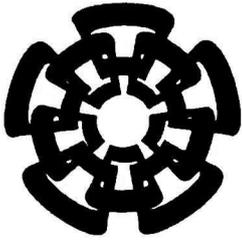


BC-653

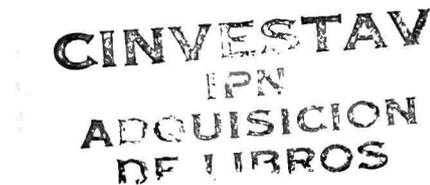
Don. - 2011

XX(179124.1)



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

Diagnóstico de Faltas Intermitentes para Sistemas de Eventos Discretos



Tesis que presenta:

Alvaro Ocegüera Valenzuela

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. Antonio Ramírez Treviño

Dr. José Javier Ruíz León

CLASIF: 7K165.68 024 2011
ADQUIS: 551-653
FECHA: 18 Agosto 2011
PROCED: Don. - 2011
\$

10.174542-1001

Diagnóstico de Faltas Intermitentes para Sistemas de Eventos Discretos

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Alvaro Ocegüera Valenzuela
Ingeniero Electrónica

Instituto Tecnológico de Querétaro 1999-2003

Becario de conacyt, expediente no. 203119

Directores de Tesis

Dr. Antonio Ramírez Treviño

Dr. José Javier Ruíz León

CINVESTAV del IPN Unidad Guadalajara, Febrero de 2011.

A mis padres, Sergio y a Yadira, a mi familia en Guadalajara, por estar a mi lado, por su apoyo, por su amor, por su comprensión.

Mi más sincero reconocimiento a mi asesor Antonio Ramírez quien me ayudo por todo este proceso de desarrollar mi tesis, desde que me invito a tomar la oportunidad que se presento hace tres años y por permitirme continuar y finalizar esta tesis.

A ti Ely, gracias por tu apoyo, tu ayuda, por tus consejos y tu motivación, gracias por estar a mi lado.

Al CONACYT por su apoyo económico y al CINVESTAV por todo el apoyo que me brindo para la realización de este proyecto de graduación.

Resumen

La detección e identificación de las faltas es de gran importancia en los sistemas tecnológicos porque ayudan a crear un sistema robusto, mejorando su rendimiento, causando clientes satisfechos como en el caso de un sistema de videojuegos, que no se reinicia mientras el usuario está jugando, o porque juegan papeles importantes como salvar vidas considerando el caso de una planta nuclear.

Debido a la importancia de este tema existen muchas metodologías para la detección de faltas y para el diseño de diagnosticadores. Estas metodologías son basadas en el modelado de sistemas utilizando autómatas y algunas otras en el uso de redes de petri. En este trabajo se utilizan redes de petri interpretadas en lugar de autómatas debido a su inherente doble representación (gráfica y matemática) que ofrece ventajas sobre el uso de autómatas para la descripción de sistemas de eventos discretos, además de reducir la complejidad computacional al resolver problemas de diagnóstico gracias a la noción de estado y acción.

El objetivo de este trabajo es relajar la condición de evento-detectabilidad para diagnosticabilidad de entrada-salida para la detección de faltas intermitentes en redes de Petri interpretadas. Este trabajo presenta un diagnosticador que detecta faltas intermitentes en sistemas que no son evento-detectable. Las restricciones del diagnosticador presentado en este trabajo se encuentran en el modelado del sistema, consecuentemente en el tipo de sistema que se está analizando.

El diagnosticador presentado aquí no necesita que todas las transiciones sean evento-detectable sino que las transiciones que son saltadas y que algunas de las transiciones que interactúan con las transiciones saltadas sean evento-detectable. El diagnosticador presentado aquí es una copia del sistema analizado más un nuevo lugar que interactúa con las transiciones evento-detectable, creando el diagnosticador y encargándose de detectar las faltas ocurridas en el sistema.

Por último el diagnosticador presentado en este trabajo permite la detección de las faltas la primera vez que ocurren, o cuenta también con la habilidad de contar faltas y reportar la falta hasta que un cierto número de ellas han pasado, funcionalidad deseada en sistemas tolerantes a faltas.

Abstract

Detection and identification of faults are topics of great importance in technological systems because the help that they can offer in the creation of robust systems, improving their performance, causing customer satisfaction as in a videogame system, that will not reset while the user is playing, or because they can play an important role in saving lives as in the case of a nuclear plant.

Based on the importance of this topic there exist several methodologies for detecting faults and for the design of diagnosers. These methodologies are based on modeling systems using automatas and some others in the use of Petri nets. This work uses interpreted Petri nets instead of automatas because of their inherent double representation (graphical and mathematical) that offers advantages over the use of automatas for describing discrete event systems, plus their help to reduce the mathematical complexity when solving diagnostics problems thanks to their notion of state and action.

The objective of this work is to relax the condition of event-detectability for input-output diagnosticability for the detection of intermittent faults on interpreted Petri nets. This work presents a diagnoser that detects intermittent faults in systems that are not event-detectable. The restrictions of this diagnoser are found in the model of the system, consequently in the type of systems that can be diagnosed.

The diagnoser presented in this work does not need to have all the transitions event-detectable but that the transitions that are jumped and some of the transitions that interact with these jump transitions are event-detectable. The diagnoser presented here is a copy of the system that is under analysis plus a new place that interacts with the event-detectable transitions, creating the diagnoser and taking care of the faults that occur in the system.

Finally, the diagnoser presented in this work allows the detection of faults the first time that they occur, or it also has the ability to count faults and report the fault until a certain number of them has occurred, a wish to have functionality in fault tolerant systems.

Contenido

1. Introducción	1
1.1. Sistemas de Eventos Discretos	2
1.2. Diagnóstico de Faltas	3
1.3. Estado del Arte para Diagnóstico de Faltas	5
1.4. Descripción del Problema	9
1.5. Panorama General de la Tesis	9
2. Redes de Petri	10
2.1. Redes de Petri	10
2.2. Redes de Petri Interpretadas	15
3. Diagnosticador de Faltas Intermitentes	19
3.1. Diagnosticabilidad de Faltas Intermitentes	20
3.2. Diseño de un Diagnosticador de Faltas Intermitentes	27
4. Casos de Estudio	30
4.1. Caso de Estudio 1	30
4.2. Caso de Estudio 2	38
4.3. Caso de Estudio 3	46
5. Conclusiones y Trabajo Futuro	56
Referencias	58
Apéndice A	60

Lista de Figuras

<u>Figura 1.1. Diagnóstico de Faltas</u>	6
<u>Figura 2.1. Red de Petri</u>	12
<u>Figura 2.2. Red de Petri Interpretada</u>	17
<u>Figura 3.1. Diagrama de Bloques del Diagnosticador</u>	20
<u>Figura 3.2. Transiciones de Falta saltando únicamente una transición</u>	22
<u>Figura 3.3. Transiciones saltadas que son Evento-detectable</u>	23
<u>Figura 3.4. Transiciones no saltadas que son Evento-detectable</u>	23
<u>Figura 3.5. Ejemplo de una RPI</u>	24
<u>Figura 4.1. Línea de Tecnología de Montaje Superficial (SMT)</u>	28
<u>Figura 4.2. Modelo de la línea SMT utilizando RPI</u>	29
<u>Figura 4.3. Modelo de la línea SMT en RPI más la transición de falta</u>	29
<u>Figura 4.4. Diagnosticador para SMT</u>	33
<u>Figura 4.5. Diagnosticador para detectar faltas la segunda vez que ocurran</u>	33
<u>Figura 4.6. Sistema de comunicación entre servidores</u>	34
<u>Figura 4.7. Modelo del Sistema de comunicación entre servidores</u>	35
<u>Figura 4.8. Modelo del Sistema de comunicación entre servidores más la falta</u>	35
<u>Figura 4.9. Diagnosticador para el Sistema de comunicación entre servidores</u>	40
<u>Figura 4.10. Sistema de Nivel de Líquido[25]</u>	41
<u>Figura 4.11. Componentes en el sistema del Tanque</u>	42
<u>Figura 4.12. Modelo del Tanque</u>	42
<u>Figura 4.13. RPI modelo del Tanque junto con la transición de falta</u>	43
<u>Figura 4.14. Diagnosticador del Sistema del Tanque</u>	48
<u>Figura A.1. Sistema para encontrar las transiciones saltadas.</u>	60

Capítulo 1

Introducción

En estos tiempos con todo el desarrollo de tecnología que existe en el mundo, los sistemas tecnológicos se están haciendo cada vez más complejos. Un teléfono celular, por ejemplo, solía ser un aparato donde las únicas tareas que desempeñaba eran el hacer y recibir llamadas. Hoy en día los teléfonos celulares pueden tomar fotografías, llevar una agenda, alarmas, recordatorios de fechas, navegar por Internet, reproducir música, etc. Todas estas tareas hacen al sistema más complejo y difícil de desarrollar para los diseñadores.

En sistemas como éstos, las faltas que se presentan pueden ser molestas para los usuarios y por consiguiente pueden representar pérdidas en los ingresos de las compañías que los desarrollan. Un caso diferente de complejidad se presenta en las plantas nucleares, en éstas, cientos de procesos necesitan ser ejecutados sin demora, porque cuando existe alguna falta en el sistema, se puede dejar sin electricidad a una ciudad completa o aún peor, si ocurre una fuga en la planta, la vida de los trabajadores y de los habitantes de las comunidades cercanas a la planta pueden correr peligro por estar expuestos a la radiación.

Debido a la complejidad de esta clase de sistemas, es una gran ventaja para los diseñadores el contar con una herramienta para detectar las faltas que ocurren en ellos. Idealmente, en la etapa de validación de los sistemas es donde se deberían detectar todos los errores que en el sistema puedan ocasionar faltas. Con este propósito existen diferentes técnicas para la validación de sistemas, como validación de caja negra, caja blanca o la inyección de faltas [18], que intentan validar la mayor parte del sistema para dejarlo libre de errores, aunque en la práctica no ocurra así. Aún con todas éstas técnicas de validación es imposible validar completamente a los sistemas, dando como resultado

que estos puedan tener errores que a su vez ocasionen faltas. Una solución a éste problema es que los sistemas puedan auto detectar sus faltas, entonces se pueden diseñar acciones de contingencia para tratar de mantener a los sistemas lo más cerca posible de su comportamiento requerido. Por ejemplo, si una falta ocurre cuando el teléfono celular está recibiendo un mensaje de texto, será mejor para el usuario que el mensaje llegue tarde debido a una de las acciones de contingencia o de manejo de faltas, a que el mensaje no llegue porque la falta no fue detectada.

Entonces, las acciones de contingencia en el caso de los teléfonos celulares hacen que el usuario esté más cómodo con el sistema, ocasionando que recomiende a la compañía o que el próximo teléfono celular que compre pertenezca a la misma compañía, incrementando así las ventas de la empresa de teléfonos celulares. Por otro lado, si una falta ocurre en la planta nuclear y el sistema puede detectarla, entonces acciones de contingencia como aislamiento, reparación o simplemente detener los procesos peligrosos pueden ser la diferencia entre la vida y la muerte.

Éstas son algunas de las razones por las cuales muchos investigadores se encuentran trabajando en el área de diagnóstico de faltas y en el área de identificación de faltas; y es ésta también la razón por la cual este tema fue escogido para este trabajo de tesis.

1.1 Sistemas de Eventos Discretos (SED)

Tomando en cuenta nuevamente al teléfono celular, es posible modelar el sistema a grandes rasgos con únicamente dos estados: *encendido* y *apagado*, {*encendido*, *apagado*}, lo que significa que el teléfono celular puede estar encendido y listo para realizar cualquiera de las tareas que tiene programadas, o puede estar apagado y ninguna de esas tareas se podrá realizar. Por otro lado, este mismo teléfono celular puede ser modelado con un conjunto diferente de estados: {*en espera*, *llamando*, *escribiendo*, *reproduciendo*, *buscando*, *descompuesto*}, haciendo el modelado del teléfono celular más descriptivo con respecto a las tareas que puede realizar, y no solamente encendido y apagado como se modelo al principio.

El teléfono celular podrá entonces, estar en cualquiera de los estados modelados, pero nunca en dos estados al mismo tiempo, por ejemplo, en el estado de *en espera* el teléfono celular está encendido, pero el teléfono celular no está realizando ninguna actividad, excepto monitoreando las demás tareas que quieran y estén listas para ejecutarse. Es el estado donde el teléfono se encontrará la mayor parte del tiempo, esperando una llamada, un mensaje, etc. Por el estado *llamando* se entiende el estado donde el usuario se encuentra llamando por teléfono, ya sea porque una llamada fue recibida o porque el usuario realizó una llamada. Por el estado *escribiendo* se entiende que el teléfono celular se encuentra en el estado donde el usuario está escribiendo un mensaje de texto para enviarlo a otro teléfono celular. El estado de *reproduciendo* es el estado en el que el teléfono celular se encuentra reproduciendo canciones. En el estado *buscando* el teléfono celular se encuentra buscando una red para conectarse y finalmente

en el estado *descompuesto*, el teléfono celular está descompuesto y no puede repararse por él mismo.

Si el teléfono celular es modelado a través de estados, donde para cambiar de estados se necesita alguna acción externa y asíncrona, entonces el teléfono puede considerarse un Sistema de Eventos Discretos (SED).

Los Sistemas de Eventos Discretos (SED) son sistemas dinámicos donde el espacio de estados es numerable, aunque puede ser infinito, y donde los estados generalmente cambian asíncronicamente en respuesta a eventos [12, 27]. Ejemplos de SED son sistemas de redes, en la aeronáutica, manufactura y procesos industriales [17]. En la actualidad mucha investigación se está llevando a cabo en diferentes áreas de SED como son el modelado, análisis, control y diagnóstico de faltas.

Definición 1.1. *Un SED es un sistema de estados discretos dirigido por eventos, i.e. su evolución está totalmente basada en la ocurrencia de eventos asíncronos [25].*

Existen una gran cantidad de ejemplos de SED en sistemas tecnológicos, por ejemplo el teléfono celular descrito anteriormente, un sistema químico que se puede modelar también por únicamente dos estados {*encendido, apagado*}, en el cual se ve a la planta o al proceso químico como activo o como apagado o deshabilitado. Pero los SED son más útiles si modelan subsistemas porque el modelado será más descriptivo ayudando a realizar un análisis más profundo de los sistemas.

1.2 Diagnóstico de Faltas

Considerando de nuevo el teléfono celular descrito anteriormente, el software embebido en él tiene muchos estados. Usando nuevamente el mismo nivel de abstracción que se presentó para el modelado del teléfono celular, se definen los mismos estados: {*en espera, llamando, escribiendo, reproduciendo, buscando, descompuesto*}, con estos estados se puede tener la situación en la que el teléfono celular se encuentra reproduciendo música, en el estado *reproduciendo*, cuando una llamada es recibida, el comportamiento normal del teléfono sería detener el reproductor de música y cambiar al estado *llamando*, pero una falta ocurrida en este cambio de estados podría ocasionar que el teléfono reciba la llamada y al mismo tiempo se encuentre reproduciendo música (cambia de estado a *llamando* pero no se detuvo la tarea del reproductor de reproducir música). El usuario del teléfono celular estaría entonces llamando por teléfono y escuchando música al mismo tiempo, éste es un ejemplo de una falta en el sistema (el no detener al reproductor de música) que lleva a una falla en el comportamiento normal del sistema (hablar por teléfono y escuchar música).

Definición 1.2. *Las Faltas pueden ser consideradas como condiciones que causan que un sistema o dispositivo fallen, (por ejemplo, un corto circuito, una conexión intermitente, una válvula atorada, un error en el software, etc). Todas estas faltas pueden ocasionar fallas en los sistemas, i.e., la terminación de la habilidad de un sistema de realizar su funcionamiento normal [8, 25].*

Definición 1.3. Análisis de Fallas, consiste en monitorear el comportamiento del sistema, analizando la ocurrencia de cualquier falta y también identificando su tipo y su origen. Si la falta puede ser detectada prontamente y ser removida, entonces las fallas pueden ser evitadas. Entonces, la detección y diagnóstico automático de faltas es un área de investigación que recibe mucha atención en el área de SED [8].

Las faltas pueden clasificarse por el tiempo que permanecen activas como faltas permanentes, transitorias o intermitentes [3, 25].

Definición 1.4. Faltas Permanentes son las faltas que pueden ocurrir a cualquier tiempo y que no se recuperaran hasta que el sistema sea reparado. Ejemplos de estas faltas pueden ser un alambre abierto, una máquina descompuesta o un error en el software que lleve a un bloqueo del mismo.

Definición 1.5. Faltas Transitorias, son las faltas con una duración limitada de tiempo ocasionadas por un mal funcionamiento del sistema o causadas por una interferencia externa.

Definición 1.6. Faltas Intermitentes son aquellas que pueden ocurrir y desaparecer durante el funcionamiento del sistema, son las faltas que ocurren, pero que permiten que el sistema continúe con su funcionamiento. Ejemplo de estas faltas son las faltas de hardware intermitentes, que son causadas normalmente por componentes eléctricos dañados (por ejemplo, relevadores dañados), componentes "pegajosos" (por ejemplo, válvulas atascadas), sobrecalentamiento de chips, mediciones con ruido de los sensores, picos en las fuentes de alimentación, etc.

Las faltas intermitentes ocurren también en software, por ejemplo, excepciones e interrupciones que son causadas por alcanzar estados desconocidos o no modelados y que pueden ocasionar mal funcionamiento del software o hasta reinicios de sistemas [5].

Definición 1.7. Los Sistemas Tolerantes a Faltas son los sistemas que continúan con un grado de operación aún bajo la presencia de faltas, y donde de acuerdo a [25] pueden tener alguna de las siguientes tolerancias:

1. **Sistemas de Tolerancia a Faltas Completa:** estos sistemas son capaces de continuar su funcionamiento en la presencia de faltas, probablemente por un tiempo limitado, pero sin perder funcionalidad o su rendimiento.
2. **Degradación Suave.** Son sistemas que continúan su funcionamiento en la presencia de faltas, pero con una degradación parcial de su funcionamiento y rendimiento mientras la falta es reparada.
3. **Detención Segura.** Son sistemas que mantienen su integridad mientras se mantienen en un paro temporal.

Un sistema tolerante a faltas puede tener alguna de las siguientes etapas [25]:

- a) *Detección del Error*, es en esta etapa donde la detección de faltas ocurre al detectar un error en alguno de los subsistemas. Si el error fue detectado es porque hubo una falta en algún componente o en algún subsistema.
- b) *Contención del daño*, esta es la etapa en donde se realiza la identificación de la falta y la contención del daño que pudiera ocasionar.
- c) *Recuperación del Error*, el propósito de esta etapa es el de mover al sistema de un estado corrupto a otro estado no corrupto dentro del mismo sistema. En este nuevo estado el sistema puede continuar con su operación aunque posiblemente con algún grado de degradación del mismo.
- d) *Tratamiento de la falla y servicio continuo*. En esta etapa se identifica al componente fallido, después de la identificación el sistema se repara al no utilizar para nada al componente fallido, o al utilizarlo con una configuración diferente. Después que el sistema es reparado, puede continuar con su operación normal.

El diagnóstico de faltas es presentado en este trabajo, siendo parte de la etapa de Detección del Error (Detección de la falta) y de la Contención del daño (identificación de la falta).

Definición 1.8. *Un Sistema de Diagnóstico de faltas es el sistema que detectará e identificará las faltas [25].*

Tomando las definiciones 1.7 y 1.8, un sistema de diagnóstico de faltas puede incluir las siguientes etapas:

- *Detección de Faltas*, etapa donde la falta es detectada.
- *Identificación de la Falta*, etapa donde la falta es identificada.

Debido a que resolver el problema de diagnóstico de faltas intermitentes basado en el modelo es el objetivo de éste trabajo, se necesita la siguiente definición:

Definición 1.9. *Diagnóstico de faltas basado en el modelo es un sistema de diagnóstico de faltas, donde el modelado del sistema será usado para observar el comportamiento normal del sistema, y éste será comparado con el comportamiento real del sistema. La detección del error ocurrirá cuando exista una diferencia entre el comportamiento del modelo del sistema y el sistema mismo. La localización del error se realiza después de detectar la diferencia entre los sistemas, i.e. el error [25].*

1.3 Estado del Arte para Diagnóstico de Faltas

Un error que se presente en el sistema, representa la presencia de una falta y debido a que los sistemas están compuestos por subsistemas, es posible describir el problema de detección de faltas como un problema de detección de faltas a nivel subsistema. Es por esta razón que el problema de detección de faltas en SED es un problema de diagnóstico de faltas en un componente del SED [25].

La Figura 1.1 muestra que cuando el diagnóstico es hecho al sistema, también se está diagnosticando a los subsistemas. En la Figura 1.1, se muestra como un sistema está compuesto de n-subsistemas en la segunda capa.

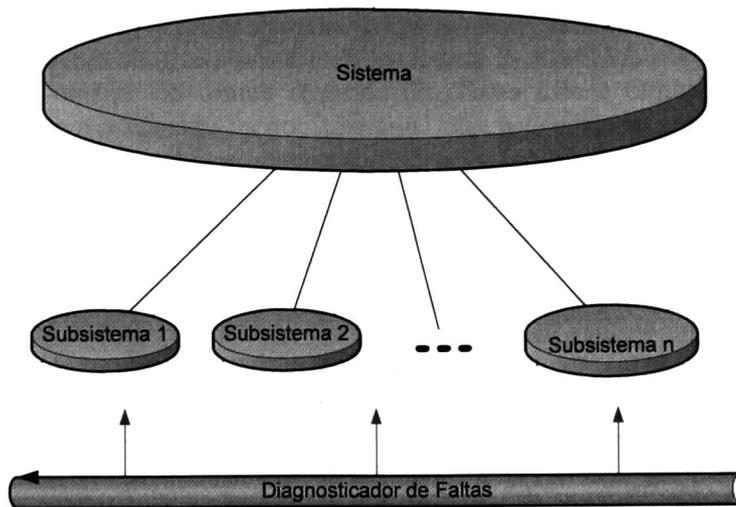


Figura 1.1 Diagnóstico de Fallas [25]

La detección de fallas y el diagnóstico de SED han sido estudiados ampliamente utilizando sistemas basados en modelos, con esta técnica el diagnóstico de fallas está basado en la especificación de la dinámica del sistema [26, 36]. Otra técnica para la detección de fallas es utilizando análisis de árboles de fallas [13], espacio de estados [15, 11] y sistemas basados en eventos [19, 26, 27].

Todas estas metodologías para la detección de fallas son basadas en el modelo de sistemas utilizando autómatas, y si para el modelo del sistema se utilizan autómatas, entonces también se utilizan autómatas para el diseño y construcción de diagnosticadores. Los diagnosticadores funcionan en paralelo con el sistema real monitoreando el comportamiento del SED con el propósito de diagnosticar la falta [8].

A diferencia de todos los artículos anteriores, las Redes de Petri (RP) son usadas en este trabajo para el modelado del sistema en lugar de autómatas. Se utilizan RP debido a su inherente doble representación (gráfica y matemática) que ofrece ventajas sobre el uso de autómatas para la descripción de SED. Otra ventaja de utilizar redes de Petri es que se reduce la complejidad computacional al resolver problemas de diagnóstico gracias a la noción de estado y acción.

El estado del arte presentado en esta sección se enfoca en diagnosticadores de fallas que utilizan RP pues es la herramienta usada en la elaboración de este trabajo.

Dotoli et. al. en [7] presentan un método para el monitoreo en línea de SED, en ese trabajo el modelo de monitoreo utiliza redes de Petri híbridas de primer orden (redes de Petri que utilizan aproximaciones de fluidos de primer orden). Se pueden observar dos

puntos en ese trabajo, el primero es que la técnica usada por Dotoli es una técnica modular que puede ser aplicada eficientemente a modelos complejos donde muchas tareas interdependientes necesitan ser ejecutadas (es decir las tareas pueden ser ejecutadas en serie, en paralelo o estar sincronizadas); mientras que el segundo punto es que su metodología asegura la detección de la falta a tiempo, es decir, antes que termine la ejecución del tiempo máximo asignado a cada tarea. La diferencia entre el trabajo de Dotoli y el presentado aquí es que las faltas que considera Dotoli son faltas basadas en tiempo, los lugares son considerados tareas que cuando son interrumpidas o cuando su tiempo de ejecución es diferente al esperado, la falta será detectada, mientras que en el presente trabajo, las faltas ocurren modeladas por transiciones, representando una desviación en el comportamiento normal del sistema.

Prock et. al., en [21] presenta también un método en línea para la detección de faltas. Prock detecta las faltas utilizando el número de marcas que existen dentro de los P-semiflujos, cuando el diagnosticador detecta que el número de marcas dentro del P-semiflujo cambia, se considera que ocurrió un error. Prock propone esa metodología para sistemas que transportan o que transforman materia. Siendo así el área de aplicación para su trabajo reactores, donde Prock realizó pruebas utilizando sensores existentes en el sistema y modelando con RP la estructura del mismo. El problema con ese método es que las faltas no pueden ser localizadas exactamente, diferencia importante al compararla con el presente trabajo.

Wu y Hadjicostis en [28] utilizan redes de Petri para introducir redundancia en el sistema. En su trabajo P-semiflujos adicionales permiten la detección y el aislamiento de marcados con faltas. El sistema se monitorea utilizando sistemas redundantes que detectarán la falta cuando el marcado sea diferente al esperado. Dos clases de faltas pueden ser detectadas utilizando redundancia, la primera se llama *falta en la transición*, este tipo de falta ocurre cuando el número de marcas recibidas y/o dadas es diferente al modelo de la RP, mientras que el segundo tipo se llama *falta en el lugar*, que es cuando el lugar corrompe el número de marcas en algún lugar de la RP. En su trabajo el marcado de la red necesita ser observable periódicamente aunque sí considera y utiliza eventos inobservables. La diferencia principal entre Hadjicostis y el trabajo presentado aquí se encuentra en el modelado de las faltas, mientras que Hadjicostis considera que las faltas se deben a que los lugares o las transiciones se corrompen añadiendo o quitando marcas, en el trabajo presentado aquí las faltas son desviaciones en el sistema modeladas por transiciones que modifican el comportamiento normal del sistema.

Lefebvre y Delherm [14] investigan cómo determinar el conjunto de lugares que pueden ser observados para la estimación exacta e inmediata de secuencias de disparo. Definen diagnosticadores mínimos para detectar y aislar las transiciones de falta inmediatamente. El trabajo propuesto por Lefebvre necesita que no existan eventos que no sean detectables mientras que en el trabajo presentado aquí, este tipo de transiciones pueden ocurrir.

Ramírez-Treviño, et. al. [23] utilizan RP interpretadas para modelar el comportamiento del sistema que incluye eventos y lugares parcialmente observables. El modelo de la red

de Petri interpretada se utiliza para detección de faltas en línea al resolver problemas de programación lineal. La diferencia principal entre el presentado por Ramírez-Treviño y el presente trabajo se encuentra en las condiciones necesarias para la creación del diagnosticador. En el trabajo presentado aquí no se necesita satisfacer la propiedad de diagnosticabilidad de entrada salida para la creación del diagnosticador, condición necesaria en el trabajo presentado en [23].

Una serie de artículos se han presentado con la suposición que ningún estado es observable [9, 27, 1, 14, 6], mientras que en el presente trabajo se necesita que algunos de los estados sean observables para poder detectar las faltas en el sistema.

En [9] Genc y Lafortune proponen un diagnosticador distribuido utilizando modularidad, el diagnosticador puede detectar faltas en línea en cada uno de los módulos del sistema. El diagnosticador reúne la información monolítica que se obtiene al combinar en un módulo individual la información obtenida al combinar todos los módulos en uno solo que conserva el comportamiento del sistema modular original. Un sistema de comunicación conecta los diferentes módulos actualizando la información de diagnóstico. El algoritmo propuesto en [9] asegura poder detectar cualquier tipo de falta debido al análisis del árbol de alcanzabilidad de cada uno de los diagnosticadores. La desventaja de este método en comparación con el presentado en este trabajo es que es necesario crear el grafo de alcanzabilidad de las RP.

Beneviste et. al., en [2] utilizan un desdoblamiento de las RP para el diseño de diagnosticadores asíncronos en línea. Estos diagnosticadores son perfectos para sistemas distribuidos y asíncronos, donde no existen estados globales ni tiempo global, pero si un modelado parcial del tiempo. Beneviste no utiliza los diagnosticadores generales utilizados en la mayoría de los artículos, si no que las redes diagnosticadoras son calculadas en línea, al utilizar la estructura original de las redes de Petri. La explosión de estados es controlada en este trabajo pero el cálculo en línea puede ser muy alto debido a la creación en línea de las estructuras de las redes de Petri. La principal diferencia entre este trabajo y el presentado en este trabajo es que el diagnosticador es calculado fuera de línea, reduciendo el cálculo en línea donde solo se realiza la detección de faltas en el sistema.

En [1], Basile utiliza un diagnosticador en línea al definir y resolver problemas de programación lineal entera (ILP). Con la suposición que las transiciones de falta no son observables, el marcado de la red es calculado con la ecuación de estado y si el marcado tiene componentes negativas entonces se concluye que una secuencia inobservable ocurrió. La programación lineal presentada en este trabajo entrega la secuencia y la ocurrencia de las faltas.

Finalmente, Dotoli et. al., [6] propone un diagnosticador en línea para evitar el rediseño y la redefinición del sistema cuando éste cambia. El diagnosticador espera por un evento observable y un algoritmo decidirá si el comportamiento es normal o si existe una falta en el sistema. Problemas ILP se definen con este propósito que darán como resultado las secuencias mínimas de transiciones inobservables que contienen las faltas que pudieron

haber ocurrido. El diagnosticador presentado por Dotoli es una técnica general que no aplica ninguna suposición al conjunto de estados alcanzados, y solamente algunas propiedades tienen que satisfacerse en el modelado de las faltas en una RP.

1.4 Descripción del Problema

La detección e identificación de las faltas es de gran importancia en los sistemas tecnológicos porque ayudan a crear un sistema robusto, mejorando su rendimiento, causando clientes satisfechos como en el caso de los teléfonos celulares o porque juegan papeles importantes como salvar vidas considerando el caso de la planta nuclear.

Muchas de las metodologías usadas para la detección y la identificación de las faltas en sistemas dinámicos asumen que una vez que la falta ocurre, el sistema se quedara ahí permanentemente, ocasionando así una falta permanente. Pero, existen muchas otras aplicaciones donde las faltas no son permanentes, donde las faltas son intermitentes como es el caso de válvulas pegadas, relevadores con faltas, errores de software, que complican la detección de las faltas porque el sistema continúa con su funcionamiento después de que ocurre la falta. Estos problemas presentan una oportunidad para mejorar la detección de faltas con diagnosticadores más rápidos y precisos.

Este tipo de faltas son las que se abordan en este trabajo, donde se tiene como objetivo la detección de faltas intermitentes. La contribución principal de este trabajo al compararlo con [23] es que las RPI no necesitan ser evento-detectable para ser diagnosticables de entrada-salida y generar el diagnosticador.

El objetivo de este trabajo es relajar la condición de evento-detectabilidad para diagnosticabilidad de entrada-salida propuestas en [23] para la detección de faltas intermitentes en RPI. Este trabajo presenta un diagnosticador que detecta faltas intermitentes en sistemas que no son evento-detectable en el sentido propuesto por Ramírez- Treviño, et. al.[23].

El diagnosticador presentado aquí, no necesita que el sistema sea evento-detectable sino que transiciones que son saltadas y transiciones que no son saltadas sean evento-detectable. Estas transiciones junto con un nuevo lugar que será el diagnosticador son las que se encargarán de detectar las faltas ocurridas en el sistema.

1.5 Panorama General de la Tesis

El resto de este reporte presenta el diseño del diagnosticador. En el Capítulo 2 se presenta un panorama general de las redes de Petri, su descripción formal y se presenta también a las redes de Petri interpretadas (RPI) que se utilizarán como herramienta de modelado y para la creación del diagnosticador. En el Capítulo 3 se presenta la solución propuesta para la detección de faltas intermitentes. El Capítulo 4 presenta ejemplos donde el

diagnosticador es capaz de detectar faltas intermitentes en RPI y finalmente en el Capítulo 5 se presentan las conclusiones y el trabajo futuro.

Capítulo 2

Redes de Petri

2.1 Redes de Petri

Las Redes de Petri fueron presentadas inicialmente por Carl Adam Petri a principios de los años sesenta en su tesis de doctorado [20]. Después de su introducción han sido ampliamente estudiadas, desarrolladas y expandidas en muchas direcciones, un ejemplo claro son las Redes de Petri Jerárquicas o aquellas en las que el tiempo forma parte de su análisis. Por su naturaleza, las Redes de Petri resultan particularmente útiles en el modelado de sistemas concurrentes, distribuidos y/o asíncronos. Asimismo son aplicables para el manejo y modelado de sistemas discretos y algunos continuos [8].

Una Red de Petri es un grafo dirigido con dos tipos de nodos: transiciones y lugares. Estos nodos están conectados por arcos, los cuales siempre unirán nodos de diferentes tipos, por ejemplo, una transición con un lugar y viceversa. La representación gráfica de los lugares son círculos mientras que la de las transiciones son rectángulos [17].

Una Red de Petri está definida formalmente como:

Definición 2.1. *La estructura de la Red de Petri es la 4-tupla $F = (P, T, I, O)$ donde:*

- $P = \{p_1, p_2, \dots, p_n\}$ es un conjunto finito de lugares,

- $T = \{t_1, t_2, \dots, t_m\}$ es un conjunto finito de transiciones,
- $I: P \times T \rightarrow \{0,1\}$ es la función que representa los arcos que van de los lugares a las transiciones
- $O: T \times P \rightarrow \{0,1\}$ es la función que representa los arcos que van de las transiciones a los lugares

Definición 2.2. La función de marcado $M: P \rightarrow \mathbb{N} \cup \{0\}$, donde \mathbb{N} es el conjunto de números Naturales, representa el número de marcas que tienen los lugares. Las marcas son representadas gráficamente por puntos en los lugares.

Definición 2.3. Una Red de Petri (RP) es el par (F, M_0) , donde F es la estructura de RP y M_0 es la distribución inicial de las marcas.

Las funciones de pre- y post- incidencia son representadas por las matrices:

$PRE = \{a_{ij}\}$ y $POST = \{b_{ij}\}$ respectivamente, donde $a_{ij} = pre(p_i, t_j)$ y $b_{ij} = post(p_i, t_j)$, PRE y $POST$ son también representadas por C^- y C^+ respectivamente.

La matriz de incidencia $C = [c_{ij}]$, $i = 1, \dots, n$; $j = 1, \dots, m$, donde $c_{ij} = b_{ij} - a_{ij}$ ($C = C^+ - C^-$)

Para una transición dada $t_i \in T$ $\bullet(t_i)$ y $(t_i)\bullet$ representan los conjuntos de lugares p_i tal que $I(p_i, t_j) \neq 0$ y $O(t_j, p_i) \neq 0$ respectivamente. Asimismo, $\bullet(p_j)$ representa el conjunto de transiciones t_j tal que $I(p_i, t_j) \neq 0$ y $(p_i)\bullet$ es el conjunto de todas las transiciones t_j tal que $O(t_j, p_i) \neq 0$.

Una transición $t_i \in T$ está habilitada en el marcado M si y sólo si $\forall p_j \in M(p_j) > PRE(t_i, t_j)$. Una transición habilitada $t_i \in T$ puede ser disparada y su disparo alcanzará un marcado diferente M_{k+1} , que puede ser calculado con la ecuación de estado de una RP.

$$M_{k+1} = M + C \cdot \vec{v}_j \quad (2.1)$$

Donde \vec{v}_j es un vector de entradas m , para el cual las componentes son definidas como sigue, $\vec{v}_j(i) = 0$, para $i \neq j$, $\vec{v}_j(j) = 1$ y m es el número de transiciones en la red. El enunciado anterior puede ser representado por $M \xrightarrow{t_j} M_{k+1}$. La ecuación 2.1 es conocida como la Ecuación de Estado de las Redes de Petri.

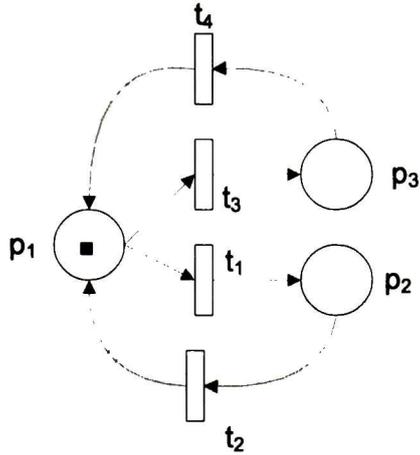


Figura 1.1 Red de Petri

Ejemplo 2.1. Considerando la Figura 2.1, denotada por (F, M_0) como se definió anteriormente. El conjunto de lugares de la RP es $P = \{p_1, p_2, p_3\}$ y el conjunto de transiciones es $T = \{t_1, t_2, t_3, t_4, t_5\}$.

La Matriz de Incidencia para esta Red de Petri es,

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

La Matriz de pre-incidencia es,

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

La Matriz de post-incidencia es,

$$C^+ = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

El marcado inicial de la RP es,

$$M_0 = [1 \ 0 \ 0]^T$$

A partir de este marcado inicial se concluye que t_1 está habilitada. Y si t_3 es disparada, el nuevo marcado será,

$$M_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Con este nuevo marcado, t_4 será habilitada y si t_4 es disparada, el nuevo marcado será,

$$M_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Para t_1 tenemos que el conjunto de lugares de entrada es $\bullet t_1 = \{p_1\}$ y el conjunto de lugares de salida es $t_1 \bullet = \{p_2\}$. Para el lugar p_1 tenemos que el conjunto de transiciones de entrada es $\bullet p_1 = \{t_2, t_4\}$ y el conjunto de transiciones de salida será $p_1 \bullet = \{t_1, t_3\}$.

Definición 2.4. Una secuencia de disparo en una (F, M_0) es una secuencia de transiciones que ocurren en la RP y puede denotarse como:

$$M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_j} \dots \xrightarrow{t_n} M_n$$

Definición 2.5. El conjunto de todas las secuencias de disparo es llamado el lenguaje de disparo

$$\Gamma(F, M_0) = \left\{ \sigma \mid \sigma = t_i t_j \dots t_n \text{ donde } M_0 \xrightarrow{t_i} M_1 \xrightarrow{t_j} \dots \xrightarrow{t_n} M_n \right\}$$

Definición 2.6. El conjunto de alcanzabilidad de una RP, $R(F, M_0)$, es el conjunto de todos los marcados que pueden ser alcanzados desde M_0 a través de cualquier secuencia de disparo.

Definición 2.7. Una Red de Petri $R(F, M_0)$ es cíclica si $\forall M_i \in R(F, M_0)$ se satisface que $\exists \sigma$, tal que $M_i \xrightarrow{\sigma} M_0$.

Definición 2.8. Una Red de Petri $R(F, M_0)$ es viva si $\forall M_i \in R(F, M_0)$ y $\forall t \in T$ se satisface que $\exists M_j$, tal que $M_i \xrightarrow{\sigma} M_j \xrightarrow{t} M_k$

Definición 2.9. Una Red de Petri $R(F, M_0)$ es k -segura si $\forall M \in R(F, M_0)$ y $\forall p \in P$, $M(p) \leq k$. Si $\forall M \in R(F, M_0)$ y $\forall p \in P$, $M(p) \leq 1$, la red es llamada 1-segura (segura o binaria).

Definición 2.10. Una Red de Petri $R(F, M_0)$ es fuertemente-conexa si y sólo si desde cualquier marcado $M_j \in R(F, M_0)$ cualquier otro marcado puede ser alcanzado $M_k \in R(F, M_0)$ por medio de una secuencia de disparo σ_k , por ejemplo. $M_j \xrightarrow{\sigma_k} M_k$.

Ejemplo 2.2. Considerando la Red de Petri de la Figura 2.1, donde el lenguaje de secuencia de disparo es,

$$\Gamma(F, M_0) = \{t_1 t_2, t_3 t_4, t_1 t_2 t_3, t_1 t_2 t_3 t_4, \dots\}$$

El conjunto de marcado alcanzable de la RP será,

$$R(F, M_0) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

La RP es cíclica debido a que para cualquier marcado alcanzable M_i existe una secuencia de disparo σ que puede llevar a la RP al marcado inicial M_0 . Y la RP es también activa ya que puede alcanzar un marcado M_k que habilita cualquier transición de la RP.

Definición 2.11. Sea C la matriz de incidencia de la RP $R(F, M_0)$. Un T-semiflujo X_i de (N, M_0) es la solución de valores racionales semi-positivos ($X_i \geq 0$) de la ecuación $CX_i = 0$. El soporte del T-semiflujo X_i , es el conjunto $\|X_i\| = \{t_j | X_i(t_j) \neq 0\}$.

Definición 2.12. Sea C la matriz de incidencia de la RP $R(F, M_0)$. Un P-semiflujo Y de $R(N, M_0)$ es la solución de valores racionales semi-positivos de la ecuación $Y^T C = 0$

Ejemplo 2.3. Considerando la RP de la Figura 2.1. Los vectores:

$$X_1 = [1 \ 1 \ 0 \ 0]^T \text{ y}$$

$$X_2 = [0 \ 0 \ 1 \ 1]^T$$

Son soluciones racionales semi-positivas para $CX_i = 0$ y son T-semiflujos de la red. Si la secuencia de disparo $\sigma = t_1 t_2$ del vector $X_1 = [1 \ 1 \ 0 \ 0]^T$ se dispara en la RP, el nuevo marcado M_j será igual al marcado inicial M_0 . De manera similar, a partir del disparo de la secuencia $\sigma = t_3 t_4$ el vector $X_2 = [0 \ 0 \ 1 \ 1]^T$ será alcanzado y el marcado será $M_k = M_0$.

El siguiente vector es la solución racional semi-positiva que satisface a $Y^T C = 0$,

$$Y_1 = [1 \ 1 \ 1 \ 1]^T$$

Es importante resaltar que para cada marcado alcanzable M_k se tiene que $M_k(p_1) + M_k(p_2) + M_k(p_3) = 1$, esta ecuación significa que para cada marcado alcanzable tenemos que en p_1, p_2, p_3 y p_4 hay solo una marca. Y $Y_1 = [1 \ 1 \ 1 \ 1]^T$ es el único componente conservativo, es decir, el único P-semiflujo de la RP.

Definición 2.18. El conjunto de las secuencias de todos los símbolos de entrada-salida que llevan a marcados de bloqueo de la RPI son denotadas como $\Lambda_B(Q, M_0) = \{w \in \Omega(Q, M_0) \mid \exists \sigma \in \Omega(w) \text{ tal que } M_0 \xrightarrow{\sigma} M_j \wedge \text{ si } M_j \xrightarrow{t_i} \text{ entonces } C(\bullet, t_i) = \vec{0}\}$

Definición 2.19. Supóngase $w = (\alpha_0, y_0)(\alpha_1, y_1) \dots (\alpha_n, y_n)$ como una secuencia de símbolos de entrada-salida. El conjunto de secuencias de marcado es definida como $S_w = \{M_0, M_1, \dots, M_k \mid M_i \in R(Q, M_0) \wedge M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} \dots \xrightarrow{t_{im}} M_k \wedge \sigma_w = t_1 t_2 \dots t_m \in \Omega(w)\}$

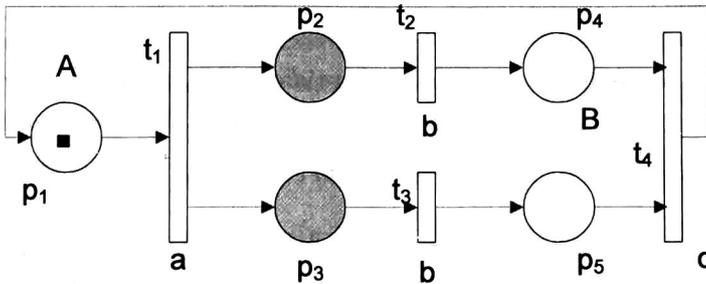


Figura 2.2 Red de Petri Interpretada

Ejemplo 2.4. La función de salida de la RPI mostrada en la Figura 2.2 es la matriz

$$\varphi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

La función de etiquetado de las transiciones es

$$\lambda(t_1) = a, \lambda(t_2) = b, \lambda(t_3) = b, \lambda(t_4) = c$$

Donde se puede ver que las transiciones t_1, t_2, t_3 y t_4 son transiciones manipuladas, y los lugares p_1, p_4 y p_5 son medibles. Los lugares coloreados en gris son los lugares no medibles.

El lenguaje de disparo de la RPI presentada es,

$$\Gamma(Q, M_0) = \{t_1, t_1 t_2, t_1 t_3, t_1 t_2 t_3, t_1 t_3 t_2, t_1 t_2 t_3 t_4, t_1 t_3 t_2 t_4, t_1 t_2 t_3 t_4 t_5, \dots\}$$

$$\Lambda(Q, M_0) = \left\{ \left(\varepsilon, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right), \left(\varepsilon, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \left(a, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right), \left(\varepsilon, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \left(a, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) \left(b, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right), \dots \right\}$$

$$\Lambda^2(Q, M_0) = \left\{ \left(\varepsilon, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \left(a, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right), \left(\varepsilon, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \left(a, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) \left(b, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right), \dots \right\}$$

$$\Lambda_B(Q, M_0) = \{ \}$$

Si tenemos la siguiente secuencia de símbolos de entrada-salida

$$w = \left\{ \left(\varepsilon, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) \left(a, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) \left(b, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \right\}, \text{ entonces}$$

$$\Omega(w) = \{t_1 t_2, t_1 t_3\} \text{ y}$$

$$S_w = \left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\}$$

Definición 2.20. Una RPI (Q, M_0) es evento-detectable si el disparo de cualquier transición puede ser detectado sólo conociendo la entrada y la salida producida por (Q, M_0) [25]

El siguiente Lema propuesto por [24] es una caracterización polinomial evento-detectable.

Lema 2.1. Una RPI viva dada por (Q, M_0) es evento-detectable si y sólo si [24]

1. $\forall t_i, t_j \in T$ tal que $\lambda(t_i) = \lambda(t_j)$ o $\lambda(t_i) = \varepsilon$ se tiene que $\varphi C(\bullet, t_i) \neq \varphi C(\bullet, t_j)$ y
2. $\forall t_k \in T$ se tiene que $\varphi C(\bullet, t_k) \neq 0$

Ejemplo 2.5. Considérese la RPI de la Figura 2.2 se tiene que

$$\varphi C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & 1 & 1 & -2 \end{bmatrix}$$

Donde puede verse que la RPI no es evento-detectable, debido a que hay dos vectores columna de las transiciones t_2 y t_3 donde el disparo de estas transiciones no puede ser distinguidas de una secuencia de símbolos de entrada-salida.

Definición 2.21. *Transiciones Evento-detectable*, las columnas que son diferentes unas de otra en φC , del Lema 2.1, representaran las transiciones que pueden ser distinguidas a partir de la secuencia de símbolos de entrada-salida. Éstas serán llamadas transiciones evento-detectable.

Capítulo 3

Diagnosticador de Faltas Intermitentes

En [25] el diagnosticador de la Figura 3.1 se utiliza para la detección de faltas permanentes e intermitentes. El diagnosticador trabaja en paralelo con el sistema real y tiene el siguiente comportamiento. Cuando entradas manipulables y no manipulables σ_i ocurren en el sistema, generaran un cambio en la salida, σ_j , del bloque del modelo del sistema, S_M , y generan también un cambio en la salida β_j del Diagnosticador, D_M .

Las salidas σ_j y β_j se comparan dentro del bloque de error e_k y cuando el sistema se encuentra libre de errores, la salida del bloque e_k será cero, indicando que no hay faltas en el sistema. Si existe un error en el sistema, las salidas del bloque e_k serán diferentes de cero, indicando así que un error ocurrió en el sistema y dando la posibilidad de identificar la falta.

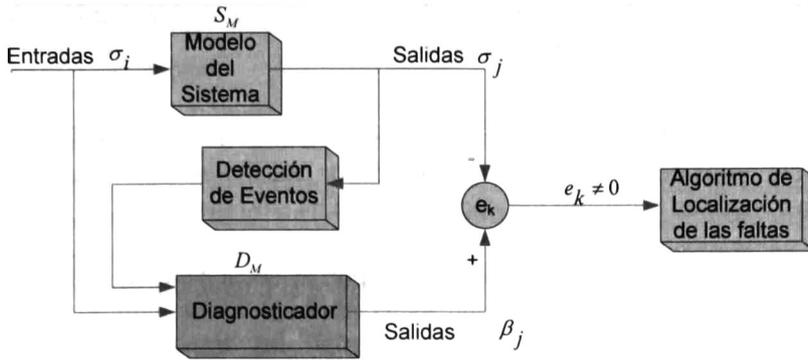


Figura 3.1 Diagrama de Bloques del Diagnosticador [23]

Donde σ_j y β_j , están dados por los vectores de observación,

$$\sigma_j = \varphi(M_{S_M})$$

$$\beta_j = \varphi(M_{D_M})$$

que representan las salidas del modelo del sistema y del diagnosticador respectivamente.

3.1 Diagnosticabilidad de Faltas Intermitentes

El diagnosticador para la detección y localización de faltas intermitentes está basado en el diagnosticador presentado en [25, 23], pero con las siguientes diferencias importantes.

- El bloque con el *modelo del sistema*, S_M , en el presente trabajo contiene únicamente el comportamiento normal del sistema, mientras que en [25] este bloque modela el comportamiento normal más el comportamiento con faltas del sistema.
- El bloque con el *Diagnosticador*, D_M , está compuesto en este trabajo por el comportamiento normal del sistema más un estado extra por cada falta intermitente en el sistema, mientras que en [25] el bloque del Diagnosticador tiene modelado únicamente el comportamiento normal del sistema.

El diagnosticador presentado en este trabajo tiene un lugar por cada transición de falta que exista en el sistema y este nuevo lugar servirá para detectar la ocurrencia de las faltas. El sistema y el diagnosticador se modelan utilizando RPI donde no importa si las transiciones son manipulables, $\lambda(t_i) = \varepsilon$ o $\lambda(t_i) \neq \varepsilon$.

La idea del funcionamiento de este nuevo lugar p_d , que será añadido por cada transición de falta, es que mientras el sistema se encuentre en su funcionamiento normal, marcas serán dadas y marcas se quitarán de este nuevo lugar p_d , esta operación de dar y quitar marcas a p_d debe ser diseñada de tal manera que en condiciones normales, la salida β_j de D_M sea igual que la salida σ_j de S_M . La diferencia se dará cuando una falta intermitente ocurra en el sistema, en este caso p_d perderá marcas y no permitirá al

Diagnosticador continuar con su funcionamiento, lo que causará a su vez que las salidas σ_j y β_j sean diferentes ocasionando un error en el bloque e_k .

El nuevo lugar p_d , será conectado únicamente por transiciones que sean distinguibles por la secuencia de símbolos de entrada-salida, es decir, por transiciones que sean evento-detectable y un nuevo lugar es necesario por cada falta que ha sido modelada.

Para encontrar las transiciones que son distinguibles por la secuencia de símbolos de entrada-salida, se utilizará el Lema 2.1 de evento-detectabilidad. Las columnas que sean diferentes entre ellas y diferentes de cero representan las transiciones que son distinguibles una de otra. Estas transiciones evento-detectables son las que darán y quitarán marcas de p_d .

La principal diferencia con respecto a [25] es que el diagnosticador presentado en este capítulo, no necesita ser evento-detectable, que era una de las condiciones necesarias para la creación del diagnosticador presentado ahí.

Para el diagnosticador presentado en este trabajo existen tres suposiciones que tienen que cumplirse antes de la creación del diagnosticador. Si estas tres suposiciones se cumplen entonces se puede decir que el sistema con faltas intermitentes es diagnosticable y que es posible la creación de un diagnosticador.

Suposición 3.1. Si t_f es la transición de falta, n representa el número total de transiciones del sistema original sin faltas y $\alpha_n \in [0,1]$, se tiene que cumplir,

$$C(\bullet, t_f) = \sum \alpha_n \cdot C(\bullet, t_n) \quad (3.1)$$

$$\sum \alpha_n \geq 2$$

Esta suposición indica que la columna correspondiente a la transición de falta t_f en la matriz de incidencia C es una combinación lineal de las demás columnas y que la transición de falta excluye por lo menos a dos transiciones.

Otra forma de entender esta suposición es que las transiciones que fueron saltadas por la transición de falta tienen que ser dos o más. Con esta finalidad es necesario encontrar las transiciones que fueron saltadas, para esto se puede notar que las transiciones que están el mismo T-semiflujo con la transición de falta no van a ser saltadas. Éstas mismas transiciones menos la transición de falta deben de pertenecer a otro T-Semiflujo. Al remover estas transiciones, que comparten T-semiflujo con la transición de falta, de cualquier otro T-semiflujo donde se encuentren, la diferencia serán las transiciones que son saltadas.

Para calcular las transiciones saltadas necesitamos calcular el T-semiflujo del sistema incluyendo las transiciones de faltas, es decir, utilizando la Definición 2.11 tenemos que encontrar las soluciones de la ecuación $C^e \bullet X^e = 0$, $X^e \geq 0$, donde C^e es la matriz de

incidencia extendida después de incluir la transición de falta, t_f , y X^e es el conjunto de vectores T-semiflujo.

Todas las transiciones que comparten T-semiflujo con una transición de falta t_f , tienen que estar en otro T-semiflujo, que es el T-semiflujo donde se encuentran normalmente cuando no existe la falta. Es en este T-semiflujo donde las transiciones que serán saltadas pueden encontrarse después de eliminar las transiciones que se comparten con T-semiflujos que incluyen a la transición de falta.

Una vez encontradas las transiciones que son saltadas por la transición de falta, para cumplir la Suposición 3.1, es necesario que el número de transiciones saltadas sean más de dos. La Figura 3.2 muestra dos ejemplos de lo que pasaría si esta suposición no se cumple.

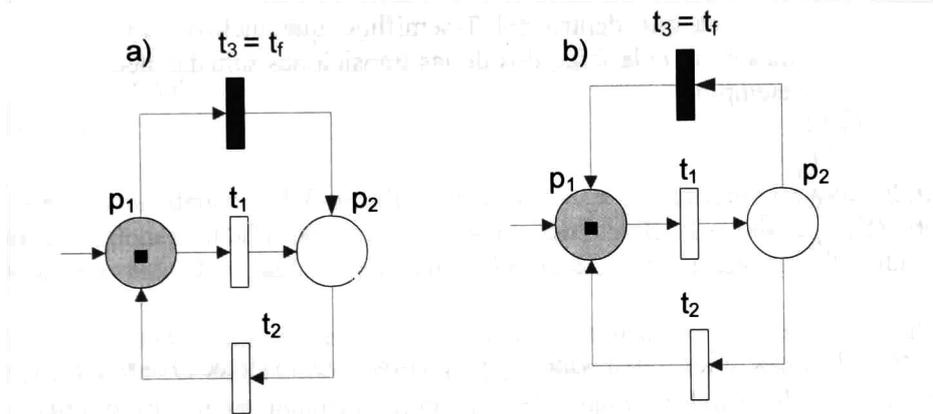


Figura 3.2 Transiciones de Falta saltando únicamente una transición.

Donde, para la red de la izquierda de la Figura 3.2,

$$C^e \bullet X^e = \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0, \text{ dando } X^e = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \right\}$$

El T-semiflujo, donde la transición de la falta se encuentra es,

$$X_1^e = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \text{ donde se puede ver que } \{t_2, t_3\} \text{ forman parte de este T-semiflujo.}$$

Ahora es necesario encontrar otro T-semiflujo que tenga t_2 como componente,

$$X_2^e = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \text{ y } \{t_1, t_2\}$$

Las transiciones en este T-semiflujo al remover las transiciones que se comparten con el T-semiflujo con la transición de falta, y el resultado será las transiciones que se saltarán debido a la transición de falta, en este caso t_1 .

Ambas RPI de la Figura 3.2, no cumplen con la Suposición 3.1, como se observa en la figura, es imposible saber si la transición de la falta, o la transición normal fueron disparadas, sin que alguna de ellas tenga su símbolo de salida diferente del vacío, $\lambda(t_1) \neq \varepsilon$ o $\lambda(t_2) \neq \varepsilon$

Suposición 3.2. Si α es el vector que tiene como coeficientes los valores de α_n encontrados con la fórmula (3.1) y E_d es un vector con coeficientes uno si la transición es evento-detectable; o cero si las transiciones no son evento-detectable, tenemos que,

$$\alpha^T \cdot E_d \geq 2, \quad (3.2)$$

Esta suposición indica que dentro del T-semiflujo que incluye la transición que fue saltada por la transición de la falta, dos de las transiciones saltadas necesitan ser evento-detectable. Por ejemplo,

$$\alpha^T \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \geq 2,$$

en la cual E_d tiene a las transiciones t_1 y t_4 como evento-detectable y a t_2 , t_3 como no evento-detectable, si el resultado de esta multiplicación es mayor o igual que dos, entonces la Suposición 3.2 se cumple.

La Suposición 3.2 fuerza a la RPI a tener por lo menos dos transiciones saltadas y evento-detectable como se muestra en la Figura 3.3, donde se tiene que $\varphi(\bullet, p_x) = e_k$, es decir, el lugar p_x tiene asociado un sensor y la lectura de este sensor es X . Entonces, p_x es un lugar medible y por consiguiente las transiciones t_{x-1} y t_{x+1} son transiciones evento-detectable.

Estas dos transiciones que son saltadas y evento-detectable serán conectadas al nuevo lugar p_d , ayudando en la detección de la transición de falta. Cuando estas transiciones no son detectadas, el diagnosticador será bloqueado por la falta de marcas en p_d , como se verá más adelante en el algoritmo para el diseño del diagnosticador.

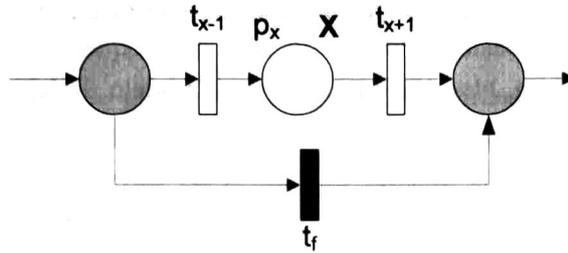


Figura 3.3 Transiciones saltadas que son Evento-detectable.

Suposición 3.3. Si C es la matriz de incidencia de la RPI, X es el T-semiflujo calculado con la matriz de incidencia sin incluir las transiciones de falta en el sistema, E_d es el vector con las transiciones evento-detectable (calculado en la Suposición 3.2) y α es el vector con los coeficientes encontrados anteriormente en la Suposición 3.1; se tiene que cumplir que

$$\begin{aligned} C \cdot X &= 0, \quad X \geq 0 \\ (X - \alpha)^T E_d &\geq 2 \end{aligned} \quad (3.3)$$

Lo cual significa que dentro del T-semiflujo que incluye a la transición de falta, al menos dos de las transiciones saltadas necesitan ser evento-detectable. Esto es, al menos dos de las transiciones evento-detectable no pueden ser brincadas por la transición de falta.

Al cumplirse esta suposición, se tendrá que dos de las transiciones evento-detectable se encuentren fuera de las transiciones saltadas, como se observa en la Figura 3.4. En esta figura t_{y-1} y t_{y+1} son transiciones evento-detectable gracias a tener $\lambda(p_y) = A$, y estas transiciones no son saltadas por ninguna transición de falta.

Estas transiciones, t_{y-1} y t_{y+1} , son necesarias para añadir/borrar marcas del nuevo lugar p_d , y así detectar cuando la transición de falta ocurrió, como se verá más adelante en el algoritmo para calcular al diagnosticador del sistema.

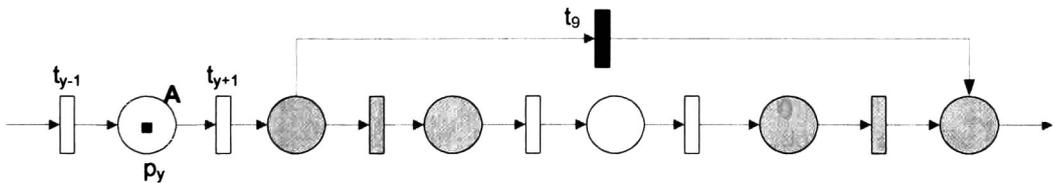


Figura 3.4 Transiciones no saltadas que son evento-detectable.

Las Suposiciones 3.2 y 3.3 hacen que la RPI tenga por lo menos dos transiciones evento-detectable que son saltadas por la transición de falta, y que la RPI tenga por lo menos dos transiciones evento-detectable que no son saltadas por la transición de falta.

Estas cuatro transiciones evento-detectable son las que se encargan de añadir y borrar marcas del lugar p_d , ocasionando que la RPI se bloquee cuando alguna de estas transiciones no se ejecuta, bloqueando entonces el modelo de RPI del diagnosticador y detectando la falta.

El diagnosticador para faltas intermitentes presentado en este trabajo es posible de crear para el Ejemplo 3.1 debido a que en este ejemplo las tres suposiciones se cumplen.

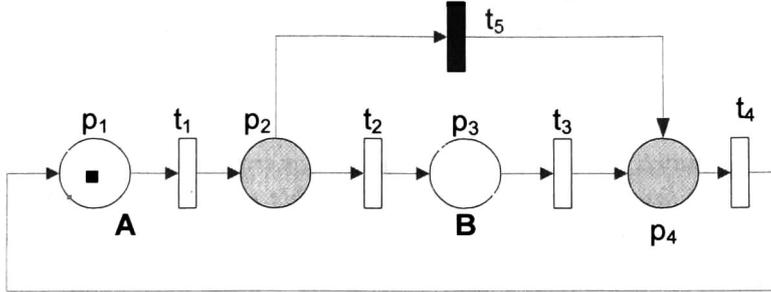


Figura 3.5 Ejemplo de una RPI

Ejemplo 3.1.

Verificando la Suposición 3.1,

$$C(\bullet, t_f) = \sum \alpha_n \cdot C(\bullet, t_n)$$

donde se tiene que cumplir que $\sum \alpha_n \geq 2$.

Entonces evaluando,

$$\begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \end{bmatrix} = \alpha_1 \cdot \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \cdot \begin{bmatrix} 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \alpha_3 \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} + \alpha_4 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix},$$

y así $\alpha_1=0, \alpha_2=1, \alpha_3=1$ y $\alpha_4=0$, cumpliéndose que $\sum \alpha_n \geq 2$, porque,

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \geq 2$$

Ahora, calculando los T-semiflujos del sistema extendido,

$$C^e \bullet X^e = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = 0, \text{ con } X^e = \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \right\}$$

A partir de este resultado, es posible ver que el segundo T-semiflujo,

$$X^e = [1 \ 0 \ 0 \ 1 \ 1]^T, \text{ incluye a las transiciones } \{t_1, t_4, t_5\}$$

y la transición de falta t_5 se encuentra en este T-semiflujo.

A partir de este resultado se puede ver también que las transiciones t_1 y t_4 no serán saltadas por la transición de falta t_5 , estas dos transiciones se encuentran en el primer T-semiflujo,

$$X^e = [1 \ 1 \ 1 \ 1 \ 0]^T \text{ que incluye las transiciones } \{t_1, t_2, t_3, t_4\}$$

Al remover las transiciones t_1 y t_4 , de este T-semiflujo, las transiciones resultantes son las transiciones que serán saltadas por la transición de falta y son la t_2 y la t_3 ; cumpliendo así la Suposición 3.1 porque se tiene que por lo menos dos transiciones son saltadas por la transición de falta.

La Suposición 3.2 también se cumple, porque el lugar p_3 es observable con símbolo de salida B , ocasionando que las transiciones t_2 y t_2 evento-detectable, y habilitando estas transiciones para ser usadas por el diagnosticador.

Utilizando φC para calcular las transiciones que son evento-detectable,

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

donde el resultado es que todas las transiciones son evento-detectable.

Verificando $\alpha^T \cdot E_d \geq 2$,

$$[0 \ 1 \ 1 \ 0] \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \geq 2, \text{ se tiene que } \alpha^T \cdot E_d \geq 2, \text{ cumpliendo también a la Suposición 3.2.}$$

La tercera suposición, la Suposición 3.3, también se cumple porque el lugar p_3 se encuentra habilitando a las transiciones t_2 y t_3 , que son saltadas por la transición de falta t_f , mientras que el lugar p_1 es el otro lugar visible en el T-semiflujo.

Para verificar la Suposición 3.3, se tiene que calcular los T-semiflujos del sistema, resolviendo entonces a $C \cdot X = 0$, se tiene,

$$C \bullet X = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = 0, \text{ con } X = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

y así $(X - \alpha)^T E_d \geq 2$

$$\left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \right)^T \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \geq 2 \text{ y } [1 \ 0 \ 0 \ 1] \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \geq 2$$

cumpliendo también a la Suposición 3.3.

Al cumplirse con este ejemplo las tres suposiciones tenemos entonces que un diagnosticador puede construirse para el sistema modelado en la Figura 3.5.

Teorema 1. *Sea (Q, M_0) una RPI con faltas de control intermitentes, donde el modelo del comportamiento normal del sistema (Q^N, M_0^N) , junto con las transiciones de falta de control es una RPI viva. Si las suposiciones 3.1, 3.2 y 3.3 se cumplen entonces el sistema es diagnosticable y es posible el diseño de un diagnosticador de faltas intermitentes.*

Demostración. *La demostración se realiza por construcción como se verá más adelante en el Capítulo 4.*

3.2 Diseño de un Diagnosticador de Faltas Intermitentes

En la sección anterior se presentaron las suposiciones necesarias para crear el diagnosticador de faltas intermitentes. En esta sección se presentarán algoritmos para diseñar dicho diagnosticador.

Algoritmo 1. Verificación de la diagnosticabilidad de la red de Petri.

Este algoritmo presenta los pasos necesarios para verificar que el diagnosticador presentado en este trabajo es posible de diseñar.

1. Modelar el sistema por el procedimiento presentado en [25] y que las transiciones de la falta se encuentren incluidas en el modelo de la RPI.
2. Encontrar las transiciones que son evento-detectable para el sistema sin el modelado de las transiciones de falta, al utilizar el Lema 2.1 y $\varphi \bullet C$
3. Encontrar las transiciones que son evento-detectable para el sistema extendido (el sistema con las transiciones de la falta en él), al utilizar el Lema 2.1 con $\varphi \bullet C^e$

4. Si las transiciones que son evento-detectable en $\varphi \bullet C$ son diferentes de las transiciones que son evento-detectable en $\varphi^e \bullet C^e$ entonces las transiciones de las faltas que se añadieron al modelo del sistema son evento-detectable o que están haciendo a una de las transiciones que solía ser evento-detectable, no evento-detectable.

Si la transición de falta es evento-detectable, entonces sería más fácil detectar cuando una falta ocurre en el sistema pues el sistema se daría cuenta cuando esta transición ocurre.

5. Encontrar los T-semiflujos del sistema extendido con

$$C^e \cdot X^e = 0, X^e \geq 0$$

Encontrar los T-semiflujos es un problema con complejidad NP, pero puede resolverse considerando las transiciones con flujo negativo que serán las transiciones saltadas (ver Apéndice A).

6. Encontrar los T-Semiflujos del sistema normal

$$C \cdot X = 0, X \geq 0$$

Encontrar los T-semiflujos es un problema con complejidad NP, pero puede resolverse considerando las transiciones con flujo negativo que serán las transiciones saltadas (ver Apéndice A).

7. Verificar que las suposiciones 3.1, 3.2 y 3.3 se cumplan.

Si las suposiciones 3.1, 3.2 y 3.3 se cumplen, entonces es posible la creación del diagnosticador presentado en este trabajo.

Algoritmo 2. Diagnosticador de faltas intermitentes

Una vez verificado que es posible diagnosticar el modelo con el algoritmo 1, el algoritmo 2 presenta los pasos a seguir para la elaboración del diagnosticador.

1. Copiar el modelo extendido del sistema y añadir un lugar por cada transición que modele faltas, este nuevo lugar será el nuevo lugar del diagnosticador, p_d , es éste lugar el que se encargará de la identificación de las faltas.
2. Las transiciones evento-detectable que están en el T-semiflujo con la transición de falta (i.e. las transiciones evento-detectable que no son saltadas) van a tener los arcos llegando hacia ellas, provenientes del nuevo lugar p_d .
3. Las transiciones evento-detectable que son saltadas por la transición de falta van a tener transiciones que salen de ellas hacia el nuevo lugar p_d .
4. El nuevo lugar, p_d , va a tener una marca inicial por cada T-semiflujo que es afectado por la transición de falta.
Esto significa que si la transición de la falta ocasiona que el sistema se salte transiciones de n T-semiflujos, entonces el marcado inicial para el lugar nuevo, p_d , añadido será n .
5. Cuando el diagnosticador necesita ser bloqueado después de que la falta ocurra i -veces, dos marcas extras necesitan ser añadidas al nuevo lugar p_d por cada vez extra que se requiera la falta ocurra antes de bloquear al diagnosticador.

En caso de que p_d tenga más marcas de las que se necesitan para bloquear al diagnosticador, estas marcas extras serán una habilidad del diagnosticador de permitir un cierto número de transiciones de faltas de que ocurran antes de bloquear al diagnosticador, es decir, el diagnosticador necesita únicamente una marca inicial, pero el diseñador puede añadir dos marcas además de la marca inicial necesaria en el diagnosticador.

Esta habilidad puede ser útil cuando el sistema modelado puede funcionar normalmente aún cuando un cierto número de faltas ocurra sin afectar su comportamiento normal. Por ejemplo, puede ser que un mensaje caído en una red de comunicación no sea suficiente para causar fallas en el sistema, pero dos mensajes caídos (dos transiciones de falta) entonces el sistema puede tener un comportamiento erróneo. En este caso al añadir dos marcas más al lugar p_d el diagnosticador perderá dos marcas al ocurrir la falta una vez, pero seguirá funcionando, cuando la falta ocurra por una segunda vez, entonces el diagnosticador se bloqueará indicando la presencia de la falta en el sistema.

Capítulo 4

Casos de Estudio

4.1 Caso de estudio 1

El primer ejemplo propuesto será una línea de sistemas de manufactura, dicha línea utiliza tecnología de montaje superficial (SMT, por sus siglas en inglés). Las líneas de producción con tecnología SMT son procesos que construyen tarjetas electrónicas para diversas aplicaciones donde los componentes son montados directamente sobre la superficie de la tarjeta de circuito impreso (PCB, por sus siglas en inglés), a diferencia de la antigua pero aún utilizada tecnología through-hole donde los componentes son unidos a las tarjetas por medio de cables que atraviesan pequeños orificios y posteriormente son soldados.

El proceso para SMT es presentado en la Figura 4.1 y puede ser descrito de la siguiente manera:

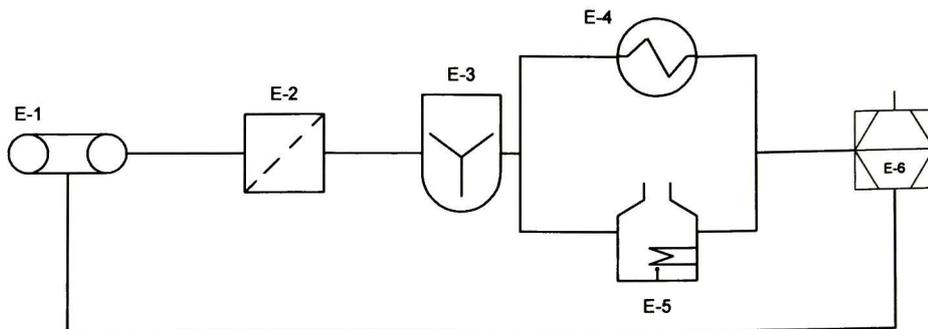


Figura 4.1 Línea de Tecnología de Montaje Superficial (SMT).

1. E-1 es la primera etapa del proceso. Se parte de la idea de que las tarjetas tienen todas las pistas y áreas de contacto necesarias, las cuales fueron fabricadas y probadas en un proceso independiente a éste. En esta etapa son lavadas para su uso en el resto del proceso. Para un mejor entendimiento del proceso, las PCBs usadas para el ejemplo son un panel compuesto de dos tarjetas.
La suposición de tener las tarjetas previamente construidas con todas las pistas y áreas de contacto es basada en el hecho de que la mayoría de las plantas de manufactura compran las PCBs de proveedores externos los cuales construyen las tarjetas con todas las conexiones internas necesarias.
2. E-2 es una máquina que colocará la soldadura de pasta sobre las áreas de contacto de las PCBs; utilizando dos plantillas para que solamente en las áreas necesarias sea depositada la soldadura de pasta, evitando la creación de defectos como impurezas y corto circuitos.
Es necesario el uso de dos plantillas porque serán dos procesos de fundición, el primero será soldadura de pasta con plomo y el restante será soldadura de pasta libre de plomo.
3. E-3 es la máquina de montaje y corte, ésta máquina se encarga de la colocación de todos los componentes sobre la PCB. La máquina no solo trabaja para un circuito integrado específico, sino que se encarga de la colocación de todos los componentes pasivos y activos. El proceso de montaje y corte en esta máquina ocurre durante varias etapas, la primera es la colocación de los elementos de mayor tamaño para continuar con los de menor tamaño hasta terminar todas las etapas para las que fue configurada. Al final hace separación de las tarjetas, mediante un corte para despanelizarlas y continuar separadas en la siguiente etapa.
4. E-4 y E-5 son hornos paralelos que calentarán la PCB de acuerdo al tipo de soldadura de pasta que fue colocada en la tarjeta. Son necesarios dos hornos porque los rangos de temperatura y el tiempo de calentamiento son diferentes entre los dos tipos de tecnología de soldadura de pasta.
5. Finalmente E-6 es la máquina para el control de calidad, una máquina de inspección óptica automatizada (AOI por sus siglas en inglés) se encarga de tomar

una imagen de la tarjeta y a partir de ésta se hace un análisis de los posibles defectos que pudieron ocurrir en cualquiera de las etapas previas. Algunos ejemplos de errores son componente mal soldado o mal colocado, componente erróneo, polaridades invertidas, etc. Para este ejemplo se considerará que el panel de 2 tarjetas es de doble lado, es decir es necesaria la colocación de componentes en ambos lados de la tarjeta, por lo tanto será enviada de nuevo al inicio del proceso y a través de todas las etapas para concluir con su fabricación.

La aplicación de una RPI para modelar el proceso de SMT se puede ver en la Figura 4.2. En este ejemplo cada estado representa uno de las etapas del proceso de SMT, por ejemplo p_1 es la primera etapa donde la PCB está limpia, p_2 es el lugar donde la soldadura de pasta es colocada en las áreas de contacto, a través de las plantillas y de manera similar para el resto de los lugares.

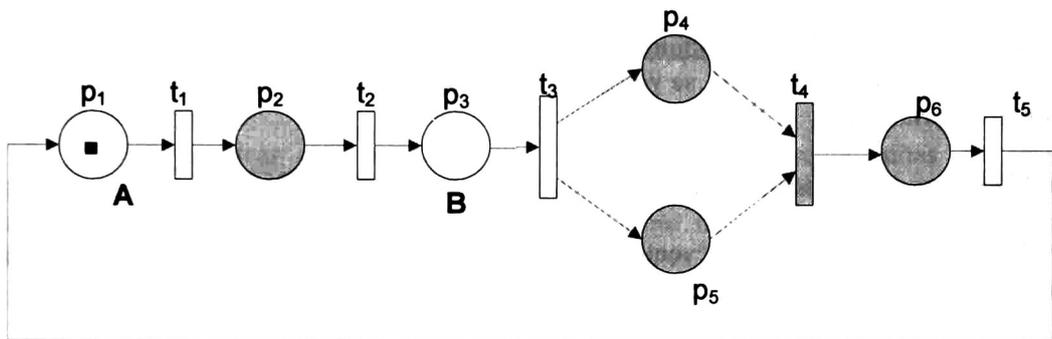


Figura4.2 Modelo de la línea SMT utilizando RPI

Es necesario colocar sensores dentro del proceso y para el diagnosticador presentado en este trabajo únicamente dos sensores serán instalados para detectar donde están las PCBs en el proceso, los sensores serán colocados en p_1 y en p_3 .

La siguiente etapa es modelar la transición de falta, durante el proceso de SMT en la etapa donde la soldadura de pasta es colocada sobre la PCB, en p_2 , es donde puede ocurrir una falta debido a un problema con la plantilla, con la máquina de inyección, etc.

Si llegara a ocurrir esta falta, entonces se tendría que las etapas del montaje de componentes y el horno serán erróneas. Esta falta es modelada con la transición t_6 en la Figura 4.3.

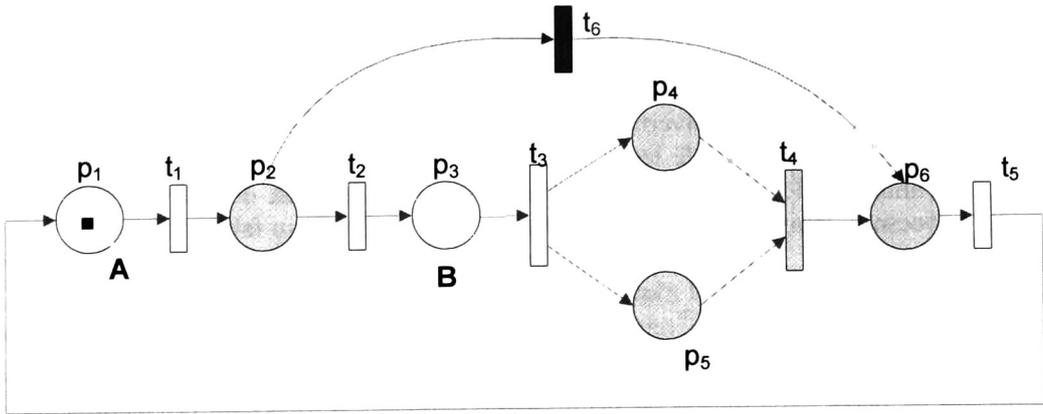


Figura 4.3 Modelo de la línea SMT en RPI más la transición de falta.

Una vez que el modelo extendido del sistema, el sistema más las transiciones de faltas son descritas usando una RPI, se verifica que el diagnosticador es posible de ser diseñado con el algoritmo 1.

1. El sistema fue modelado por el procedimiento presentado en [25].
2. Las transiciones que son evento-detectable de los sistemas normales son calculadas usando $\varphi \bullet C$.

$$\varphi \bullet C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix}$$

A partir del resultado se concluye que las transiciones diagnosticables son $\{t_1, t_2, t_3, t_5\}$ porque las columnas 1, 2, 3 y 5 son diferentes entre ellas y diferentes de cero.

3. Esta etapa es para encontrar la transición diagnosticable de sistemas extendidos, el sistema normal más las transiciones de faltas,

$$\varphi \bullet C^e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}$$

$$\varphi \bullet C^e = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix}$$

Entonces las transiciones evento-detectable del sistema extendido son $\{t_1, t_2, t_3, t_5\}$.

4. Las transiciones evento-detectable del sistema extendido son $\{t_1, t_2, t_3, t_5\}$ lo que significa que las transiciones de falta no son evento-detectable (es decir, las transiciones evento-detectable se preservan en el sistema extendido).

Vale la pena recalcar que no todas las columnas son diferentes unas de otras, consecuentemente no es posible diagnosticar el sistema utilizando el procedimiento presentado en [25, 23], sin embargo con el presente trabajo es posible diseñar un diagnosticador para la detección de las faltas intermitentes.

5. El T-semiflujo necesita ser calculado con $C^e \bullet X^e = 0$

$$C^e \bullet X^e = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix} \bullet X^e = 0$$

Dos T-semiflujos son encontrados para el sistema extendido:

$$(X_1^e)^T = [1 \ 1 \ 1 \ 1 \ 1 \ 0]$$

$$(X_2^e)^T = [1 \ 0 \ 0 \ 0 \ 1 \ 1]$$

6. Encontrar los T-semiflujos del sistema

$$C \cdot X = 0$$

$$C \cdot X = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \cdot X = 0$$

y el T-semiflujo encontrado para el sistema es

$$(X_1)^T = [1 \ 1 \ 1 \ 1 \ 1]$$

7. Las Suposiciones 3.1, 3.2 y 3.3 son verificadas en ésta etapa.

Suposición 3.1,

- $C(\bullet, t_f) = \sum \alpha_n \cdot C(\bullet, t_n)$

$$\begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \alpha_1 \cdot \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \cdot \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_3 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \alpha_4 \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ 1 \end{bmatrix} + \alpha_5 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix},$$

y así $\alpha_1 = 0$, $\alpha_2 = 1$, $\alpha_3 = 1$, $\alpha_4 = 1$ y $\alpha_5 = 0$ cumpliendo así con $\sum \alpha_n \geq 2$, porque $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 \geq 2$

- El T-semiflujo de las transiciones de falta es

$$(X_2^e)^T = [1 \ 0 \ 0 \ 0 \ 1 \ 1]$$

si t_6 es removido del T-semiflujo para ser la transición de falta, se concluye que la transiciones que no serán saltadas son $\{t_1, t_5\}$

- Buscando otro T-semiflujo donde están dichas transiciones no saltadas (este ejemplo solo tiene dos T-semiflujos haciendo más fácil el encontrar el otro T-semiflujo)

$$(X_1^e)^T = [1 \ 1 \ 1 \ 1 \ 1 \ 0]$$

- Tomando la diferencia de $(X_2^E)^T - (X_1^E)^T$ dará como resultado las transiciones saltadas

$$\{t_2, t_3, t_4\}$$

Estas transiciones son al menos dos, cumpliendo así con la Suposición 3.1.

Suposición 3.2.

- Verificando $\alpha^T \cdot E_d \geq 2$, tomando E_d del paso 1 y los coeficientes α_n calculados en la suposición anterior para el vector α .

$$[0 \ 1 \ 1 \ 1 \ 0] \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \geq 2$$

- El T-semiflujo con la transición saltada es

$$(X_1^e)^T = [1 \ 1 \ 1 \ 1 \ 1 \ 0]$$

Las transiciones saltadas en consecuencia son

$\{t_2, t_3, t_4\}$ y de éstas $\{t_2, t_3\}$ son evento-detectable.

Cumpliendo con la Suposición 3.2.

Suposición 3.3

- Verificando $(X - \alpha)^T E_d \geq 2$, tomando X, E_d y α de los puntos anteriores

$$\left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \right)^T \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \geq 2 \quad \text{y} \quad [1 \ 0 \ 0 \ 0 \ 1] \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \geq 2$$

Donde se cumple que $(X - \alpha)^T E_d \geq 2$.

- El T-semiflujo con la transición de falta es

$$(X_2^e)^T = [1 \ 0 \ 0 \ 0 \ 1 \ 1]$$

Así que las transiciones que no son saltadas son $\{t_1, t_5\}$ y ambas

transiciones con evento-detectable, cumpliendo así la Suposición 3.3

Por lo tanto al haberse cumplido las tres suposiciones el sistema es diagnosticable, y es posible diseñar el diagnosticador utilizando el algoritmo 2:

1. Un nuevo lugar p_7 , será agregado a la RPI con el diagnosticador, este lugar estará a cargo del bloqueo de la RPI cuando ocurra una transición de falta. La Figura 4.4 muestra el nuevo lugar p_7 en el diagnosticador.
2. Las transiciones evento-detectable que no son saltadas de la Suposición 3.3 son $\{t_1, t_5\}$, estas transiciones tendrán a los arcos que vendrán hacia ellas desde el nuevo lugar p_7 , como se puede ver en la Figura 4.4.
3. Las transiciones evento-detectable que son saltadas, tomando la Suposición 3.2 son $\{t_2, t_3\}$, estas transiciones tendrán arcos saliendo de ellas hacia el nuevo lugar p_7 , como se puede ver en la Figura 4.4.

4. Una marca inicial será agregada al nuevo lugar p_7 ya que sólo un T-semiflujo fue afectado por la transición de falta.

La transición de falta t_6 sólo afecta el T-semiflujo $(X_1^e)^T$, como resultado sólo un marcado inicial es agregado a él.

El diagnosticador que detectará la falta la primera vez que ésta ocurra finalmente será

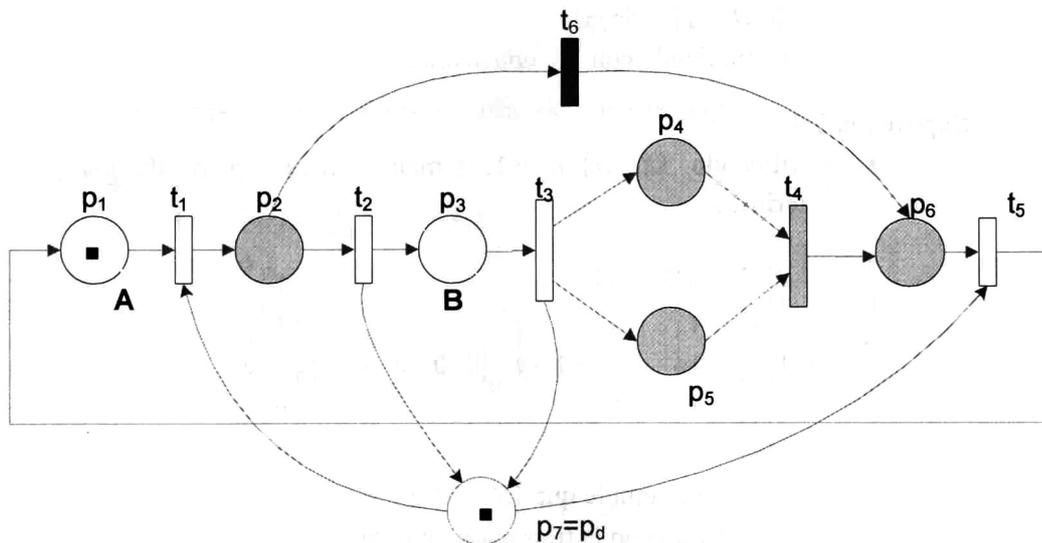


Figura 4.4 Diagnosticador para SMT

5. Figura 4.5, ejemplificará lo que pasa cuando se agregan más de las marcas necesarias al marcado inicial del nuevo lugar p_d .

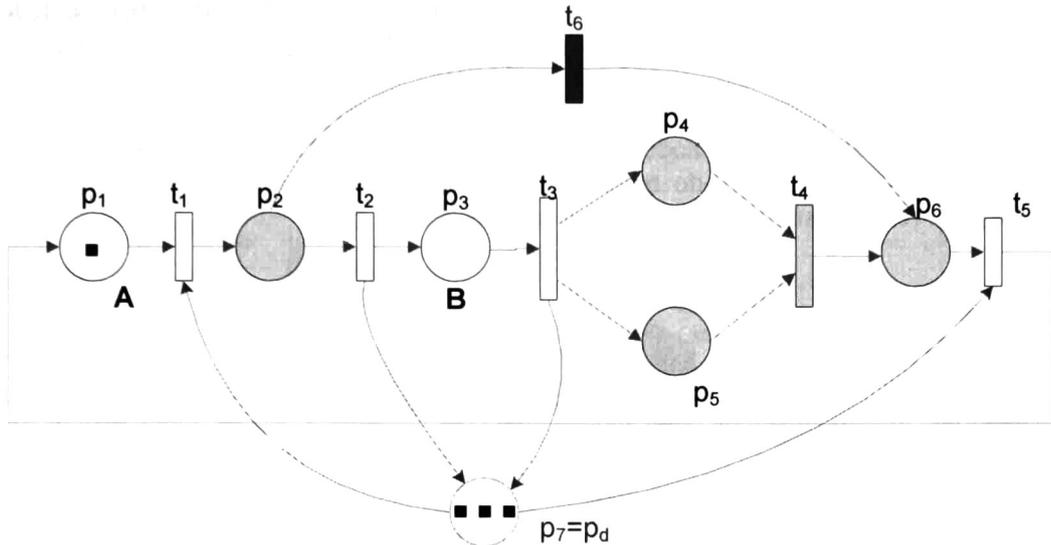


Figura 4.5 Diagnosticador para detectar faltas la segunda vez que ocurran.

Una vez creado el diagnosticador, se puede observar como funciona. El marcado inicial y la secuencia para simular que la transición de falta ocurre la primera vez son:

$$M_0^T = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 3]$$

$$\sigma = t_1 t_6$$

Después de esta secuencia el marcado es

$$M_i^T = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 2]$$

La transición t_5 está todavía habilitada y puede ser disparada, esto es por las dos marcas extra agregadas al nuevo lugar p_7 , si el marcado inicial fuese sólo uno en p_7 , el diagnosticador de RPI sería bloqueada, detectando la falta.

Suponer que se dispara t_5 , el marcado es

$$M_{i+1}^T = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]$$

y disparando de nuevo la secuencia de falta de transiciones $\sigma = t_1 t_6$, el marcado será

$$M_{i+2}^T = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]$$

Con este marcado el diagnosticador de la RPI está bloqueado y así el diagnosticador espera hasta la segunda falta es ahora que la falta ocurre dos veces en el sistema.

4.2 Caso de Estudio 2

El segundo caso de estudio es el sistema de redes presentado en la Figura 4.6, esta red modela a un sistema de comunicación entre servidores. Los mensajes son enviados del servidor S1 y se dividen en dos paquetes, estos paquetes son enviados a través de dos diferentes canales, los dos paquetes viajan un número diferente de nodos que procesarán

la información contenida en los paquetes y los enviarán al siguiente nodo. Finalmente los dos paquetes se combinarán en un router para ser enviados de regreso a S1 para un mayor procesamiento.

El sistema de redes tiene tres nodos (S1, S4 y S8) con un protocolo inalámbrico especial para comunicarse con un nodo maestro que tiene como una de sus tareas seguir los mensajes y verificar su correcta transmisión a través de la red.

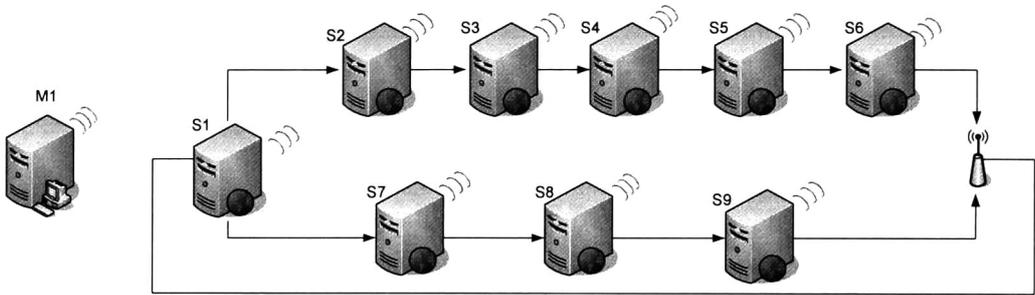


Figura 4.6 Sistema de comunicación entre servidores

El sistema de redes presentando en la Figura 4.6 puede modelarse fácilmente al utilizar RPI, como se muestra en la Figura 4.7, los lugares p_1 , p_4 y p_8 son nodos cuya función de salida es diferente de ε , esto es, porque es posible detectar al mensaje cuando se pasa a través de estos nodos

$$\lambda(p_1) = A, \lambda(p_4) = B \text{ y } \lambda(p_8) = C$$

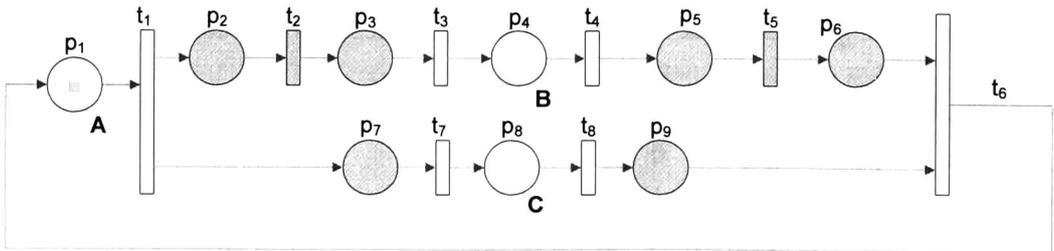


Figura 4.7 Modelo del Sistema de comunicación entre servidores

El siguiente paso es modelar las transiciones de falta en el sistema, en este caso se modela una falta que ocurre cuando el mensaje no es procesado a través de los nodos donde debería de ser procesado y se los salta para llegar a otro nodo rompiendo la secuencia normal de procesamiento del mensaje.

La Figura 4.8 muestra el modelado del sistema extendido para dos faltas, cada falta ocurre en un canal diferente, t_9 es la transición de falta que muestra cuando el mensaje se

salta los nodos S_3, S_4 y S_5 ; mientras que t_{10} es la transición de falta que muestra cuando el mensaje se salta al nodo S_8 .

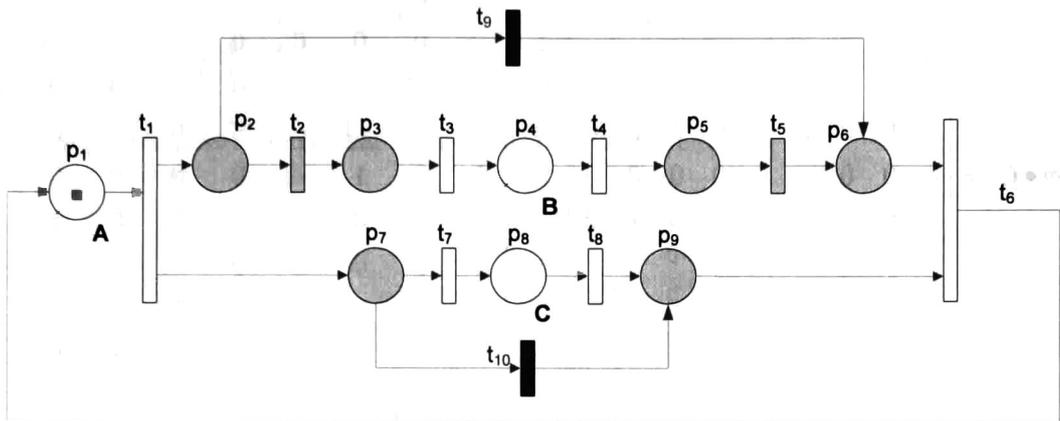


Figura 4.8 Modelo del Sistema de comunicación entre servidores más la falta

Ahora que el modelo del sistema incluye a las transiciones que modelan las faltas, podemos utilizar el algoritmo 1 presentado en la sección 3.2 para verificar que el diagnosticador presentado en este trabajo es posible de ser creado.

1. El sistema es modelado de acuerdo al procedimiento presentado en [25].
2. Este paso indicará cuales transiciones son evento-detectable

$$\varphi \bullet C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

$$\varphi \bullet C = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Dando como resultado que las transiciones evento-detectable son $\{t_1, t_3, t_4, t_6, t_7, t_8\}$.

3. En este paso se encontraran las transiciones que son evento-detectable para el sistema extendido, i.e. el sistema que tiene modeladas también las transiciones de falta.

$$\varphi \bullet C^e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\varphi \bullet C^e = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix}$$

Donde se observa que las transiciones que son evento-detectable son $\{t_1, t_3, t_4, t_6, t_7, t_8\}$

4. Las transiciones que son evento-detectable en el sistema extendido son las mismas que en el sistema normal, lo que significa que las transiciones de falta añadidas no son evento-detectable ni tampoco hacen que las transiciones que eran evento-detectable ahora no lo sean.

Vale la pena notar que no todas las columnas en 1 y 2 son diferentes unas de otras, entonces el sistema no es diagnosticable de entrada-salida de acuerdo a [25, 23], pero con el diagnosticador presentado en este trabajo es posible el diseño de un diagnosticador.

5. Los T-semiflujos del sistema extendido serán calculados con la fórmula

$$C^e \bullet X^e = 0$$

$$C^e \bullet X^e = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 1 \end{bmatrix} \bullet X^e = 0$$

Donde dos transiciones son encontradas:

$$\begin{aligned}(X_1^e)^T &= [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0] \\ (X_2^e)^T &= [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1]\end{aligned}$$

6. El siguiente paso es encontrar los T-semiflujos del sistema,

$$C \cdot X = 0$$

$$C \cdot X = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \cdot X = 0$$

Entonces, el único T-semiflujo es

$$(X_1)^T = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

7. En este paso es necesario verificar que las suposiciones 3.1, 3.2 y 3.3 se cumplan.

Es necesario verificar las suposiciones para todas las transiciones de falta que estén presentes en el sistema.

Suposición 3.1,

$$\bullet \ C(\bullet, t_f) = \sum \alpha_n \cdot C(\bullet, t_n)$$

Ésta se calcula para cada una de las transiciones de falta.

Para la transición t_9 ,

$\alpha_2 = 1$, $\alpha_3 = 1$, $\alpha_4 = 1$ y $\alpha_5 = 1$ mientras que los demás coeficientes son ceros.

Cumpliendo así con $\sum \alpha_n \geq 2$, donde $n = 8$.

Para la transición t_{10}

$\alpha_7 = 1$, $\alpha_8 = 1$, mientras que los demás coeficientes son ceros.

Cumpliendo con $\sum \alpha_n \geq 2$, donde $n = 8$.

- El T-semiflujo que contiene la transición de falta es

$$(X_2^e)^T = [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1]$$

donde se ve que $\{t_1, t_6, t_{10}\}$ son las transiciones que no serán saltadas por la transición t_9 .

- Encontrando el otro T-semiflujo donde estas transiciones son (t_{10} no es considerado debido a que es la transición de falta)

$$(X_1^e)^T = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0]$$

- La diferencia entre $(X_2^E)^T - (X_1^E)^T$ para las transiciones normales darán como resultado las transiciones que son saltadas

$$\{t_2, t_3, t_4, t_5, t_7, t_8\}$$

Debido a que las dos transiciones de falta se encuentran en el mismo T-semiflujo, todas las transiciones que son saltadas por ambas transiciones aparecerán aquí. Debido a esto se tiene que:

- Para la transición de falta t_9 , se tiene que $\{t_3, t_4\}$ son saltadas cumpliendo con esta suposición.
- Para la transición de falta t_{10} se tiene que $\{t_7, t_8\}$ son saltadas cumpliendo también con esta suposición.

Suposición 3.2

- También se verificará para cada falta, entonces $\alpha^T \cdot E_d \geq 2$, tomando E_d del paso 1.

Para la transición t_9

$$[0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0] \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \geq 2$$

y para t_{10}

$$[0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1] \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \geq 2$$

- El T-semiflujo con las transiciones saltadas es

$$(X_1^e)^T = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0]$$

Las transiciones saltadas son $\{t_2, t_3, t_4, t_5, t_7, t_8\}$.

- Para la transición de falta t_9 se tiene que $\{t_3, t_4\}$ son las transiciones evento-detectable que son saltadas cumpliendo con esta suposición.
- Para la transición de falta t_{10} se tiene que $\{t_7, t_8\}$ son las transiciones evento-detectable que son saltadas cumpliendo con esta suposición.

Suposición 3.3

- $(X - \alpha)^T \cdot E_d \geq 2$

Para la transición t_9 , tenemos que

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \geq 2 \text{ y } [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1] \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \geq 2$$

Mientras que para la transición t_{10} tenemos

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \geq 2 \text{ y } [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0] \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} \geq 2$$

Cumpliendo con la Suposición 3.2 para ambas transiciones.

- El T-semiflujo con la transición de falta es

$$(X_2^e)^T = [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1]$$

Entonces, las transiciones que no serán saltadas son $\{t_1, t_6\}$

- Cuando $\{t_1, t_6\}$ son ambas evento-detectable, y las dos transiciones se encuentran en el T-semiflujo donde se encuentra la transición de falta, cumpliendo así con la Suposición 3.3 para ambas de las transiciones de falta t_9 y t_{10} .

Al cumplirse las tres suposiciones entonces es posible el diseño del diagnosticador, este diagnosticador se diseñará utilizando el algoritmo 2 presentado en la sección 3.2.

1. Un nuevo lugar será añadido por cada transición de falta, en este caso dos nuevos lugares serán añadidos al sistema: p_{10}, p_{11} . La Figura 4.9 muestra estos dos nuevos lugares en la RPI.

2. Las transiciones evento-detectable que no son saltadas en la Suposición 3.3 son $\{t_1, t_6\}$, estas transiciones serán de donde los arcos salgan hacia los nuevos lugares p_{10}, p_{11} ; como se muestra en la Figura 4.9.

3. Las transiciones evento-detectable que son saltadas en la Suposición 3.2 son:

- para t_9 se tiene que $\{t_3, t_4\}$
- para t_{10} se tiene que $\{t_7, t_8\}$

Arcos saldrán de estas transiciones y se dirigirán hacia los nuevos lugares p_{10}, p_{11} como se puede ver en la Figura 4.9.

4. Añadir una marca inicial a los nuevos lugares p_{10}, p_{11} por cada T-semiflujo afectada por la transición de falta.

En este caso, la transición de falta t_9 se encuentra únicamente afectando a las transiciones de (X_1^e) , entonces solamente una marca será añadida al lugar p_{10} (como se muestra en la Figura 4.9).

Similarmente para t_{10} porque solamente hay un T-semiflujo afectado el X_1^e y consecuentemente solamente una marca será añadida al lugar p_{11} , como se muestra en la Figura 4.9.

5. Para este ejemplo es necesario detectar la falta cuando ocurre a la primera vez, entonces solamente una marca será añadida al marcado inicial.

Finalmente el diagrama del diagnosticador de faltas intermitentes es,

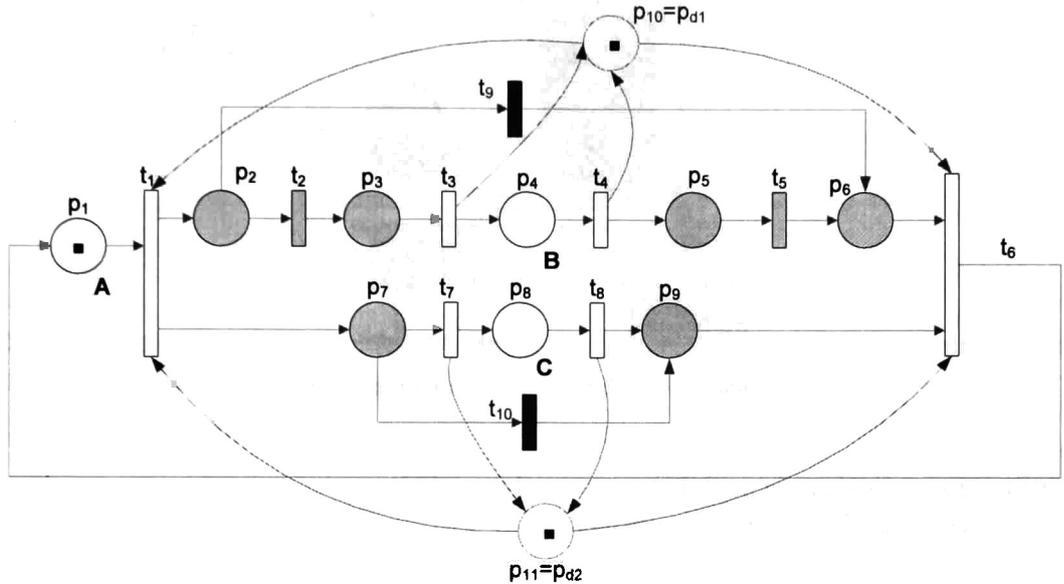


Figura 4.9 Diagnosticador para el Sistema de comunicación entre servidores

4.3 Caso de Estudio 3

Este ejemplo fue tomado de [25], el objetivo de este ejemplo es demostrar que el diagnosticador presentado en este trabajo, también funciona para sistemas diagnosticables de entrada salida como el presentado en [25].

Este caso de estudio muestra un sistema continuo que puede ser modelado como un sistema discreto al discretizar el sistema tomando algunos valores de las variables continuas para cambiar de estado.

El ejemplo presentado aquí es un nivel de líquido en un tanque. Este sistema tiene una válvula V_1 y dos sensores S_1 y S_2 . El nivel en el tanque puede ser controlado al variar V_1 , de tal forma que el nivel en el tanque tenga dos estados: (a) abierto, cuando el porcentaje de flujo de agua es mayor a cero y (c) cerrado, cuando el porcentaje de flujo de agua es cero. Existe también una válvula de salida con un flujo constante de agua para uso general. En el caso de los sensores, "on" en un sensor significa que está en contacto con el agua, mientras que de lo contrario los sensores estarán "off".

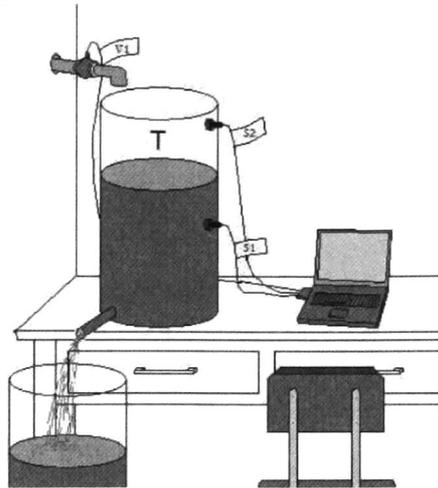
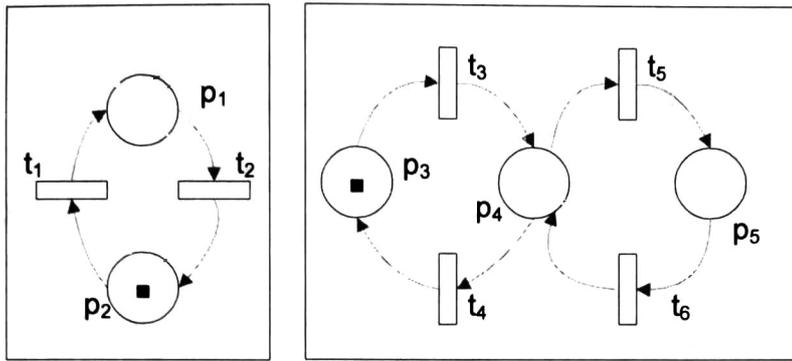


Figura 4.10 Sistema de Nivel de Líquido[25]

Utilizando las técnicas de modelado presentadas en [25],

- Identificación de los componentes:
 - Válvula = V_1
 - Tanque = T
- Variables de Estado:
 - Tareas para la Válvula V_1
 - Tareas para el tanque T
- Rango de las variables:
 - Para la válvula = {abierto, cerrado}
 - Para el tanque = {vacío, medio, alto}
- Los conjuntos anteriores son codificados usando los siguientes conjuntos de lugares:
 - $P_{tareas_V_1} = \{p_1, p_2\}$
 - $P_{tareas_T} = \{p_3, p_4, p_5\}$

Dos modelos de RP representan los componentes identificados del sistema y son representados en la Figura 4.11.



Válvula

Tanque

Figure 4.11 Componentes en el sistema del Tanque

Después de tener el modelo en RP con los componentes del sistema, en la Figura 4.12 se muestra el resultado de la composición síncrona y permisiva entre los módulos.

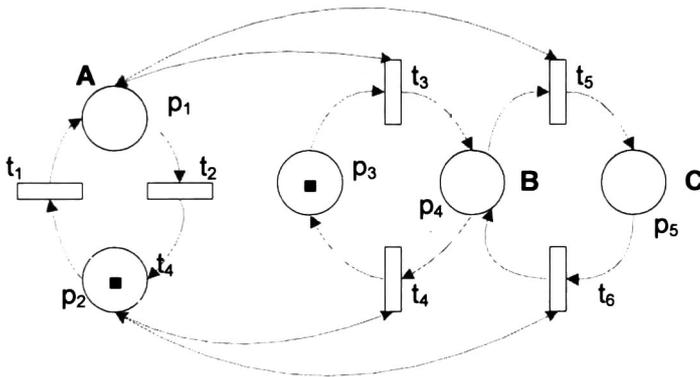


Figura 4.12 Modelo del Tanque

Una vez que se tiene el modelo del tanque, las funciones para los símbolos de salida son propuestas, en este caso en la Figura 4.12, los lugares p_1 , p_4 y p_5 son medibles

$$\varphi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Al terminar de tener el modelo del sistema, el siguiente paso es la identificación de las faltas que se van a analizar. La Figura 4.13, muestra la falta modelada para el sistema del tanque.

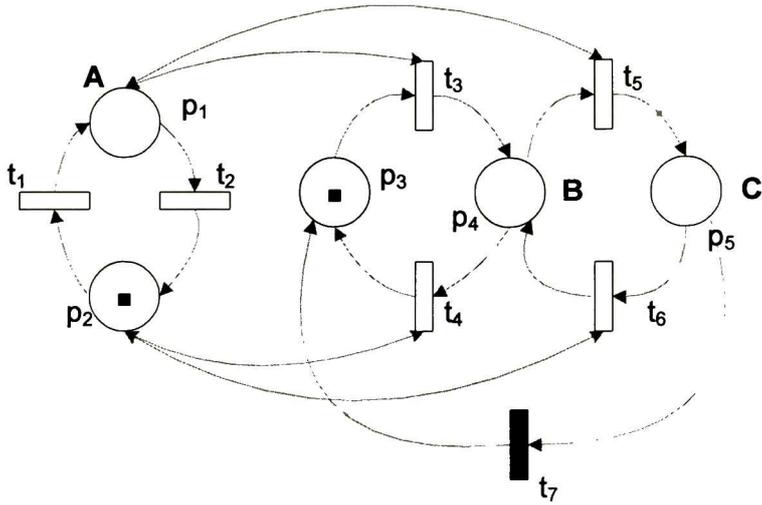


Figura 4.13 RPI modelo del Tanque junto con la transición de falta

Al tener el modelo extendido del sistema, que es el sistema normal más el comportamiento de la falta en el sistema, se puede utilizar el algoritmo 1 de la sección 3.2 para verificar si es posible la creación del diagnosticador presentado en este trabajo.

1. El sistema es modelado de acuerdo a [25]
2. Este paso es encontrar las transiciones que son evento-detectable del sistema normal.

$$\varphi \bullet C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\varphi \bullet C = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Las transiciones diagnosticables después de calcularlas $\{t_1, t_2, t_3, t_4, t_5, t_6\}$.

Todas las transiciones son evento-detectable entonces el sistema entero es sistema detectable.

3. Las transiciones evento-detectable del sistema extendido son

$$\varphi \bullet C^e = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix}$$

$$\varphi \bullet C^e = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix}$$

Las transiciones evento-detectable son entonces $\{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$. Teniendo que el sistema es evento-detectable.

4. Las transiciones evento-detectable del sistema extendido son diferentes que para el sistema normal. De hecho hay más transiciones en el sistema extendido, lo que significa que la transición de falta es evento-detectable también.

Al ser la transición evento-detectable, el diagnosticador verá el momento en que esta transición se dispare, pero el ejercicio de encontrar un diagnosticador se realizará de todas formas para verificar la funcionalidad del mismo aún en sistemas evento-detectable.

5. Los T-semiflujos del sistema extendido usando la fórmula

$$C^e \bullet X^e = 0$$

Pueden ser calculados como

$$C^e \bullet X^e = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \bullet X^e = 0$$

El siguiente conjunto de T-semiflujos es el resultado

$$X^e = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \right\}$$

6. El siguiente paso es encontrar los T-semiflujos del sistema

Por medio de la fórmula $C \bullet X = 0$

$$C \bullet X = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \bullet X = 0$$

El siguiente conjunto de T-semiflujos es el resultado

$$X = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}$$

7. Verificando si las suposiciones 3.1, 3.2 y 3.3 se cumplen tenemos,

Suposición 3.1

$$\bullet C(\bullet, t_f) = \sum \alpha_n \cdot C(\bullet, t_n)$$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ -1 \end{bmatrix} = \alpha_1 \cdot \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_2 \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \alpha_3 \cdot \begin{bmatrix} -1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \alpha_4 \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \alpha_5 \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} + \alpha_6 \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix},$$

teniendo $\alpha_1 = 1$, $\alpha_2 = 1$, $\alpha_4 = 1$ y $\alpha_6 = 1$, y las demás son cero. Con estos coeficientes se cumple que $\sum \alpha_n \geq 2$, porque

$$\alpha_1 + \alpha_2 + \alpha_4 + \alpha_6 \geq 2$$

- Hay dos T-semiflujos donde se encuentra la transición de falta

$$X_7^e = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \text{ y } X_8^e = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

De donde se ve que de X_7^e las transiciones que no serán saltadas son $\{t_3, t_5\}$; mientras que para X_8^e las transiciones que no serán saltadas por la transición de falta son $\{t_1, t_2, t_3, t_5\}$.

- Encontrando el otro T-semiflujo donde también se encuentran estas transiciones (t_7 no se considera pues es la transición de falta)

$$X_4^e = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}; \quad X_6^e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix};$$

- La diferencia entre $(X_4^E)^t, (X_6^E)^t$ y $(X_7^E)^t, (X_8^E)^t$ para las transiciones normales nos dará las transiciones que son saltadas.
- En este caso tenemos dos conjuntos de transiciones que se necesitan considerar, $\{t_4, t_6\}$ y $\{t_1, t_2, t_4, t_6\}$

Existen dos diferentes conjunto de transiciones porque con la transición de falta la RPI puede ciclarse a través de diferentes ciclos de transiciones.

Para la suposición 1, solamente pide tener al menos dos transiciones saltadas por la transición de falta, y en este caso para ambos conjuntos tenemos al mínimo dos transiciones, cumpliendo así el requisito.

Suposición 3.2

- Verificando $\alpha^T \cdot E_d \geq 2$, tomando E_d del paso 1 y los coeficientes α_n calculados en la suposición anterior, se tiene

$$[1 \ 1 \ 0 \ 1 \ 0 \ 1] \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \geq 2$$

- Los T-semiflujos con la transición de falta son,

$$X_4^e = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}; \quad X_6^e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix};$$

Las transiciones saltadas son entonces $\{t_4, t_6\}$ y $\{t_1, t_2, t_4, t_6\}$

Debido a que se cumple que el sistema es diagnosticable entrada y salida, entonces todas las transiciones son evento-detectable y la Suposición 3.2 está cumplida.

Suposición 3.3

- Verificando $(X - \alpha)^T E_d \geq 2$, tomando X, E_d y α de los puntos anteriores,

$$\begin{pmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{pmatrix}^T \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \geq 2 \quad \text{y} \quad [1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1] \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \geq 2$$

Se cumple que $(X - \alpha)^T E_d \geq 2$.

- Los T-semiflujos con las transiciones de falta son,

$$X_7^e = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \text{y} \quad X_8^e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Por lo tanto, las transiciones que no serán saltadas son $\{t_3, t_5\}$ y $\{t_1, t_2, t_3, t_5\}$.

- Debido a que el sistema es diagnosticable de entrada salida, entonces las transiciones son evento-detectable y se cumple la Suposición 3.3,

Las tres suposiciones se cumplen, lo que indica que el diagnosticador puede ser creado, para esto se utiliza el algoritmo 2 de la sección 3.2

1. Un nuevo lugar se añadirá al Diagnosticador porque solamente estamos modelando una falta, este nuevo lugar puede verse en la Figura 4.14.
2. Las transiciones evento-detectable que no son saltadas son $\{t_1, t_2, t_3, t_5\}$, estas transiciones recibirán los arcos provenientes del nuevo lugar p_6 , como se observa en la Figura 4.14.
3. Las transiciones evento-detectable que pueden ser saltadas debido a la Suposición 3.2 son $\{t_1, t_2, t_4, t_6\}$ y tendrá arcos saliendo de estas transiciones dirigiéndose al lugar nuevo p_6 , como se ve en la Figura 4.14.
4. Se añade una marca inicial al nuevo lugar por cada T-semiflujo que es afectado por la transición de falta. En este caso existen dos T-semiflujos que son afectados por la transición de falta y dos marcas iniciales serán para el nuevo lugar p_6 , como se puede ver en la Figura 4.14.

Los dos T-semiflujos afectados por la transición de falta son,

$$X_7^e = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \text{ y } X_8^e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

5. Para este ejemplo, es necesario detectar la falta la primera vez que ocurrió y por esta razón no se incluirán más marcas en el diagnosticador.

El diagnosticador finalmente será,

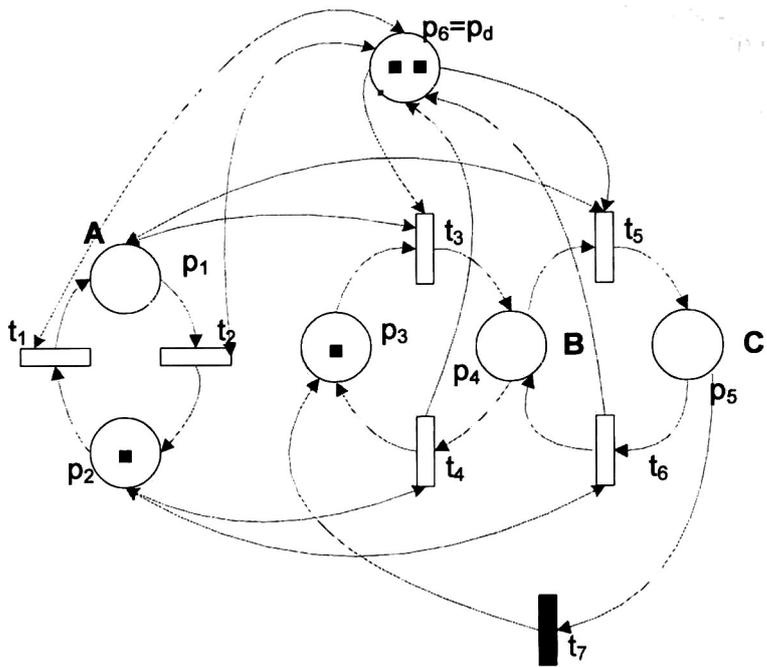


Figura 4.14 Diagnosticador del Sistema del Tanque

Capítulo 5

Conclusiones

En el presente trabajo se propone un método para la creación de un diagnosticador de faltas intermitentes utilizando Redes de Petri Interpretadas. Se desarrolló un diagnosticador con faltas intermitentes, donde la diferencia principal con el trabajo presentado en [25] es que el diseño del diagnosticador no tiene como restricción que el sistema sea evento-detectable. Las restricciones del diagnosticador presentado en este trabajo se presentan en el modelado del sistema, consecuentemente en el tipo de sistema que se está analizando.

Estas restricciones se presentan en tres suposiciones que tienen que ver con el modelado del sistema, específicamente en las transiciones, será posible construir un diagnosticador para faltas intermitentes si estas tres suposiciones se cumplen. Las suposiciones restringen a las transiciones que se ven afectadas por la transición de falta. La primera suposición restringe a que la transición de falta tiene que saltar a más de una transición, esto es debido a que si la transición de falta afectara a una sola transición con la misma salida entonces es imposible distinguir entre la transición de falta y la transición normal del sistema. La segunda y tercera suposiciones restringen a las transiciones que se encuentran saltadas por la transición de falta y a las transiciones que no son afectadas por la transición de falta. Estas dos restricciones son necesarias debido a que al hacer que ciertas transiciones sean evento-detectable el diagnosticador reconocerá cuando la ocurrencia de la transición de falta.

Por último el diagnosticador presentado en este trabajo permite la detección de las faltas la primera vez que ocurren, o cuenta también con la habilidad de contar faltas y reportar la falta hasta que un cierto número de ellas han pasado. Esta funcionalidad es deseada para aquellos sistemas que son tolerantes a cierto número de faltas, en los cuáles las acciones correctivas o preventivas se llevan a cabo después de detectar un cierto número de faltas, previo a la ocurrencia de fallas.

Trabajo Futuro

Este trabajo presenta faltas intermitentes que necesitan cumplir suposiciones 3.1, 3.2 y 3.3 para la creación del diagnosticador.

Al cumplirse estas limitaciones las transiciones de falta quedan restringidas en su modelado. Como trabajo futuro se considerará el modelado de un mayor número de transiciones de faltas y la correspondiente modificación del diagnosticador. Así como la realización de más pruebas con sistemas más grandes y con faltas intermitentes conectadas entre ellas.

Referencias

- [1] F. Basile, P. Chiacchio, and G. De Tommasi. "An efficient approach for on-line diagnosis of discrete event systems". *IEEE Transactions on Automatic Control*, 2008
- [2] A. Benveniste, E. Fabre, S. Haar and C. Jard. "Diagnosis of Asynchronous Discrete-Event Systems: A Net Unfolding Approach". *IEEE Transactions on Automatic Control*, Vol. 48. No. 5. May 2003,
- [3] A. Burns and A. Wellings. *Real-Time Systems and Programming Languages*. Eddison Wesley, 2001.
- [4] M. P. Cabasino, "Diagnosis and Identification of Discrete Event Systems using Petri Nets." *Ph.D. Thesis*, University of Cagliari, 2009.
- [5] O. Contant, S. Lafortune, D. Teneketzis. "Failure Diagnosis of Discrete Event Systems: The case of Intermittent Faults" *Proceedings of the 41st IEEE Conference on Decision and Control*, pp 4006-4011., Las Vegas, Nevada USA, December 2002.
- [6] M. Dotoli, M.P. Fanti, and A.M. Mangini. "Fault detection of discrete event systems using Petri nets and integer linear programming." In *Proceedings of 17th IFAC World Congress*, Seoul, Korea, July 2008.
- [7] M. Dotoli, M.P. Fanti, and A.M. Mangini. "Fault monitoring of discrete event systems by first order hybrid Petri nets" Workshop on Petri Nets and Agile Manufacturing, a satellite event of the 29th *International Conference on Application and Theory of Petri Nets and Other Models of Concurrency*, Xi'an, China, June 2008.
- [8] M. P. Fanti, C. Seatzu. "Fault diagnosis and identification of discrete event systems using Petri nets". *Proceedings of the 9th International Workshop on Discrete Event Systems* pp 432-435. Goteborg, Sweden, May 28-30, 2008.
- [9] S. Genc and S. Lafortune. "Distributed diagnosis of place-bordered Petri nets." *IEEE Transactions on Automation Science and Engineering*, 4(2):206–229, 2007.
- [10] A. Giua and C. Seatzu. "Fault detection for discrete event systems using Petri nets with unobservable transitions." In *Proceedings 44th IEEE Conference on Decision and Control*, Seville, Spain, pp 6323-6328. December 2005.
- [11] S. Hashtrudi Zad, R.H. Kwong, and W.M. Wonham. "Fault diagnosis in discrete-event systems: framework and model reduction." *IEEE Transactions on Automatic Control*, 48(7):1199–1222, July 2003.
- [12] J. E. Hopcroft, J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.
- [13] W.S. Lee, D.L. Grosh, F.A. Tillman, and C.H. Lie. "Fault tree analysis, methods, and applications - A review". *IEEE Transactions Reliability*, 34:194–203, 1985.
- [14] D. Lefebvre and C. Delherm. "Diagnosis of DES with Petri net models." *IEEE Transactions on Automation Science and Engineering*, 4(1):114–118, 2007.
- [15] F. Lin. *Diagnosability of discrete event systems and its applications*. Discrete Event Dynamic Systems, 4(2):197–212, 1994.
- [16] M.E. Meda, A. Ramírez-Treiviño and A. Malo. "Identification in discrete event systems". *Proceedings of the IEEE Conference on Systems, Man & Cybernetics*, pp. 740-745, 1998.

- [17] T. Murata, "Petri Nets: Properties, analysis and applications", *Proceedings of the IEEE*, vol. 77, pp. 541 – 580, 1989.
- [18] A. Ocegüera, T. Basten, L. Somers, S. Hulsenboom. "Real-Time Step Motor Emulation for Hardware-in-the-Loop Simulation". In *42nd Summer Computer Simulation Conference, SCSC 2010, Proceedings*, pages 306-313. Ottawa, Canada, 11-14 July, 2010. SCS, 2010. © SCS.
- [19] D.N. Pandalai and L.E. Holloway. "Template languages for fault monitoring of timed discrete event processes." *IEEE Transactions on Automatic Control*, 45(5), 2000.
- [20] C.A. Petri, "Kommunikation mit automaten".
- [21] J. Prock, "A new technique for fault detection using Petri nets", *Automatica*, vol. 27, No. 2, pp. 239-245, 1991.
- [22] A. Ramírez-Treviño, E. Ruiz-Beltrán, I. Rivera-Rangel, and E. López-Mellado. Diagnosability of Discrete Event Systems. A Petri Net Based Approach. *Proceedings of the IEEE International Conference on Robotic and Automation*, pp. 541-546, 2004.
- [23] A. Ramírez-Treviño E. Ruiz-Beltrán, I. Rivera-Rangel, and E. López-Mellado. "Online fault diagnosis of discrete event systems. A Petri net based approach" *IEEE Transactions on Automation Science and Engineering*, 4(1):31–39, 2007.
- [24] I. Rivera-Rangel, A. Ramírez-Treviño, L.I. Aguirre-Salas and J. Ruiz-Leon. Geometrical characterization of Observability in Interpreted Petri Nets. *Kybernetika*, vol. 41, 553-574, 2005.
- [25] E. Ruiz Beltrán; "Esquemas de Diagnóstico de Faltas para Sistemas de Eventos Discretos", *Ph.D. thesis. CINVESTAV del RPI*, Unidad Guadalajara, Noviembre 2007.
- [26] M. Sampath and S. Lafortune. "Active diagnosis of discrete-event systems." *IEEE Transactions on Automatic Control*, 43(7):908–929, 1998
- [27] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohidden, and D. Teneketzis. "Diagnosability of discrete event systems." *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.
- [28] Y. Wu and C.N. Hadjicostis. "Algebraic approaches for fault identification in discrete-event systems." *IEEE Transactions Robotics and Automation*, 50(12):2048–2053, 2005.

Apéndice A

Los algoritmos 1 y 2 del Capítulo 3.2 que requieren la búsqueda de T-semiflujos, resultan ser NP completos, por lo que su aplicación no es realista en la práctica.

Afortunadamente, el algoritmo busca los T-semiflujos para encontrar las transiciones de comportamiento normal que no se ejecutaron debido a la ocurrencia de una falta.

Así que, aprovechando esta característica, podemos buscar estas transiciones como aquellas que forman un T-flujo con la transición de falta (el T-flujo puede tener elementos negativos). Para ver cómo sería este algoritmo supongamos que se tiene la red de la Figura A.1.

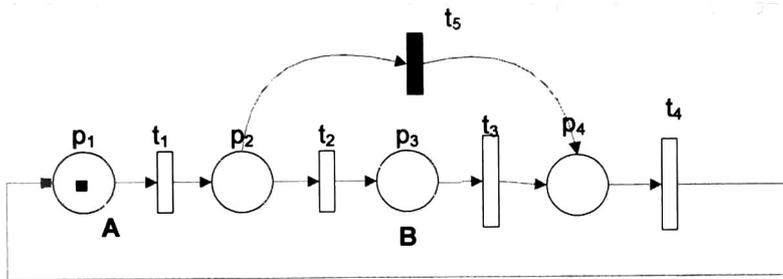


Figura A.1 Sistema para encontrar las transiciones saltadas.

Para encontrar las transiciones que no se ejecutaron debido a la ocurrencia de la falta se debe resolver el siguiente problema de programación lineal (con m igual al número de transiciones en el sistema).

$$\text{Minimizar } x(1)+x(2)+\dots+x(m).$$

s.a.

$$CX=0$$

$$x(\text{falta})=1$$

$$x(\text{anterior a falta})=0$$

Aplicando dicho algoritmo en la Figura A.1 se tiene:

$$C \bullet X = 0 \Rightarrow \begin{bmatrix} -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \bullet X = 0$$

$$\begin{bmatrix} -1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 1 \end{bmatrix} = 0,$$

$$X = \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 1 \end{bmatrix} \text{ donde } x(t_f) = x(t_5) = 1, \text{ } x(\text{antes de } t_f) = x(t_1) = 0$$

Las transiciones $\{t_2, t_3\}$ en el T-flujo son las transiciones saltadas por la transición de falta.



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N. UNIDAD GUADALAJARA

"2010, Año de la Patria, Bicentenario del Inicio de la Independencia
y Centenario del Inicio de la Revolución"

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Diagnóstico de Faltas Intermitentes para Sistemas de Eventos
Discretos

del (la) C.

Alvaro OCEGUERA VALENZUELA

el día 04 de Febrero de 2011.

Dr. Deni Librado Torres Román
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. José Javier Ruíz León
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Antonio Ramírez Treviño
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara



CINVESTAV - IPN
Biblioteca Central



SSIT0010075