido un fallo; Online, se genera la recopilación de información y la comparación con fallos en cada registro de datos, por medio de una evaluación de posibles fallos después de cada observación.

Estas dos formas se aplican a modelos que incluyen comportamiento de fallo y a modelos libres de fallo, en el primero se asume que todos los posibles estados no deseados se conocen antes de generar el diagnóstico; en el segundo se conocen todos los posibles estados deseados y los eventos de interacción entre ellos, dado el caso de tener una observación que no pertenece a la información pre-establecida se afirma la ocurrencia de un fallo [28].

El proceso de diagnosticabilidad sobre un sistema de cualquier tipo, se define como la determinación de las características necesarias y suficientes sobre el sistema, para que sean detectables posibles fallos relacionados a su funcionamiento; por lo que la diagnosticabilidad es la determinación a priori de sí la ocurrencia de un fallo en el sistema, sea diagnosticado durante la operación [25].

1.6. Evaluación de Diagnosticabilidad

Sobre cada DES que se obtenga de un sistema existen diversas formas de evaluar si es diagnosticable, estas formas quedan sujetas a los formalismo de modelado, pero de forma general en cada una de ellas se toma, los fallos como eventos no observables.

1.6.1. Diagnosticabilidad Autómatas

La diagnosticabilidad en autómatas queda sujeta a la construcción de un autómata determinista, denominado Observador, construido sólo con los eventos observables del sistema; donde la capacidad de generar un diagnóstico se da por la relación de los estados de fallo y las transiciones que se puedan presentar en ellos [29]. Por lo tanto un fallo es diagnosticable si es detectado en un número finito de eventos, es decir, del lenguaje generado por el autómata del sistema se generen observaciones distintas y suficientes para habilitar una identificación de fallos dentro de un intervalo finito de eventos, convirtiendo al autómata observador en Diagnosticador.

1.6.2. Diagnosticabilidad en Redes de Petri

El planteamiento de diagnosticabilidad para redes de Petri, es una generalización de los autómatas ya que también es necesario el uso de un observador del sistema con las mismas características, pero por la estructura del formalismo, se presenta una diversificación de que un fallo sea diagnosticable ya que: se tienen un conjunto de secuencia de transiciones que tienen una parte observable y otra no observables; y no todas las posibles secuencias tendrán fallos asociados (hay certeza de fallo si toda secuencia de transiciones tiene al menos un fallo asociado) [26]. Este crecimiento en los estados de diagnóstico es una ventaja sobre los autómatas, ya que cada una de las variantes de diagnóstico se evalúa en el grafo de alcanzabilidad base, el cual es la evaluación de todos los eventos asociados a un estado, es decir, todas las posibles trayectorias que se pueden tomar desde un punto de operación del sistema, algunas de estas conducen a estados deseados y otras a estados de fallo, la ocurrencia de fallo se afirma si y solo si todos los posibles eventos de un estado conducen a fallo.

1.7. Identificación de Sistemas

Para conocer los fallos de un sistema o saber si será posible detectarlos, antes de ejecutar cualquier proceso de evaluación, se debe obtener un modelo del sistema en relación con las características de operación establecidas. Para ello hay distintas formas de hacer la identificación del sistema:

- ✓ Modelado físico-matemático, este de modelado realiza medición directa de parámetros que se relacionen con el funcionamiento del sistema, para relacionarlos entre si por modelos físicos preestablecidos mediante un planteamiento matemático adecuado [17].
- ✓ Planteamiento de experto, este modelado consiste en recopilar la información que tenga un experto u operario respecto al sistema de interés, con el fin de plantear parámetros de funcionamiento necesarios para conocer la operación del sistema [20].
- ✓ Simulación, de forma general el simular un sistema consiste en utilizar un software de desarrollo o diseño asistido por computador que pueda implementar las características de interés para la operación del sistema [30].
- ✓ Modelo de caja negra, consiste en tomar el sistema de interés como un objeto desconocido, al cual se le aplicarán ciertas señales de entrada y se esperara posibles respuestas de salida; suponiendo una interacción directa entre entradas y salidas se determina el funcionamiento del sistema [31].

Con el objeto de tener modelos lo mas cercanos a la realidad, en la identificación de sistemas se utiliza una combinación de los modelados mencionados. En el caso los DES conviene utilizar una combinación de todos exceptuando la simulación (proceso que sirve para la validación del modelo) es decir, el experto brinda información para parametrizar las entradas que se le puedan aplicar al sistema, con el modelado físico-matemático se conocen las posibles salidas del sistema, y con el modelo de caja negra se determinan el funcionamiento del sistema.

1.7.1. Identificación por señales en Modelo de Caja Negra

El modelo de caja negra relaciona las señales de entrada con las señales de salida, aplicadas en un instante de tiempo determinado y en el caso de eventos discretos son señales

secuenciales aplicadas y registradas en un conjunto de momentos (esto es debido a que la acción sobre el sistema no genera una respuesta inmediata). Para la presente investigación se ha seleccionado un método de identificación denominado paramétrico [32][33] desarrollado por la doctora Estrada Vargas, fue elegido porque: Trabaja con datos de operación de los sistemas, Asume que se ha obtenido toda la información de operación del sistema, infiriendo todos los posibles eventos a partir de la dinámica de evolución registrada y Centra la construcción del modelo en función de las transiciones adquiriendo la PN resultante dos características relevantes: La construcción de eventos por cambios perceptibles en el sistema obteniendo el lenguaje correspondiente, y la generación de una cantidad reducida de lugares y transiciones para cumplir con el lenguaje del sistema.

Este método toma como sistema el conjunto planta-controlador, donde las entradas y salidas son determinadas desde el punto de vista de la planta, es decir, las señales de salida (Out, O) provienen de los sensores de la planta y las señales de entrada (In, I) son las acciones que ejecuta el controlador; se plantea que el sistema opera en lazo cerrado ejecutando secuencialmente tres operaciones:

1. Lectura de los sensores (señales de salida)
2. Ejecución del programa (interacción entrada-salida)
3. Escritura de los actuadores (señales de entrada)

Los valores entrada-salida son ordenados en vectores I/O, el conjunto secuencial de vectores I/O depende de la duración de la observación (al establecer un tiempo de adquisición fijo, se obtiene un número determinado de observaciones), a este conjunto se le denomina matriz de observaciones (ω), donde dos elementos consecutivos cumplen con la condición de ser distintos.

$$\omega = \left[\frac{O(1)}{I(1)} \right], \left[\frac{O(2)}{I(2)} \right], \left[\frac{O(3)}{I(3)} \right] \dots$$

La longitud de cada I(j) y O(j) depende de las r-entradas y q-salidas que tenga el controlador en cada ciclo. Para analizar la evolución de las señales (dinámica del sistema), se plantea el vector de eventos (E) siendo la resta de dos vectores I/O consecutivos:

$$E(k) = \omega(k + 1) - \omega(k) \neq 0 \tag{1.4}$$

El vector de eventos se descompone en vectores de eventos de entrada y de salida.

$$E = \begin{bmatrix} OE(k) \\ IE(k) \end{bmatrix}$$

Dependiendo los valores que tengan los vectores IE y OE, brindan información de la dinámica del sistema:

1. $OE(j) \neq 0$ y $IE(j) \neq 0$, indica que el cambio en las entradas al controlador han modificado sus salidas.
2. $OE(j) \neq 0$ y $IE(j) = 0$, este caso plantea que no hay variación en la salida del controlador, lo cual puede ser provocado porque: en el siguiente ciclo se observe el cambio, el controlador no sea sensible a los cambios en las entradas o estas variaciones modifican las salidas pero no son observables.
3. $OE(j) = 0$ y $IE(j) \neq 0$, esta relación indica tres posibilidades: primero es subsecuente al primer caso del anterior item; segundo comportamiento pre-establecido en el controlador; o puede ser generado por un fallo.

Estas observaciones cualitativas del sistema por medio de los datos cuantitativos obtenidos del mismo, son útiles para plantear su respectivo modelo en una IPN, ya que cada vector de eventos es asociado con una transición del sistema; la agrupación ordenada y secuencial de estos vectores en función de la adquisición de datos sobre el sistema, genera la secuencia de transiciones, donde los lugares asociados no tienen relación directa a la matriz de observación sino a las condiciones necesarias para cumplir con la secuencia observada, razón por la cual el conjunto de estados obtenidos del modelado se da por la generación del conjunto de estados observables e inferencia del conjunto de estados no observables.

Para continuar con la descripción del método de identificación, se ilustrará las partes ya mencionadas y las restantes. Suponga un sistema cualquiera con 6 sensores en planta y 13 señales de control, en el cual se ha observado todos los datos de operación, agrupados por ciclos de operación y eliminando los pares de columnas que se repitan, obteniendo la matriz de observaciones asociada al sistema (Figura 1.1).

Con los vectores de eventos del sistema (Figura 1.3), se efectúa una comparación entre ellos para establecer cuantos de ellos se repiten y generar un asignación de transiciones a los vectores de eventos (Figura 1.4), con el fin de obtener la secuencia de transiciones del sistema.

Matriz de Eventos

0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	-1	1	-1	0	0	0
0	0	0	1	-1	1	0	0	0	0	0	0	-1
1	0	-1	0	0	0	0	0	0	0	0	0	0
1	0	-1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	-1	1	-1	0	0	0	0	0	0	1
1	0	0	0	0	0	-1	1	-1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Sensores

Actuadores

}

}

Figura 1.3: Matriz de Eventos de un Sistema

Matriz de Eventos

0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	-1	1	-1	0	0
0	0	0	1	-1	1	0	0	0	0	0	-1
1	0	-1	0	0	0	0	0	0	0	0	0
1	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	-1	1	-1	0	1	-1	1	1	0
1	0	0	0	0	0	-1	1	-1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

t1

t2

t3

t4

t5

t4

t6

t7

t6

t7

t5

Figura 1.4: Eventos de un Sistema

La secuencia de transiciones del sistema que se ha obtenido es t1-t2-t3-t4-t5-t4-t6-t7-t6-t7-t5, donde el conjunto de las transiciones del sistema es {t1,t2,t3,t4,t5,t6,t7}. Para la creación del conjunto de lugares y su concatenación con las transiciones (Figura 1.5), el método paramétrico trabaja con dos principios de las PN: primero, un lugar queda determinado por una transición antecesora y otra posterior (que puede ser la misma) y segundo, una transición es habilitada por uno o más lugares.

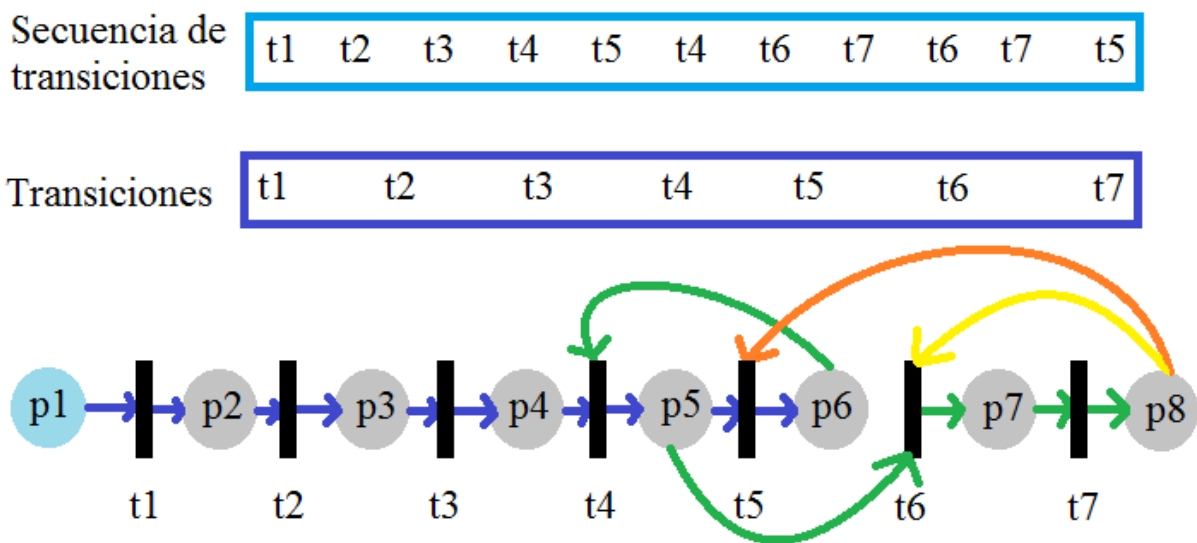


Figura 1.5: Construcción de PN en función de las Transiciones de un Sistema

De estos principios la construcción de la IPN asociada al sistema empieza con generación del primer lugar que habilita a la primera transición de la secuencia, este primer lugar se relaciona con el estado en el que se encontraba el sistema antes de empezar el proceso de adquisición de datos de operación o estado inactivo; la generación de los demás lugares queda en función de cumplir con la secuencia de transiciones en el pictórico representado en la Figura 1.5, las primeras cinco transiciones del sistema son consecutivas (t1-t2-t3-t4-t5) por lo que los lugares asociados (p1, p2, p3, p4, p5, p6) tendrán el mismo comportamiento para cumplir esa secuencias (flechas azules). Siguiendo en la secuencia se obtiene t4-t6-t7 (concatenadas con flechas verdes) , t4 representa la existencia de un ciclo (por ser una transición que se ha presentado antes), por la presencia del ciclo se reutiliza el lugar p5 para seguir con la secuencia, creando los lugares p7 y p8 de forma secuencial. Después en la secuencia de transiciones se presenta un segundo ciclo registrado por t6-t7 (representado en la flecha amarilla), por ende se utiliza los lugares p7 y p8. El último componente de la secuencia de transiciones es t5, por lo que en la PN se genera un tercer ciclo (flecha naranja).

Del pictórico de la PN (Figura 1.5), se construye una representación en limpio de la PN que cumple con la secuencia de transiciones del sistema (Figura 1.6). De la PN obtenida se deduce la matriz Pre y Post del sistema, al igual que la matriz de incidencia del sistema (Figura 1.7). La descripción sistemática del proceso de construcción de la PN y la IPN del método paramétrico, se encuentra en la etapa de análisis funcional y diseño del ciclo de vida de la sección 3.3.

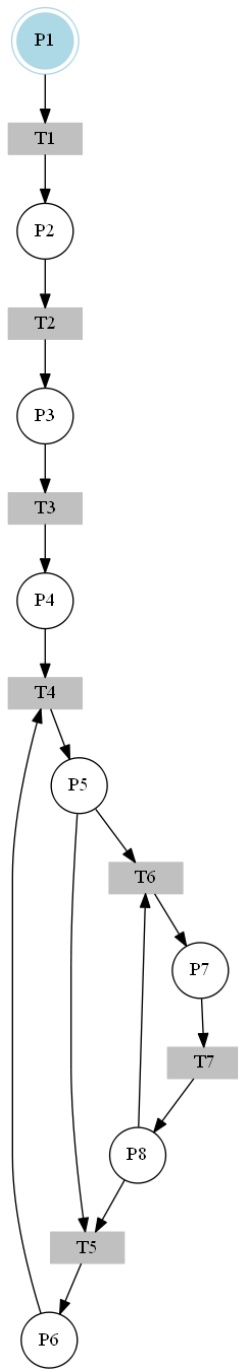


Figura 1.6: PN asociada a un Sistema

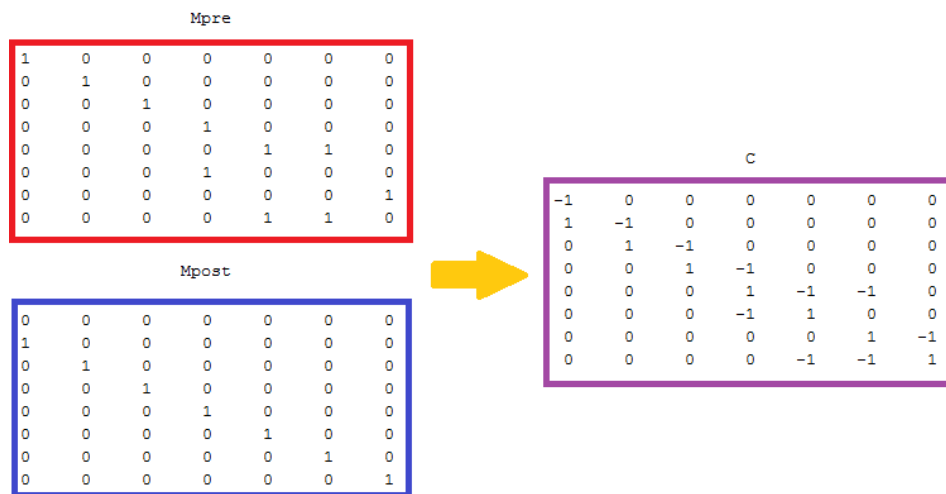


Figura 1.7: Representación matricial de la PN de un sistema

Por la resta consecutiva de las observaciones del sistema para obtener las transiciones del mismo, el método de identificación no representa componentes de sistemas con autociclo (Figura 1.8), es decir, que por la lógica de programación ellos mismos se habiliten para la operación.

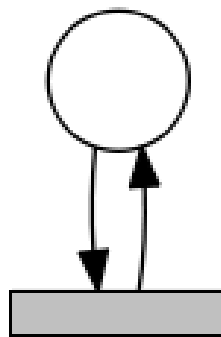


Figura 1.8: Representación de un autociclo en una PN

Capítulo 2

Verificación de Diagnosticabilidad

Las distintas investigaciones relacionadas al diagnóstico y diagnosticabilidad de DES, parten de la estructura clásica planteada para autómatas [6]; evidenciando en todas estas dos requisitos, primero que el DES a evaluar sea determinista, y segundo que todos los eventos del sistema sean diferenciables entre sí. Centrándose en IPNs las dos condiciones para la diagnosticabilidad [29] [28], quedan formuladas de la siguiente forma:

- ✓ Evento-Detectabilidad, esta característica determina que los eventos asociados a un sistema deben ser diferenciables entre si, por lo que dos transiciones no pueden tener el mismo etiquetado para llegar a dos o más lugares distintos [34]; esto se relaciona con que la IPN debe ser determinista, ya que analizando desde el lenguaje una letra (transición) solo tiene un significado (conducir a un único lugar), lo cual indica que en el proceso de construcción de una IPN relacionada a un sistema real, se deben tener los datos suficientes para distinguir todos los componentes del conjunto de eventos.
- ✓ Observabilidad, sobre una IPN se cumple la característica observabilidad si cada uno de los estados, observables y no observables, tienen una secuencia de transición fija para llegar a ellos [35], de forma general hace referencia a la diferenciable de los lugares a partir de las posibles trayectorias que conduzcan a ellos, visto desde el lenguaje una palabra (secuencia de transiciones) tiene una única interpretación (lugar); esta diferenciación de lugares se logra a partir de los datos asociados a los estados de la IPN, es decir, si dos estados ($E1$ y $E2$) tiene los mismos datos, implica que son el mismo estado ($E1=E2$).

2.1. Verificación de Diagnosticabilidad propuesta

Para la verificación de diagnosticabilidad se deben cumplir las características generales de diagnosticabilidad mencionadas. La condición para cumplir la evento-detectabilidad respecto a un conjunto de sensores predeterminado, ha sido presentado en investigaciones anteriores [21] con la ayuda de la matriz de incidencia (C) y la matriz de mapeo de salida (φ), donde la primera asocia los lugares (P) con las transiciones (TR), la segunda los lugares y los sensores (SR) del sistema, tal que:

$$C_{TR \times P} \times \varphi_{P \times SR} = C\varphi_{TR \times SR} \quad (2.1)$$

La matriz resultante que asocia transiciones con sensores ($C\varphi$), debe cumplir que cada una de sus columnas sean distintas a cero y diferentes entre sí para que el sistema sea evento-detectable; sin embargo, la construcción de la matriz de mapeo de salida no se desarrolla de forma sistemática sino por conocimiento de un experto al sistema. Las condiciones de observabilidad sobre IPN para un determinado conjunto de sensores se realizan por métodos computacionales como algoritmos genéticos [36] y programación lineal entera [37], necesitando información no solo sobre la IPN, sino del sistema al que está relacionado como ubicación de sensores, señales deseadas, y posibles perturbaciones.

Para el caso de esta investigación, la verificación de diagnosticabilidad se da en función de la identificación por modelo de caja negra; para una interpretación clara en el manejo matemático y fácil asociación con evento-detectabilidad y observabilidad, la matriz φ se denominará matriz de inspección sobre lugares y la matriz $C\varphi$ como matriz de inspección sobre transiciones.

Se iniciará con la matriz de eventos obtenida de las observaciones en la identificación del sistema, que asocia los eventos secuenciales del sistema con la secuencia de transiciones de la IPN. Para evaluar la evento-detectabilidad se dejarán solo las columnas diferentes de la matriz de eventos (si un evento se repite, este desaparece porque existiría información repetida del sistema), con el fin de obtener el conjunto de transiciones del sistema, y no la secuencia de transiciones, relacionadas a las señales In-Out del sistema analizado, tomando la submatriz que asocia los eventos con las señales de salidas (eventos de salida), no la matriz completa, porque los eventos de entrada quedan determinados por las acciones del controlador (en esta investigación no se plantean posibles fallos del controlador). De esta manera se obtiene la matriz de inspección sobre transiciones ($C\varphi$).

Para que un sistema sea evento-detectable se evalúa sobre la matriz de inspección de transiciones, las condiciones de que todas sus columnas sean diferentes entre sí y que sean distintas a una columna de ceros; la interpretación de evento-detectabilidad en función de la identificación usada indica que: El sistema debe contar con los sensores suficientes para diferenciar todas las posibles transiciones que se presenten en el sistema, mediante la dinámica que se registra en la comparación secuencial de observaciones, cada sensor dará información de activación (1), inactividad (0) o desactivación (-1) en los vectores de eventos de salida.

Para la observabilidad, al tener los lugares un etiquetado fijo, para observar si el DES obtenido es capaz de diferenciar cada lugar, se plantea el relacionar la matriz traspuesta de incidencia (C^T) con la matriz de inspección sobre transiciones, donde el resultado sería la matriz de inspección sobre lugares. Es posible utilizar las mismas condiciones de evento-detectabilidad, en la matriz de inspección sobre lugares para comprobar observabilidad, por la transitividad de correspondencia por lo que se cambia del espacio de operación de sensores-transiciones a sensores-lugares pero al ser una modificación lineal las condiciones se mantienen en el espacio nuevo, ya que la matriz de incidencia da información estable del sistema y la dinámica dependerá de la variación de las señales de sensores, partiendo de que la existencia de lugares generados están asociados a todos los estados que pueda tener el sistema real, es decir:

$$C_{P \times TR}^T \times C_{TR \times SR} = \varphi_{P \times SR} \quad (2.2)$$

Para el desarrollo de la presente investigación se toma la matriz de incidencia como una transformación lineal (transformación lineal de incidencia - TLI) para pasar de elementos del subespacio de observabilidad DES (SEO- φ) al de evento-detectabilidad DES (SEED- γ), y en viceversa, para aspectos de generalización para cualquier tipo de matrices se tomará como una pseudo inversa a la matriz de incidencia su traspuesta (transformación lineal de incidencia traspuesta - TLIT). Por la multiplicación de matrices es fácil ver que la TLI y la TLIT no son isomorfismos, en la multiplicación de un caso particular de sus representaciones es posible obtener el mismo resultado en SEO con distintos elementos del SEED o viceversa, por ende no son transformaciones lineales inyectivas. Esto indica que la validación de parámetros de un subespacio no implica la existencia de los parámetros correspondientes al otro, y ambos subespacios existen sin la necesidad del otro; sin embargo, la existencia de las dos transformaciones lineales para relacionar elementos de los dos subespacios indica una relación de correspondencia entre ambos ($\varphi \leftrightarrow \gamma$), con lo

cual en ambos subespacios hay características similares que son posibles verificar bajo los mismos métodos.

Para explicar este concepto, tómesese como ejemplo el espacio xy y el yz (representaciones espaciales en 2 dimensiones) entre los 2 se puede pasar de uno al otro con una transformación lineal correspondiente, y en ambos es posible generar mediciones de área o longitud, si un elemento de medida es válido en el espacio xy dará la misma información en el espacio yz .

De esta forma se plantea una forma indirecta de obtener la matriz de inspección sobre lugares, para evaluar sobre el sistema la capacidad de diferenciar todos los lugares por medio de los sensores con los que se cuenta, es decir, observabilidad en función de sensores. Los conceptos matemáticos utilizados para la relación entre evento-detectabilidad y observabilidad se encuentra en el anexo A.

2.2. Método de Ubicación Adecuada de Sensores para Diagnosticabilidad

En la sección anterior 2.1), se logró relacionar las dos características generales de diagnosticabilidad en función de la cantidad de sensores puestos en el sistema analizado, con esto se evidencia que del procesamiento de información obtenida por el método de identificación se infiere la matriz de transiciones en función de señales I/O, y una submatriz de esta que se denominó Inspección sobre transiciones donde se relacionan los sensores (señales O) con las transiciones. Bajo este proceso con la ejecución de un primer procedimiento experimental por tanteo, para verificación de diagnosticabilidad en función de señales de salida, y posterior uso de la propiedad matemática de correspondencia demostrada, se estableció un método de ubicación adecuada de sensores para diagnosticabilidad con el siguiente procedimiento:

1. Obtener la matriz de todas las posibles transiciones del sistema, en función de las señales I/O ordenando primero las señales de salida (de la planta) y luego las señales de entrada (a la planta).
2. Ejecutar la prueba de diagnosticabilidad propuesta.
3. Evaluar la característica de Evento-Detectabilidad en función de columnas de ceros.

4. Procesar la información sí el sistema no es evento-detectable; ubicando las mismas columnas en la matriz de transiciones obtenida. En la sección de señales de salida habrán valores de cero pero en las señales de entrada están los actuadores que generan cambios en el sistema, pero en la lógica de operación secuencial directa, que no tienen un sensor asociado; en estos actuadores es sobre los cuáles se les debe acoplar un sensor pertinente para percibir su operación.
5. Evaluar la característica de Evento-Detectabilidad en función de columnas repetidas.
6. Procesar la información sí el sistema no es evento-detectable; ubicando las mismas columnas en la matriz de transiciones obtenida. En la sección de señales de salida habrán valores idénticos, en las señales de entrada están los actuadores que generan la diferenciación en el pre-etiquetado para el método de identificación pero no tienen un sensor asociado; en estos actuadores es sobre los cuales se les debe acoplar un sensor pertinente para percibir su operación y establecer diferenciación de las columnas en la matriz de inspección sobre transiciones.
7. Ejecutar de nuevo la prueba de diagnosticabilidad propuesta, ya que la observabilidad y la evento-detectabilidad están relacionadas como se demuestra en el anexo A, la validación de una puede generar variación en la otra.
8. Evaluar la característica de Observabilidad en función de columnas de ceros.
9. Evaluar la característica de Observabilidad en función de columnas repetidas.
10. Multiplicar por la izquierda, la matriz de incidencia traspuesta por la matriz de transiciones obtenida, y ejecutar los mismos procesos de ubicación de sensores de evento-detectabilidad para observabilidad.

2.3. Aplicación de la Verificación de Diagnosticabilidad Propuesta

Para la verificación de diagnosticabilidad en función de señales de salida en un sistema en lazo cerrado (Planta-Controlador), en la presente investigación se han determinado tres pasos: Primero, realizar la identificación por el método de caja negra del sistema para obtener la representación matricial de la IPN asociada; Segundo, aplicar el método de ubicación de adecuada de sensores desarrollado, evaluando la diagnosticabilidad del

sistema y en caso de no cumplirse esta característica, arroja las señales de salida necesarias para alcanzarla; y Tercero, la representación gráfica de la IPN del sistema.

Para ejemplificar el proceso que se ha mencionado, se ilustrará con una Máquina Dispensadora de bebidas (descrita con mayor detalle en la sección C.1, en un ciclo de operación de gaseosa fría; para el proceso de identificación, de diagnosticabilidad y de representación gráfica se utilizarán los aplicativos software desarrollados en Labview y Matlab (El proceso de diseño, construcción, y utilización de ambos se encuentra en el capítulo 3).

El proceso de identificación inicial indica que la IPN asociada al sistema tiene 19 lugares y 18 transiciones (Figura 2.1).

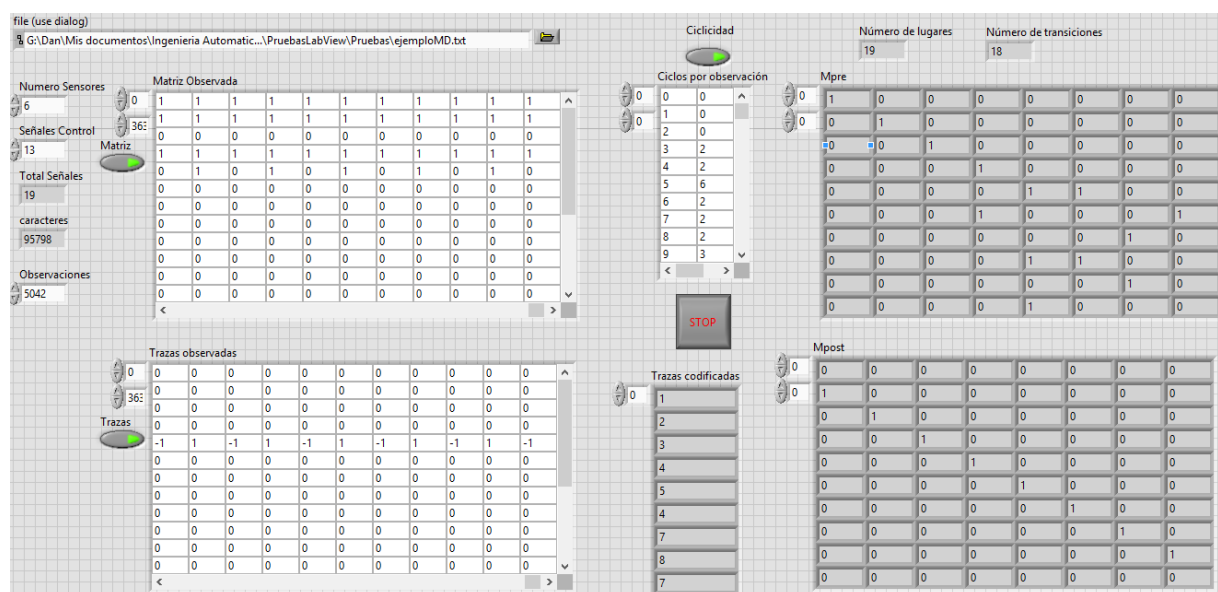


Figura 2.1: Proceso de identificación con 6 sensores

En la primera verificación de diagnosticabilidad (Figura 2.2), de la matriz de inspección sobre transiciones se observa que: Las columnas 1, 9, 10, 12, 13, 14, 16 y 18 iguales a cero (rojo); la primera columna no es relevante ya que es originada en el comienzo de la simulación de la máquina dispensadora de bebidas; las columnas 4 y 11 (verde) junto con 5 y 17 (azul) son iguales entre sí. Por su parte en la matriz de inspección sobre lugares se aprecia que: las columnas 1 y 14 son iguales a cero (rojo) y las columnas 10, 11, 12, 13 y 17 son iguales entre sí (azul).

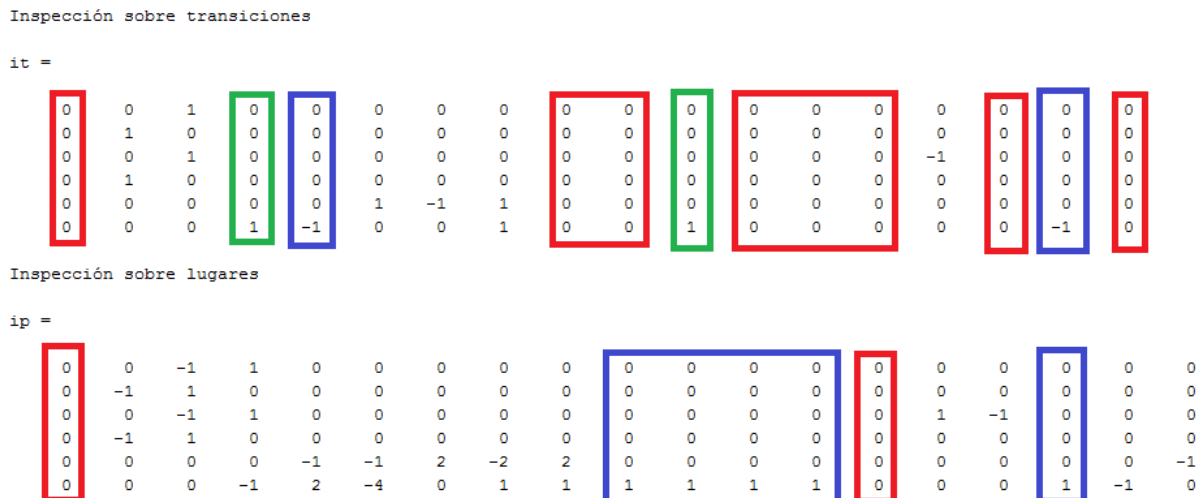


Figura 2.2: Primera verificación de diagnosticabilidad

Con el número de las columnas iguales a cero de la matriz de inspección sobre transiciones se revisa la matriz de transiciones/señales In-Out (Figura 2.3), se observa que en las transiciones 9, 10, 12, 13, 14, 16 y 18 están variando las señales de los actuadores 4, 6, 8 y 12 correspondientes a v4, vg, ve y dg; son a estas señales de entrada que se les debe asociar un sensor pertinente para observar su operación. En la simulación se añaden las señales correspondientes a los actuadores 4, 6, 8 y 12 a las señales de sensor, para volver a ejecutar el proceso de identificación (Figura 2.4), donde la IPN obtenida mantiene el mismo número de lugares y transiciones.

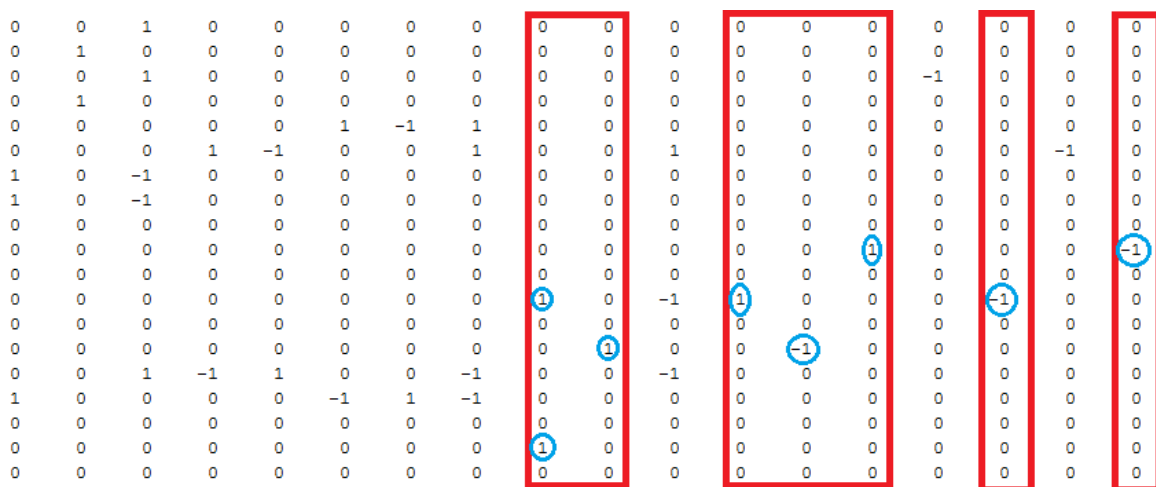


Figura 2.3: Interacción de señales In-Out y transiciones con 6 sensores

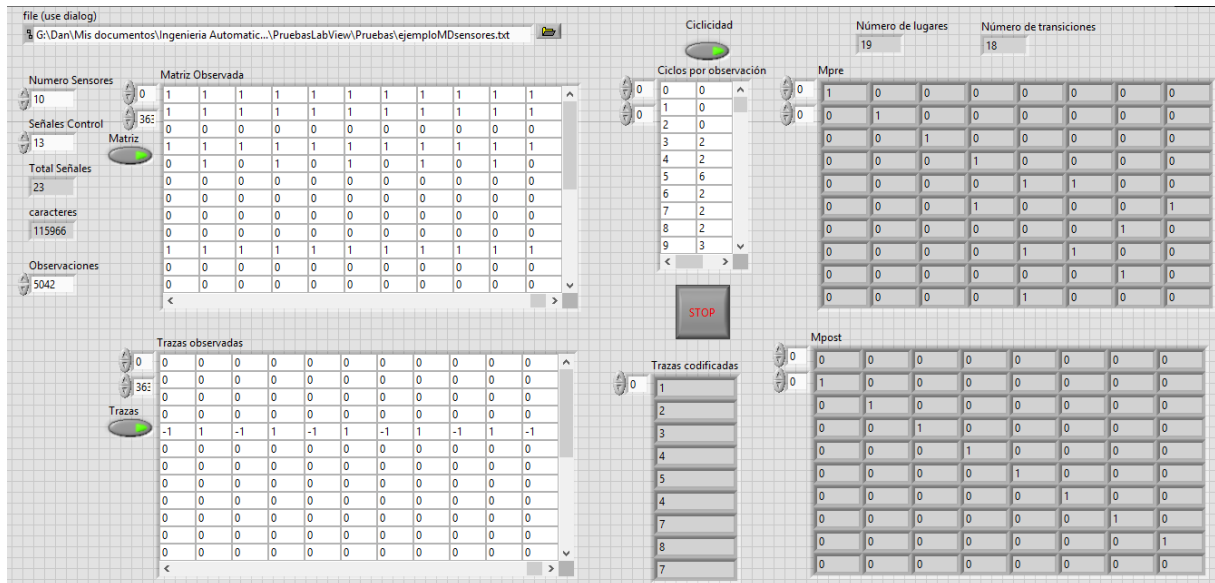


Figura 2.4: Proceso de identificación con 10 sensores

En la verificación de diagnosticabilidad (Figura 2.5), se observa que la matriz de inspección sobre transiciones ya no tiene columnas iguales a cero, pero las columnas 5 y 17 son iguales; por su parte la matriz de inspección sobre lugares tiene todas sus columnas diferentes a cero y distintas entre sí. (Figura 2.6).

Se procede a revisar las columnas 5 y 17 de la matriz de transiciones/señales In-Out, en la comparación de ambas columnas de la matriz, se ve que en la columna 5 se activa el actuador 9 correspondiente a f; en la simulación se añade la señal correspondiente al actuador 9 al conjunto de señales de sensor, para volver a ejecutar el proceso de identificación (Figura 2.7), donde la IPN obtenida mantiene el mismo número de lugares y transiciones.

En la verificación de diagnosticabilidad (Figura 2.8), se observa que las condiciones de evento-detectabilidad y de observabilidad se cumplen en las matrices de inspección sobre transiciones y lugares; la representación gráfica asociada a este sistema diagnosticable es la Figura 2.9, donde se asigna los sensores se a las transiciones y los actuadores a los lugares; si un lugar tiene una etiqueta s1s8, indica que los sensores 1 y 8 están activos, y si en un lugar hay una etiqueta a3a6, los actuadores 3 y 6 están operando; si aparece la etiqueta UT en una transición o UP en un lugar indica que son no observables, es decir, ningún elemento del sistema está activo en ese estado o evento.

Inspección sobre transiciones

it =

0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	-1	0	0	1	0	0	1	0	0	0	0	0	-1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1	0
0	0	0	0	0	0	0	0	1	0	-1	1	0	0	0	-1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Inspección sobre lugares

ip =

0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-1	1	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0
0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	-1	2	-2	2	0	0	0	0	0	0	0	0	0	-1
0	0	0	-1	2	-4	0	1	1	1	1	1	1	1	0	0	0	1	-1
0	0	0	0	0	0	0	0	0	0	-1	0	0	1	1	0	0	0	-1
0	0	0	0	0	1	0	-1	0	1	0	-2	1	0	0	1	-1	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0	0	1	0
0	0	0	0	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0

Figura 2.5: Segunda verificación de diagnosticabilidad

0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	-1	0	0	1	0	0	1	0	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-1
0	0	0	0	0	0	0	0	1	0	-1	1	0	0	0	-1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0
1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	-1	1	0	0	0	-1	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0	0	0
0	0	1	-1	0	0	0	-1	0	0	-1	0	0	0	0	0	0	0	0
1	0	0	0	0	-1	1	-1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 2.6: Interacción de señales In-Out y transiciones con 10 sensores

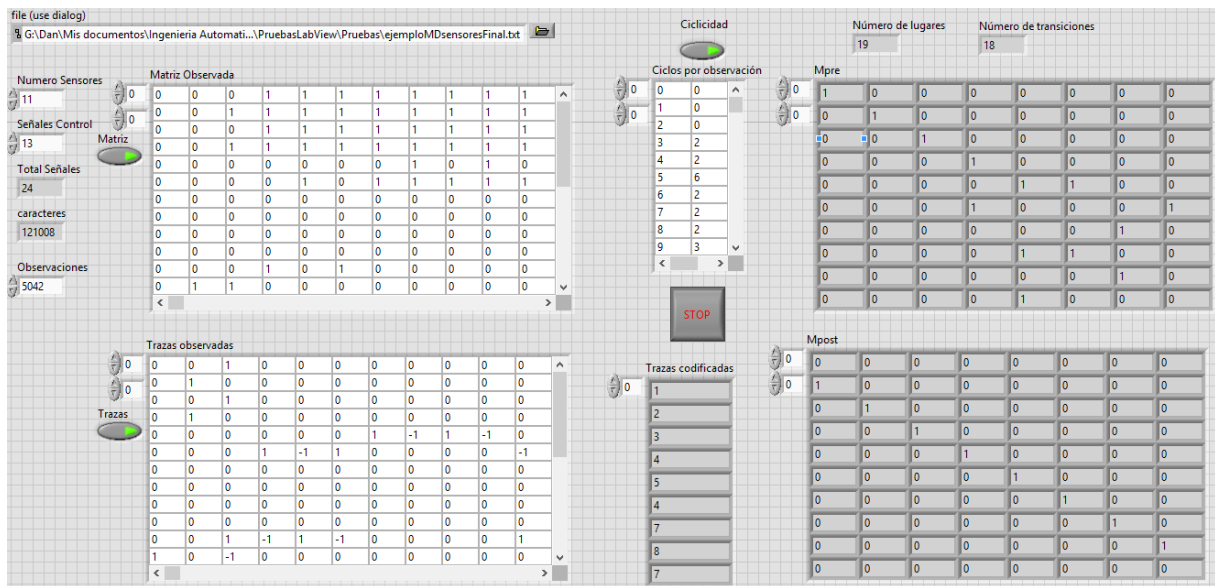


Figura 2.7: Proceso de identificación con 11 sensores

Inspección sobre transiciones

it =

0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	0	0
0	0	0	1	-1	0	0	1	0	0	1	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	-1
0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	-1	1	0	0	-1	0	0	-1	0	0	0	0	0	0	0

Inspección sobre lugares

ip =

0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	-1	1	0	0	0	0	0	0	0	0	0	1	-1	0	0	0
0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	-1	-1	2	-2	0	0	0	0	0	0	0	0	0	-1
0	0	0	-1	2	-4	0	1	1	1	1	1	1	1	0	0	1	-1
0	0	0	0	0	0	0	0	0	0	-1	0	0	1	1	0	0	-1
0	0	0	0	0	1	0	-1	0	1	0	-2	1	0	0	1	-1	0
0	0	0	0	0	0	0	0	0	0	1	0	0	-2	0	0	0	1
0	0	0	0	0	0	0	-1	0	1	0	0	0	0	0	0	0	0
0	0	-1	2	-2	4	0	-1	-1	-1	-1	-1	-1	0	0	0	0	0

Figura 2.8: Tercera verificación de diagnosticabilidad

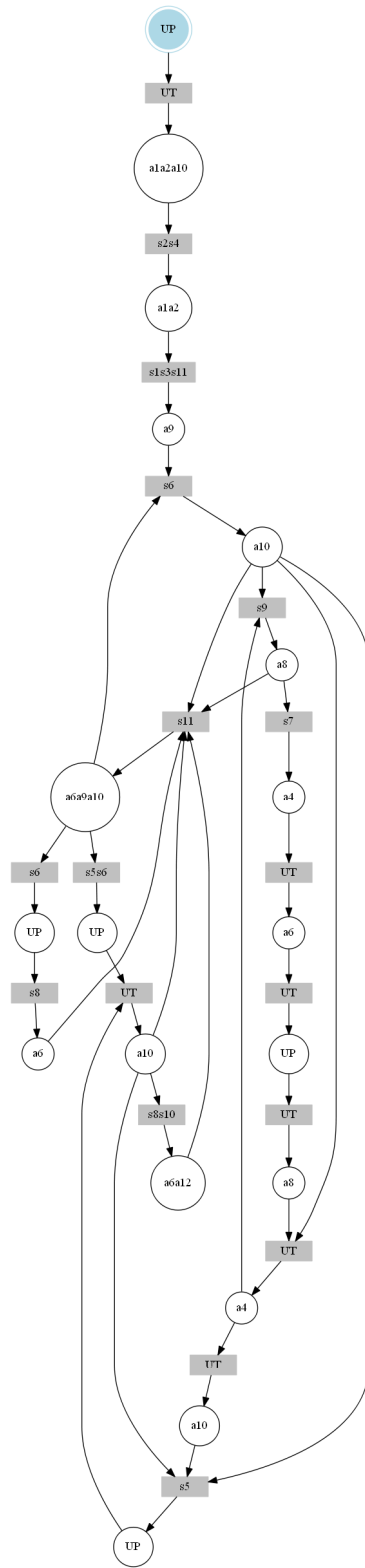


Figura 2.9: IPN asociada al sistema diagnosticable

Capítulo 3

Aplicativo Software de Evaluación

En este capítulo se explica la metodología utilizada para el desarrollo software implementado para la verificación de diagnosticabilidad en IPN por medio de señales de entrada-salida.

Iniciando con la construcción de herramientas software, de forma general existen dos métodos para su ejecución:

- ✓ Desarrollo caótico [38].
- ✓ Desarrollo metódico [39].

El desarrollo caótico, comúnmente denominado Programar y Corregir (Ensayo y Error), es el método más utilizado para aplicaciones software por su facilidad de ejecución inicial, pero no por esto el más adecuado; ya que no se tiene una claridad desde el comienzo de la funcionalidad de cada una de las partes del software, y solo por posibles errores en ejecución o compilación éstas son modificadas; otro aspecto en contra es que el incremento de funcionalidades del aplicativo software final se ve casi invalidado; pero el factor de mayor peso es que los plazos de entrega en relación a un proyecto no son precisos, ya que la fase de prueba o depuración del software final son largos.

Por otra parte, el desarrollo metódico aunque presenta mayor complejidad en la fase inicial de ejecución en comparación al caótico, brinda diversas ventajas al aplicarlos a la construcción de software necesario para un proyecto, asegurando que el desarrollo software sea previsible y eficiente, estas son: Documentación detallada de las funcionalidades del

software, y énfasis en la planificación (proceso inspirado en áreas de la ingeniería); pero su utilización es baja porque es necesario cumplir diversas características haciendo que el desarrollo software sea demorado.

Por el continuo uso de estos dos métodos y sus deficiencias, entre 1980-90 se han desarrollado métodos que combinan características de ambos, e incorporan soluciones de otros ámbitos como la ingeniería o la gerencia, dependiendo de las necesidades que se presenten para proyectos o problemas con requerimientos específicos; a los nuevos métodos obtenidos se denominan metodologías ágiles [40], [41].

3.1. Metodologías Ágiles

Las metodologías ágiles buscan compensar las falencias tanto del desarrollo caótico como del metódico, buscando un equilibrio óptimo del trabajo; para ello se reduce la documentación necesaria, se basa en la adaptación y mejoramiento con los cambios, y se orienta a las personas no a los procesos, todo con el fin de generar la percepción de lo realmente necesario [42]. Existen diversas metodologías entre las cuales se encuentran:

- ✓ Adaptive Software Development (ASD) [43], el desarrollo software adaptativo está compuesto por la fase de especificación o planificación del proyecto, la fase de colaboración o desarrollo y la fase de aprendizaje donde se verifica la calidad del software.
- ✓ Agile Modeling [44], el modelado ágil se basa en obtener un software funcional por encima de excesiva documentación, y prima la respuesta adecuada a cambios sobre seguir un plan específico.
- ✓ Extreme Programming (XP) [45], la programación extrema potencia las relaciones interpersonales para un desarrollo adecuado del software planteando roles específicos con obligaciones preestablecidas, estableciendo canales de comunicación entre cada uno de los grupos, y generando realimentación continua por parte del cliente.
- ✓ Feature-Driven Development (FDD) [45], el desarrollo centrado en características toma como fases primordiales en diseño y desarrollo, para los procesos iterativos mediante los cuales se generan entregas constantes al cliente de las soluciones software.
- ✓ Dynamic Systems Development Methodology (DSDM) [46], la metodología de desarrollo de sistemas dinámicos propone un proceso iterativo e incremental para tener un

balance entre la funcionalidad del producto final, el tiempo impuesto por el cliente y el uso de los recursos disponibles.

Ya que para el presente trabajo de grado se cuenta con un tiempo establecido de 9 meses, unos objetivos ya establecidos desde el anteproyecto y recursos limitados, se ha optado por implementar la DSDM para el desarrollo del aplicativo software.

3.2. Dynamic Systems Development Methodology

La DSDM se originó en 1994 por un consorcio de vendedores y expertos en el ámbito del software; dando un enfoque iterativo e incremental en el desarrollo de sistemas de información [47], (SI:conjunto de elementos orientados a tratar y administrar datos e información, organizados para su posterior uso, generados para cubrir una necesidad [48]); abordando los problemas comunes en los proyectos relacionados al desarrollo software: falta de tiempo, presupuestos ajustados, falta de cargos claros y requisitos flexibles. Para solucionar los problemas mencionados hay prerequisites para el uso de la DSDM:

- Interacción del equipo de desarrollo, clientes y gestores del proyecto.
- Facilidad de descomposición modular, dividir el proyecto en partes más pequeñas posibilita el enfoque iterativo e incremental.
- Definición clara de los requisitos del SI, con orden de prioridad en su implementación.

Con los prerequisites de la DSDM, se plantean sus principios de operación y cumplimiento:

1. Obtener un sistema que se aproxime a las necesidades actuales del proyecto, no a las que se plantearon en su inicio.
2. No hay una solución perfecta sino adecuada, por lo que un 80 % del proyecto puede hacerse en un 20 % del tiempo previsto.
3. Entrega en fecha estipulada, presupuesto previsto y con buena calidad.
4. Exigencias para un SI flexible.

5. Trabajo en paralelo de las fases, definiendo grupos o roles adecuados en cada caso.
6. Comunicación de las partes interesadas.
7. Participación constante de los usuarios.
8. Existen equipos de desarrollo y de gestión.
9. Los equipos integrantes tienen la capacidad de decidir.
10. El producto final se puede mejorar o expandir en proyectos futuros.

3.2.1. Fases de la DSDM

Pre-proyecto

Como toda etapa inicial de un proyecto, en la fase de pre-proyecto de la DSDM se realiza la identificación del problema para establecer una idea base; se establece la financiación de la solución. Al ser un problema que tiene relevancia, en su análisis se deben buscar los medios para la solución del mismo y se determinan los compromisos de realización de todas las partes, con el fin de definir responsabilidades y beneficios en la solución del problema que se ha identificado. El pre-proyecto se debe ejecutar de forma meticulosa y clara para evitar problemas futuros.

Ciclo de vida

En esta fase se tienen etapas o niveles comunes a todos los proyectos, pero de forma adicional se trabaja de forma iterativa tal que se pueda volver o replantear las etapas anteriores Figura 3.1 [1]; a continuación se describen los niveles del ciclo de vida del DSDM, junto con las actividades más relevantes de cada uno:



Figura 3.1: Ciclo de vida DSDM (adaptado de [1])

– Nivel 1: Estudio de Viabilidad

- ✓ Evaluación de cumplimiento de los requisitos de la DSDM.
- ✓ Informe y primer prototipo o bosquejo de viabilidad.

– Nivel 2: Estudio de Mercado

- ✓ Financiación de todo el proyecto.
- ✓ Necesidades y deseos de los usuarios afectados.
- ✓ Talleres con los stakeholders.
- ✓ Escalonamiento para el plan de desarrollo bajo el método MoSCoW (Most-Should-Could-Wont) [49].
- ✓ Delimitación temporal para los objetivos (Timeboxing).
- ✓ Arquitectura del sistema y plan de desarrollo.
- ✓ Identificar posibles riesgos.

Para los niveles 3, 4 y 5 se llevan a cabo 4 procesos similares: Identificar, Planear, Crear y Revisar; los cuales se adaptan a las características de cada nivel.

– Nivel 3: Análisis Funcional

- ✓ Creación del modelo funcional a partir de los requisitos del plan de desarrollo del nivel 2.
 - ✓ Ciclo de prototipado.
 - ✓ Obtención del modelo y prototipo funcional, mediante una lista de solicitudes crecientes por parte del cliente o los usuarios.
- Nivel 4: Diseño
- ✓ Integración de los componentes funcionales.
- Nivel 5: Implementación
- ✓ Ejecución de pruebas sobre el producto final.
 - ✓ Documentación lista para usuarios finales (manuales o guías) y entrenamiento o capacitación a los mismos.

Post-proyecto

En la etapa final del DSDM se debe asegurar un sistema de acción eficiente, dejar un listado de posibles mejoras y mantenimiento necesario, al igual que las posibles actualizaciones que se puedan aplicar al sistema desarrollado o sobre uno ya existente.

3.3. Desarrollo Software

Por lo que todo trabajo de grado es en sí un proyecto, y en este se tiene como uno de sus objetivos específicos el desarrollar un aplicativo software para la evaluación de sistemas de eventos discretos; es pertinente el aplicar el DSDM al proyecto que se denominará “Software DES”, donde los distintos participantes con su respectivo roles son:

- Mariela Muñoz (Cliente, Supervisora, y Asesora).
- Daniel Valencia (Desarrollador, Investigador y Evaluador de Desempeño).
- Departamento de Electrónica, Instrumentación y Control (Evaluador de Viabilidad).
- Consejo de la Facultad de Ingeniería Electrónica y Telecomunicaciones (Evaluador de Mercado).

- Estudiantes del programa de Ingeniería en Automática Industrial (Usuario final).

Con los roles del proyecto Software DES definidos, se da pasó a la ejecución de las fases del proyecto:

3.3.1. Pre-proyecto

Esta fase es desarrollada entre la cliente, el desarrollador y la asesora, llegando a un acuerdo común a cada parte, documentando los resultados en el formato de presentación de trabajo de grado al departamento. Teniendo como fecha de inicio el 16 de abril de 2017 (16/04/2017) y concluyendo el 24/05/2017.

3.3.2. Ciclo de vida DSDM

Estudio de Viabilidad

La evaluación de los prerequisites del DSDM se lleva a cabo por parte del Investigador, dando el visto bueno para la aplicación de la metodología en el proyecto Software DES. El grupo evaluador de viabilidad toma el documento desarrollado en la fase del pre-proyecto como informe de viabilidad, revisando que la idea del proyecto sea pertinente, dando aprobación para la continuación del mismo bajo modificaciones pequeñas del informe presentado y planteando una investigación más específica. Dando fin al nivel 1 del DSDM el 29/05/2017.

Estudio de Mercado

Con la información recolectada por el investigador y las observaciones del grupo evaluador de viabilidad, el desarrollador y la asesora se reunieron en 14 ciclos de talleres semanales con duración de una hora para el análisis de usuarios afectados (estado del arte y aporte), el escalonamiento del plan de desarrollo y timeboxing (actividades y cronograma), el plan de desarrollo (objetivos y condiciones de entrega), y la financiación del proyecto (recursos, presupuesto y fuentes de financiación); la información se registró en el anteproyecto de trabajo de grado. La arquitectura del sistema a desarrollar en Software DES se fue elaborando de forma paralela con la información de las otras actividades junto con los

posibles riesgos, a cargo del investigador. Finalizando este nivel con la aprobación del grupo evaluador de mercado el 19/09/2017.

Análisis funcional y Diseño

Se decidió trabajar de forma conjunta el nivel 3 y 4 para utilizar de forma eficiente el tiempo, porque para este proyecto todos los módulos desarrollados (Análisis funcional) deben trabajar en conjunto (Diseño y Construcción); tomando como base el conocimiento del investigador y la asesora sobre diagnosticabilidad.

En el nivel de modelo o análisis funcional, se utilizó como herramientas de desarrollo software LabView [50] de Nationals Instruments para la identificación del sistema y MatLab [51] de MathWorks para verificar la diagnosticabilidad del mismo. Se eligieron estos dos programas por sus características offline y online, velocidad de operación, requerimientos hardware, funcionamiento modular y capacidad de exportar e incorporar todo el desarrollo en lenguaje C o Java. Para el análisis funcional se tomó como piloto de prueba la planta Fisher del grupo de automática industrial, se eligió esta planta: por estar disponible en el laboratorio; por simular un proceso de manufactura donde se involucran pistones, suministro de piezas y bandas transportadoras; por generar su respuesta en tiempo discreto; por brindar facilidad en la extracción de datos y por ser una planta de uso pedagógico.

Las responsabilidades de los roles se definieron involucrados en la etapa de construcción del proyecto de la siguiente forma: la fase de reconocer (identificar) y planear está a cargo del investigador y la asesora, la de crear por el desarrollador y la de revisar por la supervisora y el desarrollador. Para lograr los objetivos de los niveles 3 y 4 se realizaron 7 ciclos incrementales de Reconocer-Planear-Crear-Revisar, descritos a continuación.

✓ Reconocer

- a. Se inicia con el proceso de identificación del modelo de un sistema a partir de un método de caja negra de un sistema dado, en el cual se tiene como entrada una serie de datos binarios (1: operación - 0: inactivo) de las señales de entrada-salida del sistema; y bajo el procesamiento se tiene como salida la matriz de observaciones.
- b. El tiempo de muestreo de los datos no es fijo y se pueden repetir los datos obtenidos, se necesita un módulo funcional que se encargue de eliminar los vectores columna consecutivos de la matriz de observaciones para observar los datos que generan cambios en el sistema.

- c. Ya que el aplicativo software será destinado para analizar sistemas de manufactura, en ellos se pueden presentar ciclos de operación es decir, después de un determinado número de estados el sistema volverá a un estado determinado; ganando importancia la determinación de los ciclos que se observen en ω .
- d. Siguiendo con el proceso de identificación de caja negra, se procede al cálculo de la matriz de vectores de eventos del sistema.
- e. El manejo de vectores presenta falencia, ya que su diferenciación se debe hacer componente a componente, y consumen mayor memoria que un valor escalar; por estas falencia se desarrolló un módulo capaz de procesar los vectores de eventos y convertirlos de forma única a un valor escalar.
- f. 1. Logrando diferenciar de forma clara cada una de las transiciones del sistema, se obtiene el lenguaje o secuencia de transiciones del mismo, a partir del cual se llega a la identificación del sistema construyendo de forma directa la matriz pre y post, y calculando la matriz de incidencia C. Dado que el proceso de construcción es complejo de entenderlo directamente en los bloques de LabView se optó primero implementarlo en MatLab, bosquejando el quehacer del módulo.
2. El investigador en la documentación de LabView (Manuales de usuario y manuales) encontró dos bloques, el primero sirve para implementar los script de MatLab, pero el consumo de máquina es alto; el segundo utiliza el lenguaje C y su velocidad de procesamiento es rápida.
- g. Con la identificación del sistema mediante señales entrada-salida, se continua con implementar el método propuesto para verificar diagnosticabilidad (sección 2.2), en MatLab realizándose de forma offline, ya que la diagnosticabilidad se determina de forma apriori a la operación online y continua del sistema.

✓ Planear

- a. Para el primer módulo se estipuló un tiempo de 2 semanas de desarrollo en LabView, para la construcción de la matriz de observaciones ω la información necesaria es la cadena de datos secuenciales del sistema, el total de señales que se tienen del sistema para saber la longitud de las columnas de ω , el número de observaciones que se hicieron del sistema delimitando la cantidad de filas de ω .
- b. Ya que se identificó el problema del primer módulo se da 1 semana para los nuevos requisitos, donde las entradas son el número total de señales, la matriz de observaciones ω y el número de observaciones; y la información relevante de salida es la

matriz de observaciones mínima (sin vectores consecutivos repetidos) y el verdadero total de observaciones del sistema, ya que se están eliminando datos repetidos.

- c. Para determinar la ciclicidad de ω se colocó como tiempo límite 2 semanas, la idea a seguir es comparar las columnas de ω con todas las sucesoras, hasta que se llegue a la última columna; registrando en un vector las iteraciones que se demora una columna en repetirse, sino se repite el valor es cero.
- d. Para obtener un vector de evento se debe restar dos vectores consecutivos de ω (vector $k+1$ menos vector k), la creación de su respectiva matriz es el agrupar en orden de observación cada uno de estos vectores, de esta forma se obtienen todas las transiciones de la planta por señales entrada-salida; se da 1 semana para obtener el módulo que haga este proceso.
- e. Debido a la complejidad del proceso que manifestó el investigador, y por un consenso entre la supervisora y el desarrollar se dieron 3 semanas, donde el proceso a desarrollar es generar un etiquetado para las transiciones asociadas a los vectores de eventos; sumando término a término 1 a cada columna porque los valores de estas son 1(activación), 0(inactividad) y -1(desactivación) quedando en 2,1 y 0 respectivamente para así pasarlo a formato string. El etiquetado se da en la comparación de las cadenas de 2,1 y 0, en donde sí no existe ese valor en la iteración presente o antecesoras se da una etiqueta (empezando desde 0,1,2,...).
- f.
 1. Por la dificultad en la construcción del módulo se dio 4 semanas, donde los pasos generales a seguir son: Definir el vector de transiciones (todas las posibles palabras del sistema) y el vector de secuencia de transiciones (el lenguaje obtenido en el sistema); Por el modelado en IPN una transición siempre debe tener un lugar antecesor y un lugar predecesor; Un lugar queda determinado por una transición antes y después de este, generando cada uno de los lugares en función de donde viene y hacia donde va. Respecto a los vectores de transiciones antecesoras y predecesoras a los lugares, se obtiene la información necesaria para obtener la matriz Pre, Post y de Incidencia; logrando la identificación del sistema por sus señales entrada-salida.
 2. Con el nuevo bloque de LabView, la tarea dada es el convertir todos los comandos y lógica de MatLab a lenguaje C.
- g. La verificación de diagnosticabilidad desarrollada es compleja desde el punto de vista de la formulación, pero en la implementación software es de menor dificultad dando una semana para este módulo; el proceso se inicia con la matriz de vectores de eventos, extrayendo la submatriz solo con las señales de salida (sensores) siendo la matriz de inspección sobre transiciones; esta nueva matriz se multiplica por la

matriz de incidencia traspuesta para obtener la matriz de inspección sobre lugares, evaluando las condiciones de evento-detectabilidad en la matriz de inspección sobre transiciones y las de observabilidad sobre la matriz de inspección sobre lugares. Si estas dos matrices cumplen con sus respectivas condiciones se dice que el sistema es diagnosticable.

✓ Crear

Los items a, b, c y d se trabajaron de forma paralela, al igual que e y f al terminar el primer grupo, y de forma individual g.

- a. Del tiempo estipulado de 2 semanas, se utilizó solo una.
- b. Se cumplió con el tiempo estipulado para el segundo módulo; añadiendo la exportación de la matriz de observaciones en un archivo txt para su posterior análisis, si es necesario.
- c. La idea base se logró implementar en una semana, pero se utilizó el tiempo restante en depuración antes de presentar el módulo, ya que se presentaban carencias al incrementar el número de datos.
- d. Fue necesario el utilizar dos días mas de lo asignado para este item.
- e. Se llegó a un primer prototipo de módulo en 1 semana, sin embargo presentaba fallas en columnas de longitudes superiores a los 15 componentes; observando que se generaba un desbordamiento de memoria en la estructura de LabView, se compensó el problema en una semana más y se depuró hasta con vectores de 200 componentes sin error alguno, y exportando los datos de etiquetado-vector y la secuencia de etiquetas en hojas de calculo xls.
- f.
 1. De los pasos a seguir para este módulo se logró en 2 semanas obtener en MatLab, que trabaja con la información en las hojas de cálculo de la secuencia de etiquetas; se verificó cada uno de los modelos identificados del sistema por parte del desarrollador de manera manual.
 2. En las siguientes 2 semanas, se pasó todo el script de MatLab a lenguaje C tomando como medida patrón el funcionamiento del módulo en MatLab.
- g. El desarrollar este módulo consumió un día. De forma complementaria por observaciones del investigador y la cliente, se desarrolló un segundo módulo encargado de la representación gráfica de la IPN asociada al sistema identificado, generando el código en MatLab para implementarlo en GraphViz [52]. Este módulo tiene como entrada la matriz Pre y Post, genera el conjunto de lugares con el número de filas y

el conjunto de transiciones con el número de columnas de las matrices; a partir de la matriz Pre en la posición de columna y fila con valores iguales a 1, se crean los arcos que van de los lugares a las transiciones, con la matriz Post en la posición de columna y fila con valores iguales a 1, se crean los arcos que van de las transiciones a los lugares; generando la secuencia entre lugares y transiciones que necesita GraphViz para su funcionamiento.

Para el etiquetado de lugares y transiciones, el módulo extrae de la matriz de vectores de eventos la submatriz de las señales de salida es decir, solo la dinámica de los sensores comparando si es mayor a 0 para saber si un sensor está activo o no en una transición (s1s2...) determinada; lo mismo se hace con las señales de entrada, pero con la matriz resultante de la multiplicación de la submatriz de señales de entrada y la matriz de incidencia traspuesta, para conocer la activación de los actuadores en cada lugar (a1a2...). En el caso de que todos los actuadores en un lugar sean 0 se dice que es un estado no observable (UP), y si sucede con los sensores en una transición se presenta una transición no observable (UT).

✓ Revisar

- a. Revisando este primer bloque se observó que había muchos vectores consecutivos que se repetían, esto debido a que no hay un tiempo de muestreo de datos establecido.
- b. El módulo cumplió con su funcionalidad y no se observaron carencias de operación.
- c. Por la depuración se cumplieron las expectativas del módulo mencionado.
- d. Se revisó el módulo en el tiempo estipulado donde no cumplía con la funcionalidad, el desarrollador tuvo 2 días más logrando alcanzar los requerimientos establecidos.
- e. Se evaluó al cabo de las 3 semanas el módulo de etiquetado, cumpliendo los requerimientos establecidos.
- f.
 1. Se vio que la funcionalidad del módulo en MatLab es la adecuada, pero no la requerida, ya que se debe implementar en LabView.
 2. El módulo en LabView alcanzó todas las expectativas necesarias para el proyecto.
- g. Sobre el módulo solicitado se decide que cumple con sus objetivos, y el módulo adicional ayuda a resaltar que se está trabajando con el formalismo de IPN e identificar los estados y transiciones no observables que varían en función a las señales que se tengan de la planta.

Implementación

I. Identificar

Se dio la aprobación por parte de la cliente y del investigador, valorando que cumplía todos los requerimientos establecidos, para iniciar con el proceso de capacitación del usuario final.

II. Planear

Para la capacitación, se planeó tomar una muestra de la población de estudiantes del programa de ingeniería en automática industrial(PIAI), utilizando una guía de usuario para el uso del software. Para de forma posterior, evaluar el desempeño del software y de la guía esperando una aceptación mayor a 8 en una escala del 1 al 10.

III. Crear

Se desarrolló la guía de usuario (anexo B), y para la evaluación global de desempeño se formuló una encuesta conformada por dos preguntas:

1. ¿De 0 a 10 califique la utilidad del aplicativo software (donde 0 es malo y 10 es excelente), de acuerdo al método de identificación paramétrico?
2. ¿De 0 a 10 califique la utilidad de la guía de usuario para el uso del aplicativo software (donde 0 es malo y 10 es excelente)?

IV. Capacitar

La muestra de la población fue conformada por 17 estudiantes del PIAI que en el primer semestre del 2018 estaban cursando la electiva “Identificación de sistemas de eventos discretos” en la fecha 30/04/2018 en el horario de la asignatura de 6-8 pm; a los cuales se les socializó el método de identificación del aplicativo software durante una hora y en la siguiente se realizó la entrega del software, bajo 4 ejemplos de aplicación; después del uso por parte de los estudiantes del software se les entregó a cada uno la encuesta formulada para que la entregaran de forma anónima, obteniendo los siguientes resultados:

Los resultados obtenidos (Tabla 3.1) indican que por parte del usuario final hay una aceptación de 9,8 sobre 10 del aplicativo software y de 9,5 de la guía de usuario.

Tabla 3.1: Resultados de Desempeño de Implementación DSDM

Estudiante	Pregunta	
	1	2
1	10	10
2	10	10
3	10	10
4	10	10
5	10	10
6	10	10
7	10	10
8	10	10
9	10	10
10	10	10
11	10	10
12	10	9
13	10	9
14	10	9
15	9	8
16	9	8
17	9	8
Promedio	9,8	9,5

3.3.3. Post-proyecto

Al culminar con la entrega del proyecto se tuvo una última reunión entre el grupo investigador, la asesora y la cliente del proyecto, tratando los siguientes temas:

- ✓ **Utilidad en otros proyectos:** En el momento de entrega del aplicativo software no hay proyectos similares y a fines donde se pueda utilizar por completo o parte de los módulos desarrollados.
- ✓ **Tiempos de operación:** Con 5000 vectores de datos de una longitud de 30 datos por vector, el tiempo de respuesta del aplicativo software fue de 4 minutos.
- ✓ **Mantenimiento y Actualización:** Por el uso de Labview 2012 y Matlab 2013a, al pasar su implementación a versión posteriores no se debe realizar ningún cambio para su funcionamiento normal. Por parte del mantenimiento no es necesario realizarlo.

Capítulo 4

Resultados

En este capítulo se presentan los resultados experimentales, al implementar la prueba de diagnosticabilidad descrita en la sección 2.1 en conjunto con el aplicativo software obtenido de la sección 3.3. El proceso de experimentación queda resumido en 4 etapas:

1. Adecuación y adquisición de las señales In y Out del sistema, Planta + Controlador a analizar, en un tiempo establecido por el funcionamiento del sistema.
2. Identificación del comportamiento del sistema, a partir de los datos obtenidos con el módulo software desarrollado en LabView.
3. Representación del comportamiento como una red de Petri interpretada del sistema, obtención de las matrices de inspección sobre transiciones y lugares para verificación de diagnosticabilidad; por medio del módulo software desarrollado en MatLab con soporte en Graphviz.
4. Evaluación de resultados; si el sistema es diagnosticable se termina el proceso, caso contrario se incrementa el número de señales Out respecto a señales In y volver a la etapa 1.

La etapa 1 es capaz de implementarse en plantas reales como las del laboratorio de control de procesos del PIAI, sin embargo no se asocian a sistemas de manufactura (caso de interés), por lo que se eligieron 10 sistemas para simular en Simulink (modelado físico asociado a la operación del sistema), y la librería Stateflow (lógica de controlador y máquinas de estado asociadas a la planta). Se presenta un sistema para ilustrar el proceso de experimentación, los 9 sistemas restantes quedan registrados en el anexo C.

4.1. Sistema de Embotellamiento y Clasificación

Un sistema de embotellamiento de agua (Figura 4.1) realiza diferentes operaciones de llenado y clasificación de botellas, dependiendo de las dimensiones de éstas. El operador tiene que poner en funcionamiento el sistema, colocar botellas, retirar y colocar cajas. Las operaciones se dividen entre las botellas de 0,5 litros (Operación A), botellas de 2 litros (Operación B) y garrafas de 5 litros (Operación C). Las operaciones consisten en poner en funcionamiento el sistema apretando el botón de marcha (M). Al principio de la cinta 1 (M1), se encuentran los sensores que determinarán el programa a seguir en el resto de la operación. Para una botella de 0,5L se activa el sensor S1, para una botella de 2L se activa S2, y para una garrafa de 5L se activa S3. Cuando estén los sensores activados se pondrá en funcionamiento la cinta 1 (M1), hasta que la botella llegue al sensor de posición S4, que se encuentra debajo del depósito de agua. Si se trata de la operación A, el depósito estará abierto 5 segundos; si se trata de la operación B, el depósito estará abierto 10 segundos; y si es la operación C, el depósito estará abierto 20 segundos. Pasado el tiempo de operación indicado, se pondrá en funcionamiento la cinta transportadora(M1), la cual detendrá su movimiento cuando llegue al sensor S5. En este punto se debe tapar la botella con el tapón. Para ello está el pistón, que se encuentra en su posición inicial y donde un brazo posiciona un tapón; dependiendo del tipo de botella se activará el pistón 5, 10 y 20 segundos, cuando se coloque el tapón el piston vuelve a la posición inicial. Justo después la cinta transportadora (M1) continuará avanzando hasta el sensor S6, donde se encuentra el brazo B, que llevará las botellas a sus respectivas cajas. Para ello el brazo se encuentra en su posición inicial; cogerá las botellas cerrando las garras (activando su sensor de sujeción sj) y llevará hasta los sensores de las cajas (S7, S8 o S9), una vez ha dejado la botella el brazo volverá a su posición inicial. El depósito de agua se considera que siempre está lleno.

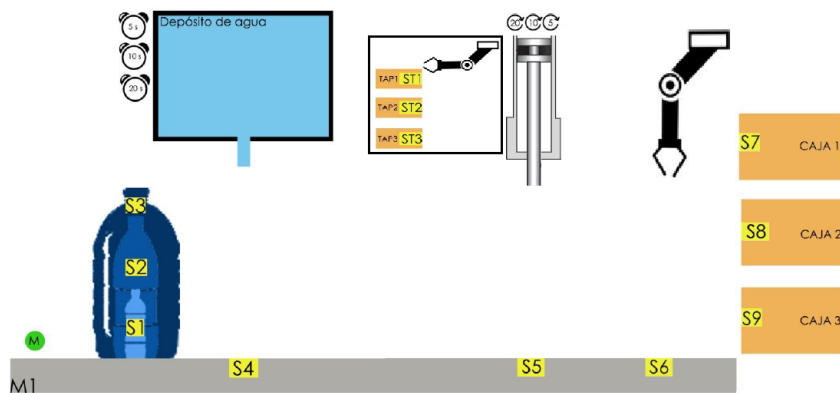


Figura 4.1: Sistema de Embotellamiento y Clasificación

Con la descripción del funcionamiento del Sistema de Embotellamiento y Clasificación (SEC) se implementó en simulink (Figura 4.2), con su respectivo controlador (Figura 4.3), cinta transportadora (Figura 4.4) y brazo robot con pinza (Figura 4.5); el funcionamiento del pistón y el llenado de las botellas no se simula ya que es temporizado y solo tiene dos estados de operación (On/Off). El control del brazo robot tiene como salidas BR1 y BR2 para codificar respectivamente las posiciones: Inicial (0-0), Caja botellas pequeñas(1-0), Caja botellas medianas(0-1) y Caja botellas grandes(1-1); de igual forma el control de la pinza tiene Pz1 y Pz2 para representar: Desactivación (0-0), Activación (1-1), y Stand By (1-0).

El grupo de signals builders en la Figura 4.2 se utilizó para simular la llegada de botellas, para simular un proceso aleatorio el bloque "Botellas", genera las señales de activación de los sensores S1, S2 y S3 (Figura 4.6).

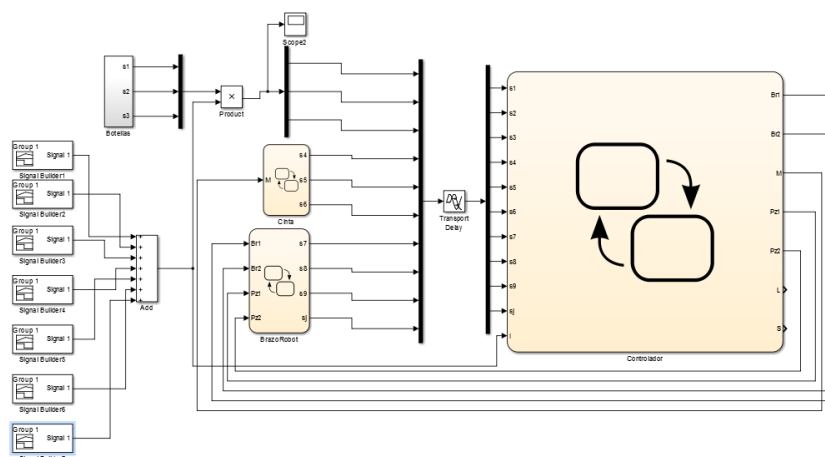


Figura 4.2: Sistema de Embotellamiento y Clasificación en Simulink

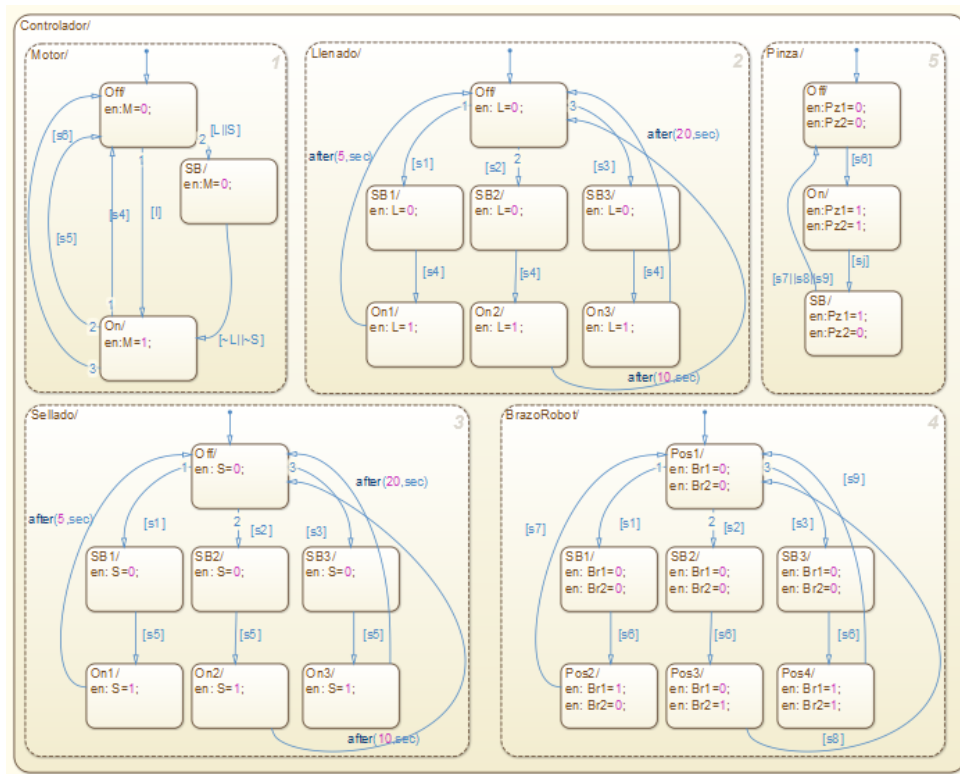


Figura 4.3: Controlador del SEC

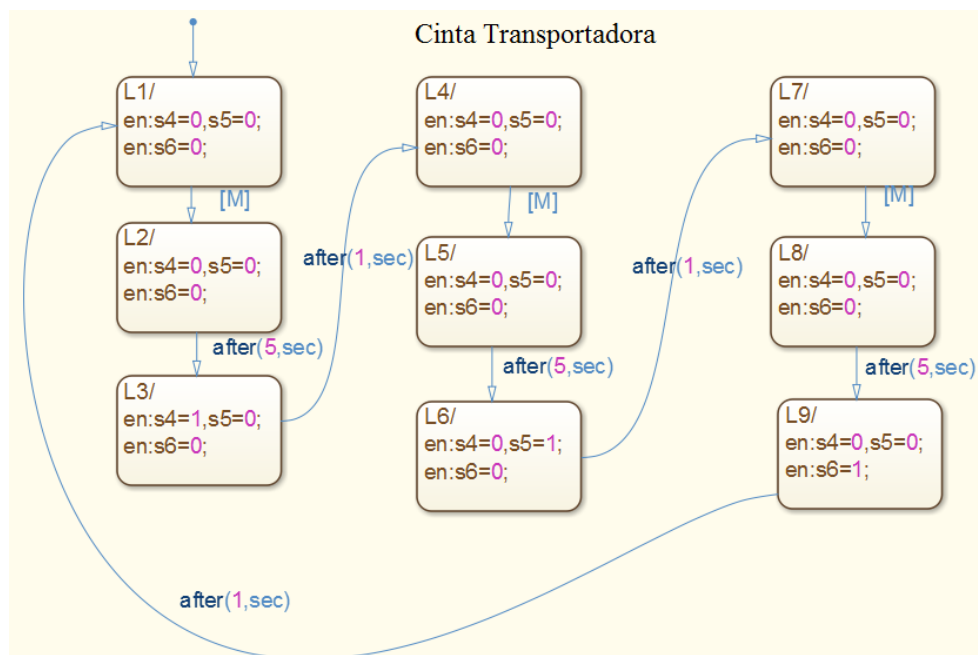


Figura 4.4: Cinta transportadora del SEC

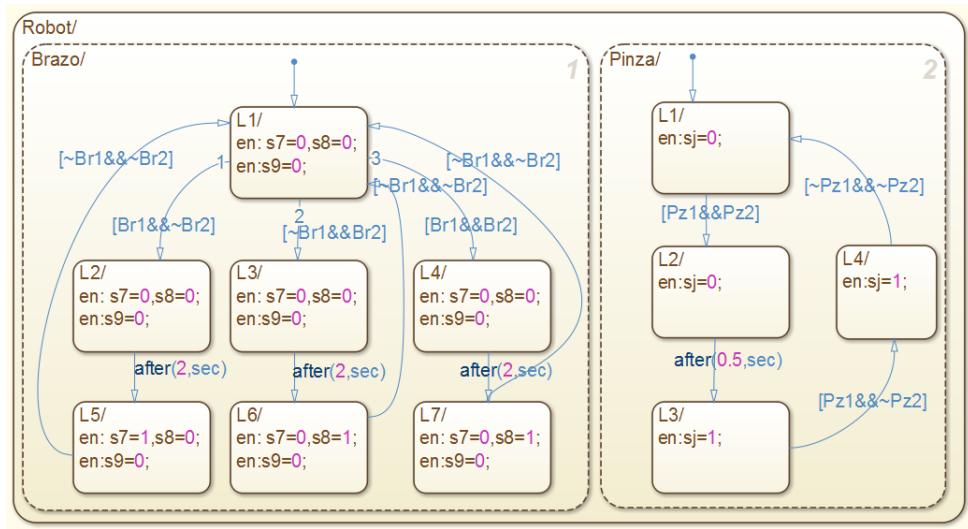


Figura 4.5: Brazo Robot del SEC

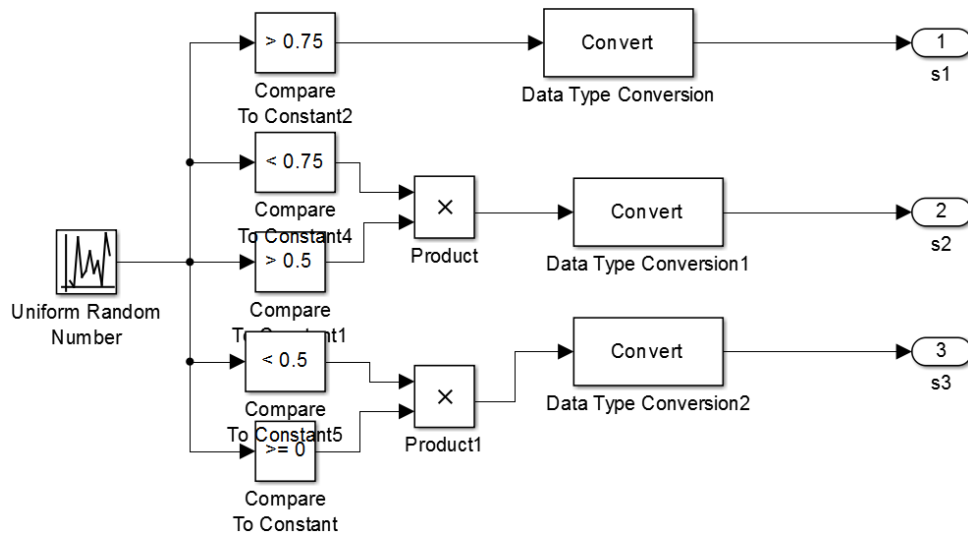


Figura 4.6: Llegada de Botellas del SEC

Se realizó el primer proceso de diagnóstico del SEC con las señales iniciales (10 señales de sensor y 7 señales de actuador) para un ciclo de operación C (Tabla 4.1), la identificación en Labview obtuvo una IPN de 15 lugares y 14 transiciones (Figura 4.7), la prueba de diagnosticabilidad indicó que el sistema no era observable y tampoco evento-detectable, por lo tanto no diagnosticable. Para una visualización clara de las IPN representadas en este capítulo, su comportamiento gráfico ha sido dividido en segmentos indicados en la estructura principal; que posteriormente son presentados de forma individual, en este primer caso la Figura 4.7 es precedida por Figura 4.8 y Figura 4.9. Como ya se mencionó en el capítulo 3, la IPN obtenida (Figura 4.8) representa las operaciones, acciones y sucesos del sistema, en el primer segmento de la IPN(Figura 4.8) se aprecia que el lugar inicial (circulo azul) no tiene ningún actuador en operación asociado por la etiqueta UP (Unknown Place), al pasar por la primera transición ,en la cual ningún sensor esta mandando información ya que tiene asociada la etiqueta UT (Unknown Transition), llega al segundo lugar donde están operando los actuadores 1, 2 y 3; dicho lugar puede activar 3 transiciones: la primera cuando el sensor 1 se activa llegando a un lugar UP, para luego para por una transición UT y terminar en un lugar donde solo esta operando el actuador 1; la segunda opción es pasar por una transición UT para llegar a un lugar que indica la operación del actuador 1, en el cual al activarse el sensor 5 se llega al lugar donde el actuador 3 está operando; la última opción es pasar por una transición UT para llegar a un lugar que indica la operación del actuador 1, en el cual al activarse el sensor 6 se llega a un lugar donde los actuadores 4, 6 y 7 están operando

Tabla 4.1: Señales In/Out de la SEC

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	M1	Cinta transportadora	s1	s1	Sensor botella A
a2	L	Señal de llenado	s2	s2	Sensor botella B
a3	S	Señal de sellado	s3	s3	Sensor botella C
a4	BR1	Señal 1 de brazo robot	s4	s4	Sensor posición llenar
a5	BR2	Señal 2 de brazo robot	s5	s5	Sensor posición sellar
a6	PZ1	Señal 1 de pinza	s6	s6	Sensor final de linea
a7	PZ2	Señal 2 de pinza	s7	s7	Sensor almacenamiento A
-	-	-	s8	s8	Sensor almacenamiento B
-	-	-	s9	s9	Sensor almacenamiento C
-	-	-	s10	sj	Sensor de sujeción

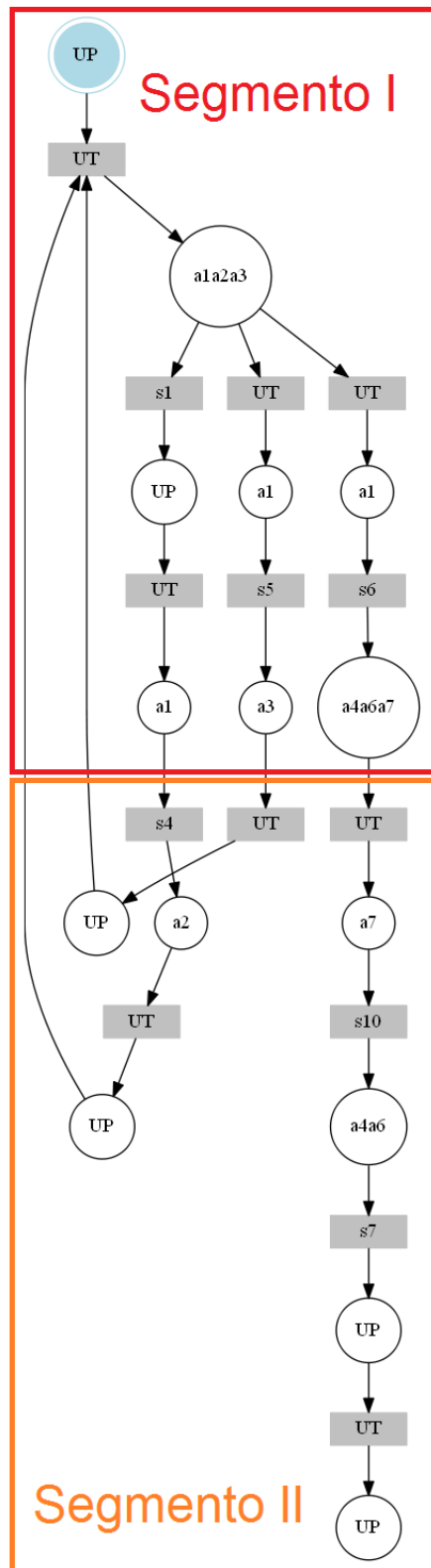


Figura 4.7: IPN #1 asociada al SEC

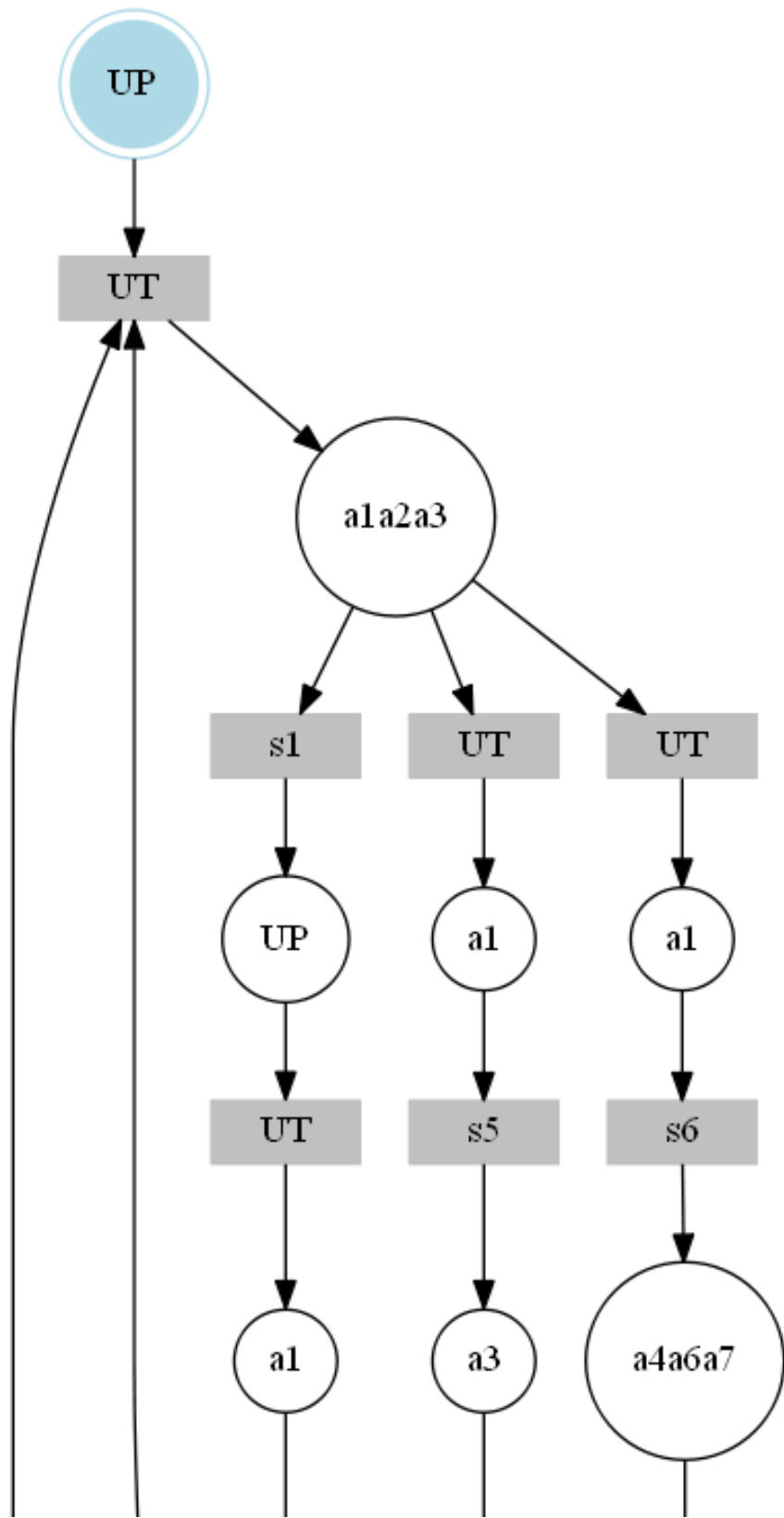


Figura 4.8: Segmento I de la IPN #1 asociada al SEC

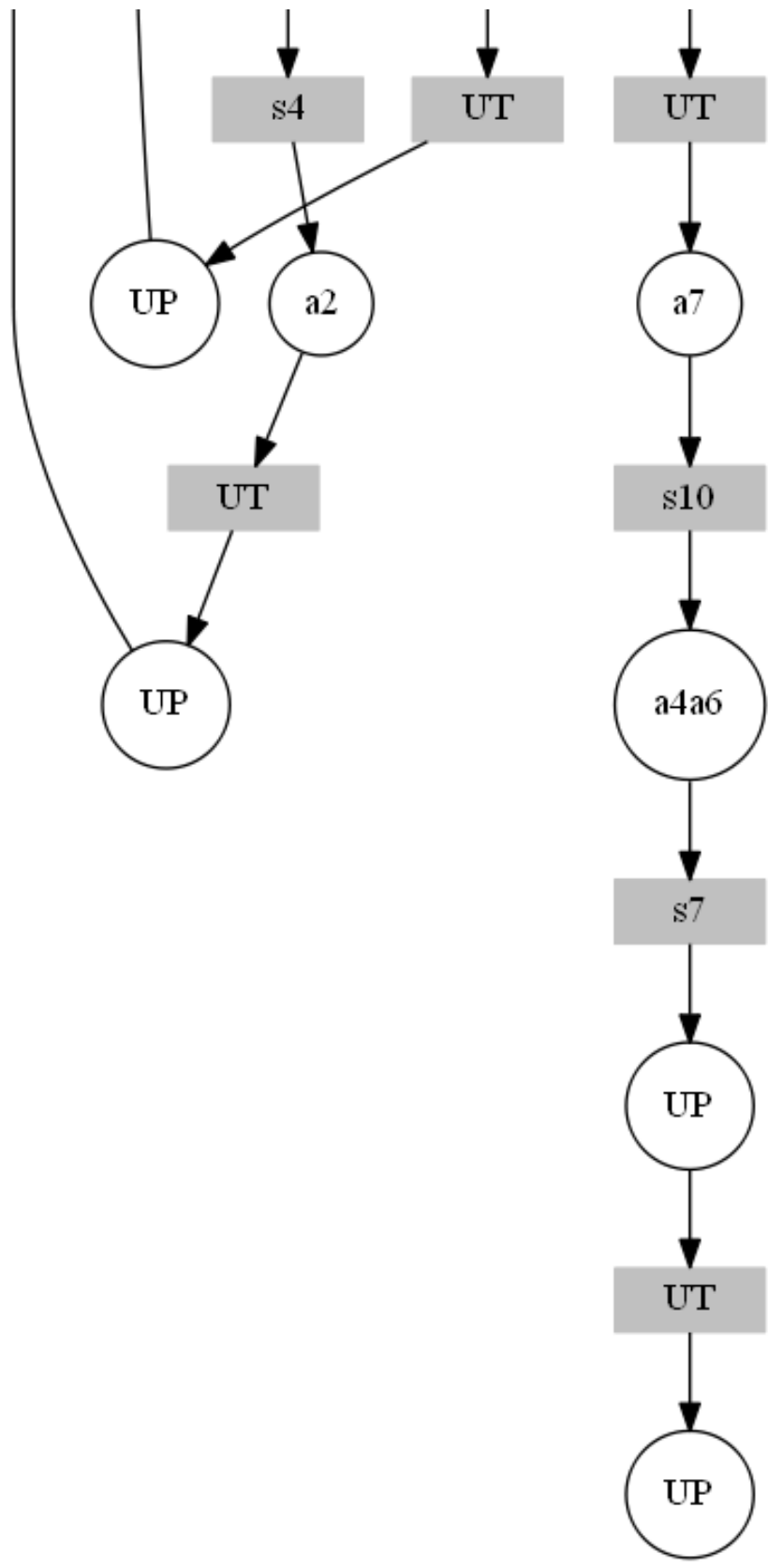


Figura 4.9: Segmento II de la IPN #1 asociada al SEC

Con el método de ubicación de sensores se determinó los sensores adicionales para que el sistema sea diagnosticable, estos deben estar asociados a la activación de la cinta transportadora, el llenado y sellado de las botellas; en la identificación de este sistema se obtuvo una IPN de 15 lugares y 14 transiciones (Figura 4.10, 4.11, 4.12).

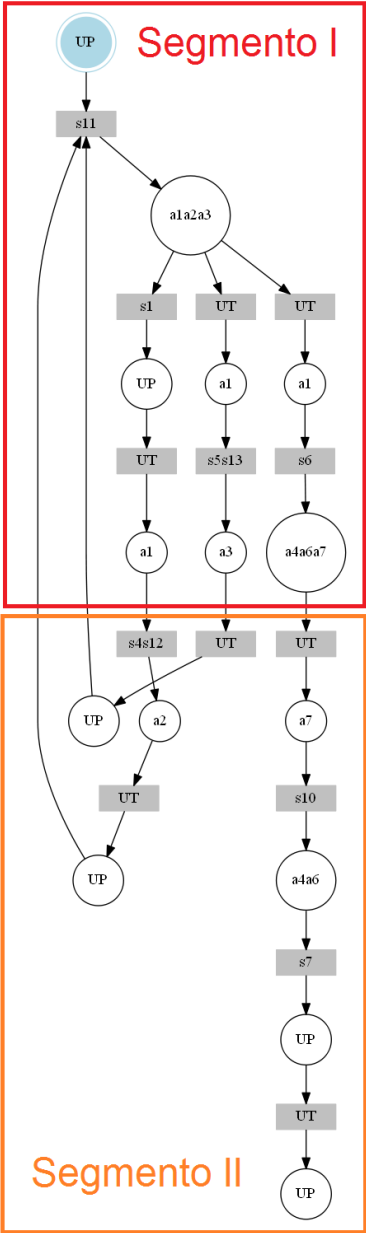


Figura 4.10: IPN #2 asociada al SEC con 13 sensores

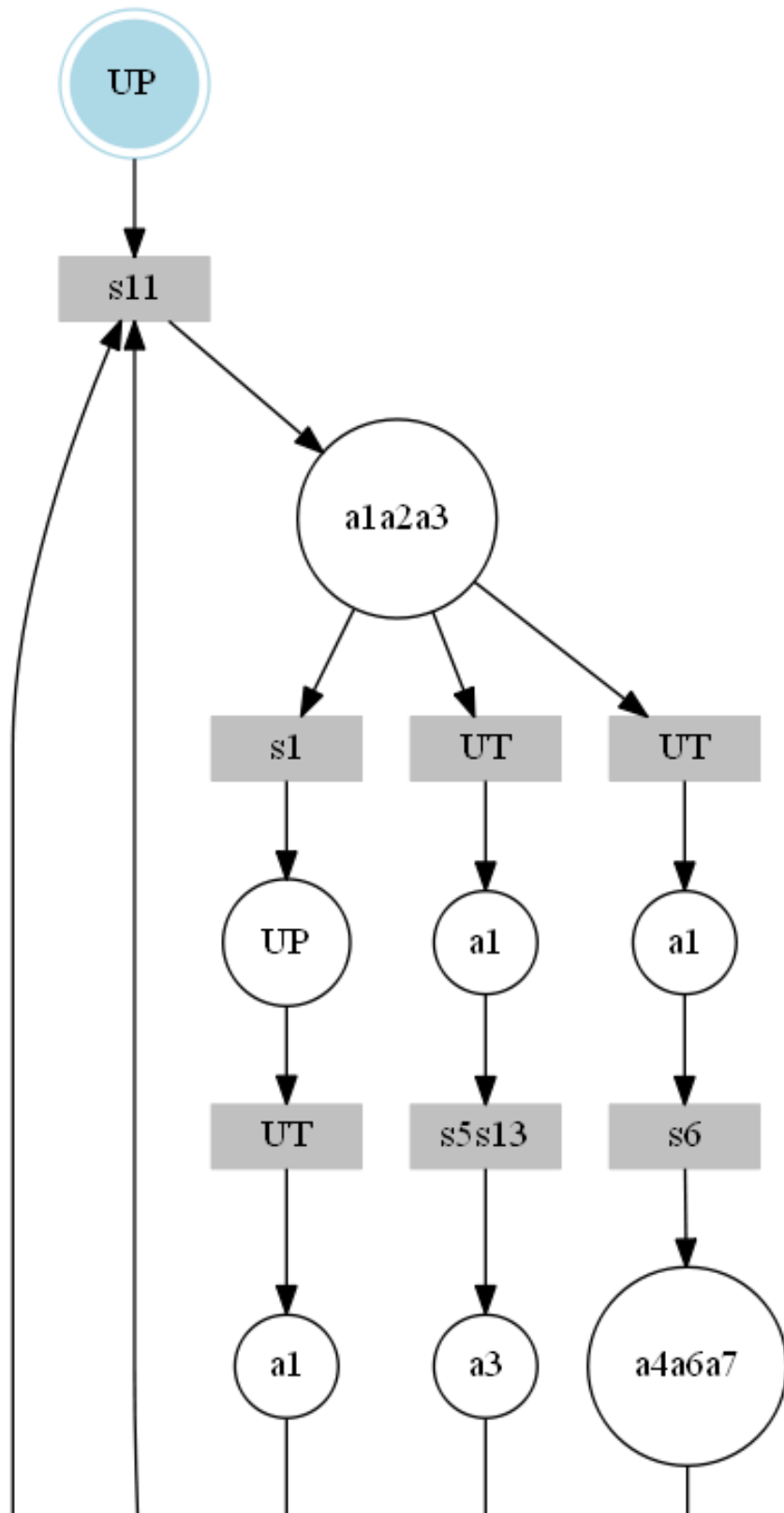


Figura 4.11: Segmento I de la IPN #2 asociada al SEC

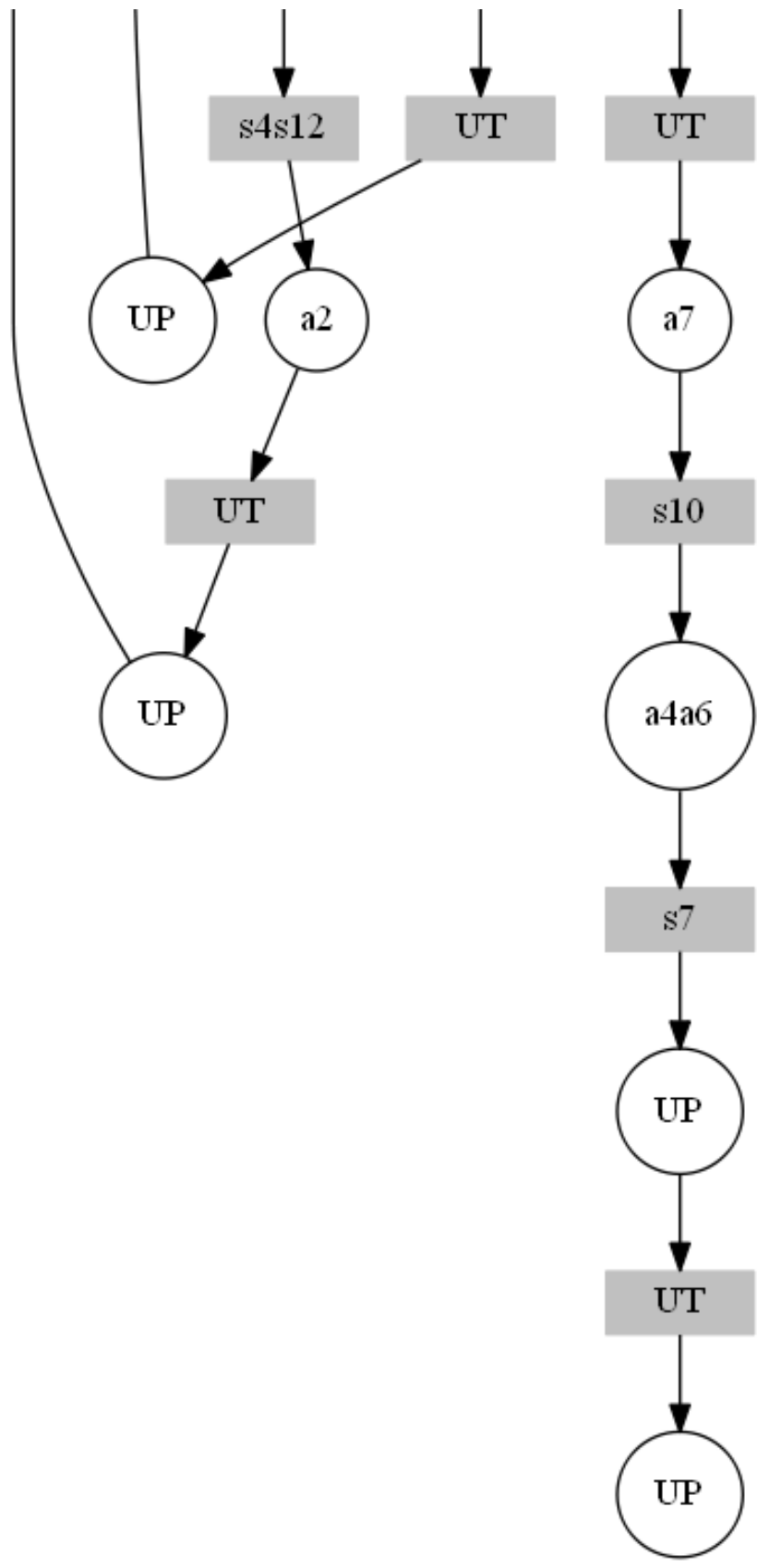


Figura 4.12: Segmento II de la IPN #2 asociada al SEC

Después se configuró el SEC para ejecutar 3 ciclos de operación C-A-C obteniendo una IPN de 21 lugares y 20 transiciones, con la cantidad de sensores de la segunda prueba el sistema no era diagnosticable (Figura 4.13, 4.14, 4.15, 4.16), pero el método indicó que para ello se necesitaba un sensor asociado a BR1 y Pz1 (Figura 4.17, 4.18, 4.19, 4.20).

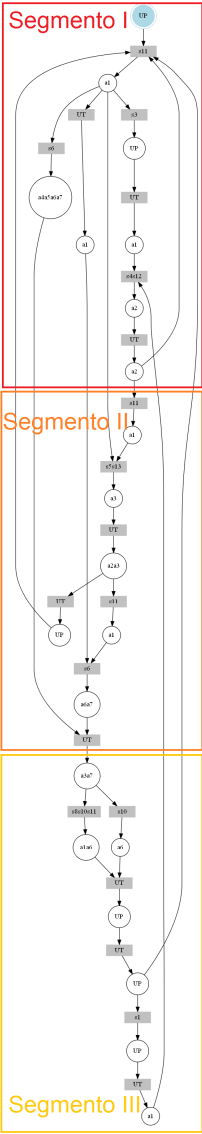


Figura 4.13: IPN #3 asociada al SEC con 13 sensores - Operación CAC

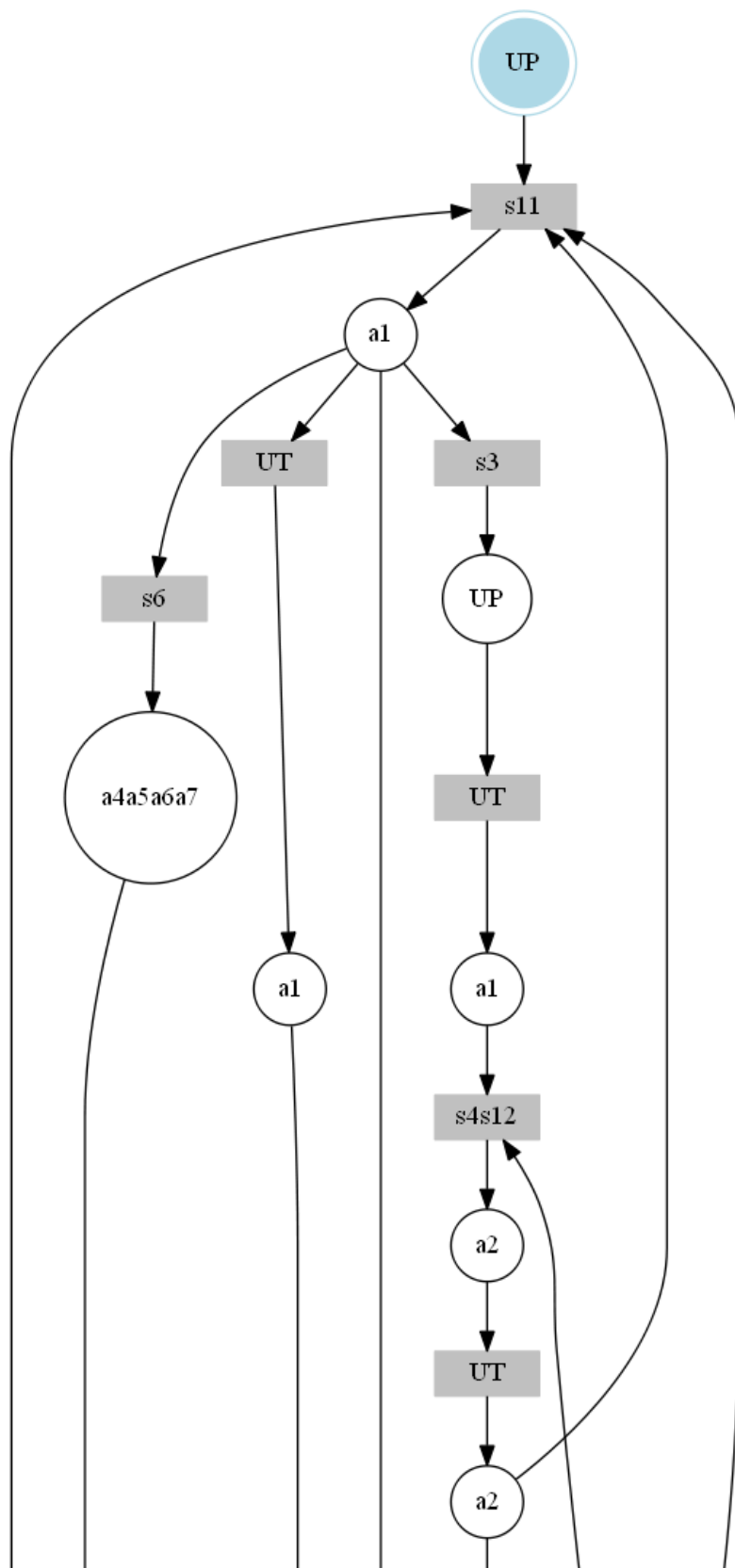


Figura 4.14: Segmento I de la IPN #3

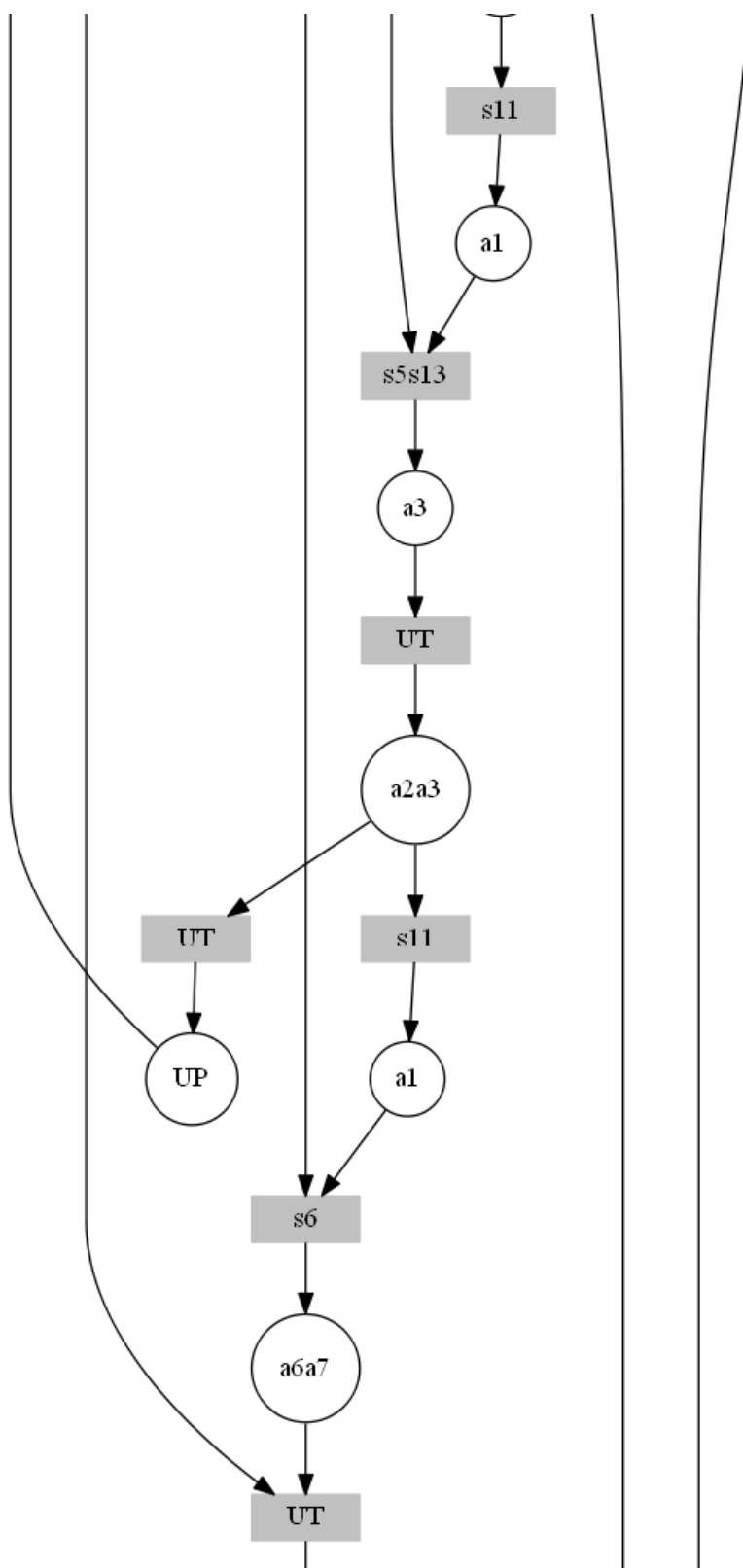


Figura 4.15: Segmento II de la IPN #3

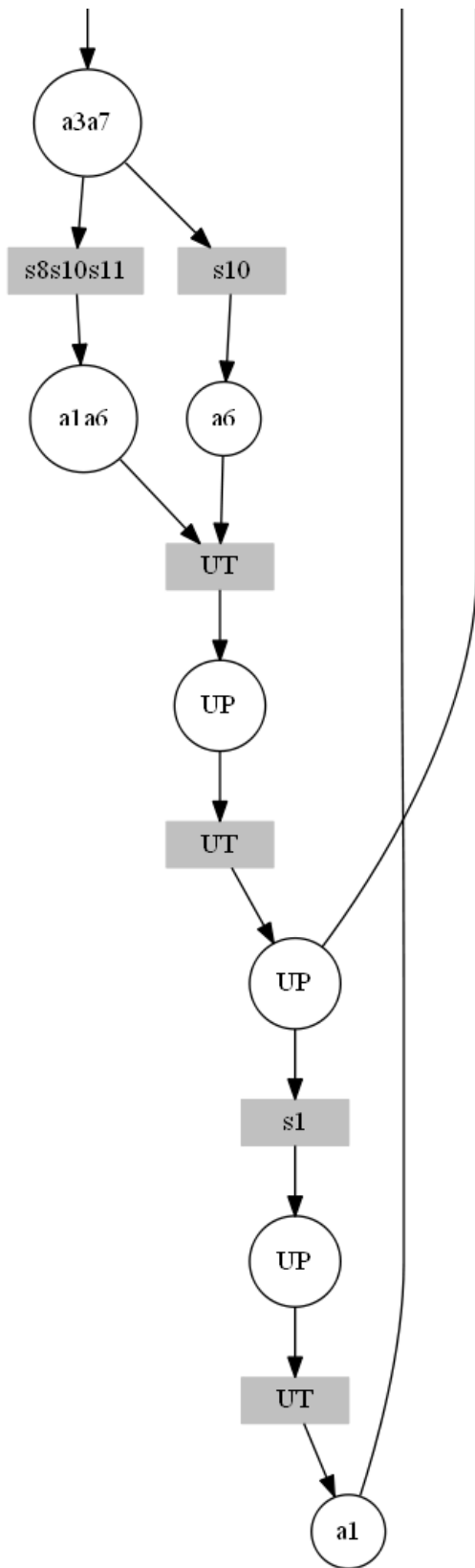


Figura 4.16: Segmento III de la IPN #3

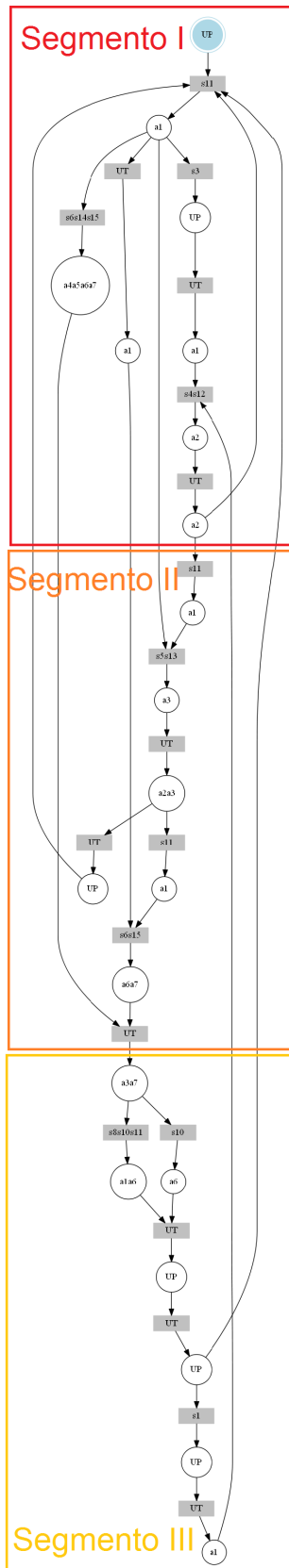


Figura 4.17: IPN #4 asociada al SEC con 15 sensores - Operación CAC

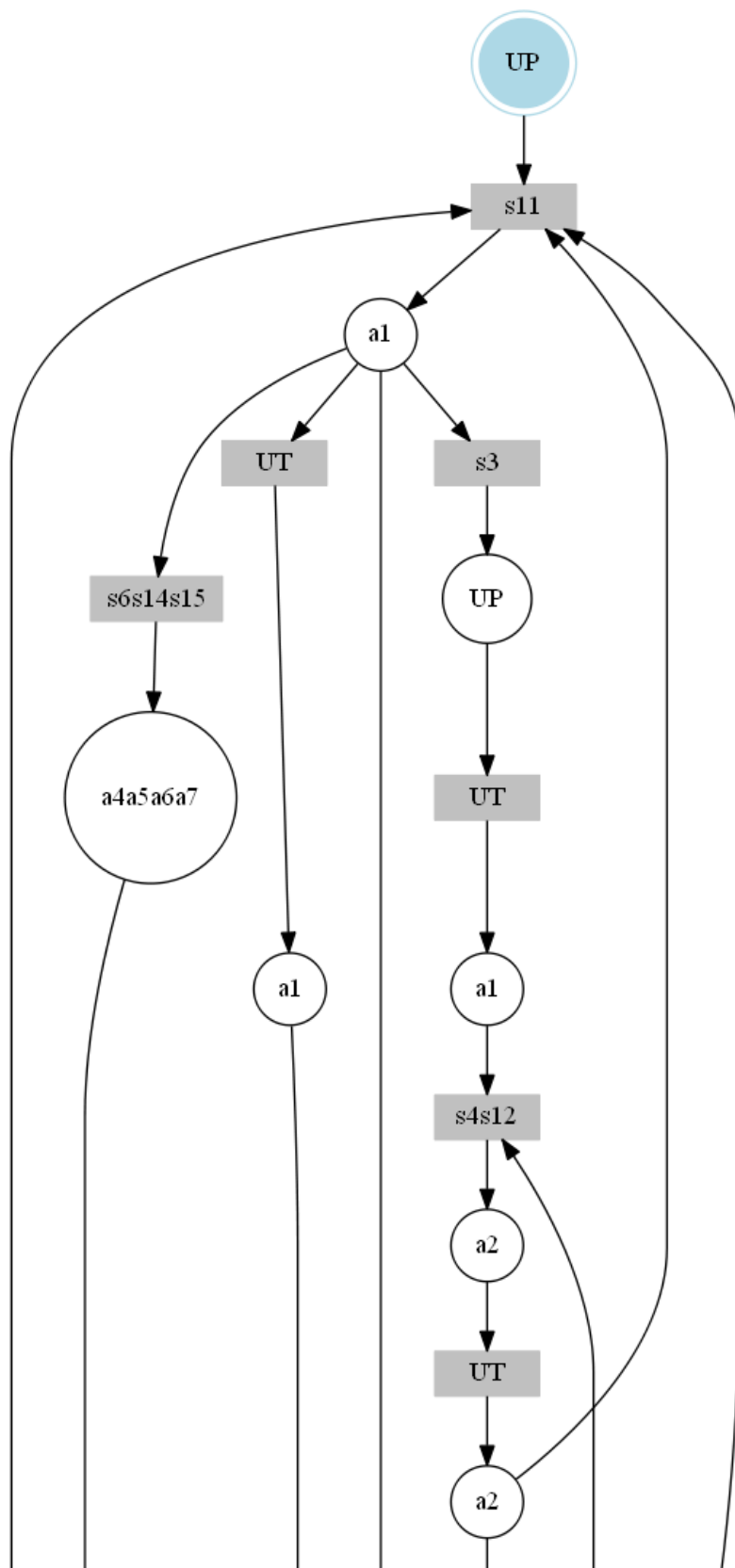


Figura 4.18: Segmento I de la IPN #4

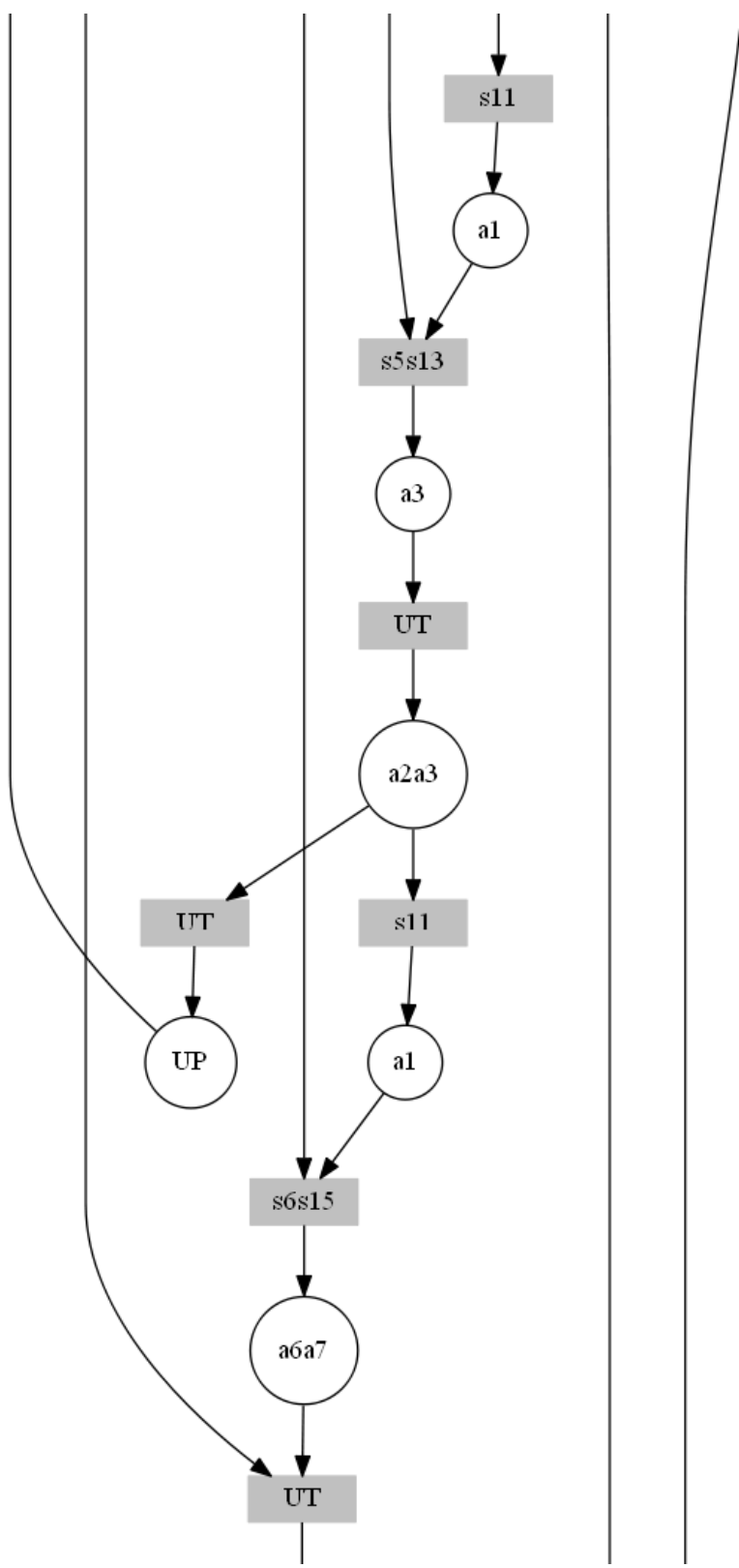


Figura 4.19: Segmento II de la IPN #4

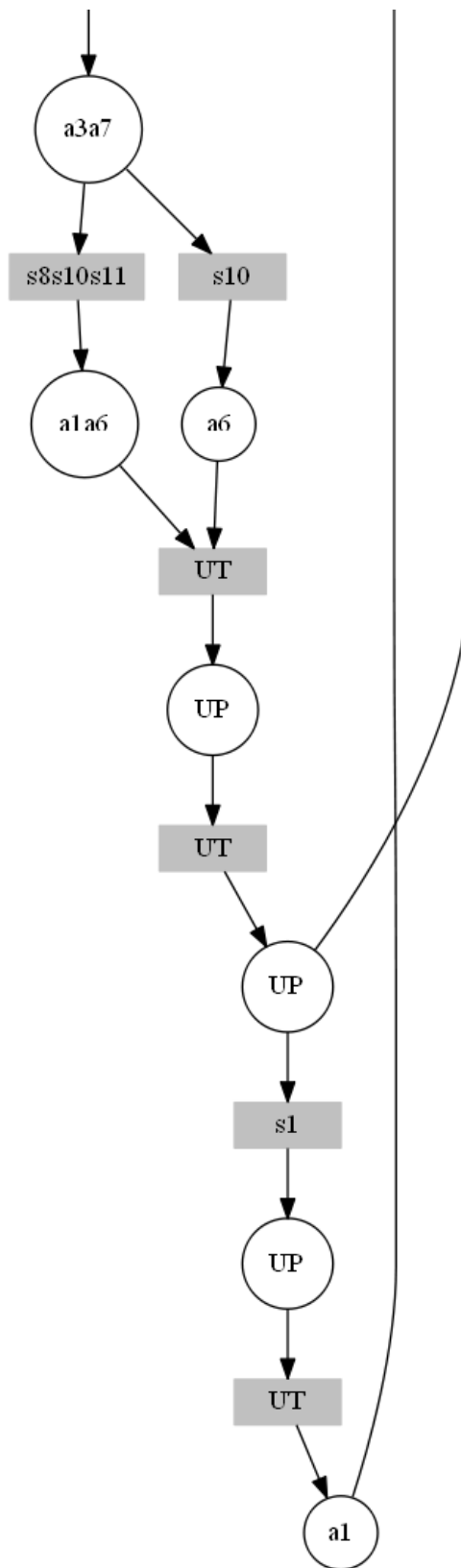


Figura 4.20: Segmento III de la IPN #4

Para observar toda la operación del sistema se configuró el SEC con 3 ciclos de operación A-C-B obteniendo una IPN de 30 lugares y 29 transiciones (Figura 4.21, 4.22, 4.23, 4.24), pero el sistema no era evento-detectable con 15 sensores, el método indicó que se necesitaba adicionar un sensor asociado a BR2 (Figura 4.25, 4.26, 4.27, 4.28).

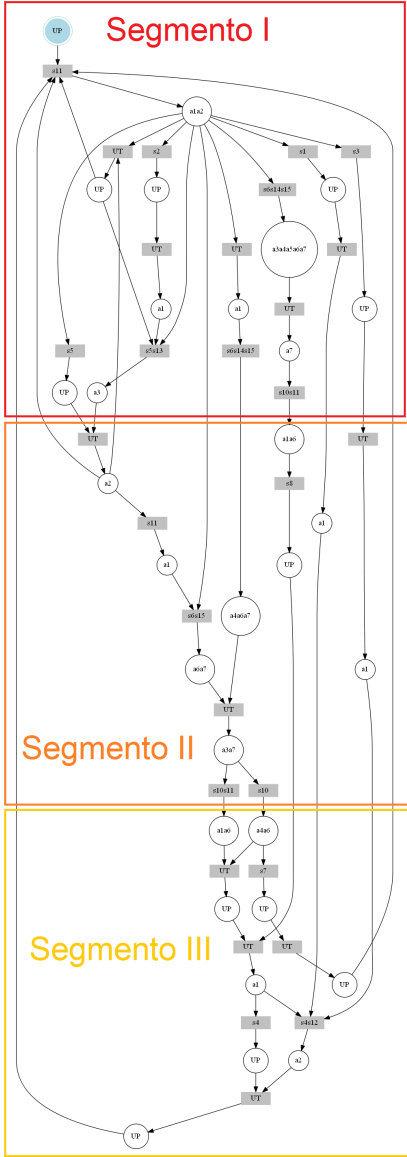


Figura 4.21: IPN #5 asociada al SEC con 15 sensores - Operación ACB

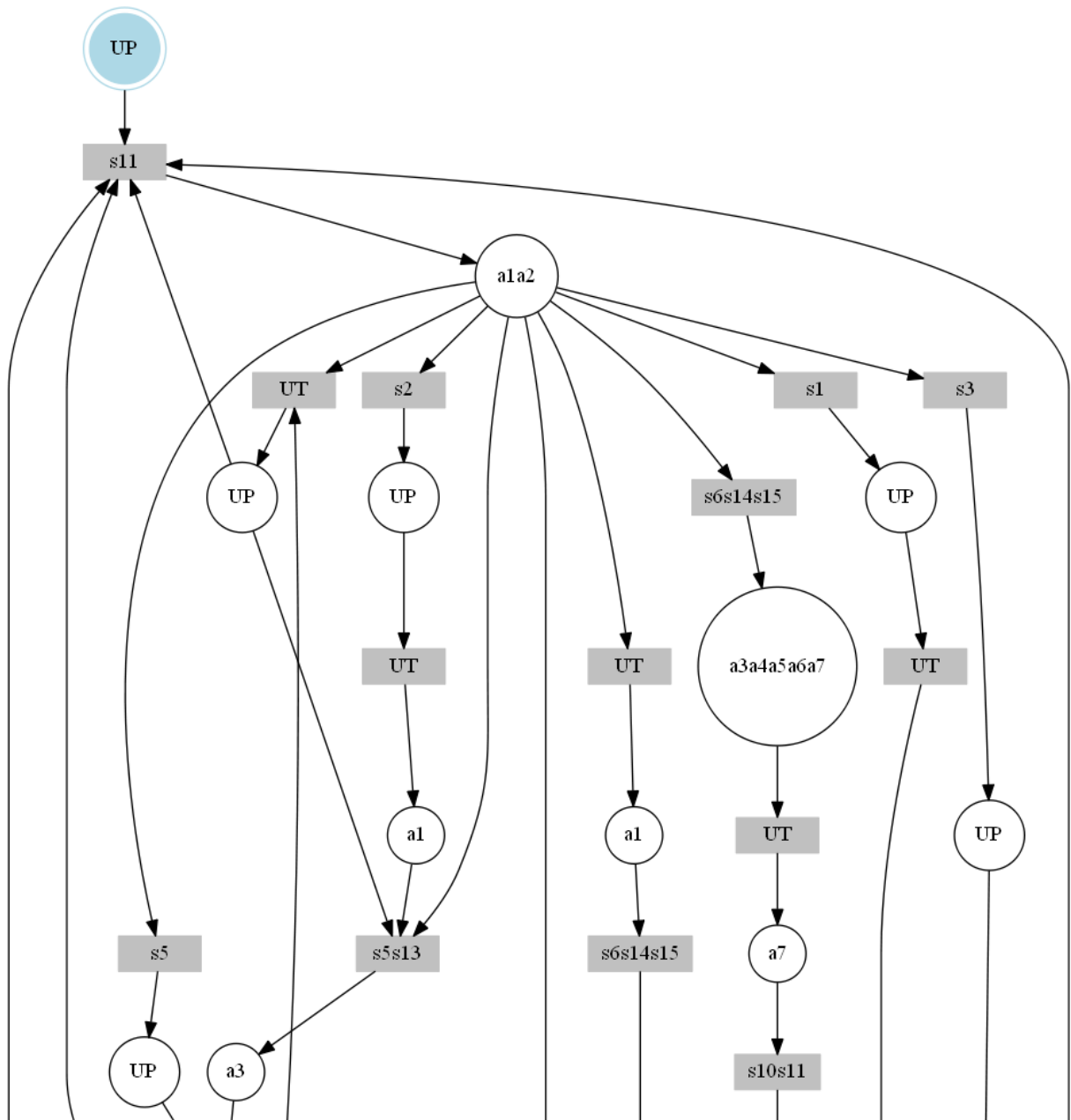


Figura 4.22: Segmento I de la IPN #5

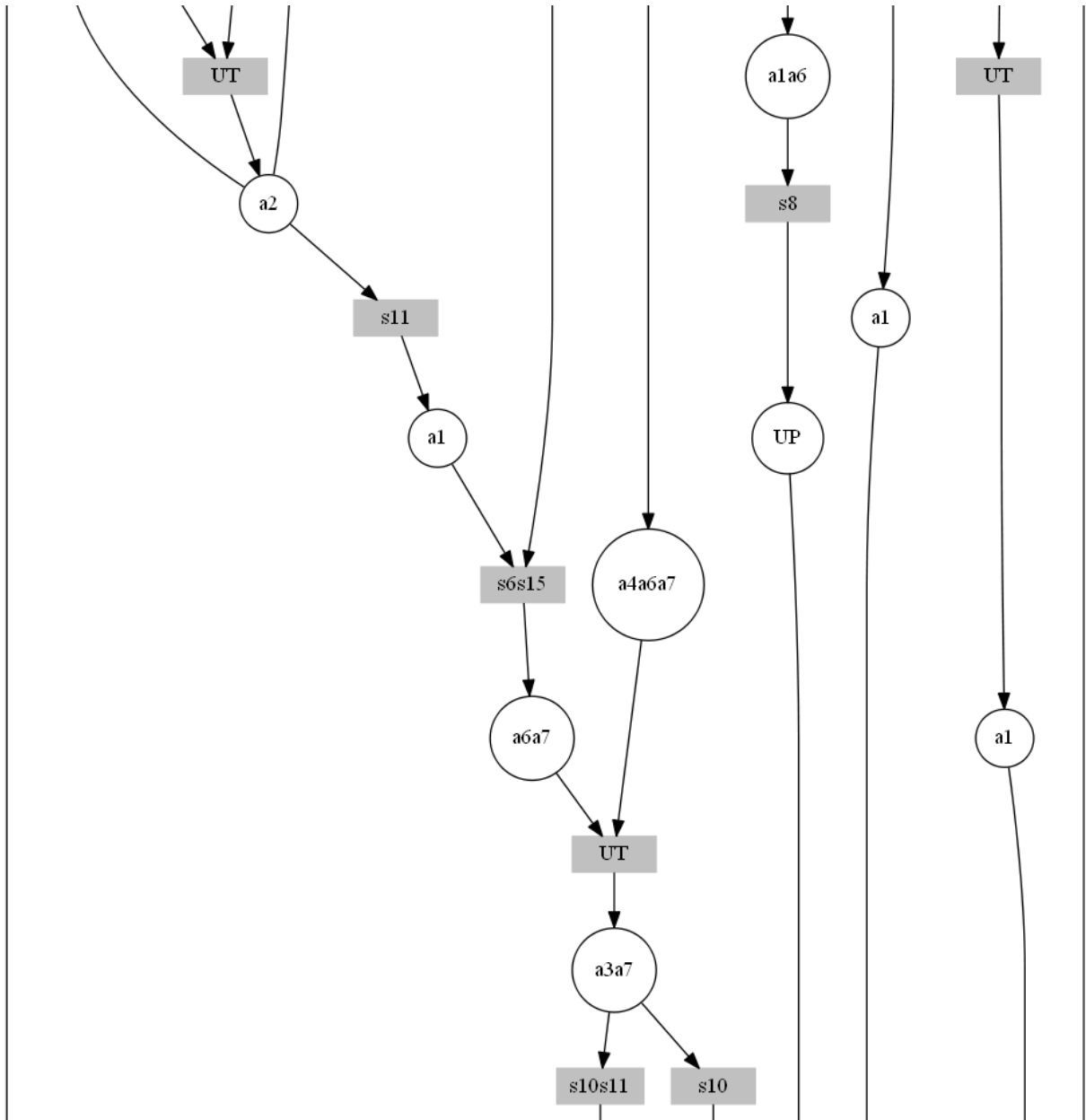


Figura 4.23: Segmento II de la IPN #5

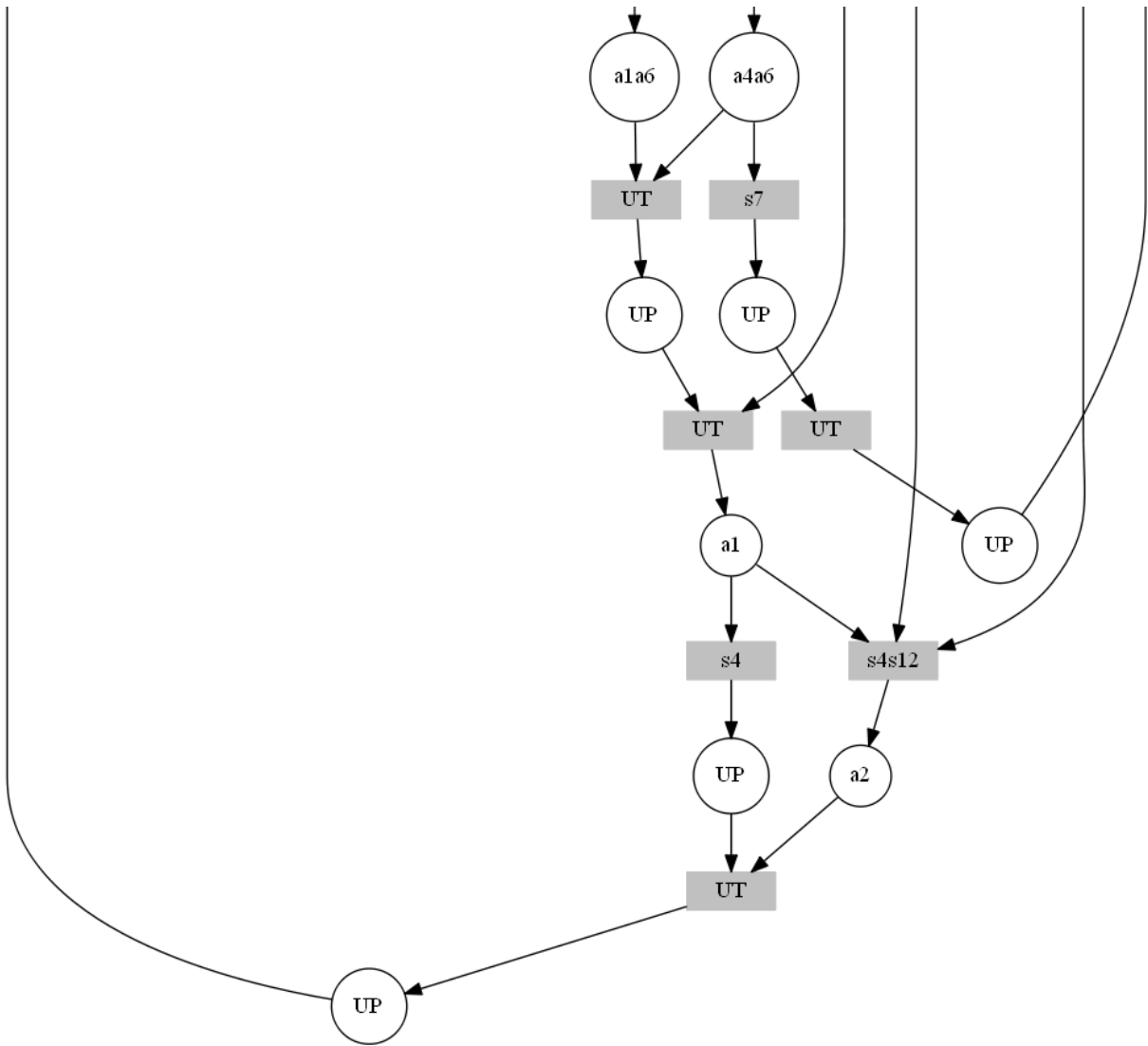


Figura 4.24: Segmento III de la IPN #5

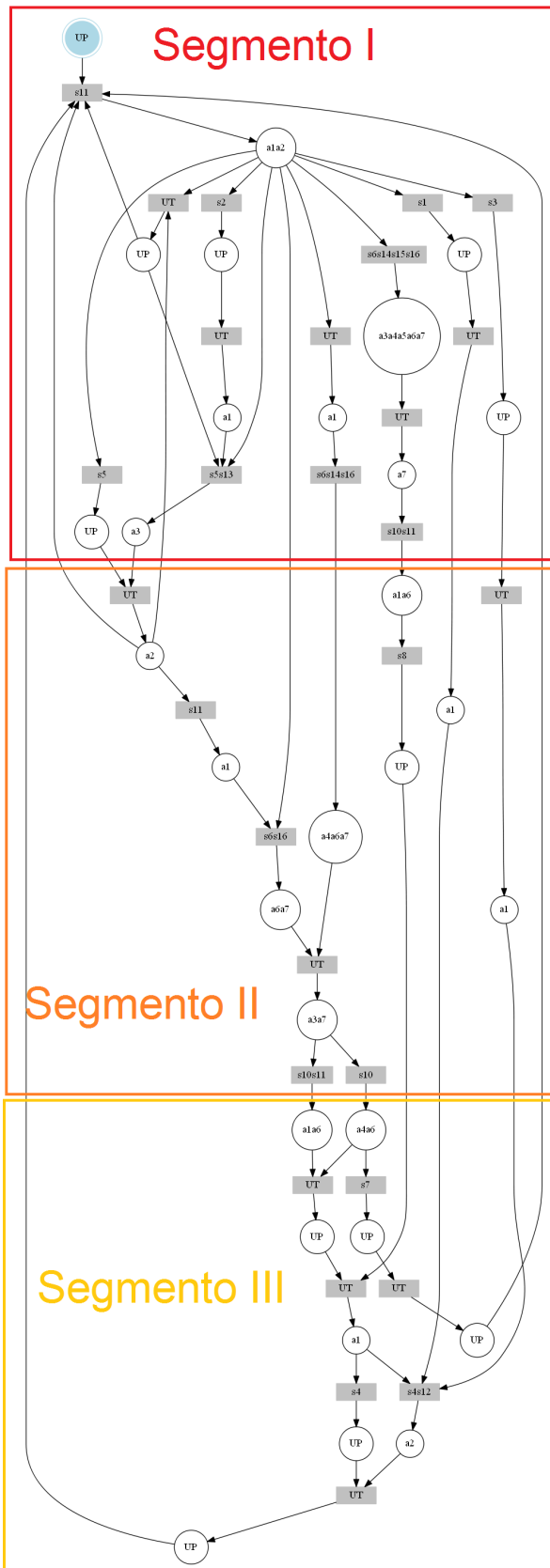


Figura 4.25: IPN #6 asociada al SEC con 16 sensores - Operación ACB
66

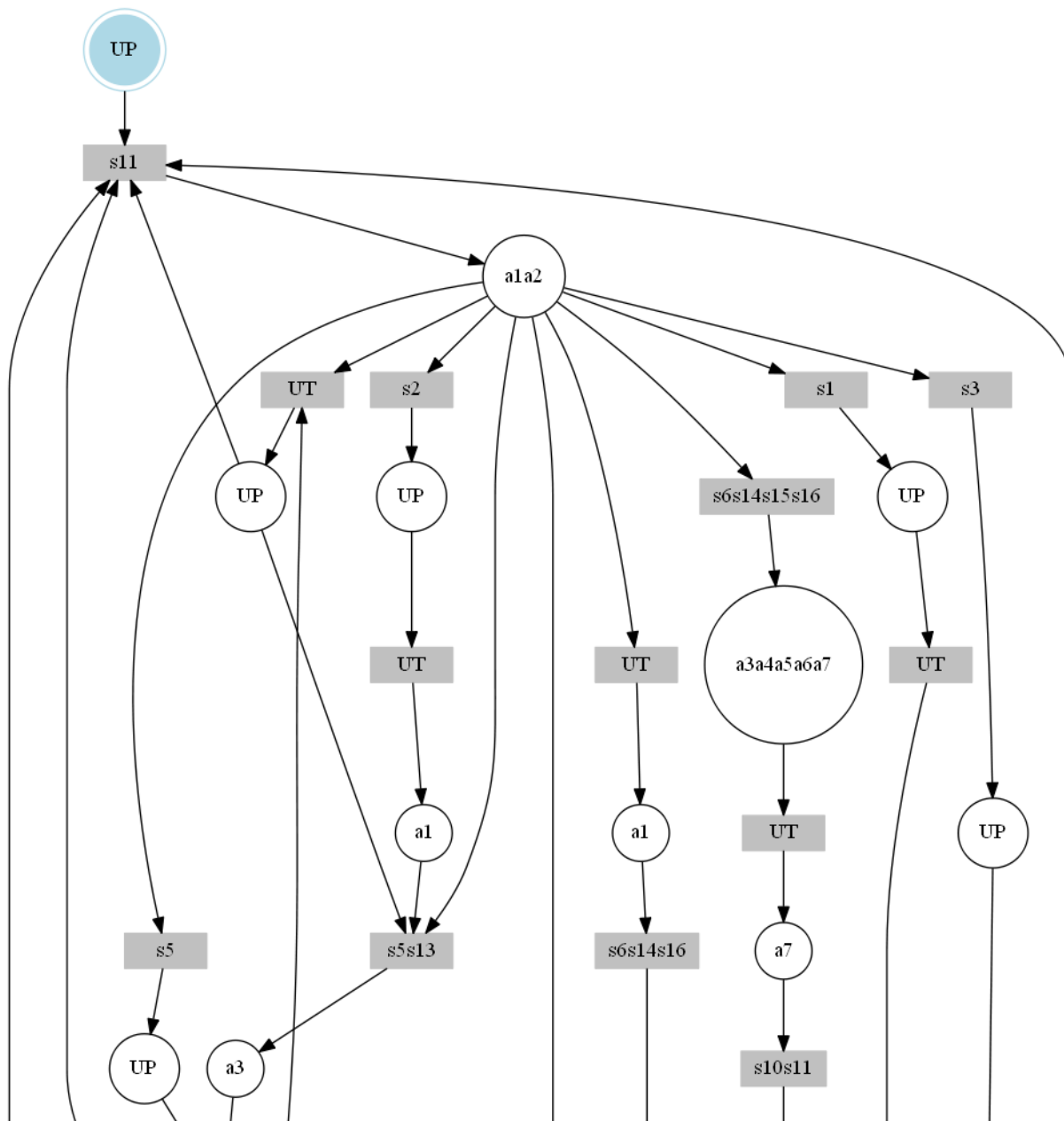


Figura 4.26: Segmento I de la IPN #6

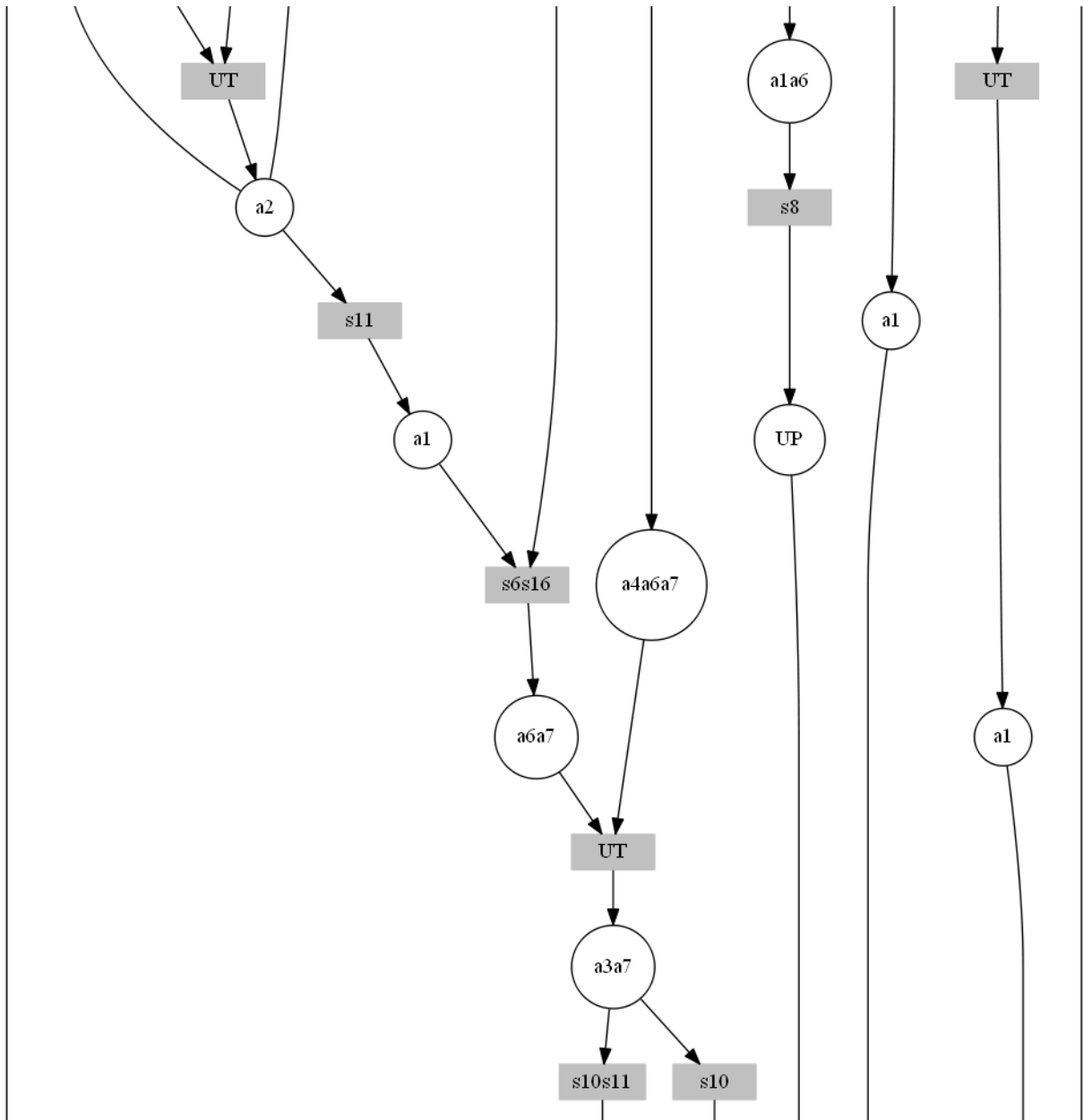


Figura 4.27: Segmento II de la IPN #6

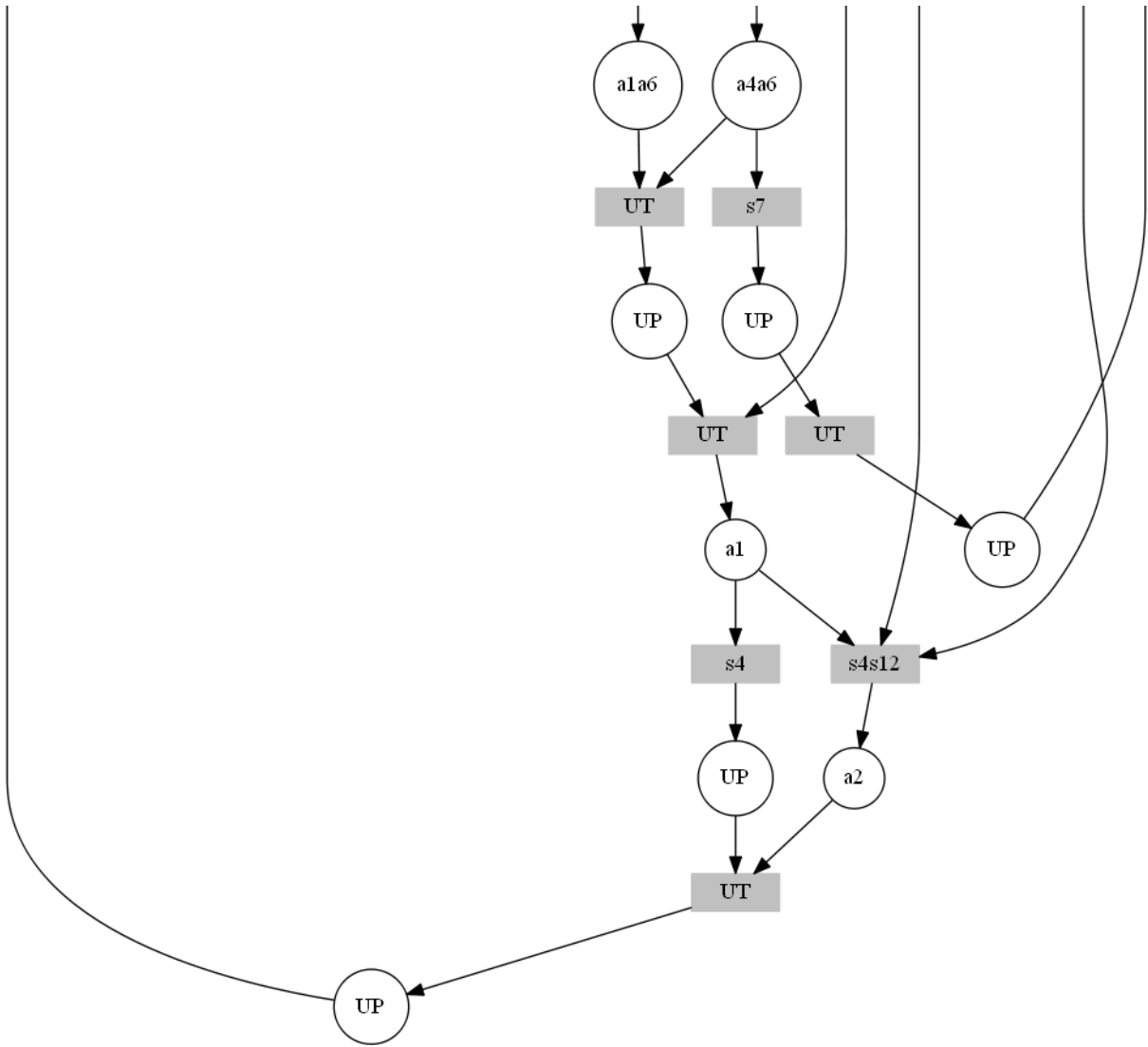


Figura 4.28: Segmento III de la IPN #6

Como últimas 2 pruebas de diagnosticabilidad sobre el SEC, se configuró 7 ciclos de operación A-C-B-B-C-C-B para observar los posibles crecimientos de la IPN, junto con los 16 sensores pre-establecidos (Figura 4.29, 4.30, 4.31, 4.32) y los 17 sensores de límite (Figura 4.33, 4.34, 4.35, 4.36)(sensores en planta y sensores asociados a señales de actuador), en ambos caso se obtuvo un sistema diagnosticable junto con una IPN de 31 lugares y 30 transiciones.

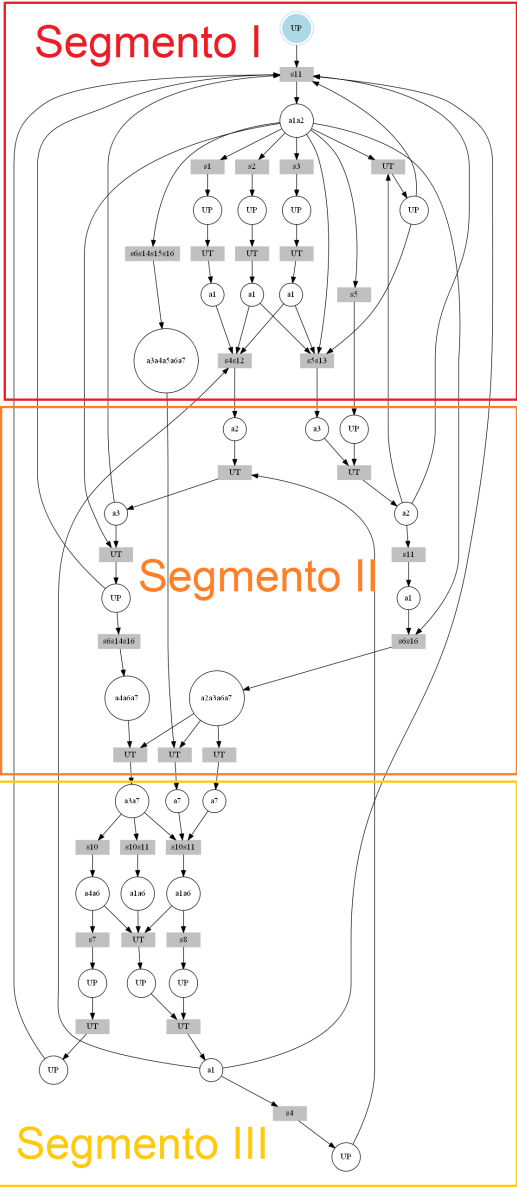


Figura 4.29: IPN #7 asociada al SEC con 16 sensores - Operación ACBBCCB

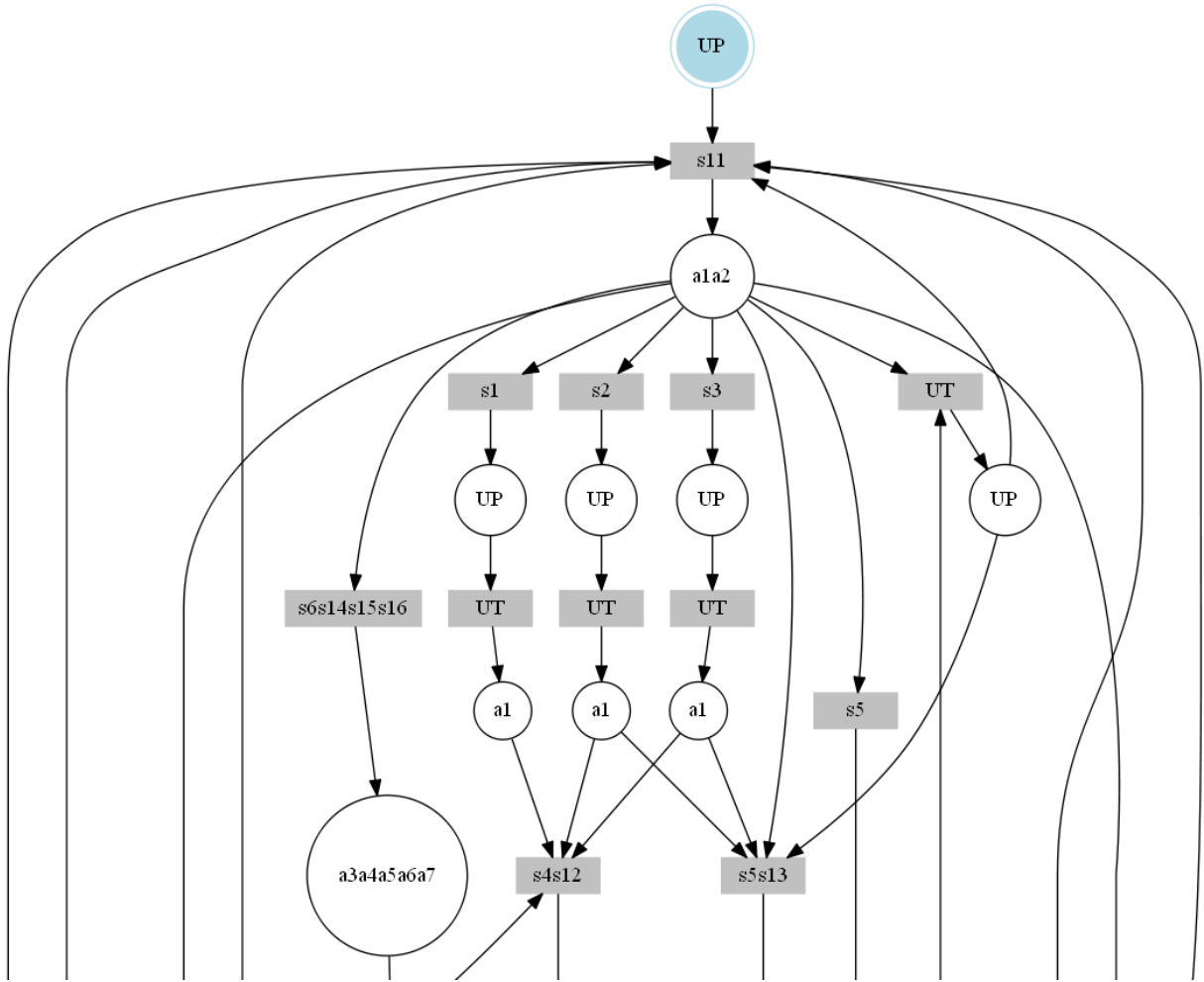


Figura 4.30: Segmento I de la IPN #7

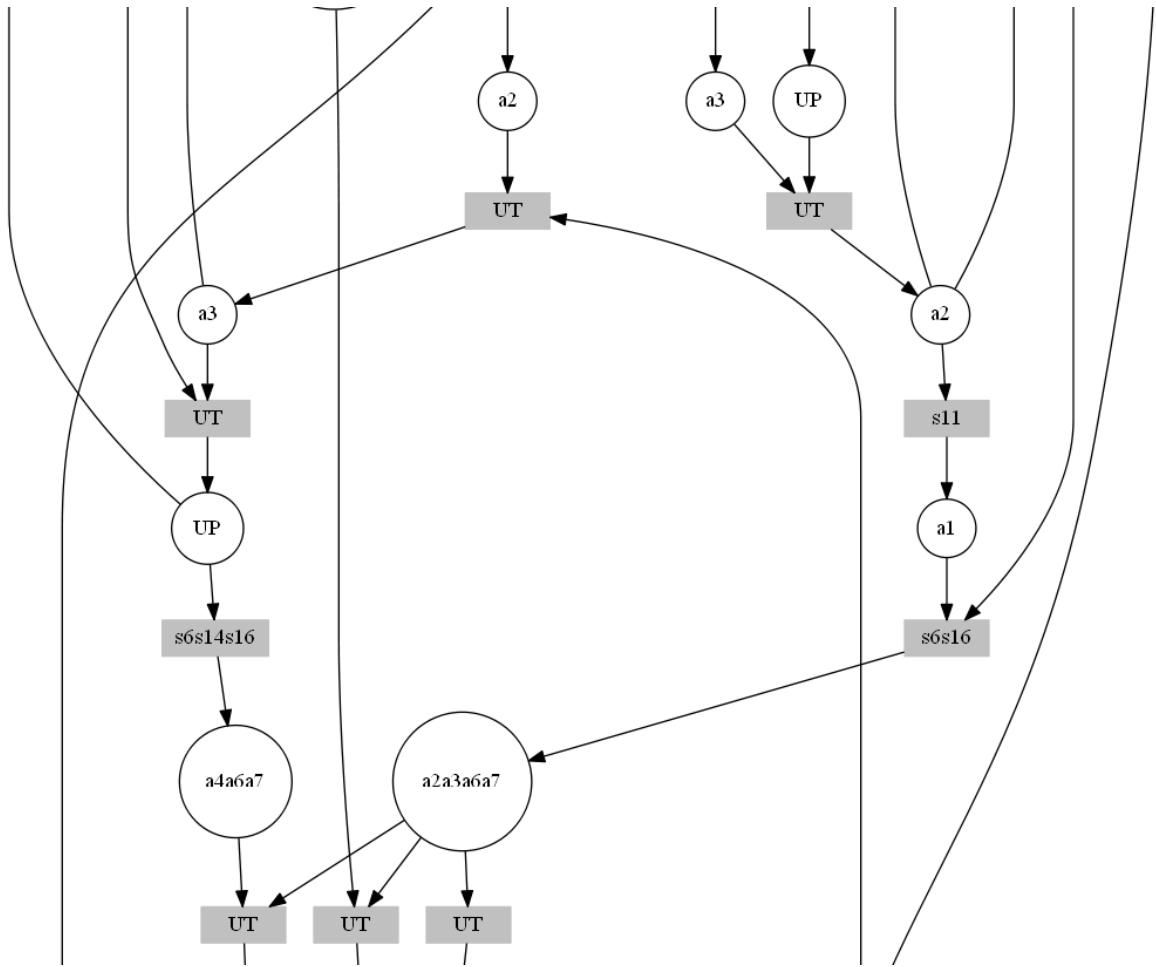


Figura 4.31: Segmento II de la IPN #7

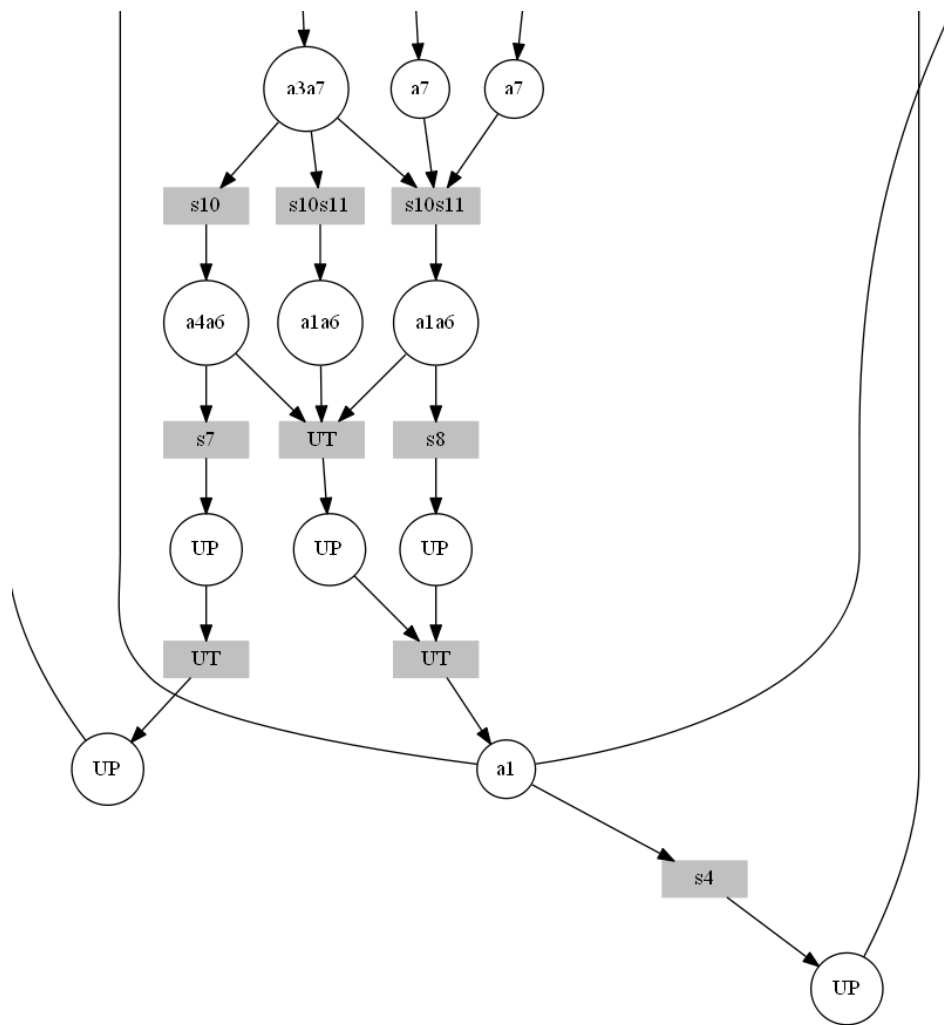


Figura 4.32: Segmento III de la IPN #7

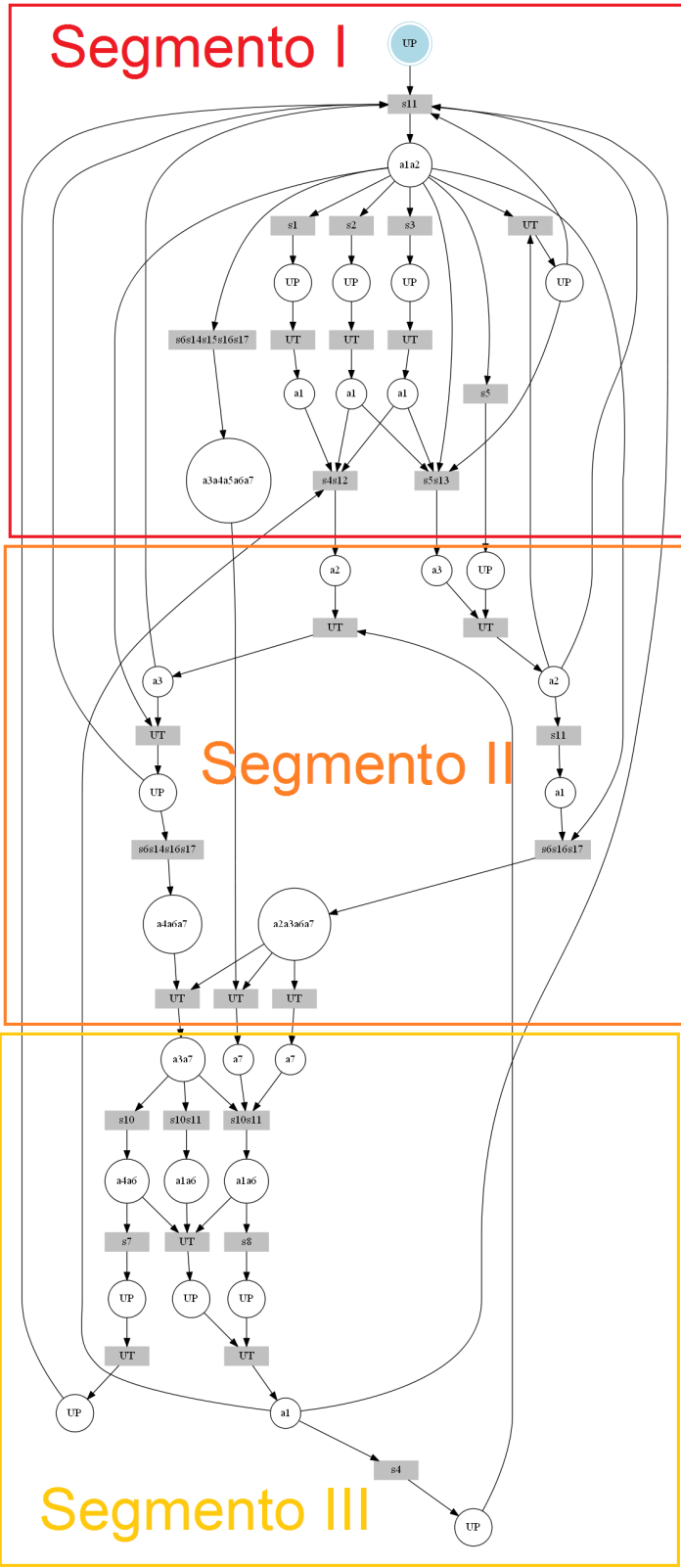


Figura 4.33: IPN #8 asociada al SEC con 17 sensores - Operación ACBBCCB

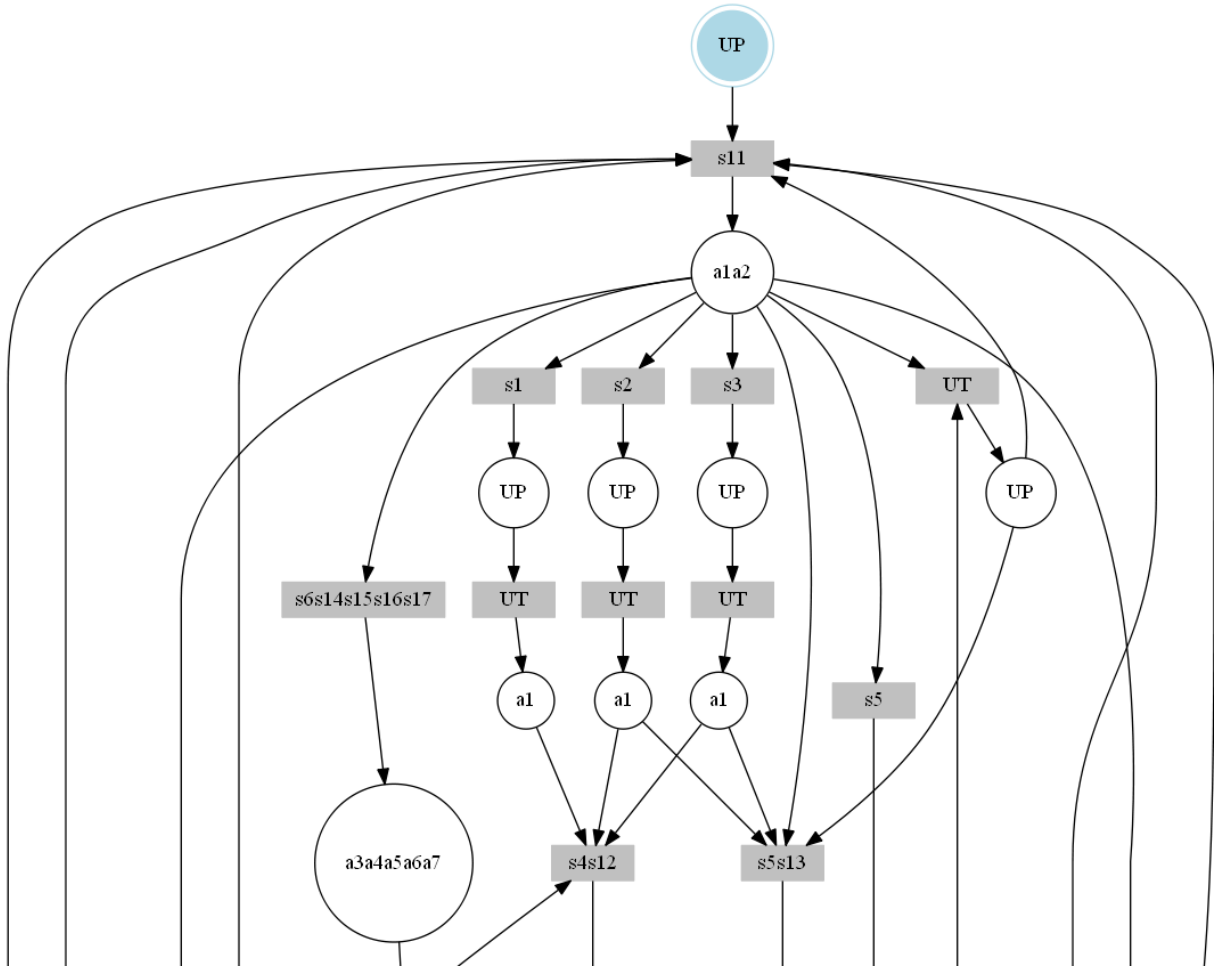


Figura 4.34: Segmento I de la IPN #8

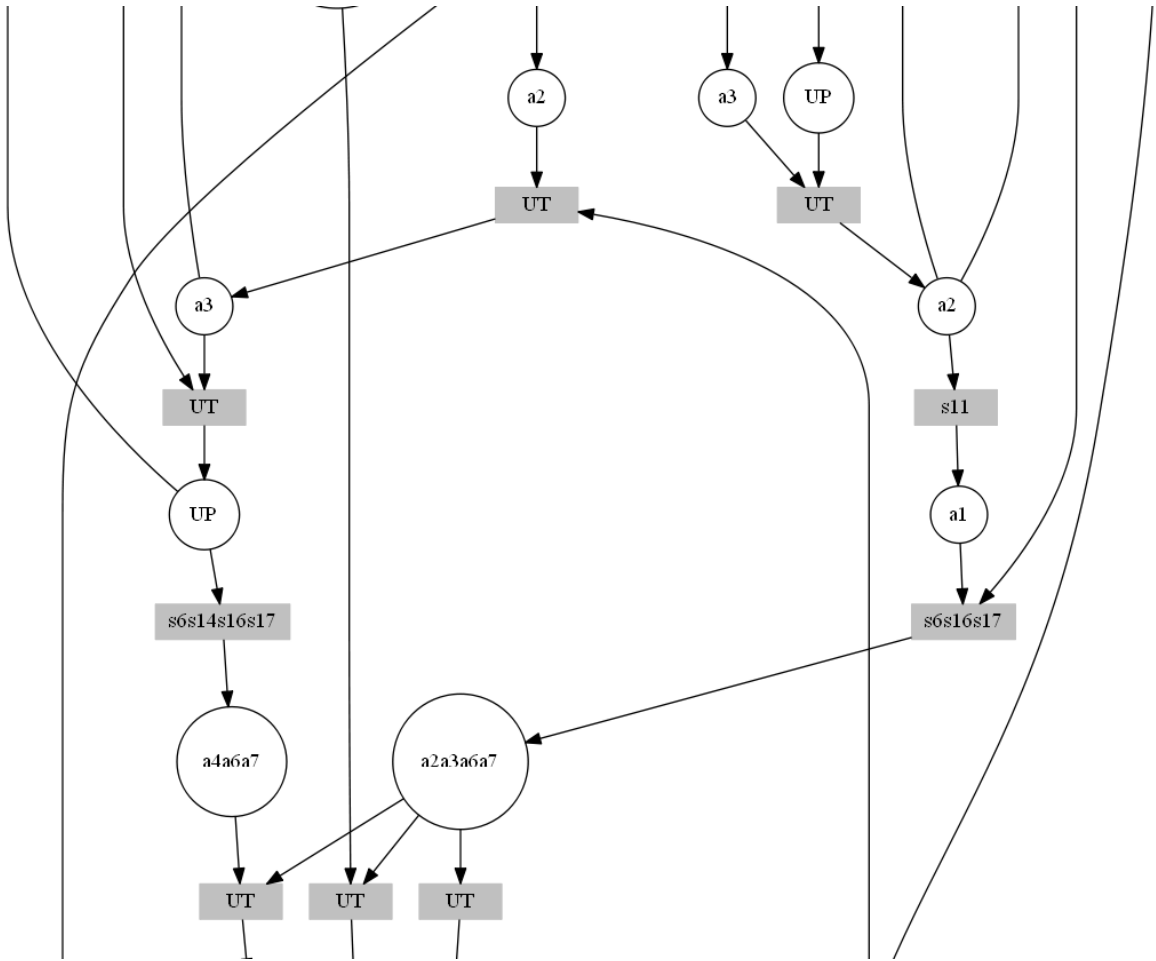


Figura 4.35: Segmento II de la IPN #8

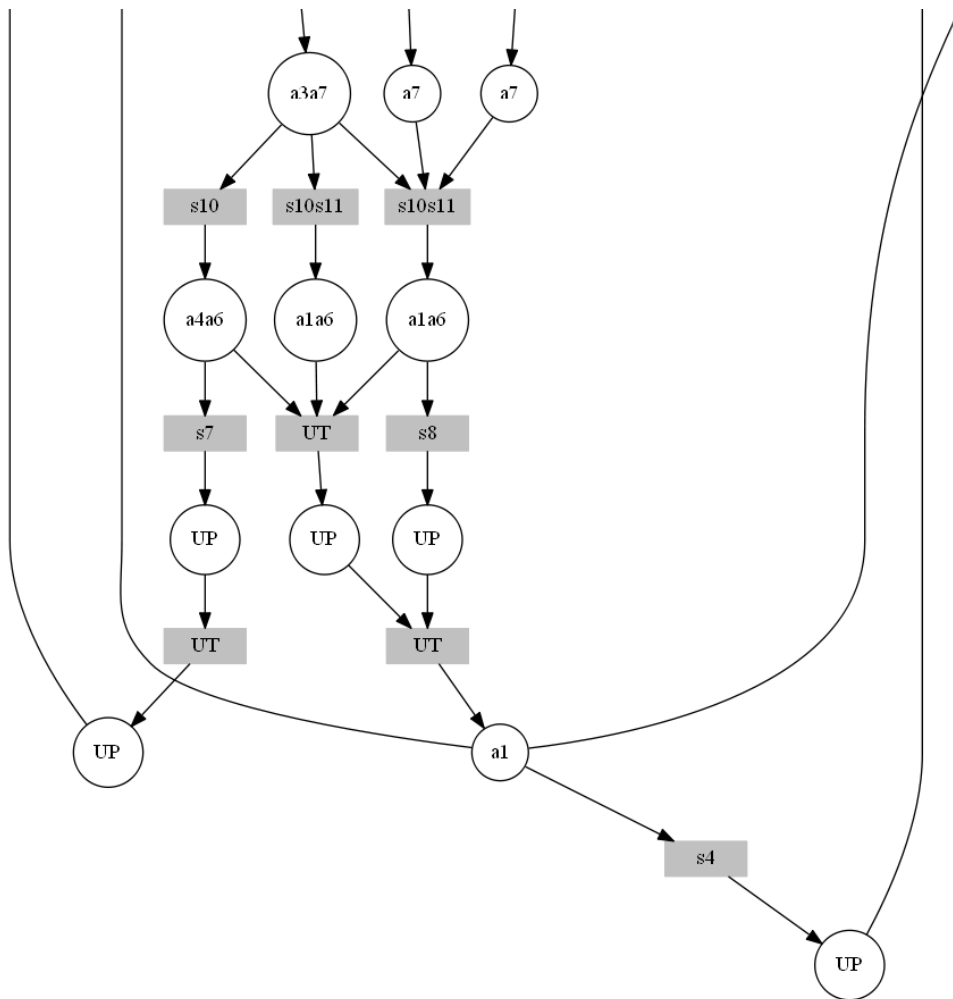


Figura 4.36: Segmento III de la IPN #8

En este ejemplo se evidencia que para cada modo de operación hay un determinado grupo de sensores para que el sistema analizado sea diagnosticable, además la unión de cada uno de los grupos asegura la diagnosticabilidad para todas las posibles operaciones del proceso, es decir, hay conjuntos locales para la diagnosticabilidad de cada uno de los modos de operación presentes, y un conjunto global de diagnosticabilidad del sistema con todos los modos de operación. Esto resalta uno de los principios del método de identificación paramétrica, el asumir que se ha registrado toda la información del sistema, ya que sino se cumple se obtiene un bloque o segmento representativo de operación del sistema, en vez de su representación completa; por lo que todas las modificaciones que se realicen al sistema (en este caso relacionadas a la diagnosticabilidad), serán acciones locales que no logran el cumplimiento de condiciones globales. En este caso se observó todos los modos de operación del sistema de forma consecutiva, lo que permite ver el crecimiento del conjunto de lugares y transiciones no solo por la ejecución de un modo de operación u otro, sino por la interconexión que genera el método de identificación empleado, estableciendo cierto número de lugares y transiciones no observables para unir todos los bloques representativos de funcionamiento entre sí, es decir, para ir desde la finalización de un ciclo de operación al inicio de otro. Con lo que se obtiene una representación general del sistema, permitiendo que el cumplimiento de las condiciones de diagnosticabilidad en este, aseguren que el sistema es diagnosticable de forma global.

Para aclarar, los resultados obtenidos no indican la existencia de posibles conjuntos diferentes entre sí para lograr un sistema diagnosticable, ya que el primer conjunto de sensores que cumplen las condiciones de diagnosticabilidad son todos los posibles sensores que se pueden asociar a la planta, el segundo grupo de sensores es un subconjunto de todos los posibles sensores; por lo que el primer conjunto contiene al segundo. Por lo tanto todos los conjuntos de sensores que aseguran la diagnosticabilidad global de un sistema, en su intersección deben tener un número reducido o conjunto mínimo de sensores con los cuales se alcanza un sistema diagnosticable; significando que después de contener el conjunto mínimo de sensores cualquier posible agrupación de estos cumple las condiciones de diagnosticabilidad.

En la investigación con el método de ubicación adecuada de sensores se obtiene un conjunto de sensores para asegurar la diagnosticabilidad, sin poder afirmar que no hay otra posible agrupación de sensores. Para probar la existencia de un conjunto mínimo de sensores para alcanzar un sistema diagnosticable y que esta agrupación es única dentro del conjunto global de sensores; en las pruebas de verificación de diagnosticabilidad realizadas en este capítulo y las presentadas en el anexo C, después de alcanzar un sistema diagnosticable,

se realizaba la reducción en un sensor, brindado por el método, de forma aleatoria en todas las pruebas el sistema era no diagnosticable; esto afirma la existencia de un número mínimo de sensores. Como una segunda prueba, en las plantas simuladas posterior a la obtención del conjunto de sensores brindado por el método, se mantenía la cantidad de sensores fija y se formaban otros 10 posibles grupos de sensores aleatorios (exceptuando en la planta presentada en la sección 4.1, donde se ejecuto la prueba con los 17 posibles grupos de 16 sensores), en ninguna de las pruebas de agrupación aleatoria el sistema evaluado era diagnosticable; con este último resultado es posible sustentar que hay un conjunto único de sensores para cumplir las condiciones de diagnosticabilidad, de igual forma con el método de ubicación adecuada de sensores propuesto se obtiene el único conjunto mínimo de sensores para que los sistemas en lazo cerrado analizados fueran diagnosticables.

4.2. Análisis

En los resultados anteriormente presentados y los expuestos en el anexo de simulaciones, se observan comportamientos a resaltar:

1. En los sistemas de lazo cerrado (Planta-Controlador), la existencia de diagnosticabilidad está acotada entre un número límite superior de señales de salida (sensores iniciales y sensor asociado por cada actuador), y un límite inferior; la cota superior asegura que todo lugar o transición sea medible evitando que algún fallo o evento no sea perceptible, la cota inferior brinda la característica que todos los cambios en estados o eventos (observables o no observables) sean diferenciables y distinguibles; después de superar estos límites la diagnosticabilidad se mantiene, exceptuando los casos de redundancia de información que pueden indicar estados innecesarios o ubicación de señales de operación intermedia no deseada. El límite inferior de señales de salida se alcanzó, por medio del método de ubicación de sensores desarrollado en esta investigación; el cual fue corroborado manteniendo el número de sensores pero variando que señales se toman para el proceso de verificación de diagnosticabilidad, de dichas pruebas se observó que el método determina la cantidad mínima de señales excluyentes (número de sensores adecuados) es decir, para que un sistema sea diagnosticable se debe tener un número específico de señales, y estas señales deben provenir de determinados puntos de operación del sistema.

2. Las IPN obtenidas incrementan lugares y transiciones dependiendo del número de señales salida que se tengan de la planta hasta un punto umbral. Esto es debido a que entre mayor información se tenga de la planta las combinaciones de los vectores I/O aumentan (por cada señal adicional se multiplica por 2 las combinaciones, con un total de m -señales las posibles combinaciones serían 2^m), y por ende aumentaría la matriz de observación y los vectores de eventos; pero el punto umbral de crecimiento se presenta porque, de las combinaciones de vectores I/O, no todas son posibles por la operación propia del sistema de lazo cerrado de control; lo que determina la existencia de un número máximo de lugares y transiciones en un sistema diagnosticable, sin redundancia de sensores o información innecesaria.

3. El etiquetado de las transiciones en las IPN varía en función de los sensores en la planta, aun después de que el sistema es diagnosticable. Este resultado está reportado en la literatura [12], donde se plantea que para que en un sistema no diagnosticable con función de etiquetado L_{init} exista diagnosticabilidad, se construye una nueva función de etiquetado L_{new} , con base a L_{init} y un etiquetado adicional L_{total} tal que: $L_{new} = L_{init} \cup L_{total}$.

La función de etiquetado L_{new} , representa una variación sobre la adquisición de las señales del sistema con las cuales se está obteniendo información del mismo; la verificación de la diagnosticabilidad en este caso se plantea como un problema de programación lineal entera, en cada ciclo de prueba se observa los posibles eventos detectables con las señales de adquisición establecida, se revisa si los posibles eventos detectados están relacionados al comportamiento de fallo establecido, y se genera las modificaciones pertinentes a la función de etiquetado L_{new} ; obteniendo un número óptimo de señales en este proceso. Para la presente investigación, se empieza por el proceso de variación de señales de salida y su consecuente etiquetado en la IPN asociada al sistema, la verificación está descrita en la sección 2.1

4. Con lo anteriormente descrito, bajo el método de ubicación de sensores desarrollado se obtiene el conjunto mínimo de señales de salida del sistema bajo el cual es diagnosticable; la ubicación adecuada de sensores para cualquier proceso industrial o de manufactura sobre el cual se quiere asegurar una característica como observabilidad o diagnosticabilidad, se busca el menor número necesario de sensores (bajo costo) que aseguren el mejor desempeño posible (alta eficiencia).

Con este criterio es posible afirmar que se ha obtenido en cada una de las plantas simuladas el número óptimo de sensores para diagnosticabilidad, por la existencia de un conjunto único y exclusivo que asegura esta característica. De la misma forma, el cumplir con la observabilidad y evento-detectabilidad sobre todo el modelo del sistema es de mayor peso que una condición suficiente y necesaria, ya que aunque el modelo es libre de fallo puede trabajar en paralelo con un modelo de fallo, ambas características generales de diagnosticabilidad se deben asegurar sólo en los tramos de la IPN asociados a comportamiento de fallo [21]; es decir, se excluye la redundancia de sensores en sistemas de lazo cerrado para que sean diagnosticables.

Capítulo 5

Conclusiones y Trabajos Futuros

5.1. Conclusiones

Con el desarrollo obtenido a lo largo de la investigación y sus diferentes resultados, se concluye que:

- ✓ Se adecuó una prueba de diagnosticabilidad en función de las señales de salida de sistemas de eventos discretos, la cual sirvió como herramienta de verificación de diagnosticabilidad para sistemas de manufactura. La prueba evalúa los efectos de una determinada cantidad de señales de salida sobre las transiciones o eventos (evento-detectabilidad, existencia de un etiquetado sobre la IPN para diferenciar en qué lugar se encuentra el sistema) y sobre los lugares o estados (observabilidad, certeza de que el conjunto de lugares observables y no observables, tienen una secuencia fija para llegar a ellos); asegurando que todos los componentes de estos conjuntos de lugares y transiciones de las IPN sean distinguibles y diferenciables cada uno en su respectivo conjunto; para ello se debe contar con la matriz de inspección sobre lugares y de inspección sobre transiciones, en cada matriz se evalúa: Distinguibilidad, verificando la no existencia de columnas de cero ya que en el caso contrario, no se tendrían los sensores adecuados para poder indicar la existencia del lugar o transición asociados a dichas columnas de cero; Diferenciabilidad, corroborando que todas las columnas de la matriz son diferentes, dado el caso contrario indicaría que no hay sensores suficientes para indicar que se está en un lugar o se activó una transición asociados a las columnas que se repiten.

- ✓ Se implementó el método de identificación de caja negra para IPN por medio de señales de entrada-salida de DES, la prueba de diagnosticabilidad planteada y la representación gráfica de la IPN, mediante un aplicativo software modular elaborado bajo la metodología de desarrollo de sistemas dinámicos (DSDM) usando como herramientas de desarrollo Labview (Identificación) y MATLAB (Prueba de diagnosticabilidad y Representación gráfica); con el cual se determinaron: Todos los involucrados en el proyecto, junto con las responsabilidades y beneficios de cada rol; El alcance del proyecto, con sus ciclos de desarrollo y recursos utilizados; y Registro de fechas de cada etapa del proyecto para evaluar el progreso en cada una de ellas. Al aplicar la DSDM se logró un proceso de creación software eficiente, reducción en la cantidad de posibles inconvenientes, aumento de los ciclos de depuración, documentación detallada de la evolución y uso del aplicativo software; y una aceptación promedio del usuario final de 9,65 en la escala de 0-10.

- ✓ Se determinó una dependencia entre la existencia de diagnosticabilidad y la cantidad de señales de salida de planta de sistemas en lazo cerrado de control, por medio del procesamiento del aplicativo software a los datos recopilados en las simulaciones de procesos de manufactura, efectuando variación directa de la cantidad de sensores. Esta dependencia se determinó bajo un umbral máximo y mínimo, el máximo se presenta cuando en el sistema analizado todo el proceso es medible es decir, se adquieren las señales provenientes de la planta y las señales de los actuadores, este umbral se determina para evitar redundancias en las señales adquiridas e información innecesaria para el proceso; el umbral mínimo se halló en todos los sistemas bajo un método de ubicación de sensores desarrollado, cabe aclarar que el umbral mínimo no solo indica una cantidad determinada de sensores sino un conjunto exclusivo de estos, por lo que la existencia de diagnosticabilidad no sólo esta sujeta a cuantas señales de salida se estén adquiriendo, sino también a qué señales se están adquiriendo.

- ✓ Se formuló una relación entre evento-detectabilidad y observabilidad, a partir de desarrollos recientes reportados en la literatura se determinó que existe una relación de correspondencia entre estas dos características contrario a la unicidad que generalmente se asume, por lo que para las dos hay propiedades similares o iguales pero la existencia de una no es una doble implicación sobre la existencia de la otra. La formulación se construyó a partir del componente matemático de las IPN, determinando dos espacios vectoriales en Z^2 , que representan la relación de sensores-transiciones y la relación sensores-lugares; estos dos espacios interaccionan por matriz de incidencia o su traspuesta que actúan como una transformación lineal de un espacio a otro.

- ✓ Se construyó un método de ubicación adecuada de sensores para diagnosticabilidad, obtenido a partir de las pruebas de diagnosticabilidad, este método consiste en la verificar si el sistema identificado es diagnosticable, caso contrario se evalúa la evento-detectabilidad con la matriz de inspección sobre transiciones, si no se cumple por distinguibilidad se ve las columnas donde son cero y se pasa a la matriz de transiciones - señales I/O, analizando esas mismas columnas y se ve qué actuadores están activos a los cuales será necesario asociarles un sensor, si la evento-detectabilidad no se cumple por diferenciabilidad se registran las columnas que se repiten y se comparan en la matriz transiciones-señales I/O, las cuales tendrán diferencia en los actuadores a cada uno de los actuadores diferentes se les asociará un sensor; se procede a la observabilidad efectuando un procedimiento similar para la distinguibilidad y diferenciabilidad en la matriz de inspección de lugares, solo que las columnas registradas se evalúan en la matriz resultante de la multiplicación por la izquierda de la matriz incidencia traspuesta con la matriz de transiciones-señales I/O como filas. Después de este proceso se obtiene un conjunto exclusivo de sensores, en el cual la eliminación de uno de ellos o cambio por un sensor diferente, el sistema no es diagnosticable; ya que en todos los procesos industriales se busca tener la cantidad mínima de sensores que brinden el mayor desempeño (bajo costo y alta eficiencia), es posible afirmar que el método de ubicación adecuada de sensores, da el número óptimo de señales de salida de planta para diagnosticabilidad.

5.2. Trabajos Futuros

Desde el alcance logrado en este trabajo de grado, es posible seguir desarrollando investigación relacionadas tales como:

- ✓ Efectuar pruebas online con el módulo software de identificación sobre las plantas del Laboratorio de Control de Procesos, siendo necesario generar los códigos de intercomunicación entre los controladores de cada planta y Labview.
- ✓ Desarrollar un módulo software de diagnóstico en Labview, para expandir el aplicativo software existente.
- ✓ Construir plantas discretas de manufactura con Arduino, ya que es una herramienta académica en continuo crecimiento, y su conexión a Labview y MATLAB permite expandir el alcance de uso del aplicativo software realizado.

- ✓ Implementar otros métodos de identificación para IPN en aplicativos software, con el fin de comparar con la identificación ya desarrollada, y enriquecer el proceso de aprendizaje de los estudiantes del programa de Ingeniería en Automática Industrial con herramientas de apoyo por sistematización y visualización de procesos.

Anexo A

Relación matemática entre Observabilidad y Evento-Detectabilidad

La Observabilidad y Evento-Detectabilidad son características atribuibles a las IPN, como tal se generan variaciones en la representación gráfica de las mismas o en su lenguaje, pero al pertenecer a las IPN adquieren un formalismo matemático implícito desde conceptualización y es con este mismo con el cual se ha logrado relacionar estas dos propiedades. Antes de plantear su relación se utilizarán algunas definiciones del álgebra lineal [53]:

Definición A.1 Espacio Vectorial: Sea V un conjunto no vacío en el cual se han definido dos operaciones, llamadas Suma y Producto por escalar, dados los elementos u, v y w de V y los escalares α y β de \mathbb{R} , la suma de u y v se denota como $u + v$ y el producto escalar de α por u como αu . Se dice que V es un espacio vectorial si se cumplen las siguientes propiedades:

1. Propiedad clausurativa para la suma: $u + v \in V$.
2. Propiedad conmutativa para la suma: $u + v = v + u$.
3. Propiedad asociativa para la suma: $(u + v) + w = u + (v + w)$.
4. Propiedad modulativa para la suma: Existe un elemento $\mathbf{0} \in V$: $u + \mathbf{0} = u$, $\forall u \in V$.
5. Existencia del opuesto para la suma: $\forall u \in V, \exists v \in V : u + (v) = \mathbf{0}$.
6. Propiedad clausurativa para el producto por escalar: $\alpha u \in V$.

7. Propiedad distributiva respecto a la suma de vectores: $\alpha(u + v) = \alpha u + \alpha v$.
8. Propiedad distributiva respecto a la suma de escalares: $(\alpha + \beta)u = \alpha u + \beta u$.
9. Propiedad asociativa para el producto por escalares: $\alpha(\beta u) = (\alpha\beta)u$.
10. Propiedad modulativa para el producto por escalar: $1u = u$.

Definición A.2 Subespacio Vectorial: Sea V un espacio vectorial y sea H un subconjunto de V . Si H es espacio vectorial con los mismos escalares y las mismas operaciones que V , decimo que H es un subespacio de V .

Teorema A.1 Caracterización de subespacio: Sea V un espacio vectorial y H un subconjunto no vacío de V . H es un subespacio vectorial de V , si y solo si, los elementos de H satisfacen las propiedades clausurativas para la suma y el producto por escalar (propiedades 1 y 6 de los espacios vectoriales).

Definición A.3 Transformación Lineal: Dados dos espacios vectoriales V y W , se dice que la función $T : V \rightarrow W$, es una transformación lineal de V en w , si y solo si:

1. Propiedad aditiva: $T(v_1 + v_2) = T(v_1) + T(v_2), \forall v_1, v_2 \in V$.
2. Propiedad homogénea: $T(\lambda v_1) = \lambda T(v_1), \forall v_1 \in V, \forall \lambda \in \mathbb{R}$

Definición A.4 Imagen de una transformación lineal: Dada una transformación lineal $T : V \rightarrow W$, se define la imagen de T $Im(T)$, como el conjunto de todos los vectores de w de W para los cuales existe un vector v de V , tal que $T(v) = w$, es decir: $Im(T) = \{w \in W \exists v \in V : T(v) = w\}$.

Definición A.5 Transformación Lineal Inyectiva: Dada una transformación lineal $T : V \rightarrow W$, se define como inyectiva, si y solo si, para cada w de $Im(T)$, existe un único $v \in V$ tal que: $T(v) = w$ (Relación uno a uno entre los dos espacios vectoriales asociados por la transformación lineal).

Definición A.6 Transformación Lineal Inyectiva: Dada una transformación lineal $T : V \rightarrow W$, se define como inyectiva, si y solo si, para cada w de $Im(T)$, existe un único $v \in V$ tal que: $T(v) = w$ (Relación uno a uno entre los componentes de dos espacios vectoriales asociados en la transformación lineal).

Definición A.7 Transformación Lineal Sobreyectiva: Dada una transformación lineal $T : V \rightarrow W$, se define como sobreyectiva, si y solo si, $Im(T) = W$ (Relación uno a uno entre los espacios vectoriales asociados en la transformación lineal).

Definición A.8 Isomorfismo: Dada una transformación lineal $T : V \rightarrow W$ es un isomorfismo, si y solo si, es inyectiva y sobreyectiva.

Definición A.9 Espacios Vectoriales Isomorfos: Dados dos espacios vectoriales V y W y un isomorfismo $T : V \rightarrow W$, se dice que V y W son isomorfos denotado por $V \cong W$

Para el manejo matemático de Observabilidad y Evento-Detectabilidad se establece estas dos características como espacios vectoriales en \mathbb{R}^2 , por lo que se tiene el comportamiento global para modelados continuos (Números reales) y discretos (Números enteros), generando la evaluación de ambas características en eventos discretos relacionados a posibles "subespacios"* correspondientes en \mathbb{Z}^2 Lugares x Sensores(Observabilidad) y Transiciones x Sensores (Evento-Detectabilidad). En investigaciones anteriores [21], se relacionan la matriz de sensores para lugares y para transiciones (elementos de estos subespacios) por medio de la matriz de incidencia y se establecen parámetros de evaluación para asociarlos a los sistemas de donde provienen dichos elementos.

*Los subespacios son reales y no enteros por la existencia de campos escalares en su formulación, pero el reducir los posibles valores de los campos escalares solo a números enteros, permite la existencia de un pseudo-espacio entero.

Anexo B

Guia de Usuario del Aplicativo de Identificación DES

Para el uso adecuado del aplicativo software de identificación por método de caja negra, se deben seguir los siguientes pasos:

1. Cargar los datos de operación del sistema a identificar (Figura B.1).
2. Iniciar el aplicativo software (Figura B.2).
3. Ingresar el numero de sensores o señales de salida, asociados a los datos de operación del sistema (Figura B.3).
4. Ingresar el numero de señales de control o entrada asociados a los datos de operación del sistema (Figura B.4).
5. Al ingresar los datos se mostraran el total de señales de operación del sistema, junto con la cantidad de datos o caracteres con los que se cuentas para la identificación (Figura B.5).
6. Con la información suministrada se elige cuantas observaciones se quieren hacer del sistema (Figura B.6).
7. Activar la opción Matriz, para desplegar la matriz de observaciones del sistema (Figura B.7).
8. Activar la opción Trazas, para desplegar la matriz de trazas del sistema (Figura B.8).

9. Activar la opción Ciclicidad, para observar si en la matriz de observaciones se presentan ciclos en las iteraciones (Figura B.9).
10. Después de observar la información nueva, pulsar Stop para terminar el proceso de identificación (Figura B.10).
11. Al detener el proceso de identificación; se obtienen las trazas de la matriz de observación, con las etiquetas por los vectores de trazas (Figura B.11).
12. Como identificación del sistema, el aplicativo software brinda la matriz pre y post asociadas a la red de petri del sistema (Figura B.12).
13. Si se desea realizar otra identificación del mismo sistema o de otro, desactivar las opciones de Matriz, Trazas y Ciclicidad, después repetir desde el paso 1 (Figura B.13).
14. Si se desea salir del aplicativo software, cerrar la ventana principal (Figura B.14).

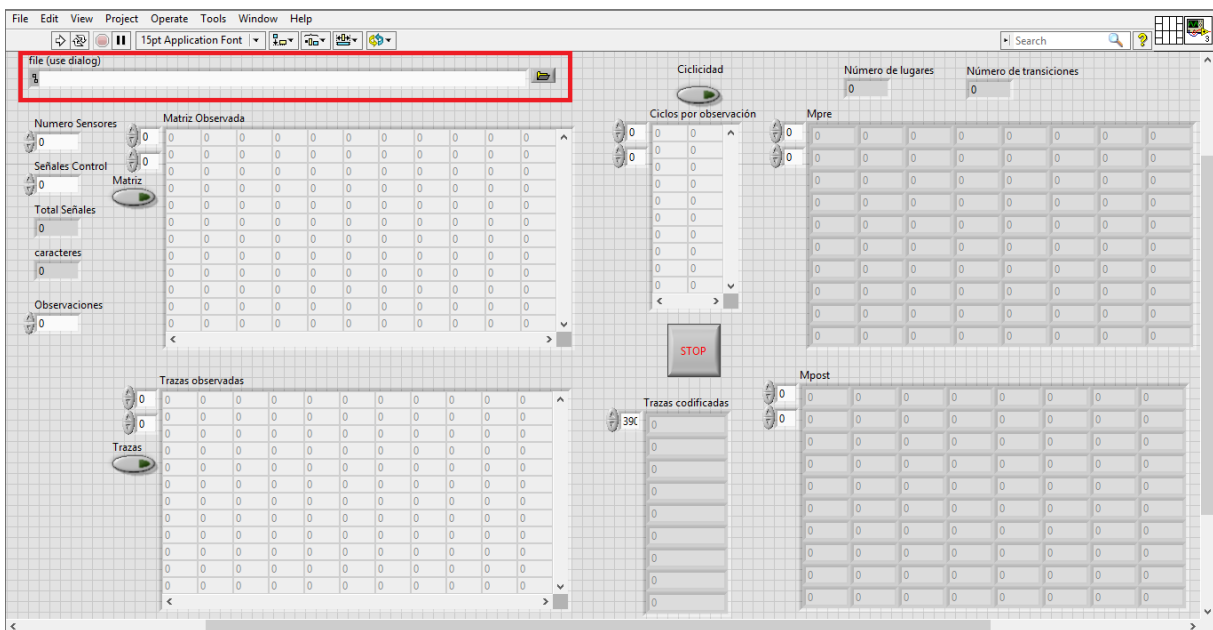


Figura B.1: Carga de datos

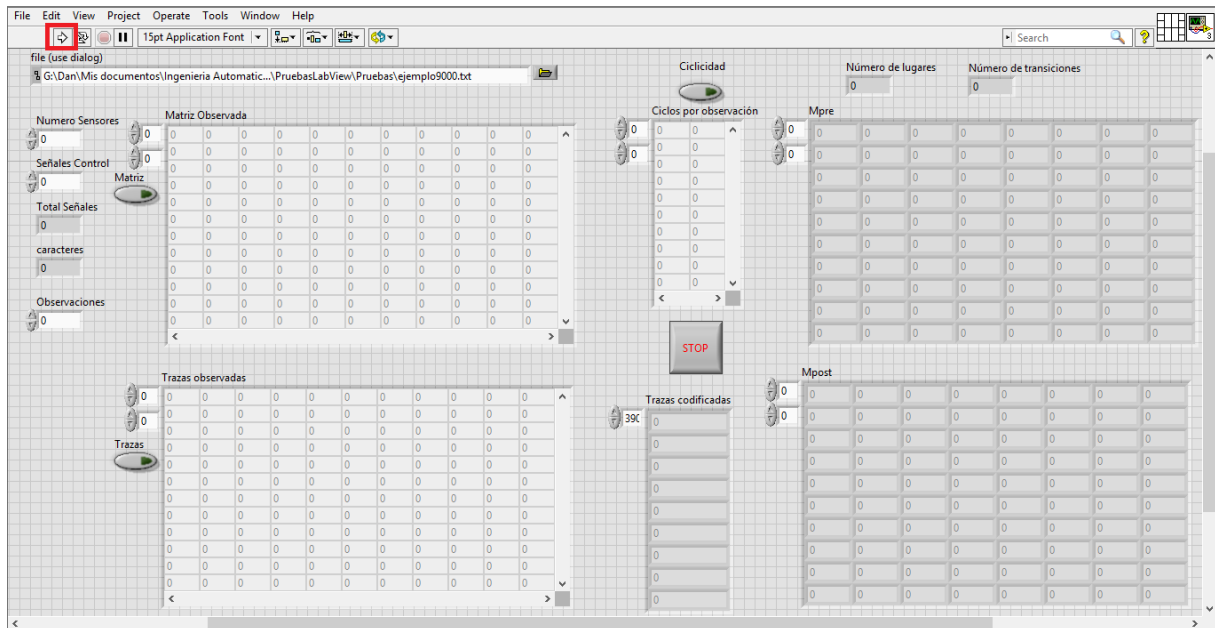


Figura B.2: Ejecución del Aplicativo Software

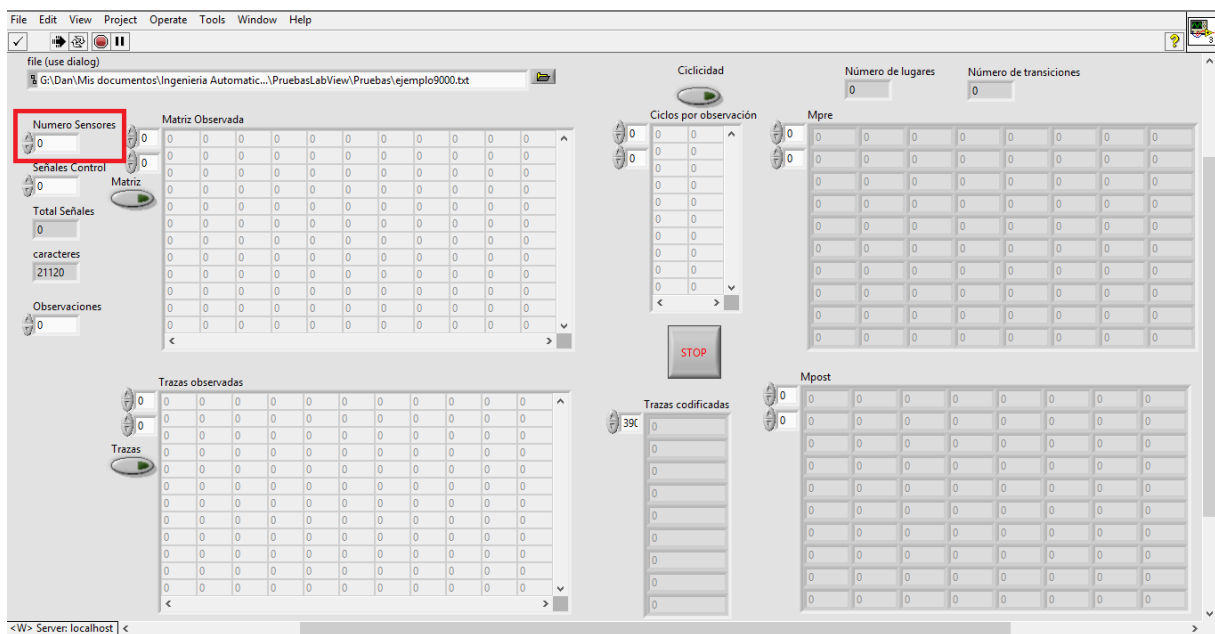


Figura B.3: Ingreso de número de señales de salida

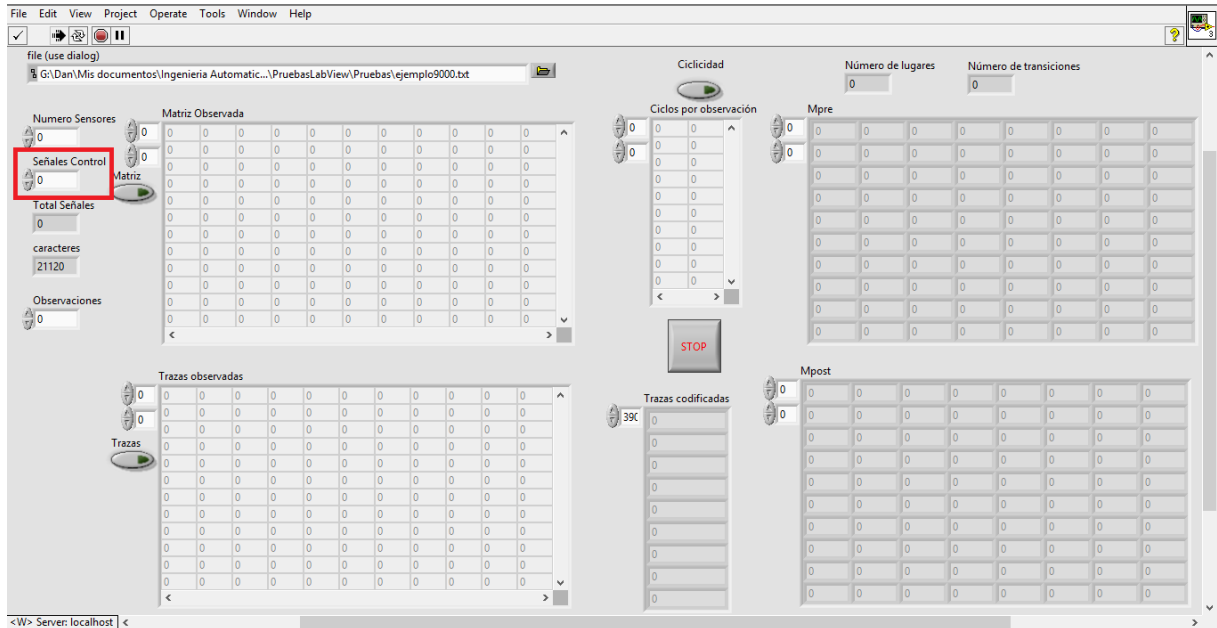


Figura B.4: Ingreso de número de señales de entrada

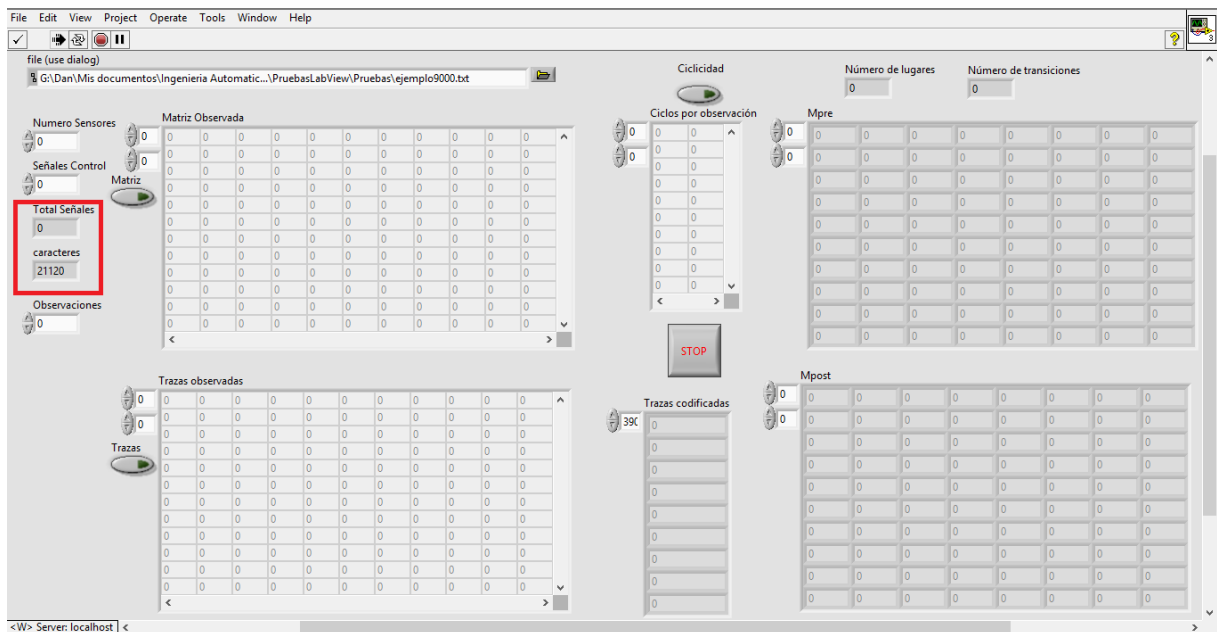


Figura B.5: Número total de señales del sistema

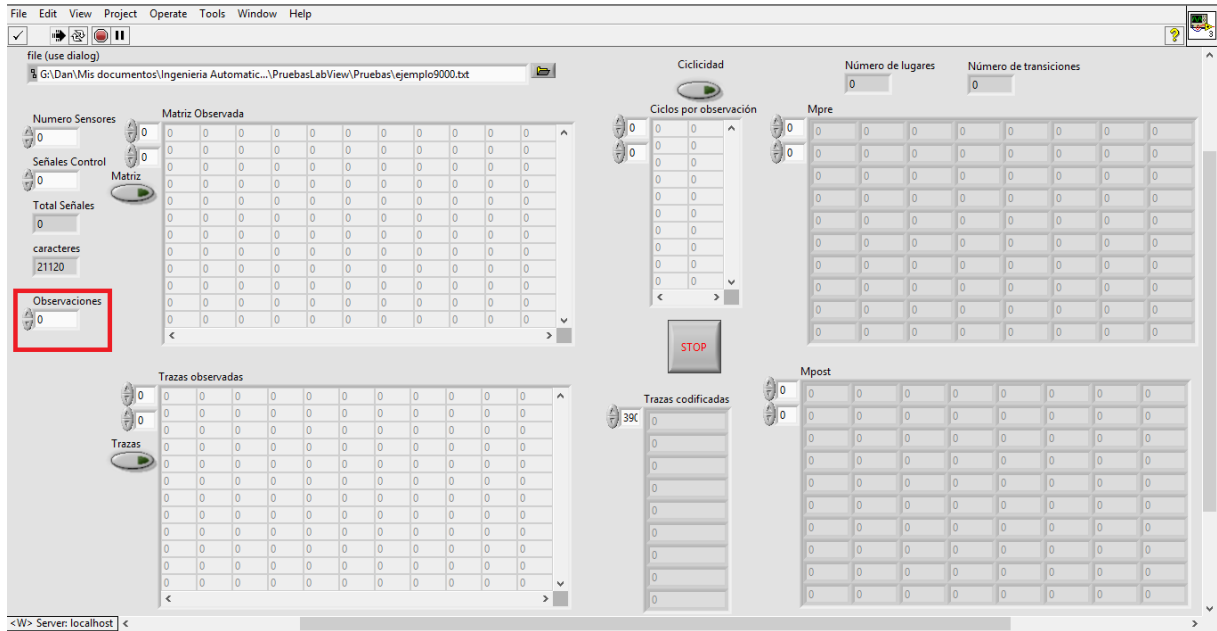


Figura B.6: Ingreso de número de observaciones

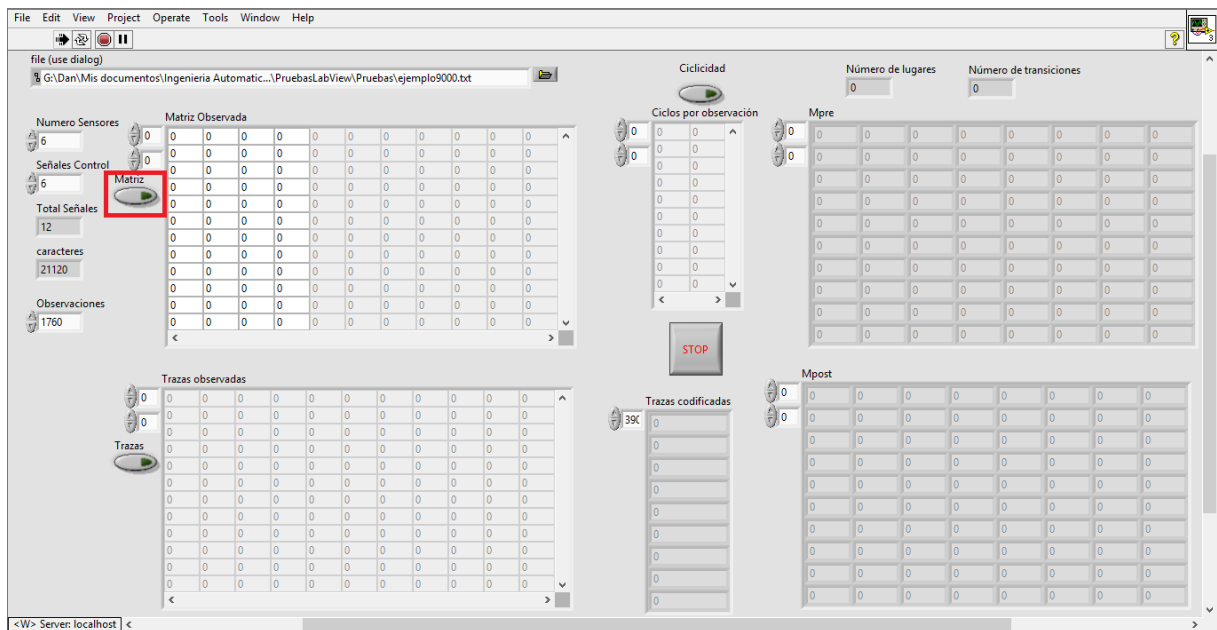


Figura B.7: Activación de la matriz de observaciones

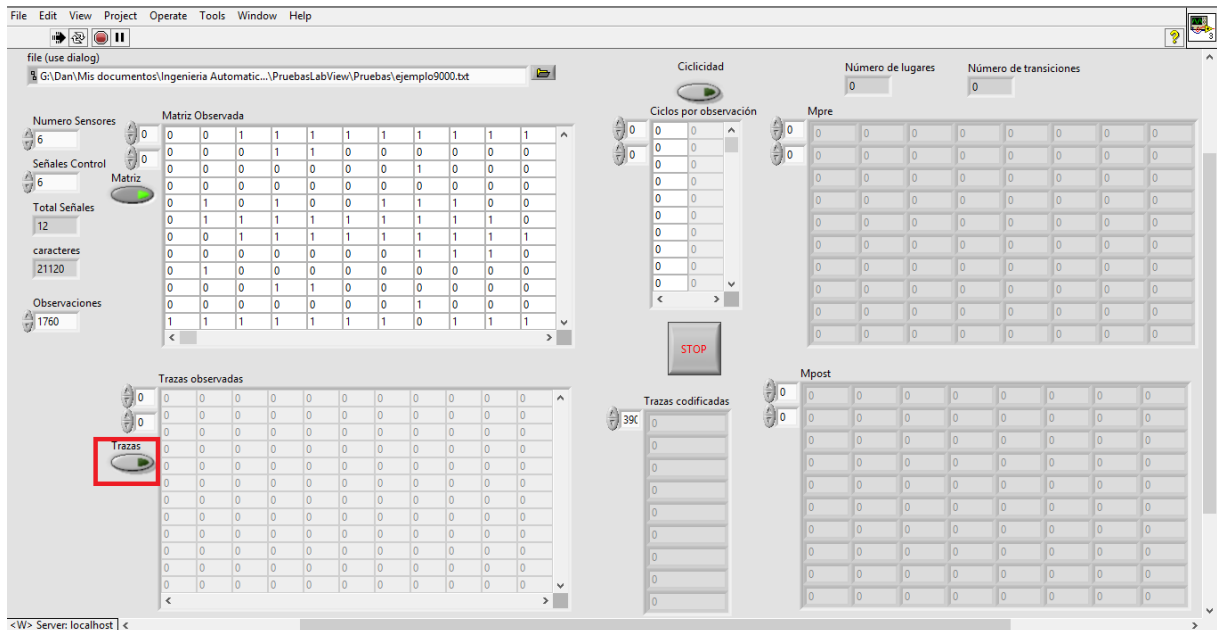


Figura B.8: Activación de la matriz de trazas

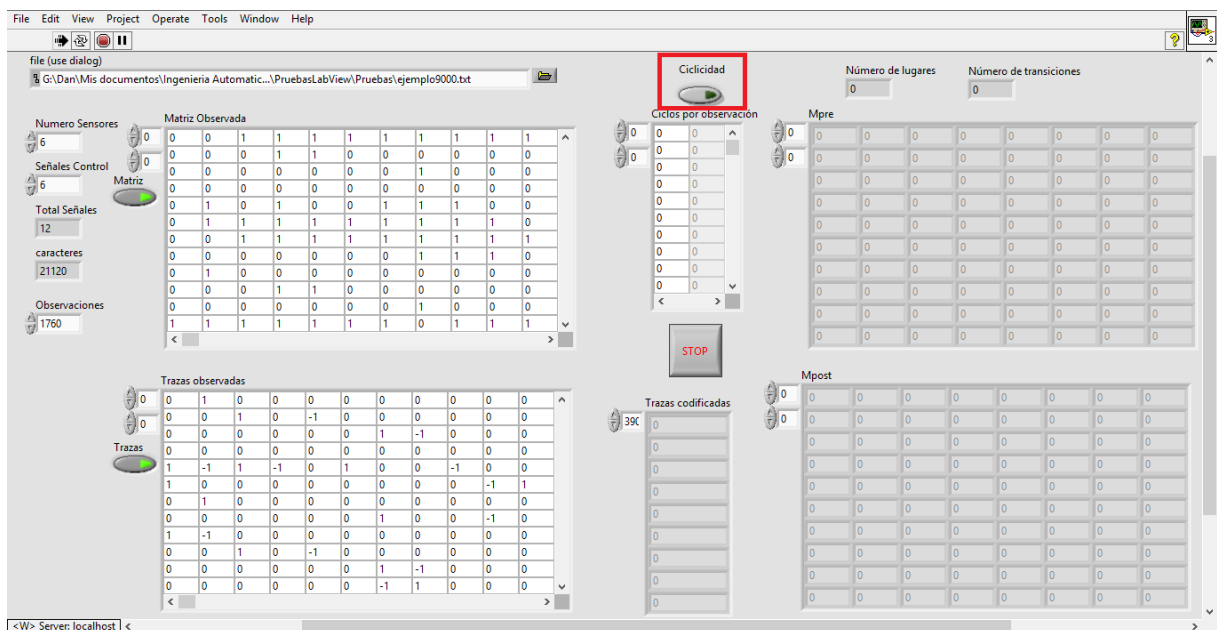


Figura B.9: Verificación de ciclicidad en la matriz de observaciones

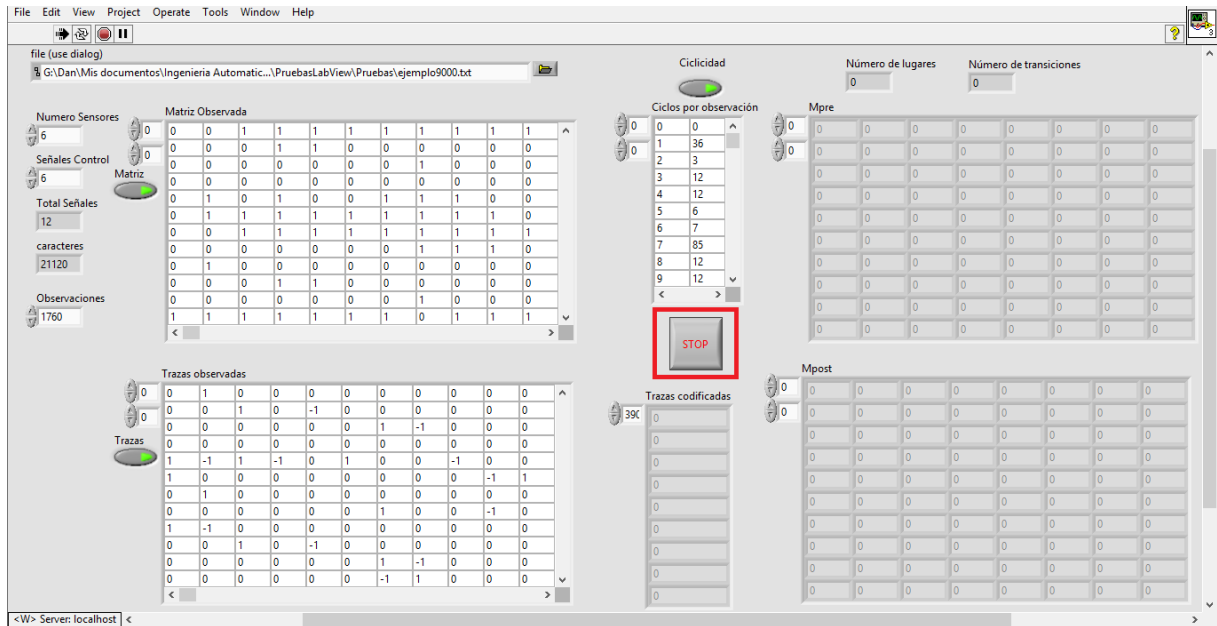


Figura B.10: Finalización del proceso de identificación

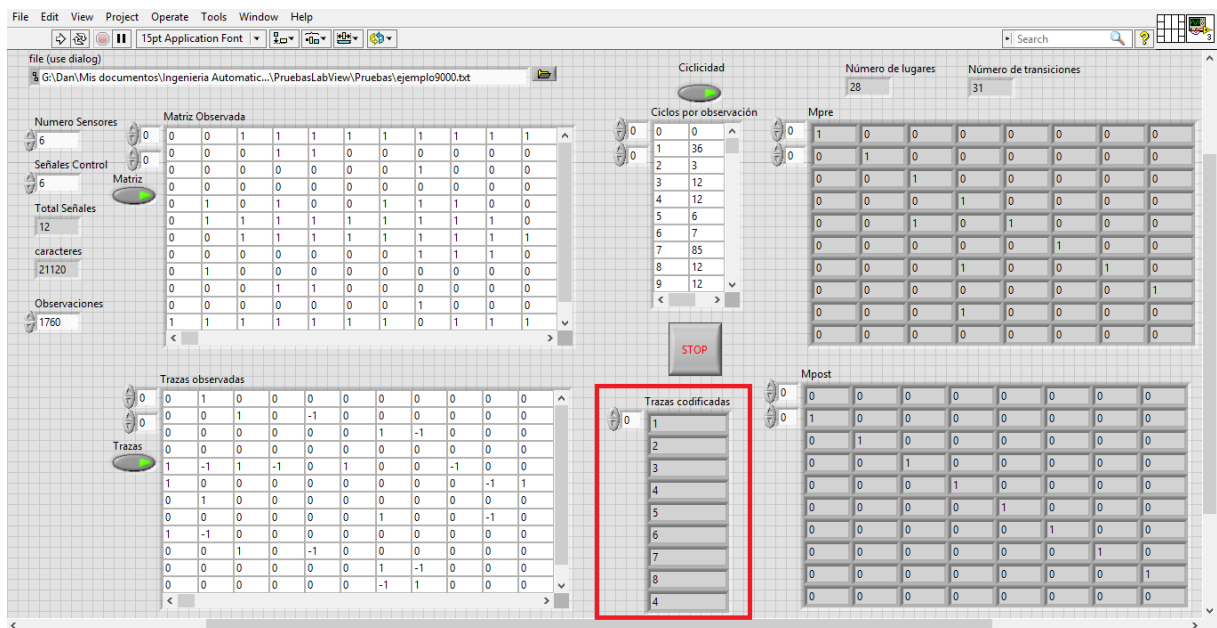


Figura B.11: Despliegue de trazas codificadas

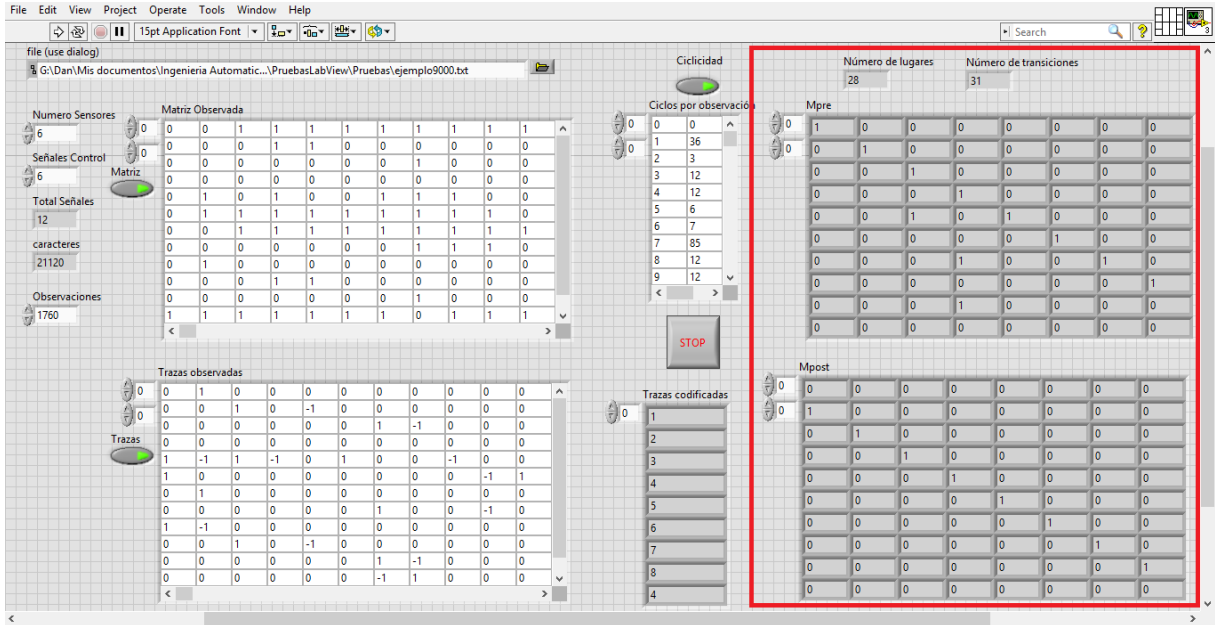


Figura B.12: Matriz Pre y Post del sistema identificado

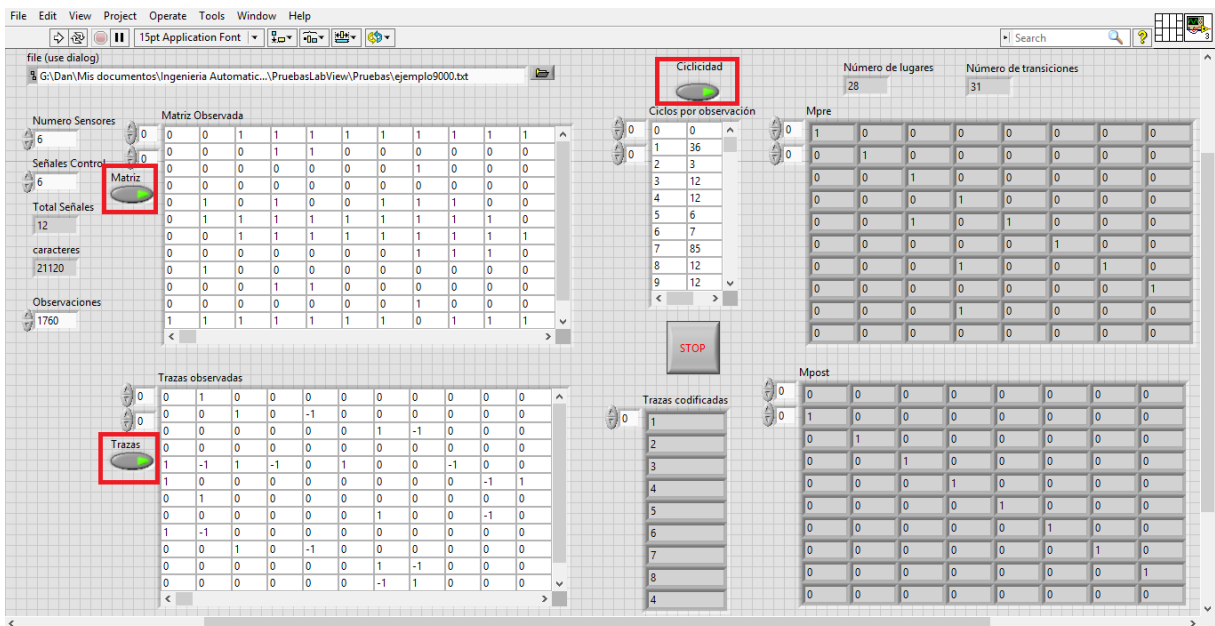


Figura B.13: Desactivación de opciones para reinicio

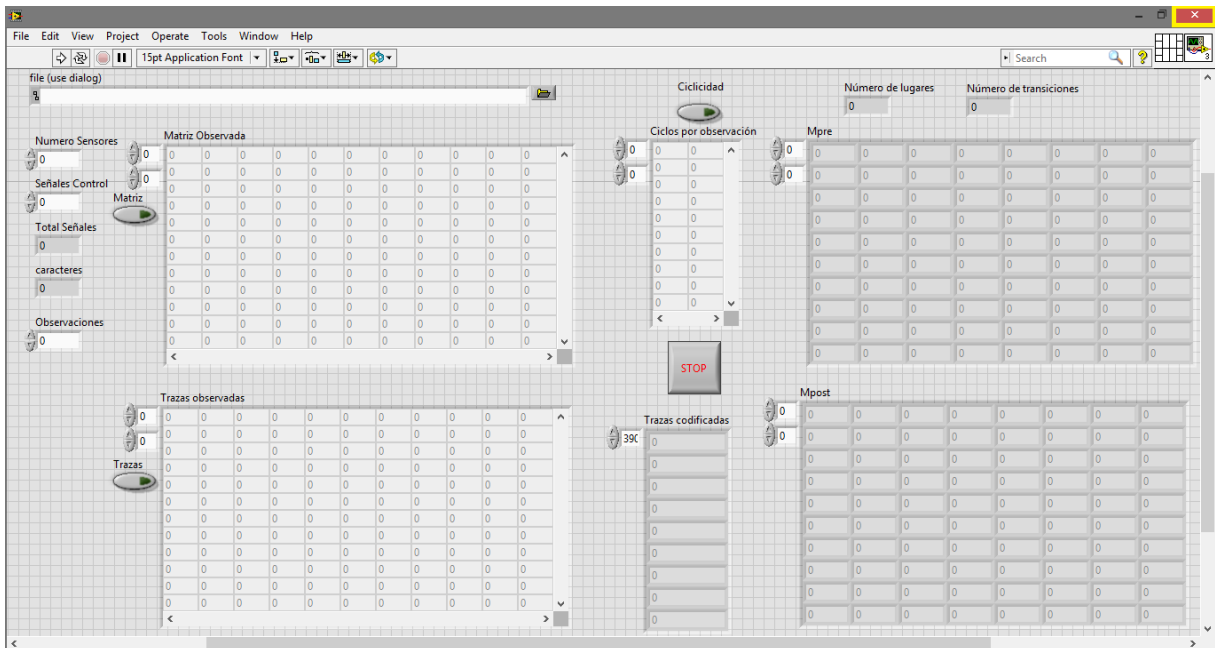


Figura B.14: Cierre del aplicativo software

Anexo C

Sistemas Simulados para Verificación de Diagnosticabilidad

En este anexo se presenta los 9 sistemas adicionales a los presentados en el capítulo 4, sobre los cuales se realizó verificación de diagnosticabilidad en función de sensores. Para lograr una visualización adecuada, el anexo tiene un componente virtual de imágenes ordenadas en carpetas de los sistemas expuestos, se presenta las simulaciones hechas en Simulink con una vista general del sistema en lazo cerrado, el controlador utilizado, junto con componentes adicionales de cada planta; al igual que las IPN obtenidas con la ayuda de Matlab y Graphviz en las pruebas 3 de diagnosticabilidad, la prueba 1 corresponde a la representación del sistema con los sensores de operación establecidos, la prueba 2 es la representación con la cantidad máxima de sensores posibles en el sistema, y la prueba 3 es la representación del sistema bajo el método de ubicación de sensores desarrollados. Todos los demás requerimientos para analizar el componente virtual será expuesto en la sección correspondiente de cada sistema.

C.1. Máquina Dispensadora de Bebidas

El funcionamiento de una máquina dispensadora de café, té y gaseosa (Figura C.1); consta de dos recipientes en los cuales se mantiene agua caliente y fría. Para ello se utilizan una resistencia calefactora (R) y un refrigerador (F). Las válvulas V1 y V2 son utilizadas para agregar agua y los sensores N1, N2, N3, N4 y T1, T2 para indicar los niveles máximos de agua y las temperaturas adecuadas, respectivamente. Se selecciona la bebida mediante los

pulsadores C (café), T (té) o G (gaseosa); para preparar la bebida debe abrirse la válvula de agua correspondiente (V3 o V4) durante 5 segundos. Simultáneamente, por 1 segundo, se activará la válvula del dispensador adecuado (Vc, Vt o Vg) luego la mezcla se agitará, a una velocidad respectiva a la bebida (dc, dg, dt), hasta terminado el agregado de agua. La operación se completa llenando el vaso abriendo por 5 segundos la válvula Ve. Es imprescindible controlar en todo momento tanto la temperatura de los depósitos de agua como el nivel de los mismos (deben mantenerse a la temperatura adecuada y cargarse cuando el nivel descienda por debajo de N3 y N4). Debido a los 3 modos de operación que presenta la máquina dispensadora (café, té o gaseosa), es la única excepción de los sistemas simulados con 7 pruebas de diagnosticabilidad.

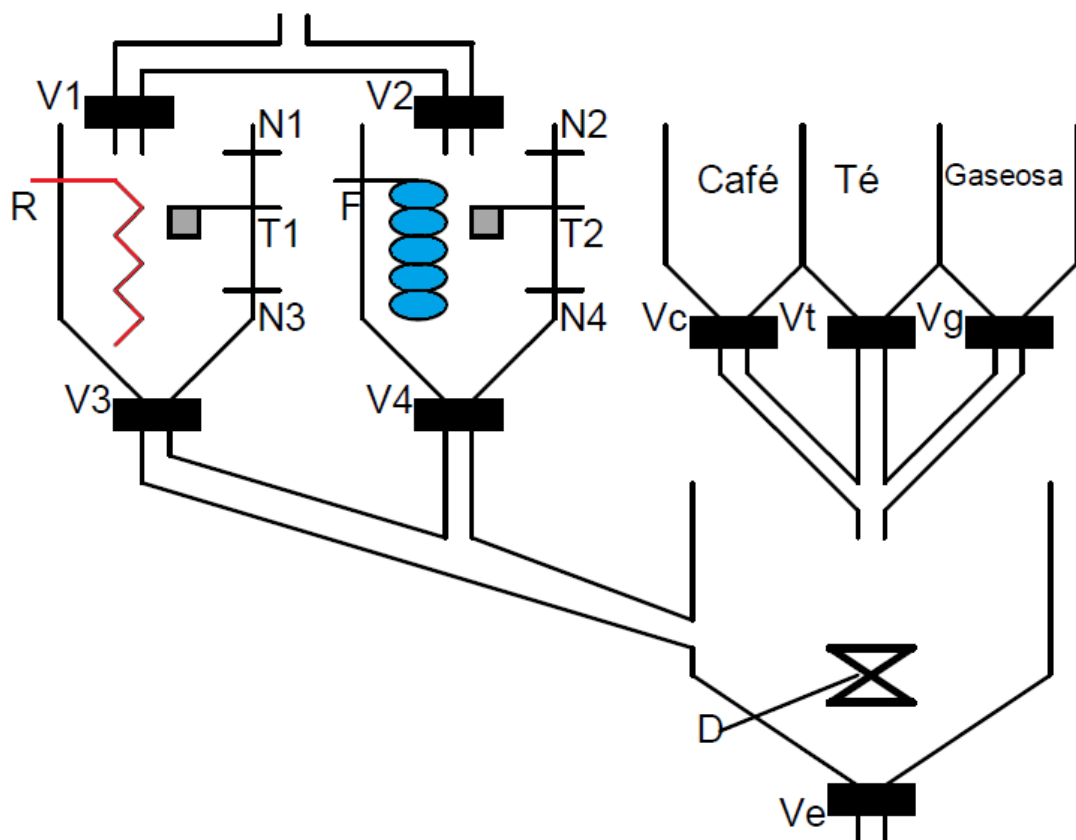


Figura C.1: Máquina Dispensadora de Bebidas

Con la simulación de la máquina dispensadora de bebidas, se configuró el modo de operación de Café Caliente y se adecuaron las 6 señales de sensor iniciales (Out) y las 13 de actuadores (In) (Tabla C.1), para exportarlas al programa de identificación en Labview obteniendo una IPN de 21 lugares y 20 transiciones; para posteriormente ejecutar la primera prueba de diagnosticabilidad, la cual indica que el sistema no es evento-detectable ni observable y por ende no es diagnosticable.

Tabla C.1: Señales In/Out de la Máquina Dispensadora de Bebidas

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	v1	Válvula entrada tanque 1	s1	N1	Nivel alto tanque 1
a2	v2	Válvula entrada tanque 2	s2	N2	Nivel alto tanque 2
a3	v3	Válvula salida tanque 1	s3	N3	Nivel bajo tanque 1
a4	v4	Válvula salida tanque 2	s4	N4	Nivel bajo tanque 2
a5	vc	Válvula suministro café	s5	T1	Temperatura tanque 1
a6	vg	Válvula suministro gaseosa	s6	T2	Temperatura tanque 2
a7	vt	Válvula suministro té	-	-	-
a8	ve	Válvula entrega bebida	-	-	-
a9	f	Señal refrigeración	-	-	-
a10	r	Señal calefacción	-	-	-
a11	dc	Señal mezclado café	-	-	-
a12	dg	Señal mezclado gaseosa	-	-	-
a13	dt	Señal mezclado té	-	-	-

Como segunda prueba de diagnosticabilidad bajo el mismo modo de operación, se adecuaron 19 señales de sensor (Out)(las 6 iniciales y 13 asociadas a los actuadores) y las 13 de actuadores (In) para exportarlas al programa de identificación en Labview obteniendo una IPN de 21 lugares y 20 transiciones; esta prueba se realiza ya que si se pueden medir de forma directa todas las señales de entrada y salida, se pueden inferir todos los eventos y estados que tenga el sistema, es decir, hay un límite máximo en el número de sensores para diagnosticabilidad. La segunda prueba indicó que el sistema es diagnosticable.

Por el método de ubicación de sensores desarrollado y corroborado por la prueba de diagnosticabilidad; se determinó que los sensores adicionales, necesarios para que la máquina dispensadora en modo de operación Café Caliente fuera diagnosticable, deben estar asociados a las válvulas Vc, Ve y V3, junto con la señal del mezclador Dc. Sin embargo al cambiar el modo de operación a Gaseosa Fría estos sensores no eran los adecuados, el método indicó que los sensores necesarios están asociados a las válvulas Vg y V4, señal de operación del refrigerador f y la señal del mezclador Dg, su IPN está conformada por 19 lugares y 18 transiciones.

Con el último modo de operación (Te Caliente), los sensores adicionales para diagnosticabilidad deben estar asociados a la válvula Vt y la señal del mezclador Dg, su IPN está conformada por 19 lugares y 18 transiciones; para ver de forma global de diagnosticabilidad se probaron los otros dos modos de operación Café caliente con IPN de 21 lugares y 20 transiciones; y Gaseosa Fría con IPN de 19 lugares y 18 transiciones. En todos los modos

de operación el sistema es diagnosticable con un total de 16 sensores (6 sensores de planta y 10 adicionales asociados a señales u operación de actuadores). Como se mencionó la máquina dispensadora de bebidas tiene 3 modos de operación, los cuales fueron simulados de forma individual por limitaciones del recurso computacional, en sus respectivas identificaciones el número de lugares y transiciones varia; ya que cada modo de operación tiene un conjunto de actuadores y sensores relacionados para ejecutar el proceso determinado.

C.2. Fabricación de Llaves de Tuercas

Para la fabricación de llaves de tuercas (Figura C.2), se requiere de una serie de operaciones divididas en dos módulos. El proceso se inicia con un botón de Marcha (M), que indica la llegada de una barra cilíndrica de acero para corte activando la cinta transportadora 1, en la máquina existe un sensor de medida s1, una vez que la barra activa el sensor y se detiene la cinta 1, se activa un brazo robot 1 para quedar en posición con la barra, sujetando la barra con pinzas (Pz1) hasta la activación un sensor de sujeción (sj1), llevando la barra a la primera posición de operación (pos1); activando la cierra 2 segundos después del brazo robot 1, para después iniciar el avance de la cierra durante 1.5 segundos hacia el robot con la barra (pos1) y luego retornar a su posición original, el avance de la cierra solo se activará si el sensor de sujeción (sj1) está activo. En el momento que la cierra ha terminado su trabajo, el brazo robot se pone en marcha nuevamente con la barra y la lleva a una nueva posición (pos2), se desactivan las pinzas de sujeción y la barra queda en una nueva estación donde hay una cinta transportadora que se activa moviendo la barra a través de un horno, que se mantiene encendido todo el tiempo, hasta que se activa un sensor de posición (s2) para indicar la salida de la barra del horno, en ese momento se detiene la cinta 2.

En el segundo módulo, una vez que la barra llega al final de la cinta y se activa (s2), se esperan 1.5 segundos para que la barra se estabilice, transcurrido este tiempo se pone en marcha el segundo brazo robot (BR2) hasta que alcanza la posición correcta donde está la barra (pos3), se activan las pinzas del segundo robot (PZ2) hasta que se active la sujeción 2 (sj2), llevando la barra a la siguiente posición (pos4) donde un dado monoestable (D1) con presión, da forma a la barra durante 1 segundo; el brazo con la barra va a la última posición (pos5), donde se le quita el exceso de material con otro dado monoestable (D2) durante 2 segundos. Para después desactivar las pinzas y el robot vuelve a su posición original; cayendo la llave de tuercas en un contenedor, para luego realizarle trabajos manuales de

acabado.

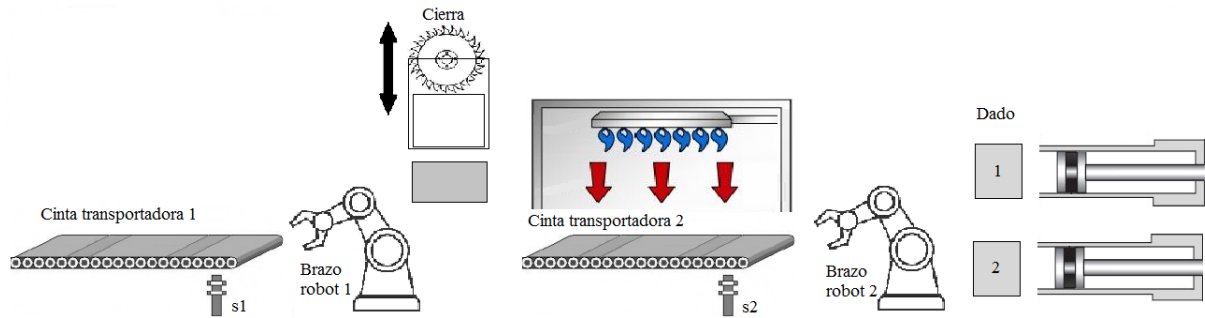


Figura C.2: Proceso de Fabricación de Llaves de Tuerca

Con la descripción del Proceso de Fabricación de Llaves de Tuercas (PFLT) se implementó en simulink, con su respectivo controlador, el funcionamiento de la cinta transportadora 1 y 2, la operación del brazo robot 1 que tiene 3 posiciones representadas en órdenes de control como $j1$ y $j2$: posición inicial (0-0), posición 1 (1-0) y posición 2(1-1), la operación del brazo robot 2 que tiene 4 posiciones representadas en órdenes de control como $l1$ y $l2$: posición inicial (0-0), posición 3 (1-0), posición 4(0-1), posición 5 (1-1), el funcionamiento de las pinzas de los brazos robots 1 y 2, determinadas por las acciones de control $x1-x2$ y $y1-y2$ respectivamente, con tres acciones: activación (1-1), desactivación (0-0) y espera (1-0); estas señales quedan registradas en la Tabla C.2 como actuadores. Por la operación de control en la cierra compuesta por corte (cc) y movimiento (cm), no es necesaria simular la planta de corte.

Tabla C.2: Señales In/Out de la Fabricación de Llaves de Tuerca

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	c1	Cinta transportadora 1	s1	p1	Sensor posición 1
a2	c2	Cinta transportadora 2	s2	p2	Sensor posición 2
a3	cc	Señal corte de cierra	s3	p3	Sensor posición 3
a4	cm	Señal movimiento de cierra	s4	p4	Sensor posición 4
a5	j1	Señal 1 brazo robot 1	s5	p5	Sensor posición 5
a6	j2	Señal 2 brazo robot 1	s6	S1	Final de carrera 1
a7	l1	Señal 1 brazo robot 2	s7	S2	Final de carrera 2
a8	l2	Señal 2 brazo robot 2	s8	sj1	Sensor sujeción 1
a9	x1	Señal 1 pinza 1	s9	sj2	Sensor sujeción 2
a10	x2	Señal 2 pinza 2	-	-	-
a11	y1	Señal 1 pinza 2	-	-	-
a12	y2	Señal 2 pinza 2	-	-	-
a13	d1	Señal activación dado 1	-	-	-
a14	d2	Señal activación dado 2	-	-	-

Con los datos iniciales de simulación del PFLT (9 señales de sensor y 14 señales de actuadores)(Tabla C.2), se inició la evaluación de diagnosticabilidad del sistema con 2 ciclos de operación, obteniendo una IPN de 21 lugares y 20 transiciones, pero el sistema no cumplía las características de observabilidad y de evento-detectabilidad, por ende tampoco era diagnosticable. La segunda prueba de diagnosticabilidad se realizó con el número máximo de sensores (23- 9 señales de sensores y 14 adicionales asociados a actuadores), se obtuvo la misma cantidad de lugares y transiciones en la IPN, pero en este caso el sistema es diagnosticable (comportamiento esperado). En la tercera prueba de diagnosticabilidad, se añadieron a los 9 sensores iniciales, sensores asociados a la señal de control j2 del brazo robot 1 y la l1 del brazo robot 2, la señal de corte de la cierra, y activación del dado 1 y 2, indicados por el método de ubicación de sensores; obteniendo un sistema diagnosticable y con una IPN asociada de 21 lugares y 20 transiciones.

C.3. Cadena de Pintado

La cadena de pintado (Figura C.3) consta de una rampa de suministro de piezas, para llevar al túnel de pintado y secado. Para el ingreso de la pieza para pintar se usa un pistón monoestable(H), empujando la pieza hacia la cinta transportadora T1. La entrada de la pieza al túnel de pintado se registrara con el sensor pin, desactivando el piston H, y

poniendo en marcha así las pistolas de pintura P que no dejaran de funcionar hasta que la pieza pase por el sensor psec; el cual da inicio al proceso de secado, los ventiladores de secado V seguirán en funcionamiento hasta que la pieza rebase el final del túnel (sensor pout). Cuando el túnel este libre se podrá reiniciar la operación de selección, pintado y secado.

La verificación y clasificación comienza cuando una pieza sale del túnel (sensor pout), poniendo en marcha la cinta T2, el cual parara cuando la pieza sea detectada por el sensor ver que representa la posición idónea para el verificado de la pieza pintada I, la maquina responsable de ello emitirá una de las dos señales posibles (bueno o malo). Para la clasificación de piezas buenas y piezas malas se usará una puerta diagonal que dirigirá la pieza donde corresponda; esta puerta esta accionada por un pistón biestable con las posiciones de fin de carrera a+ y a-, y las correspondientes acciones (Ap, para ir de a- hasta a+ y An, para ir de a+ hasta a-). Puesto que se supone que la mayor parte de las piezas están bien pintadas, esta puerta se mantendrá en la posición a- (piezas buenas). Si una pieza es buena, se activara otra vez la cinta T2 mientras cae por la rampa correspondiente. Por otro lado, si la pieza es mala, la puerta se moverá para dirigirla a la rampa correspondiente mediante el accionamiento de la cinta T2, para posterior caída en la rampa de piezas malas y parada de la cinta. Este proceso entero se accionara cuando el operario pulse M y no parara hasta que se pulse de nuevo M y no haya piezas dentro del túnel de pintado y secado.

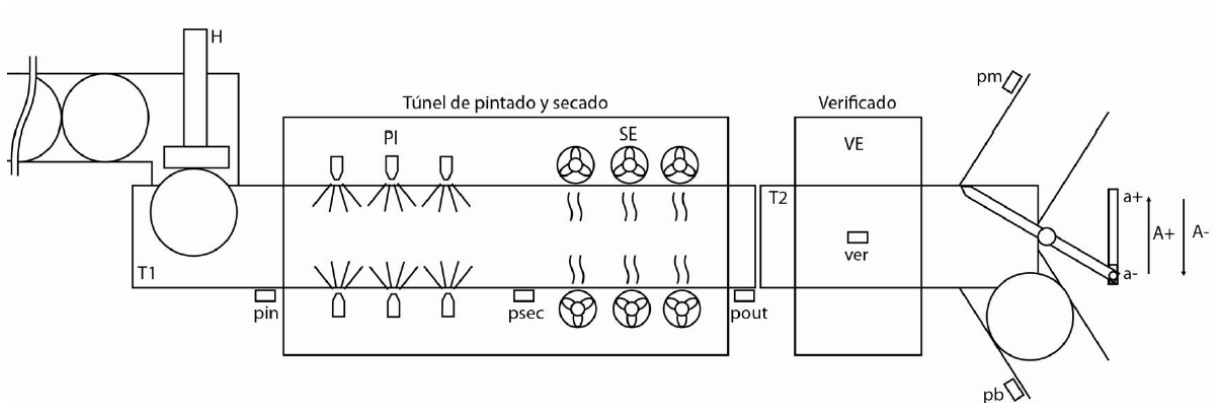


Figura C.3: Cadena de Pintado

Con los datos iniciales del proceso para un ciclo de operación (7 señales de sensor y 8 señales de actuador)(Tabla C.3), se realizó la primera prueba de diagnosticabilidad obteniendo que el sistema no era observable ni evento-detectable, con una IPN de 13 lugares y 12 transiciones. En la segunda prueba con un máximo de 15 señales de sensor (7 iniciales y 8 sensores adicionales asociados a los actuadores), el sistema se volvió diagnosticable y se mantuvo la misma cantidad de lugares y transiciones. En la última prueba se aplicó sobre el sistema inicial el método de ubicación de sensores con un total de 9 sensores (7 iniciales y un sensor asociado a cada banda transportadora), con estas señales el sistema es diagnosticable con una IPN de 13 lugares y 12 transiciones.

Tabla C.3: Señales In/Out de la Cadena de Pintado

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	H	Señal activación pistón 1	s1	s	Posición 1
a2	T1	Cinta transportadora 1	s2	Pin	Posición entrada
a3	P	Señal activación pintura	s3	Psec	Posición secado
a4	V	Señal activación ventilador	s4	Pout	Posición salida
a5	T2	Cinta transportadora 2	s5	o	Posición observación
a6	I	Señal activación inspección	s6	b	Sensor pieza buena
a7	Ap	Posición positiva pistón 2	s7	m	Sensor pieza mala
a8	An	Posición negativa pistón 2	-	-	-

C.4. Línea de Embalaje

En una línea de embalaje se tiene un producto que consiste en un paquete dos cajas (A) y (B) dentro de una caja grande (C)(Figura C.4). Las cajas A, B y C circularán respectivamente por las cintas MA, MB y MC. Las cajas que llegan por las cintas se detienen cuando los sensores (sA), (sB) o (sC) detectan alguna caja. Se procede al llenado de las cajas A y B con las órdenes AL y BL, los sitios de llenado tienen sensores (AF) y (BF) de posición que advierten de la finalización del llenado de las cajas. Cuando las cajas están llenas se conducen hasta el final de las cintas hasta los sensores (FA) y (FB). Un brazo robot se encargará de recoger A y B y meterlas dentro de C. El brazo robot se activa siempre y cuando haya cajas A o B y C posicionadas; tiene prioridad la caja que llegue antes al final de la cinta. Se toman las cajas A y B para meterlas dentro de la C con las acciones (TA-TC y TB-TC), hasta la activación de sus sensores de llegada(AC y BC).

Las cajas llenas C continuarán hasta el final de la cinta (FC) y llevar al proceso de sellado. El sistema se activa mediante un pulsador de Marcha (M). Cuando el sistema se inicia empiezan a llegar cajas por las tres cintas transportadoras MA, MB y MC.

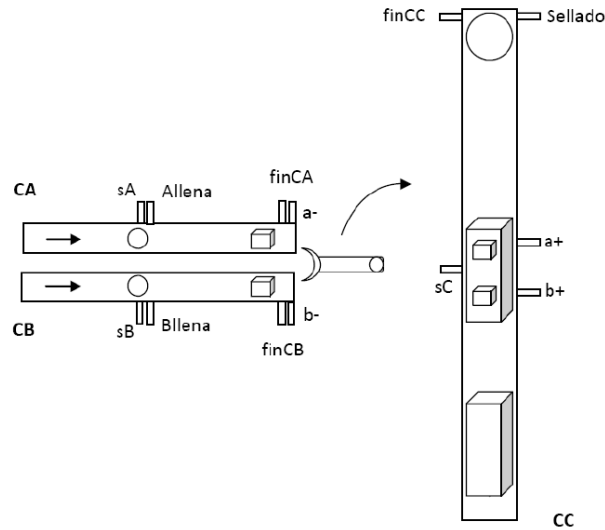


Figura C.4: Línea de Embalaje

Con los datos iniciales del proceso para un ciclo de operación (10 señales de sensor y 8 señales de actuador)(Tabla C.4), se realizó la primera prueba de diagnosticabilidad obteniendo que el sistema era diagnosticable, con una IPN de 16 lugares y 15 transiciones. En la segunda prueba con un máximo de 18 señales de sensor (10 iniciales y 8 sensores adicionales asociados a los actuadores), el sistema permaneció diagnosticable y con la misma cantidad de lugares y transiciones. En este caso no fue necesario realizar una tercer prueba para aplicar el método de ubicación de sensores.

Tabla C.4: Señales In/Out de la Linea de Embalaje

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	MA	Cinta transportadora A	s1	SA	Sensor llegada caja A
a2	MB	Cinta transportadora B	s2	AF	Sensor llenado caja A
a3	MC	Cinta transportadora C	s3	FA	Final de carrera A
a4	AL	Señal llenar caja A	s4	SB	Sensor llegada caja B
a5	BL	Señal llenar caja B	s5	BF	Sensor llenado caja B
a6	TA	Señal A brazo robot	s6	FB	Final de carrera B
a7	TB	Señal B brazo robot	s7	SC	Sensor llegada caja C
a8	TC	Señal C brazo robot	s8	FC	Final de carrera C
-	-	-	s9	AC	Sensor llegada caja A a C
-	-	-	s10	BC	Sensor llegada caja B a C

C.5. Equipo de Mezclado

El funcionamiento de un equipo de mezclado de componentes (Figura C.5), para formar la resina para introducirlos en un molde para que reaccionen con una matriz de fibras de basalto y formar finalmente el material compuesto (Proceso RIM); se detalla a continuación: Se activa el equipo (M), ambos componentes que reaccionan (caprolactama + iniciador y caprolactama + activador) para formar la poliamida, se añaden a cada uno de los tanques (Tin1 y Tin2), los cuales incorporaran un sensor (Sl1 y Sl2) la cantidad idónea de cada uno de ellos (TA y TB), una vez alcanzada la cantidad exacta, se activara el sistema neumático que actúa como atmósfera protectora (atm), unido directamente a cada uno de los tanques, alcanzando los 2 bares de presión, no activándose los motores del sistema de mezclado del interior de los tanques (D1 y D2) hasta alcanzar la presión (Pin).

Estos tanques se conectan al sistema de pistones dispensadores, el cual determinara la proporción de mezclado, siendo necesaria para el proceso de obtención de composites a base de fibras de basalto una relación 1 : 1. Unas válvulas neumáticas (v1 y v2) demandan la cantidad exacta (cl) de cada uno de los reactivos a los pistones, estos pistones no impulsaran el material hasta que el perfil basculante que los une se encuentre en una posición horizontal de 0° (Ph), obteniendo de este modo una relación de mezclado 1 : 1. Los pistones (p1 y p2) impulsan los componentes a través de unas mangueras conectoras a la mezcladora neumática de comportamiento dinámico, de manera que esta incorpora unos sensores de presión que detectan la cantidad necesaria de cada uno de los componentes,

de manera que si detecta un presión (S_{vm1} y S_{vm2}), significaría que los componentes están llenando el molde, reducirá la velocidad de la mezcladora de alto (D_h) a bajo (D_l), recirculando a los tanques el exceso de material (R_1 y R_2); otro sensor detecta que cantidad sigue siendo optima (s_{vo1} y s_{vo2}). Al llegar a la presión máxima en el interior del molde (S_p) la válvula dejara de impulsar los componentes, desconectando la atmosfera protectora (atm), y esta parara el sistema de pistones y pasado un segundo el sistema de mezclado del interior de los tanques del mismo modo se detendrá. Los componentes en el interior del molde (ambas soluciones de caprolactama) entran en contacto con la matriz de fibra de basalto, reaccionando químicamente y formándose finalmente el composite.

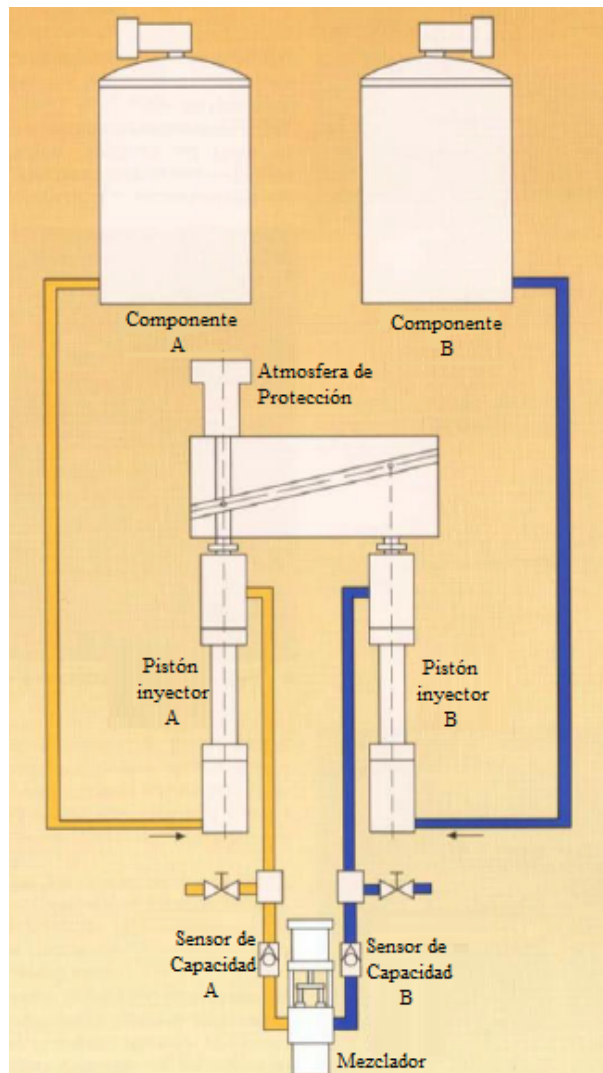


Figura C.5: Equipo de Mezclado

Se realizó la primera prueba de diagnosticabilidad con los datos iniciales del proceso para dos ciclos de operación (9 señales de sensor y 15 señales de actuador)(Tabla C.5), obteniendo que el sistema no era observable ni evento-detectable, con una IPN de 19 lugares y 18 transiciones. En la segunda prueba con un máximo de 24 señales de sensor (9 iniciales y 15 sensores adicionales asociados a los actuadores), el sistema se volvió diagnosticable y se mantuvo la misma cantidad de lugares y transiciones. En la ultima prueba se aplicó sobre el sistema inicial el método de ubicación de sensores con un total de 23 sensores (9 iniciales, todos los posibles sensores asociados a los actuadores exceptuando la velocidad de mezcla alta), el sistema es diagnosticable con una IPN de 13 lugares y 12 transiciones.

Tabla C.5: Señales In/Out del Equipo de Mezclado

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	Tin1	Señal llenar tanque 1	s1	sl1	Sensor nivel tanque 1
a2	Tin2	Señal llenar tanque 2	s2	sl2	Sensor nivel tanque 2
a3	Tout1	Señal vaciar tanque 1	s3	Pin	Sensor presión de entrada
a4	Tout2	Señal vaciar tanque 1	s4	cl	Sensor cantidad exacta
a5	D1	Motobomba tanque 1	s5	ph	Sensor posición horizontal
a6	D2	Motobomba tanque 2	s6	sm1	Sensor desbalance 1
a7	Dh	Velocidad mezclado alto	s7	svo1	Sensor cantidad optima 1
a8	Dl	Velocidad mezclado bajo	s8	svm2	Sensor desbalance 2
a9	atm	Señal atmósfera protectora	s9	svo2	Sensor cantidad optima 2
a10	p1	Señal activación pistón 1	s10	sp	Sensor de presión máxima
a11	p2	Señal activación pistón 2	-	-	-
a12	v1	Válvula neumática 1	-	-	-
a13	v2	Válvula neumática 2	-	-	-
a14	R1	Señal retorno tanque 1	-	-	-
a15	R2	Señal retorno tanque 2	-	-	-

C.6. Línea de Producción de Yogurt

Para el inicio del proceso de producción de yogurt (Figura C.6), se pulsa el botón de marcha (M) empezando el llenado del tanque de suministro de leche con chaqueta térmica por medio de un bombeo inicial(Ti), después de que un sensor de nivel (FL) indica que el tanque esta lleno se empieza el proceso de pasteurización con la calefacción de la leche (Cl) hasta alcanzar un temperatura establecida (Tl). Con el lote de leche pasteurizado se

da paso al proceso de fermentado, inyectando la leche por medio de una válvula (V1) y una motobomba (B1) hacia el tanque de fermentado hasta que el tanque de leche llega a un nivel establecido (E1), al mismo tiempo se empieza el flujo desde el tanque de enzimas (la reserva de enzimas se considera constante) por medio de una válvula (Ve) y una motobomba (Be) hasta que un sensor de nivel (Ff) indica el llenado del tanque de fermentado.

Para la fermentación se ejecutan dos acciones rápidas, incremento la temperatura de la mezcla (leche+enzimas) por medio de un calefactor (Cf) y homogenización de la mezcla con un mezclador (D1), terminada la fermentación se pasa al llenado del tanque de yogurt con el fermentado y aditivos, para el flujo del tanque de fermentado al de yogurt por seguridad se deben activar dos válvulas (V1 y V2) junto con una motobomba (Bf), hasta que un sensor de nivel indique que el tanque de fermentado este vacío; para la adición de aditivos se activa una válvula (Va) y una motobomba (Ba) hasta que el tanque de yogurt este lleno (Fy). Para la elaboración de yogurt se debe homogenizar la mezcla con un mezclador (D2) hasta que se llegue a un punto de temperatura deseado (Ty) por medio de un calefactor (Cy); cuando el lote de yogurt esta finalizado se pasa al area de embotellamiento por medio de una válvula (Vem) y una motobomba (Bem) hasta que el tanque de yogurt esta vacio (Ey).

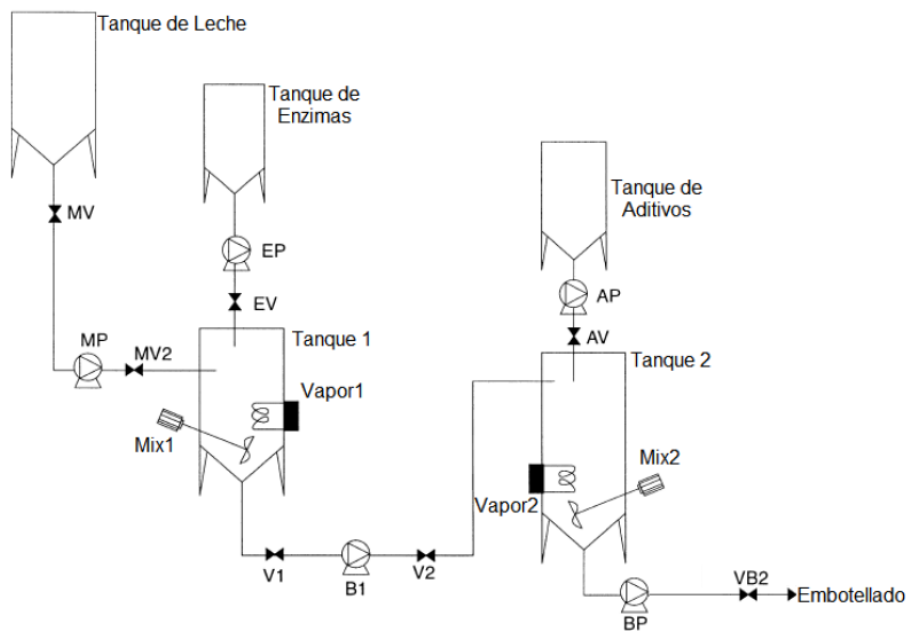


Figura C.6: Línea de Producción de Yogurt

Se realizó la primera prueba de diagnosticabilidad con los datos iniciales del proceso para

un ciclo de operación (9 señales de sensor y 17 señales de actuador)(Tabla C.6), obteniendo que el sistema no era evento-detectable y por ende no diagnosticable, con una IPN de 21 lugares y 20 transiciones. En la segunda prueba con un máximo de 26 señales de sensor (9 iniciales y 17 sensores adicionales asociados a los actuadores), el sistema se volvió diagnosticable y se mantuvo la misma cantidad de lugares y transiciones. En la última prueba se aplicó sobre el sistema inicial el método de ubicación de sensores con un total de 10 sensores (9 iniciales y uno asociado a la motobomba de aditivos), el sistema es diagnosticable con una IPN de 21 lugares y 20 transiciones.

Tabla C.6: Señales In/Out de la Línea de Producción de Yogurt

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	Tl	Señal llenar leche	s1	F1	Nivel alto leche
a2	Cl	Señal calentar leche	s2	E1	Nivel bajo leche
a3	Vl	Señal vaciar leche	s3	FF	Nivel alto fermentado
a4	Bl	Motobomba leche	s4	EF	Nivel bajo fermentado
a5	Ve	Válvula enzimas	s5	FY	Nivel alto yogurt
a6	Be	Motobomba enzimas	s6	EY	Nivel bajo yogurt
a7	D1	Señal mezclador 1	s7	TL	Temperatura leche
a8	Cf	Señal calentar fermentado	s8	TF	Temperatura fermentado
a9	Va	Válvula aditivos	s9	Ty	Temperatura yogurt
a10	Ba	Motobomba aditivos	-	-	-
a11	V1	Válvula seguridad 1	-	-	-
a12	Bf	Motobomba fermentado	-	-	-
a13	V2	Válvula seguridad 1	-	-	-
a14	D2	Señal mezclador 2	-	-	-
a15	CY	Señal calentar yogurt	-	-	-
a16	Vem	Válvula embotellado	-	-	-
a17	Bem	Motobomba embotellado	-	-	-

C.7. Planta Clasificadora de Piezas

Se tiene un proceso de descarga y clasificación de material de ferretería (Figura C.7), de una empresa que recibe containers con el material embalado en cajas. El proceso pretende que tras la descarga de la mercancía en una tolva por un operario, se ingrese a través de una cinta transportadora en la cual actúan diversos pistones temporizados que distribuyen cada producto. El proceso empieza con el accionamiento del botón de inicio (I) que activa la cinta transportadora (C), para llevar la pieza a un sistema de visión de máquina donde

se clasifica como tuerca, tornillo, llave inglesa, martillo o alicate (s1, s2, s3, s4 y s5); dependiendo la clasificación se activara el pistón correspondiente (P1, P2, P3, P4 y P5) en el tiempo adecuado. Sí no se ha clasificado la pieza, esta llegará al final de la banda transportadora y se prenderá una alarma luminosa (A).

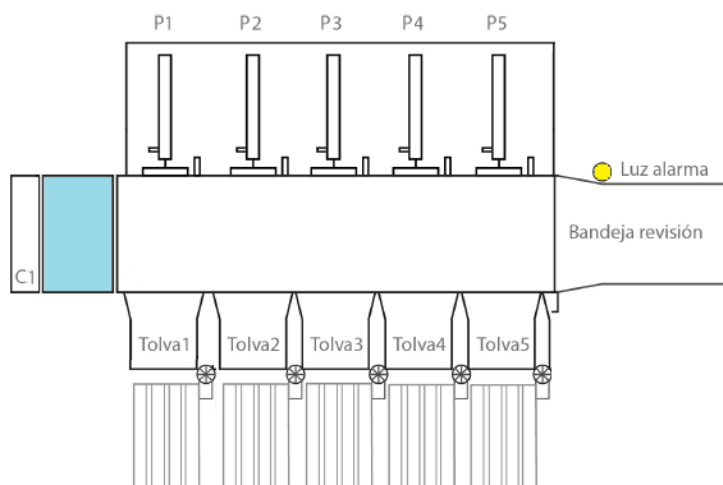


Figura C.7: Equipo de Mezclado

Con los datos iniciales del proceso para dos ciclos de operación (5 señales de sensor y 7 señales de actuador)(Tabla C.7), se ejecutó la primera prueba de diagnosticabilidad obteniendo que el sistema no era diagnosticable, con una IPN de 32 lugares y 31 transiciones. En la segunda prueba con un máximo de 12 señales de sensor (5 iniciales y 7 sensores adicionales asociados a los actuadores), el sistema se volvió diagnosticable y se mantuvo la misma cantidad de lugares y transiciones. En la ultima prueba se aplicó sobre el sistema inicial el método de ubicación de sensores con un total de 11 sensores (5 iniciales y los posibles sensores adicionales asociados a los actuadores exceptuando el de la alarma luminosa), el sistema es diagnosticable con una IPN de 32 lugares y 31 transiciones.

Tabla C.7: Señales In/Out de la Planta Clasificadora de Piezas

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	C	Cinta transportadora	s1	s1	Sensor tuerca
a2	P1	Señal pistón 1	s2	s2	Sensor tornillo
a3	P2	Señal pistón 2	s3	s3	Sensor llave inglesa
a4	P3	Señal pistón 3	s4	s4	Sensor martillo
a5	P4	Señal pistón 4	s5	s5	Sensor alicate
a6	P5	Señal pistón 5	-	-	-
a7	A	Señal alarma	-	-	-

C.8. Fabricación de Lapices

Se presenta un sistema para la fabricación de lápices por medio de un grupo de procesos (Figura C.8). Para el inicio del ciclo se precisa de una preforma laminar de madera de unas medidas $18 \times 7 \times 0.5 \text{ cm}^3$, el operario coloca la tablilla de madera en el sitio correspondiente, y pulsando el botón de marcha (M) el primer sensor capta la entrada de las tablillas (s1), activando la cinta (C1) y el pistón (P1) durante 0,5 segundos. Este proceso lleva las tablillas hasta la cinta, que se mantiene puesto que es monoestable hasta llegar al sensor de activación de corte (s2). Después de que el sensor de activación de corte capte las tablillas, se activa la sierra (Sr) hasta llegar al sensor de finalización de mecanizado de las tablillas (s3). Se mantiene la cinta otra vez y continua hasta llegar al sensor (s4) que activa la encoladora (Pg) que inserta cola en las ranuras creadas por la sierra, activándose hasta el sensor de finalización de la encoladora (s5) y continua manteniéndose la cinta durante todo el proceso hasta llegar al sensor para el siguiente pistón (s6).

Una vez se activa el sensor que la llegada de las tablillas (s6), se activa el pistón (P2) y la segunda cinta (C2) durante 0,5 segundos. Este proceso lleva las tablillas hasta la cinta, que se mantiene puesto que es monoestable hasta llegar al sensor de activación del siguiente pistón separador (s7). Después de que se el sensor de activación de la separadora capte las tablillas, se activa el siguiente pistón (P3) y mediante el uso de un contador y un sensor de posición (s8), separa las tablillas pares de las impares, activando el pistón durante 0,5 segundos. Una vez separadas las tablillas, las que se mantienen en la cinta continúan hasta una posición determinada (s9) y posteriormente hasta unirse a las otras en el (s13). Las que se han separado activan las cintas siguientes para su transporte (C2 y C3), durante 0,5 segundos. A continuación se mantiene la tercera cinta (C3) y se activa la siguiente (C4), hasta llegar al sensor que activa la primera rueda (s10). Una vez el sensor de la primera rueda capta la llegada de las tablillas, se activa la rueda (R1) y se mantiene la cuarta cinta (C4) hasta llegar al sensor de la siguiente rueda que voltea las tablillas (s11). Una vez el sensor capte las tablillas, activa la siguiente cinta (C5), la siguiente rueda (R2) y mantiene la cinta anterior (C4), hasta llegar al sensor de activación del siguiente pistón (s12).

Después de que se el sensor de activación del pistón capte las tablillas, se activa el cuarto pistón (P4), y la última cinta (C6) durante 0.5 segundos. A continuación se mantiene la cinta (C6) hasta llegar al siguiente sensor (s13), que capta el momento exacto donde las tablillas separadas se vuelven a unir con las que se habían mantenido. Una vez se

haya cumplido se llega al final del primer proceso, un recipiente donde se depositan las tablillas, con un sensor de posición (s14) se activa el ultimo pistón (P5) que se encarga de comprimir las tablillas durante media hora hasta el secado del pegamento, un sensor de presión (s15) indica que se tiene la presión adecuada en los lapices y activa una luz de emergencia (A) para retirar el recipiente lleno y una vez retirado se vuelve a reiniciar el primer proceso.

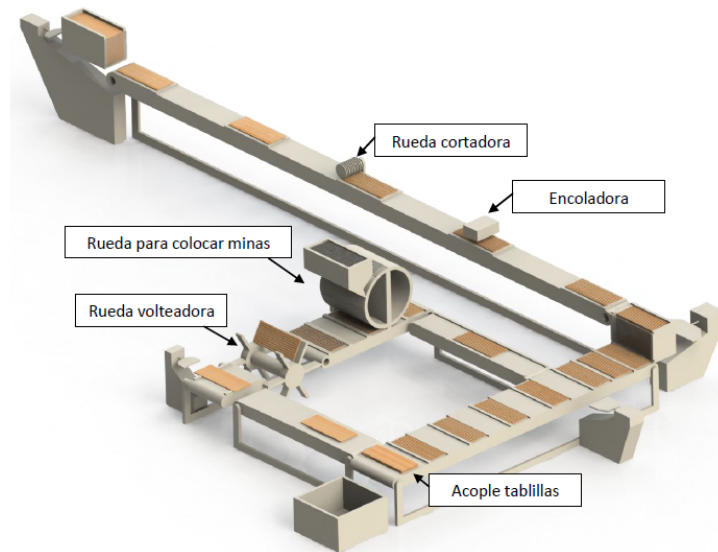


Figura C.8: Fabricación de Lapices

Se realizó la primera prueba de diagnosticabilidad con los datos iniciales del proceso para un ciclo de operación (15 señales de sensor y 16 señales de actuador)(Tabla C.8), obteniendo que el sistema no era diagnosticable, con una IPN de 31 lugares y 30 transiciones. En la segunda prueba con un máximo de 31 señales de sensor (15 iniciales y 16 sensores adicionales asociados a los actuadores), el sistema se volvió diagnosticable y se mantuvo la misma cantidad de lugares y transiciones. En la última prueba se aplicó sobre el sistema inicial el método de ubicación de sensores con un total de 18 sensores (15 iniciales, un sensor asociado a C1, P5 y A), el sistema es diagnosticable con una IPN de 31 lugares y 30 transiciones.

Tabla C.8: Señales In/Out de la Fabricación de Lapices

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	C1	Cinta transportadora 1	s1	s1	Sensor posición 1
a2	C2	Cinta transportadora 2	s2	s2	Sensor posición 2
a3	C3	Cinta transportadora 3	s3	s3	Sensor posición 3
a4	C4	Cinta transportadora 4	s4	s4	Sensor posición 4
a5	C5	Cinta transportadora 5	s5	s5	Sensor posición 5
a6	C6	Cinta transportadora 6	s6	s6	Sensor posición 6
a7	P1	Señal pistón 1	s7	s7	Sensor posición 7
a8	P2	Señal pistón 1	s8	s8	Sensor posición 8
a9	P3	Señal pistón 1	s9	s9	Sensor posición 9
a10	P4	Señal pistón 1	s10	s10	Sensor posición 10
a11	P5	Señal pistón 1	s11	s11	Sensor posición 11
a12	Pg	Señal pegamento	s12	s12	Sensor posición 12
a13	Sr	Señal sierra	s13	s13	Sensor posición 13
a14	R1	Señal rueda 1	s14	s14	Sensor posición 14
a15	R2	Señal rueda 1	s15	s15	Sensor posición 15
a16	A	Señal alarma	-	-	-

C.9. Control de Trafico Vial

Se describe una zona de baja circulación automatizada con un cruce entre 2 rutas A y B, después un otro cruce entre la ruta A y una vía férrea (Figura C.9). En la ruta principal A, antes del primer cruce entre las dos rutas, y después del cruce con la vía férrea, hay dos paneles de señalización digital que pueden mostrar las palabras GOOD o TOO FAST, según la velocidad del coche. En el primer cruce todos los semáforos son automatizados y cambian de color según la presencia de los coches, para hacer una regulación inteligente del tráfico. En el segundo cruce, con la vía férrea, hay un dispositivo de seguridad para alertar los coches de la llegada del tren. Todos los actuadores son monoestables. Se asume que esta zona es muy poco frecuentada, y que nunca hay un coche detrás de otro, y nunca hay un coche por cada lado de una ruta al mismo tiempo. También se asume que cuando un captador de presencia de coche se activa, queda activado durante 10 segundos, y cuando un captador de presencia de tren se activa, queda activado durante 60 segundos (se supone que cuando un tren esta detectado necesita 40 segundos para exceder la ruta A). Este sistema no trata del procedimiento de emergencia o de los diferentes problemas que pueden ocurrir, por ejemplo una avería de un coche en frente de un captador, o una bombilla de semáforo que deja de funcionar. El sistema tiene tres subsistemas principales:

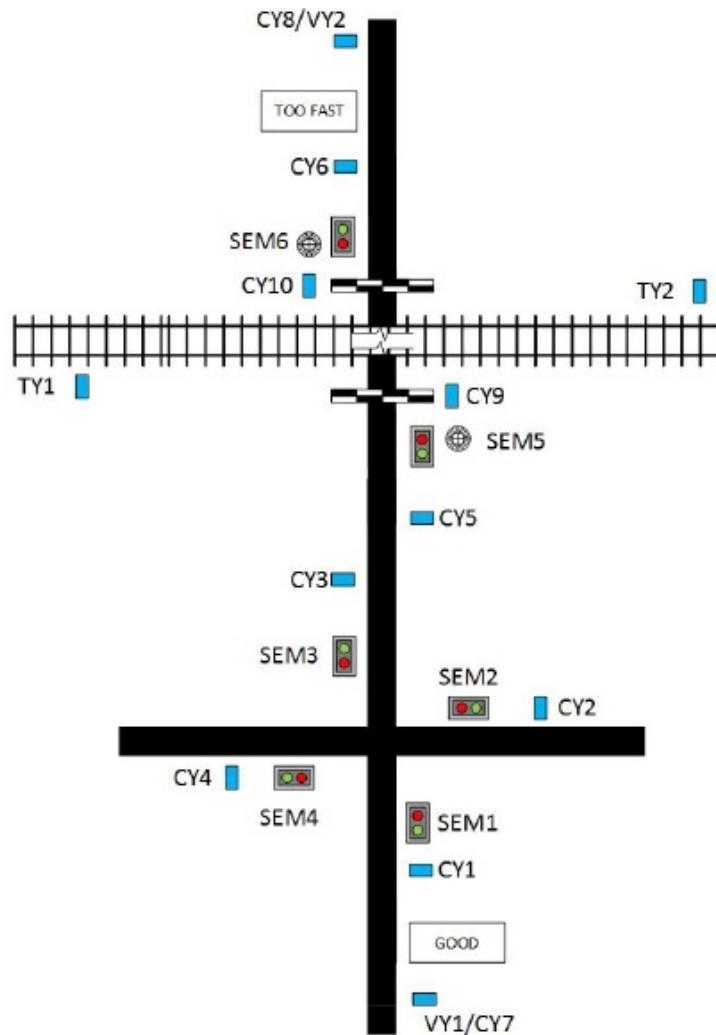


Figura C.9: Control de Trafico Vial

El primer subsistema se encarga del exceso de velocidad en la ruta A, antes del primer cruce, y en el otro lado de la vía férrea, cuando un coche avance hasta la vía férrea. Por cada lado, tenemos un captador de velocidad de coche, y un panel de visualización digital que puede mostrar GOOD o TOO FAST, los posibles casos son:

- Antes del cruce de las rutas A y B, si se detecta un coche (cy7), cuando el captador indica que la velocidad del coche es menor o igual a 50 km/h (no vy1), el panel muestra la señalización GOOD durante 5 segundos ($T_{11}=1$ y $T_{12}=0$).
- Antes del cruce de las rutas A y B, si se detecta un coche (cy7), cuando el captador indica que la velocidad del coche es más de 50 km/h (vy11), el panel muestra la señalización TOO FAST durante 5 segundos ($T_{11}=0$ y $T_{12}=1$).

- En el otro lado de la vía férrea, si se detecta un coche (cy8), cuando el captador indica que la velocidad del coche es menor o igual a 50 km/h (no vy2), el panel muestra la señalización GOOD durante 5 segundos (T21=1 y T22=0).
- En el otro lado de la vía férrea, si se detecta un coche (cy8), cuando el captador indica que la velocidad del coche es más de 50 km/h (vy2), el panel muestra la señalización TOO FAST durante 5 segundos (T21=1 y T22=0).

El segundo subsistema concierne el cruce entre las rutas A y B. Tenemos 4 semáforos en este cruce, que pueden estar verdes, o rojos, y por defecto son de color rojo. También hay 4 captadores en el cruce, por cada lado de las dos rutas, los casos que se evalúan son:

- En la ruta A, cuando un captador detecta la presencia de un coche (cy1 o cy3), el sistema verifica el estado de la ruta B. Si no hay un coche en los dos lados de la ruta B (no cy2 y no cy4), los semáforos de la ruta A se cambian al color verde durante 15 segundos (S1 y S3).
- En la ruta A, cuando un captador detecta la presencia de un coche (cy1 o cy3), el sistema verifica el estado de la ruta B. Si hay un coche en uno de los lados de la ruta B (cy2 o cy4), la ruta B tiene prioridad. Entonces los semáforos de la ruta A quedan rojos, el sistema espera 20 segundos y después los semáforos de la ruta A se cambian al color verde durante 15 segundos (S1 y S3).
- En la ruta A, cuando un captador detecta la presencia de un coche (cy1 o cy3), el sistema verifica el estado de la ruta B. Si los semáforos de la ruta B son verdes, el sistema espera hasta que los semáforos de la ruta B pasan al color rojo, espera 5 segundos y después los semáforos de la ruta A se cambian al color verde durante 15 segundos (S1 y S3).
- En la ruta B, cuando un captador detecta la presencia de un coche (cy2 o cy4), el sistema verifica el estado de la ruta A. Si los semáforos de la ruta A son rojos, los semáforos de la ruta B se cambian al color verde durante 15 segundos (S2 y S4).
- En la ruta B, cuando el captador detecta la presencia de un coche (cy2 o cy4), el sistema verifica el estado de la ruta A. Si los semáforos de la ruta A son verdes, el sistema espera hasta que los semáforos de la ruta A pasan al color rojo, espera 5 segundos y después los semáforos de la ruta B se cambian al color verde durante 15 segundos (S2 y S4).

El tercer subsistema controla el cruce entre la ruta A y la vía férrea. En este cruce, tenemos 2 semáforos, uno de cada lado de la vía, que pueden estar en rojo, verde, o apagados; por defecto están apagados. También hay dos barreras, una en cada lado de la vía, que están cerradas por defecto, y dos dispositivos de alarma audible, que están apagados por defecto. Sobre la vía férrea, a distancia de seguridad, hay un captador de presencia de tren, a la izquierda y a la derecha. Por cada lado de la vía férrea, sobre la ruta A, hay un captador de presencia de coche. Se recuerda que los captadores de presencia del tren están activados por 60 segundos, las situaciones que se plantean son:

- En la ruta A, cuando un captador detecta la presencia de un coche (cy5 o cy6), el sistema verifica el estado de la vía férrea. Si no hay un tren a la izquierda y a la derecha (no ty1 y no ty2) de la vía, las barreras se elevan (b1 y b2) y después los semáforos se cambian al color verde durante 15 segundos (sv1 y sv2). Después de esos 15 segundos, y si el captador no detecta un coche debajo de la barrera (no cy9 y no cy10), las barreras bajan.
- En la ruta A, cuando un captador detecta la presencia de un coche (cy5 o cy6), el sistema verifica el estado de la vía férrea. Si un tren ha sido detectado a la izquierda o a la derecha (ty1 o ty2) de la vía, las barreras quedan abajo (no b1 y no b2), los semáforos se cambian al color rojo (sr1 y sr2) y las alarmas audibles se encienden (a1 y a2). El sistema queda así hasta que el sensor de presencia de tren se desactiva (no ty1 y no ty2), y en este momento las alarmas audibles y los semáforos se apagan, las barreras se elevan (b1 y b2) y después los semáforos se cambian al color verde durante 15 segundos (sv1 y sv2). Después de esos 15 segundos, y si el captador no detecta un coche debajo de la barrera (no cy9 y no cy10), las barreras bajan.

Con los datos iniciales del proceso para tres ciclos de operación (12 señales de sensor y 16 señales de actuador) (Tabla C.9), se ejecutó la primera prueba de diagnosticabilidad obteniendo que el sistema no era diagnosticable, con una IPN de 30 lugares y 29 transiciones. En la segunda prueba con un máximo de 28 señales de sensor (12 iniciales y 16 sensores adicionales asociados a los actuadores), el sistema se volvió diagnosticable y se mantuvo la misma cantidad de lugares y transiciones. En la última prueba se aplicó sobre el sistema inicial el método de ubicación de sensores con un total de 18 sensores (12 iniciales junto con los sensores asociados a sr1, sv1, sr2, sv2, T11 y T12), el sistema es diagnosticable con una IPN de 30 lugares y 29 transiciones.

Tabla C.9: Señales In/Out del Control de Trafico Vial

Señales de Actuador			Señales de Sensor		
IPN	Planta	Nombre	IPN	Planta	Nombre
a1	S1	Semaforo 1	s1	cy1	Sensor posición 1
a2	S2	Semaforo 2	s2	cy2	Sensor posición 2
a3	S3	Semaforo 3	s3	cy3	Sensor posición 3
a4	S4	Semaforo 4	s4	cy4	Sensor posición 4
a5	sr1	Señal rojo semaforo tren 1	s5	cy5	Sensor posición 5
a6	sv1	Señal verde semaforo tren 1	s6	cy6	Sensor posición 6
a7	sr2	Señal rojo semaforo tren 2	s7	cy7	Sensor posición 7
a8	sv2	Señal verde semaforo tren 2	s8	cy8	Sensor posición 8
a9	a1	Señal alarma 1	s9	vy1	Sensor velocidad 1
a10	a2	Señal alarma 2	s10	vy2	Sensor velocidad 2
a11	b1	Señal barrera 1	s11	ty1	Sensor posición tren 1
a12	b2	Señal barrera 2	s12	ty2	Sensor posición tren 2
a13	T11	Señal 1 tablero 1	-	-	-
a14	T12	Señal 2 tablero 1	-	-	-
a15	T21	Señal 1 tablero 2	-	-	-
a16	T22	Señal 2 tablero 2	-	-	-

Bibliografía

- [1] D. D. Teixeira, F. J. A. Pires, J. P. G. de Sousa, and T. A. G. P. S. Pinto, “DSDM-Dynamic Systems Development Methodology,” *Faculdade de Engenharia da Universidade do Porto*, vol. 27, pp. 5–9, 2005.
- [2] S. Matsumura, K. Ogata, S. Fujii, H. Shioya, and H. Nakamura, “Adaptive control for the steam temperature of thermal power plants,” *Control Engineering Practice*, vol. 2, no. 4, pp. 567–575, 1994.
- [3] L. Zhao, B. Zhang, H. Yang, and Q. Li, “Optimized linear active disturbance rejection control for pneumatic servo systems via least squares support vector machine,” *Neurocomputing*, vol. 242, pp. 178–186, 2017.
- [4] M. Pertl, T. Weckesser, M. Rezkalla, K. Heussen, and M. Marinelli, “A decision support tool for transient stability preventive control,” *Electric Power Systems Research*, vol. 147, pp. 88–96, 2017.
- [5] R. Jacob, J.-J. Lesage, and J.-M. Faure, “Overview of discrete event systems opacity: Models, validation, and quantification,” *Annual Reviews in Control*, vol. 41, pp. 135–146, 2016.
- [6] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis, “Failure diagnosis using discrete-event models,” *IEEE transactions on control systems technology*, vol. 4, no. 2, pp. 105–124, 1996.
- [7] F. Schumacher and A. Fay, “Formal representation of GRAFCET to automatically generate control code,” *Control Engineering Practice*, vol. 33, pp. 84–93, 2014.
- [8] B. Liu and A. Verbraeck, “Multi-resolution modeling based on quotient space and DEVS,” *Simulation Modelling Practice and Theory*, vol. 70, pp. 36–51, 2017.

- [9] A. Boussif and M. Ghazel, “Diagnosability Analysis of Input/Output Discrete-Event Systems Using Model-Checking,” *IFAC-PapersOnLine*, vol. 48, no. 7, pp. 71–78, 2015.
- [10] S. Reshmila and R. Devanathan, “Diagnosis of power system failures using observer based discrete event system,” in *Control, Measurement and Instrumentation (CMI), 2016 IEEE First International Conference on.* IEEE, 2016, pp. 131–135.
- [11] S. Haar, S. Haddad, T. Melliti, and S. Schwoon, “Optimal constructions for active diagnosis,” *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 101–120, 2017.
- [12] M. P. Cabasino, S. Lafortune, and C. Seatzu, “Optimal sensor selection for ensuring diagnosability in labeled Petri nets,” *Automatica*, vol. 49, no. 8, pp. 2373–2383, 2013.
- [13] A. G. Anbulagan, “Incremental Diagnosis of DES with a Non-Exhaustive Diagnosis Engine,” *International Workshop on Principles of Diagnosis (DX)*, 2009.
- [14] D. Han, J. Wu, H. Zhang, and L. Shi, “Optimal sensor scheduling for multiple linear dynamical systems,” *Automatica*, vol. 75, pp. 260–270, 2017.
- [15] W. Wang, S. Lafortune, A. R. Girard, and F. Lin, “Optimal sensor activation for diagnosing discrete event systems,” *Automatica*, vol. 46, no. 7, pp. 1165–1175, 2010.
- [16] G. S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, “Empirical study of agile software development methodologies: A comparative analysis,” *ACM SIGSOFT Software Engineering Notes*, vol. 40, no. 1, pp. 1–6, 2015.
- [17] P. Laszczyk, “Simplified modeling of liquid-liquid heat exchangers for use in control systems,” *Applied Thermal Engineering*, vol. 119, pp. 140–155, 2017.
- [18] Z.-J. Fu, W.-F. Xie, S. Rakheja, and D.-D. Zheng, “Adaptive optimal control of unknown nonlinear systems with different time scales,” *Neurocomputing*, vol. 238, pp. 179–190, 2017.
- [19] M. Šnipas, V. Radziukynas, and E. Valakevičius, “Modeling reliability of power systems substations by using stochastic automata networks,” *Reliability Engineering & System Safety*, vol. 157, pp. 13–22, 2017.
- [20] J.-H. Zhang, J.-J. Xia, J. M. Garibaldi, P. P. Groumpos, and R.-B. Wang, “Modeling and control of operator functional state in a unified framework of fuzzy inference petri nets,” *Computer Methods and Programs in Biomedicine*, vol. 144, pp. 147–163, 2017.

- [21] A. Ramírez-Treviño, E. Ruiz-Beltran, J. Aramburo-Lizarraga, and E. Lopez-Mellado, “Structural diagnosability of DES and design of reduced Petri net diagnosers,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 2, pp. 416–429, 2012.
- [22] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer Science & Business Media, 2009.
- [23] S. Reshmila and R. Devanathan, “Modeling a system using observations in discrete event system for failure diagnosis,” in *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in*. IEEE, 2015, pp. 280–284.
- [24] J. Zaytoon and M. Sayed-Mouchaweh, “Discussion on fault diagnosis methods of discrete event systems,” *IFAC Proceedings Volumes*, vol. 45, no. 29, pp. 9–12, 2012.
- [25] X. Yin and S. Lafortune, “On the decidability and complexity of diagnosability for labeled Petri nets,” *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5931–5938, 2017.
- [26] M. P. Cabasino, A. Giua, and C. Seatzu, “Fault detection for discrete event systems using Petri nets with unobservable transitions,” *Automatica*, vol. 46, no. 9, pp. 1531–1539, 2010.
- [27] F. Basile, P. Chiacchio, and G. De Tommasi, “On K-diagnosability of Petri nets via integer linear programming,” *Automatica*, vol. 48, no. 9, pp. 2047–2058, 2012.
- [28] S. Lafortune, F. Lin, and C. N. Hadjicostis, “On the history of diagnosability and opacity in discrete event systems,” *Annual Reviews in Control*, 2018.
- [29] J. Zaytoon and S. Lafortune, “Overview of fault diagnosis methods for discrete event systems,” *Annual Reviews in Control*, vol. 37, no. 2, pp. 308–320, 2013.
- [30] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System dynamics: modeling, simulation, and control of mechatronic systems*. John Wiley & Sons, 2012.
- [31] L. Ljung, “System identification,” in *Signal analysis and prediction*. Springer, 1998, pp. 163–173.
- [32] A. P. Estrada-Vargas, E. López-Mellado, and J.-J. Lesage, “Identification of partially observable discrete event manufacturing systems,” in *Emerging Technologies & Factory Automation (ETF A), 2013 IEEE 18th Conference on*. IEEE, 2013, pp. 1–7.

- [33] A. P. Estrada-vargas, E. López-Mellado, and J.-j. Lesage, “A Black-box Identification Method for Automated Discrete Event Systems To cite this version : A Black-box Identification Method for Automated Discrete Event Systems,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, pp. 1321–1336, 2016.
- [34] A. Ramírez-Treviño, E. Ruiz-Beltrán, I. Rivera-Rangel, and E. Lopez-Mellado, “On-line 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.
- [35] D. M. Muñoz, A. Correcher, E. García, and F. Morant, “Generación Determinística de Lenguajes Legales para Sistemas de Eventos Discretos,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 13, no. 2, pp. 207–219, 2016.
- [36] L. Aguirre-Salas, “Sensor selection for observability in interpreted Petri nets: a genetic approach,” in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 4. IEEE, 2003, pp. 3760–3765.
- [37] K. Hernández-Rueda, M. E. Meda-Campaña, and J. Arámburo-Lizárraga, “Enforcing Diagnosability in Interpreted Petri Nets,” *IFAC-PapersOnLine*, vol. 48, no. 7, pp. 58–63, 2015.
- [38] A. Cockburn and J. Highsmith, “Agile software development, the people factor,” *Computer*, vol. 34, no. 11, pp. 131–133, 2001.
- [39] H.-J. Bullinger, K.-P. Fähnrich, and T. Meiren, “Service engineering–methodical development of new service products,” *International Journal of Production Economics*, vol. 85, no. 3, pp. 275–287, 2003.
- [40] J. Highsmith and A. Cockburn, “Agile software development: The business of innovation,” *Computer*, vol. 34, no. 9, pp. 120–127, 2001.
- [41] M. Brhel, H. Meth, A. Maedche, and K. Werder, “Exploring principles of user-centered agile software development: A literature review,” *Information and Software Technology*, vol. 61, pp. 163–181, 2015.
- [42] T. Dingsøyr and C. Lassenius, “Emerging themes in agile software development: Introduction to the special section on continuous value delivery,” *Information and Software Technology*, vol. 77, pp. 56–60, 2016.

- [43] L. R. Vijayasarathy and C. W. Butler, “Choice of software development methodologies: Do organizational, project, and team characteristics matter?” *IEEE Software*, vol. 33, no. 5, pp. 86–94, 2016.
- [44] Y. Lindsjörn, D. I. K. Sjøberg, T. Dingsøy, G. R. Bergersen, and T. Dybå, “Teamwork quality and project success in software development: A survey of agile development teams,” *Journal of Systems and Software*, vol. 122, pp. 274–286, 2016.
- [45] G. S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, “Empirical study of agile software development methodologies: A comparative analysis,” *ACM SIGSOFT Software Engineering Notes*, vol. 40, no. 1, pp. 1–6, 2015.
- [46] F. S. Silva, F. S. F. Soares, A. L. Peres, I. M. de Azevedo, A. P. L. F. Vasconcelos, F. K. Kamei, and S. R. de Lemos Meira, “Using CMMI together with agile software development: A systematic review,” *Information and Software Technology*, vol. 58, pp. 20–43, 2015.
- [47] S. Nerur, R. Mahapatra, and G. Mangalaraj, “Challenges of migrating to agile methodologies,” *Communications of the ACM*, vol. 48, no. 5, pp. 72–78, 2005.
- [48] C. P. Lo and A. K. W. Yeung, *Concepts and techniques of geographic information systems (2eds)*. Upper Saddle River (NJ): Prentice-Hall, Inc, 2007.
- [49] R. Baxter, “Software engineering is software engineering,” in *26th International Conference on Software Engineering, W36 Workshop Software Engineering for High Performance System (HPCS) Applications*. IET, 2004, pp. 4–18.
- [50] P. Tiernan, “Enhancing the learning experience of undergraduate technology students with LabVIEW software,” *Computers & Education*, vol. 55, no. 4, pp. 79–88, 2010.
- [51] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu, “Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems,” *Control Engineering Practice*, vol. 19, no. 9, pp. 989–1001, 2011.
- [52] E. R. Gansner, “Drawing graphs with Graphviz,” *Technical report, AT&T Bell Laboratories, Murray, Tech. Rep, Tech. Rep.*, 2009.
- [53] H. J. Martinez and A. M. Sanabria, *Algebra lineal*. Universidad del Valle, 2010.