



Generación Determinística de Lenguajes Legales para Sistemas de Eventos Discretos

Muñoz, Doyra Mariela^a, Correcher, Antonio^b, García, Emilio^b, Morant, Francisco^b

^aGrupo de Automática Industrial, Universidad del Cauca, Popayán, Colombia.

^bInstituto de Automática e Informática Industrial, Universitat Politècnica de València, Camino de Vera, nº14, 46022, Valencia, España.

Resumen

En este artículo se propone una red de Petri, interpretada, estocástica, (st-IPN), como modelo para representar el lenguaje regular obtenido a partir de la combinación de señales de entrada - salida, en un sistema de eventos discretos (SED) en lazo cerrado. Las señales de entrada, son las señales externas que afecten al sistema y las órdenes de control emitidas por el controlador a la planta y las señales de salida son las respuestas de los sensores a las órdenes de control. La st-IPN propuesta, es un generador determinista del lenguaje legal del sistema, capaz de representar secuencias de eventos temporizadas de naturaleza estocástica. El modelo propuesto puede ser aplicado a sistemas de gran escala, a partir de la división del sistema en subsistemas, ya que el modelo global puede ser encontrado con base en la composición de los modelos de los subsistemas.

Palabras Clave:

Modelado de sistemas de eventos discretos, Redes de Petri, Observabilidad temporal.

1. Introducción

En (Guasch et al., 2005) los autores definen un sistema como una colección de objetos o entidades que interactúan entre sí para alcanzar un cierto objetivo y el estado de un sistema como un conjunto mínimo de variables necesarias para caracterizar o describir todos aquellos aspectos de interés del sistema en un cierto instante de tiempo. El problema de modelado de un sistema dinámico se puede ver, de acuerdo a Lunze (1998) como: “Dado un sistema dinámico S , con un conjunto de preguntas B , sobre su comportamiento, encontrar una representación M que ayude a responder a las preguntas dadas; entonces, M se llama el modelo de S ”. Desde este enfoque se precisa que dado que el modelo se utiliza para resolver un problema, éste no necesariamente es único.

El modelado de sistemas de eventos discretos (SED) ha sido estudiado desde hace muchos años, tanto por la academia como la industria; puesto que la evolución exponencial de las tecnologías industriales de computación, comunicación y de sensores ha traído nuevos sistemas dinámicos, complejos y flexibles que se caracterizan por acontecimientos discretos, algunos controlados y otros no, algunos observados por sensores y otros no,

algunos aparecen de forma automática a partir de los procesos físicos remotos y algunos se generan manualmente por los usuarios. Los SED tienen un comportamiento que se representa por una secuencia finita o infinita de estados delimitados por eventos que ocurren de manera asíncrona o síncrona (Cassandras y Lafortune, 2008).

Varias propuestas de modelado han sido dadas usando diferentes enfoques: autómatas, máquinas de estado finito, Redes de Petri (PN) y cadenas de Markov. En el caso de los sistemas de manufactura, donde pueden existir dinámicas determinadas por eventos discretos, si se utilizan autómatas como formalismo de modelado, las dimensiones de tales modelos pueden resultar intratables para su análisis en tiempo real, particularmente cuando se aplica para detectar y diagnosticar fallos. (González-Miranda, 2014).

Las PN han sido reconocidas como un modelo apropiado para describir SED (Ichikawa y Hiraishi, 1988; Girault y Valk, 2003; Silva, 1993), particularmente cuando se trata de sistemas asíncronos (Fanti et al., 2012; Hu et al., 2012). Las PN incorporan la noción de estado distribuido y de reglas que permiten pasar de un estado a otro, lo cual captura tanto el comportamiento estático como el comportamiento dinámico.

Teniendo en cuenta que un sistema en lazo cerrado es una interrelación de señales, se han desarrollado propuestas de modelado bajo PN que se han denominado modelos de sistemas de condiciones; estos modelos se representan mediante PN con entradas y salidas explícitas llamadas condiciones (Ashley y

* Autor en correspondencia.

Correos electrónicos: mamunoz@unicauca.edu.co (Muñoz, Doyra Mariela), ancorsal@upv.es (Correcher, Antonio), egarciam@isa.upv.es (García, Emilio), fmorant@isa.upv.es (Morant, Francisco)

Holloway, 2004); esas representaciones permiten la interacción entre subsistemas así como la interacción de la planta con el controlador (Holloway et al., 2000).

(Sreenivas y Krogh, 1991) definen una clase de sistemas dinámicos de eventos discretos (DEDS), en tiempo continuo, llamados sistemas condición/evento (C/E) y también definen modelos para estos sistemas basados en una extensión de las PN y los llaman C/E PN (por sus siglas en inglés) con dos tipos de valores de señales discretas de entrada - salida (E/S): condiciones de señales y señales de eventos; las señales de condición E/S llevan información del estado y las señales de evento E/S llevan información de transición de estado. La idea general es modelar un sistema como un conjunto de módulos con un comportamiento dinámico particular y su interconexión a partir de sus señales, (Patil et al., 2012). Las condiciones y los eventos de entrada pueden ser conectados con algunas transiciones dentro del módulo por condiciones y arcos de eventos. Los módulos de lugares pueden ser conectados a las condiciones de salida por condiciones de arco, y las transiciones a los eventos de salida por eventos de arco. Estos conceptos proveen una base para un enfoque de composición para construir modelos de sistemas grandes a partir de componentes pequeños, (Patil et al., 2012). Este tipo de propuestas de modelado se han desarrollado básicamente para ser utilizadas en verificación de propiedades como: alcanzabilidad, acotabilidad y vivacidad; aunque Ashley (2004), presenta una propuesta basada en sistemas de condición para realizar diagnóstico de fallos.

Otra forma que ha sido ampliamente estudiada en el modelado de SED utilizando PN son las redes de Petri temporizadas (t-PN), que fueron introducidas por Merlin y Farber (1976); Berthomieu y Diaz (1991). Las t-PN, son PN extendidas con tiempo en las transiciones; poseen la expresividad necesaria para modelar sistemas en lazo cerrado y presentan beneficios para realizar análisis de verificación de propiedades. (Berthomieu et al., 2004; Gardey et al., 2005). Para modelar sistemas complejos, se han realizado propuestas de división en subsistemas; pero la composición de PN no ha sido trivial. Berthomieu et al. (2006) presentan una propuesta de composición, siempre que las transiciones sean sincronizadas. En Peres et al. (2011) se presenta una solución que facilita la especificación de los sistemas dependientes del tiempo, en un planteamiento de composición de componentes modelados como t-PN. En la mayor parte de las t-PN las transiciones tienen un tiempo asociado, por lo que la evolución queda determinada por el disparo de las transiciones. Una transición se puede disparar, si el tiempo está dentro de su intervalo de disparo estático (Boucheneb y Hadjidj, 2006). En Salum (2008) se propone una PN conducida por eventos en lugar del paso del tiempo, denominada EDPNs, (por sus siglas en inglés). Las EDPNs emplean arcos inhibidores, arcos de ensayo y lugares con capacidad finita; como el tiempo es considerado un estado (marcado) del sistema (PN), su paso se corresponde con el juego de marcas que genera un árbol de alcanzabilidad. Al introducir arcos inhibidores existen muchas dificultades para utilizar las herramientas de análisis cualitativo de las PN.

Por otro lado, como la transición de estados en un SED es un evento (Ramadge y Wonham, 1989), éste se puede modelar a

partir de su lenguaje. El lenguaje que modela el comportamiento de un sistema se puede representar por su expresión regular, (Sampath et al., 1996), que es una síntesis de todas las combinaciones válidas de eventos. En Ramadge y Wonham (1989) un SED es modelado como un autómata y su comportamiento es representado por su lenguaje; pero este tipo de modelos pueden presentar inconvenientes en sistemas con gran cantidad de dispositivos debido a la explosión combinatoria de estados; suele ser más conveniente emplear formalismos con mayor capacidad de condensación de estados, como las PN.

Una PN tienen un poder más descriptivo que las máquinas de estados finitos en el sentido que el conjunto del lenguaje de la PN, es un superconjunto de los lenguajes regulares y permite un modelado más conciso, (Kumar y Holloway, 1996; Valk y Vidal-Naquet, 1981). Algunas propuestas para modelar SED bajo el lenguaje en PN se pueden ver en Nakamura et al. (1998); Sreenivas (1993, 2006); Gaubert y Giua (1996) o para encontrar ciclos en sistemas de estados repetitivos en Desel (2013).

Respecto a la utilización de PN para identificación y diagnóstico, varias propuestas de modelado se han generado utilizando PN extendidas como: PN interpretadas en Cabasino et al. (2011); Dotoli et al. (2008a); Estrada-Vargas et al. (2012) o como PN temporizadas en Basile et al. (2011).

1.1. Enfoque del trabajo

La aplicación de la presente propuesta de modelado es el diagnóstico de fallos en SED; por lo tanto ésta debe permitir explorar los eventos observables para detectar la ocurrencia de eventos no-observables. Los eventos no-observables son eventos que ocurren por algún tipo de fallo o son eventos que generan cambios en el sistema sin que sean registrados por los sensores.

Como se describió en la introducción, de acuerdo a Lunze (1998), un modelo debe generarse para resolver preguntas sobre el comportamiento del sistema al que representa; por lo tanto, las preguntas a solucionar teniendo en cuenta que el modelo se genera para realizar diagnóstico son:

- Dado el lenguaje legal de un SED, ¿cómo representarlo bajo una PN de tal manera que el lenguaje legal sea igual al lenguaje de la PN?.
- ¿Qué tipo de eventos no-observables pueden ser explicados? y bajo ¿qué condiciones?, a partir de algunas secuencias de eventos observados.

Para solucionar las preguntas se propone un generador determinista del lenguaje legal de un sistema, basado en una red de Petri interpretada temporizada (st-IPN), cuyo lenguaje generado es el mismo lenguaje legal y representa el conjunto de todas las posibles secuencias de eventos del sistema en estado libre de fallos.

Para sistemas de gran escala, se propone dividirlo en subsistemas. El sistema global se modela a partir de modelos locales (en cada subsistema) que representan las mismas características que el modelo global. Un sistema se define como en la Figura 1.



Figura 1: Sistema a Modelar

En la Figura 1, se puede observar que el sistema es una interrelación de señales de E/S en lazo cerrado. Es un sistema de condiciones en donde la ocurrencia de eventos es estocástica y se consideran como señales de entrada tanto las señales externas que afecten al sistema como las órdenes de control y como señales de salida, la respuesta de los sensores.

El modelado de un SED como st-IPNs presenta ventajas frente a otras propuestas porque es escalable, es decir, modela sistemas independiente de su tamaño, genera modelos deterministas, permite realizar composición sincronizada a través del lenguaje generado y se puede aplicar a problemas de diagnóstico basándose en las propiedades de observabilidad en sistemas estocásticos, que se demuestran en este trabajo.

El presente artículo describe en la sección 2 las definiciones de operaciones básicas sobre teoría de lenguaje y la simbología a utilizar en las secciones siguientes, así como los requerimientos del sistema a modelar; en la sección 3 se relacionan los fundamentos teóricos respecto a PN necesarios para el diseño de la propuesta de modelado; en la sección 4 se define la propuesta de modelado como una st-IPN; en la sección 5 se presenta la propuesta de modelado para sistema de gran escala; en la sección 6 se valida la propuesta a partir de un ejemplo de aplicación a un proceso y finalmente se describen las conclusiones y trabajos futuros.

2. Eventos y Lenguaje del sistema

Una manera de estudiar el comportamiento de los SED es emplear la teoría de lenguajes. Un evento es la representación de un cambio instantáneo en alguna parte del sistema, puede caracterizarse por un valor y un instante en el que ocurre, (Giua, 2013); un evento constituye una letra y el conjunto de eventos es el alfabeto; una secuencia de eventos es una palabra, (Cassandras y Lafourche, 2008). El conjunto de eventos, Ω , también incluye el evento nulo, ε , que modela la situación en la que no ocurre ningún evento, (Correcher, 2005).

Sea Ω un conjunto de eventos y Ω^* un conjunto finito de palabras sobre Ω ; un lenguaje \mathcal{L} es un subconjunto de Ω^* . Dadas dos palabras s y s' , ss' es la concatenación de s y s' . $|s|$ es el tamaño de la palabra s .

Definición 1. Operación de Proyección.

Dados dos conjuntos de eventos Ω_1, Ω tales que $\Omega_1 \subseteq \Omega$, se define la operación de proyección de palabras, $P_{\Omega_1} : \Omega^* \rightarrow \Omega_1^*$, así $P_{\Omega_1}(\varepsilon) = \varepsilon$ siendo ε un evento nulo, y para $a \in \Omega$, $s \in \Omega^*$, $P_{\Omega_1}(as) = aP_{\Omega_1}(s)$ si $a \in \Omega_1$ y $P_{\Omega_1}(as) = P_{\Omega_1}(s)$ si $a \notin \Omega_1$, (Cassez y Tripakis, 2008).

Esta operación permite eliminar palabras en Ω^* que no pertenecen a Ω_1^* . Dentro del contexto del presente trabajo es importante definir la proyección sobre lenguajes compuestos.

Definición 2. Operación de Proyección sobre conjuntos de eventos compuestos P_C .

Dados dos conjuntos de eventos Ω_p, Ω_q y dados dos eventos e_p y e_q , tal que, $e_p \in \Omega_p$ y $e_q \in \Omega_q$, un evento compuesto ω es la concatenación de e_p y e_q , es decir, $\omega = e_p e_q$, entonces $\Omega = \Omega_p \Omega_q$ es un conjunto de eventos compuestos. La operación P_C se define como: $P_C : \Omega \rightarrow \Omega_q$, es $P_{C_{\Omega_q}}(\varepsilon) = \varepsilon$; $P_{C_{\Omega_q}}(e_p e_q) = P_{\Omega_q}(e_p) P_{\Omega_q}(e_q)$.

Esta operación elimina palabras en Ω^* que no pertenecen a $\Omega_p^* \Omega_q^*$ y eventos Ω_p (o Ω_q) que no pertenecen a Ω_p^* (o Ω_q^*).

El sistema a considerar en el contexto del presente trabajo tiene una estructura basada en la lectura de señales de E/S, como el mostrado en la Figura 1; el comportamiento del sistema se ve afectado por señales de entrada tales como: externas al controlador, externas a la planta e internas al controlador (órdenes de control) agrupadas en: $Entradas = \{EExC, EExP, EInP\}$ y por señales de salida representadas por las lecturas sensoriales como: $Salidas = \{sr_1, \dots, sr_{n_{sr}}\}$, siendo n_{sr} el número de lecturas sensoriales. Todas las señales se considerarán binarias.

Definición 3. Representación de señales binarias.

Dado un conjunto de ns señales, cuyos valores se representan en cadenas de ceros o unos; la notación x_{dec} representa la combinación de una cadena tal que x es el nombre de la señal y dec es la representación decimal de la cadena. El conjunto X agrupa todas las combinaciones posibles de las señales; es decir $X = \{x_0, \dots, x_{2^{ns}-1}\}$.

Por ejemplo: si el número de señales binarias es $ns = 3$, entonces $X = \{x_0, x_1, \dots, x_7\}$; donde $x_0 = [000] \dots x_7 = [111]$.

Las entradas externas al controlador, $EExC = \{ec_1, \dots, ec_{n_{ec}}\}$ con cardinalidad n_{ec} tales como comandos del SCADA o los requerimientos del operario, entre otros, afectan el funcionamiento del sistema. Al conjunto de todas las combinaciones posibles de entradas externas al controlador se le ha denominado modo de operación (OM).

Definición 4. Modo de Operación.

Un modo de operación es una cadena de ceros o unos que representan los valores de las señales de entrada externas al controlador. Un modo de operación se representa como o_{om} , con base en la Definición 3. El conjunto de todos los modos de operación se denomina OM . $OM = \{o_0, \dots, o_{2^{n_{ec}}-1}\}$.

$EExP$ son señales de entrada que afectan directamente a la planta y corresponden a comportamientos legales o a comportamientos de fallo; pueden ser observables o no-observables, entonces $EExP = \{EExP^o, EExP^{no}\}$, con $EExP^o = \{eop_1, \dots, eop_{n_{eop}}\}$ donde n_{eop} es el número de entradas externas observables a la planta y $EExP^{no} = \{euop_1, \dots, euop_{n_{euop}}\}$, donde n_{euop} es el número de entradas externas no-observables a la planta.

Las entradas internas a la planta están constituidas por el conjunto de acciones de control, $EInP = \{Cc\} = \{cc_1, \dots, cc_{n_{cc}}\}$ siendo n_{cc} el número de órdenes de control.

Para modelar el comportamiento de la planta controlada se interrelacionan sus entradas y sus salidas. Las entradas a la planta son $U \subset \text{Entradas}$, donde $U = \{EExP^o, EExP^{no}, Cc\}$, $EExP^o$ y Cc son entradas observables y $EExP^{no}$ son entradas no-observables. Por lo tanto el número total de entradas es $m = m_o + m_{no}$, donde m_o es el número de entradas observables, $m_o = n_{eop} + n_{cc}$ y $m_{no} = n_{euop}$ es el número de entradas no-observables.

Las salidas de la planta es el conjunto $Y = \{\text{Salidas}\}$; el número total de salidas es n , $n = n_{sr}$.

El conjunto de entradas es $U = \{u_0, \dots, u_{2^m-1}\}$, donde u_s es una cadena valores de entrada en un instante determinado; $u_s = [euop_1 \dots euop_{n_{euop}} eop_1 \dots eop_{n_{eop}} cc_1 \dots cc_{n_{cc}}]$, u_s es un símbolo de entrada cuya representación se basa en la Definición 3.

El conjunto de salidas es $Y = \{y_0, \dots, y_{2^n-1}\}$, donde y_j es una cadena de valores de salida en un instante determinado; $y_j = [sr_1 \dots sr_n]$, y_j es un símbolo de salida cuya representación se basa en 3.

Un símbolo de E/S relaciona un símbolo de entrada y uno de salida: $(u_s y_j)$, donde $u_s \in U$ siendo $U = \{u_0, \dots, u_{2^m-1}\}$ y $y_j \in Y$ siendo $Y = \{y_0, \dots, y_{2^n-1}\}$.

El lenguaje es un conjunto finito de secuencias de símbolos, $\sigma = e_1 e_2 \dots e_n$, conocidos como palabras. De acuerdo a (Cassandra y Lafourche, 2008), el lenguaje generado por un SED puede ser considerado con tres niveles de abstracción: lenguaje estocástico temporizado, lenguaje temporizado y lenguaje sin tiempo. La adición de tiempo a una palabra puede hacerse dando a cada símbolo un valor de tiempo, resultando una secuencia de pares de símbolos: símbolo-tiempo. Entonces una secuencia temporizada es: $\sigma = (e_1, t_1) (e_2, t_2) \dots (e_n, t_n)$ donde si $t_i \sim f(t_i)$, siendo $f(t_i)$ una función de densidad, donde el tiempo se considera estocástico.

Dado que el comportamiento de los sistemas industriales no es determinístico y es el enfoque del presente trabajo, se considera modelar con base en eventos temporizados estocásticos cuyo símbolo se define a continuación.

Definición 5. Evento temporizado.

Un evento temporizado ω^i es un símbolo de E/S; que se genera cuando al menos una señal de E/S cambia de un instante τ_{i-1} a τ_i ; el cual para cada modo de operación om , tiene asociado un tiempo t_{om}^{ev} . Se representa como: $\omega^i = (u_s y_j) \cdot f(t_{om}^{ev})$, donde si t_{om}^{ev} no es constante, entonces $t_{om}^{ev} \sim f(t_{om}^{ev})$, es decir t_{om}^{ev} tiene a una función de densidad de probabilidad.

Por lo tanto:

$$\omega^i = (u_s y_j) \cdot f(t_{om}^{ev}) \quad (1)$$

en el instante τ_i . t_{om}^{ev} , es el tiempo transcurrido entre dos eventos consecutivos en un modo de operación om .

$$t_{om}^{ev} = |\tau_i| - |\tau_{i-1}| \quad (2)$$

Definición 6. Operación proyección de evento temporizado Pt .

Sea un evento temporizado de la forma $\omega^i = a \cdot t^{ev}$ donde a es un evento. Dados dos conjuntos de eventos temporizados Ω_1, Ω tal que $\Omega_1 \subseteq \Omega$, se define la operación de proyección de evento temporizado, $Pt_{\Omega_1} : \Omega^* \rightarrow \Omega^*$, así: $Pt_{\Omega_1}(\varepsilon) = \varepsilon$ siendo ε un

evento nulo y $Pt_{\Omega_1}(a \cdot t^{ev}) = P_{\Omega_1}(a) P_{\Omega_1}(t^{ev})$ donde $P_{\Omega_1}(a) = a$ si $\omega^i \in \Omega_1$ y $P_{\Omega_1}(a) = \varepsilon$ de lo contrario; pero $P_{\Omega_1}(t^{ev}) = t^{ev}$ si $\omega^i \in \Omega_1$ o $\omega^i \notin \Omega_1$.

Esta proyección elimina de Ω los eventos temporizados que $\notin \Omega_1$; pero si tiene en cuenta el tiempo que los eventos temporizados eliminados gastan en Ω .

Definición 7. Operación proyección de secuencia temporizada Pt .

Sea s una secuencia de eventos temporizados donde $s = \omega^0 \dots \omega^k$. Dados dos conjuntos de eventos temporizados Ω_1, Ω tal que $\Omega_1 \subseteq \Omega$, se define la operación de proyección de secuencia temporizada, $Pt_{\Omega_1} : \Omega^* \rightarrow \Omega^*$, así, $\forall \omega^i \in s$, donde $s \in \Omega^*$, $Pt_{\Omega_1}(s) = Pt_{\Omega_1}(\omega^0) \dots Pt_{\Omega_1}(\omega^k)$. Para encontrar el tiempo en la proyección, se inicializa una variable $t_{proy} = 0$. Para $i = 0 : k$, si $Pt_{\Omega_1}(a \cdot t^{ev}) = a \cdot t^{ev}$ entonces $t_{proy} = t^{ev}$ y la variable regresa a cero $t_{proy} = 0$; de lo contrario, si $Pt_{\Omega_1}(a \cdot t^{ev}) = \varepsilon \cdot t^{ev}$, $t_{proy} = t_{proy} + t^{ev}$.

Esta definición permite hallar el tiempo de los eventos proyectados.

Definición 8. Alfabeto del sistema.

El alfabeto del sistema Ω , es el conjunto de símbolos de E/S temporizados ω^i . Con $\omega^i = (u_s y_j) \cdot f(t_{om}^{ev})$, donde $u_s y_j \in U \cdot Y$, $|U \cdot Y| = |2^m| \times |2^n|$ y $f(t_{om}^{ev}) \in \delta$ es la función de densidad del evento, δ es una función de asignación de las funciones de densidad. Es decir $\omega^i \in (U \cdot Y)^* \cdot \delta$.

El lenguaje estocástico temporizado \mathcal{L} , para el sistema a modelar es un subconjunto de secuencias de eventos temporizados, tal que $\mathcal{L} \subset \Omega^*$, donde, $\Omega : \{\omega^i = (u_s y_j) \cdot f(t_{om}^{ev})\}$ que se generan en instantes τ_i . Es decir:

$$\mathcal{L} = \{s \mid s \in \Omega^*\} \quad (3)$$

Por lo tanto $\mathcal{L} = \{\omega^0, \dots, \omega^k\}$ a instantes $\tau_0 \leq \dots \leq \tau_k$.

3. Bases sobre Redes de Petri.

Las redes de Petri (PN) constituyen una familia de formalismos bien conocidos para el modelado, análisis y síntesis de sistemas concurrentes y distribuidos formalizados como discretos, (Girault y Valk, 2003; Silva y Recalde, 2007); proveen modelos compactos y capturan características importantes de los SED, como concurrencia, sincronismo, relaciones causales, recursos compartidos, etc. A continuación se presentan algunas definiciones sobre el tema.

Definición 9. PN.

Una PN N , es un grafo bipartito representado por la tupla $N = (P, TR, Pre, Post, M_0)$ (Murata, 1989) donde: P es el conjunto de lugares con cardinalidad np , TR es el conjunto de transiciones con cardinalidad ntr , $Pre : P \times TR \rightarrow \mathbb{N}$ y $Post : TR \times P \rightarrow \mathbb{N}$ son la matrices que contienen los arcos que conectan lugares y transiciones y M_0 es el marcado inicial; la función de marcado $M : P \rightarrow \mathbb{N}$ representa el número de marcas en cada lugar. (Murata, 1989; Dotoli et al., 2008b).

La matriz de incidencia $I = Post - Pre$ es una matriz de $np \times ntr$. Los conjuntos de lugares previos y posteriores a una transición, se denotarán como: $\bullet tr = \{p \in P : Pre(p, tr) > 0\}$ $tr\bullet = \{p \in P : Post(p, tr) > 0\}$. (Dotoli et al., 2008b).

Definición 10. Conjunto de alcanzabilidad de una PN.

Sea N una PN, la transición tr_r está habilitada si $\forall p_q \in \bullet tr_r$, $m(p_q) \geq I(p_q, tr_r)$. La transición habilitada tr_r puede ser disparada alcanzando un nuevo marcado M_{k+1} que se calcula por $M_{k+1} = M_k + I \cdot \vec{tr_r}$, donde $\vec{tr_r}$ es un vector de ceros de tamaño $|TR|$, excepto en la posición r , que es igual a 1. La evolución del marcado de una PN, dada una secuencia de disparo, $\sigma = tr_1 tr_2 \dots tr_k$, se denota como $M \lceil \sigma > M_k$. El conjunto de alcanzabilidad de una PN es el conjunto de todos los posibles marcados alcanzables desde M_0 , disparando solo transiciones habilitadas; este conjunto es denotado por $R(N, M_0)$. (Ramirez-Treviño et al., 2012)

Tomando solo la parte observable de las entradas, es decir, $U_o = \{EExP^o, Cc\}$ y con base en las definiciones de símbolo de entrada, símbolo de salida, alfabeto del sistema y de acuerdo a (Hernández y Meda-Campana, 2012), a continuación se define una red de Petri interpretada (IPN).

Definición 11. Red de Petri Interpretada (IPN).

Una IPN es una tupla $Q = (N, U_o, Y, \lambda, \varphi)$, donde, N es una PN basada en la Definición 9, $U_o = \{u_0, \dots, u_{2^{m_o}-1}\}$ es el alfabeto de entrada donde u_s es un símbolo de entrada y m_o es el número de entradas observables, $Y = \{y_0, \dots, y_{2^n-1}\}$ es el alfabeto de salida donde y_j es un símbolo de salida y n es el número de salidas, $\lambda : TR \rightarrow U_o$ es una función de etiquetado que asigna un símbolo de entrada a cada transición, $\varphi : R(N) \rightarrow Y$ es una función de salida que asigna un símbolo de salida a cada marcado alcanzado.

Una $tr_r \in TR$ de una IPN está habilitada si $\forall p_q \in \bullet tr_r$, $m(p_q) \geq I(p_q, tr_r)$. Si $\lambda(tr_r) = u_s \neq \varepsilon$ está presente y si tr_r está habilitada entonces tr_r se dispara y se alcanza un nuevo marcado M_{k+1} , el cual se calcula mediante la ecuación de estado:

$$\begin{aligned} M_{k+1} &= M_k + I \cdot \vec{tr_r} \\ y_j &= \varphi(M_k) \end{aligned} \quad (4)$$

donde $I \cdot \vec{tr_r}$ se define como en la Definición 10 y y_j es un símbolo de salida, tal que $y_j \in Y$.

Definición 12. Determinismo.

Una IPN es determinística, si dada una secuencia de transiciones $\sigma_i = tr_1 \dots tr_k \dots$ tal que $M_0 \lceil \sigma_i > M_k$, $M_0 \lceil \sigma_i > M'_k$ implica que $M_k = M'_k$. es decir, el marcado alcanzado después del disparo de una secuencia de transiciones es única, (Gaubert y Giua (1996)).

El lenguaje de disparo de una IPN (Ramirez-Treviño et al., 2007) se define como:

Definición 13. Lenguaje de Disparo de una IPN.

Sea Q una IPN y $M_0 \lceil \sigma_i > M_k$, el conjunto de todas secuencias de disparo se llama lenguaje de disparo: $\mathcal{L}_F(Q) = \left\{ \sigma | \sigma = tr_1 \dots tr_k \wedge M_0 \xrightarrow{tr_1} \dots M_w \xrightarrow{tr_k} M_k \mid M_0, \dots, M_w \in R(N) \right\}$.

4. Modelo del Sistema

El siguiente modelo tiene como objetivo representar el comportamiento legal de un SED, a partir del lenguaje legal del sistema; para ello se propone un generador determinístico de lenguaje, el cual es una PN extendida que se ha denominado IPN temporizada estocástica (st-IPN).

Las características agregadas al concepto básico de PN son: en primer lugar la adición de etiquetas en las transiciones y en las funciones de salida de los lugares, con el objeto de relacionarlos con las señales de E/S y generar una interpretación en el comportamiento y cambio de estado del modelo y en segundo lugar se tiene en cuenta el tiempo en el disparo de las transiciones y el comportamiento de este tiempo se asume bajo una función de densidad de probabilidad no establecida, es decir puede ser normal, o exponencial, o etc; por este motivo se remarca que sea “temporizada - estocástica”.

El modelo propuesto pretende servir de modelo de referencia para el diagnóstico en SED estocásticos y además busca ser determinista; es por ello que en esta propuesta se ha incluido, por un lado, en la función de salida, un elemento que dotará a la st-IPN de memoria en la función de salida. Se trata del *diferencial del símbolo de salida*, (dy); y por otro lado, se ha incluido el tiempo de disparo de las transiciones, el cual es el tiempo que transcurre desde que una transición es habilitada hasta que es disparada (Piera y Music, 2011), considerándolo de naturaleza variable.

Definición 14. Diferencial de símbolo de salida. dy .

Dados dos símbolos de salida $y_j^i, y_{j*}^{i-1} \in Y^1$, generados en los instantes τ_i y τ_{i-1} respectivamente; dy_j se define como:

$dy_j(y_j^i \times y_{j*}^{i-1}) \rightarrow \{-1, 0, 1\}$, donde $dy_j = y_j^i - y_{j*}^{i-1}$ con $dy_j \in \{-1, 0, 1\}$; así los posibles valores de dy_j son: $0 \times 0 \rightarrow 0$; $0 \times 1 \rightarrow 1$; $1 \times 0 \rightarrow -1$; $1 \times 1 \rightarrow 0$.

Esto significa que, dado que se trabaja con señales binarias, dy representa el cambio del valor de la señal de salida, de 0 a 1, o de 1 a 0; por lo tanto, dy representa el comportamiento histórico de las señales de un instante τ_{i-1} a un instante τ_i . De esta manera la función de salida, está compuesta por el valor de las señales de salida y_j y su diferencial dy_j , es decir y_j/dy_j . Esta condición de la función de salida genera una característica particular de las señales, la cual se ha denominado característica de direccionalidad de las señales.

Teniendo en cuenta el diferencial y empleando funciones de densidad de probabilidad en las transiciones, una st-IPN se define de la siguiente manera:

Definición 15. Red de Petri Interpretada Temporizada Estocástica (st-IPN).

Una st-IPN es una estructura representada por:

$$stQ = (Q, \Omega, \delta, OM) \quad (5)$$

Donde $Q = (N, U_o, Y, \lambda, \varphi)$ es una IPN, donde N, U_o, Y tienen el mismo significado que en la Definición 11; $\lambda : TR \rightarrow$

¹Nota: Se adicionan los superíndices con el objeto establecer el orden de los eventos

$U_o \times \delta$ es una función de etiquetado que asigna un símbolo de entrada y una función de densidad a cada transición, φ es la función de salida definida como: $\varphi : (RN) \rightarrow Y/dY$; φ es isomórfica sobre Y/dY .

$\Omega := (U_o \times Y) \cdot \delta$ es el alfabeto del sistema. Un símbolo del alfabeto es $\omega^i = (u_s y_j) \cdot f(t_{r,om})$, donde $(u_s y_j)$ es el símbolo de E/S, $f(t_{r,om})$ es la función de densidad de probabilidad del tiempo de la transición r , asociada al símbolo de E/S, en el modo de operación om .

$\delta := TR \times OM \rightarrow f(t_{TR \times OM})$ es una función de asignación de las funciones de densidad del tiempo de disparo de cada transición por cada modo de operación.

OM es el conjunto de modos de operación.

Dado un evento $\omega^i = (u_s y_j) \cdot f_{om}^{ev}$; una transición $tr_r \in TR$ con $\lambda(tr_r) = u_s \cdot f(t_{r,om})$, está habilitada en om , si $\forall p_q \in \bullet tr_r$, $m(p_q) \geq I(p_q, tr_r)$ y si $(a \leq t_{om}^{ev} \leq b)$ siendo $\int_a^b f(t_{r,om}) \geq (1 - \alpha)$, donde $1 - \alpha$ es el nivel de confiabilidad y $[a, b]$ el intervalo de confianza.

Si $f(t_{r,om}) = N(\mu, \sigma)$, entonces $a = \mu - z_{\alpha/2} * \sigma$ y $b = \mu + z_{\alpha/2} * \sigma$, una tr_r está habilitada si cada lugar de entrada cumple con los requerimientos de marcado y si $t_{om}^{ev} \in [a, b]$.

Si $\lambda(tr_r) = u_s \cdot f(t_{r,om})$ está presente y si tr_r está habilitada entonces tr_r se dispara y se alcanza un nuevo marcado M_{k+1} , calculado a partir de (4); pero $y_j/dy_j = \varphi(M_k)$. Así cada lugar de la st-IPN representa no solo el estado actual, sino que también incluye información de direccionalidad, por lo tanto la red evoluciona a partir de la ecuación de estado:

$$\begin{aligned} M_{k+1} &= M_k + I \cdot \overrightarrow{tr_r} \\ y_j/dy_j &= \varphi(M_k) \end{aligned} \quad (6)$$

Definición 16. Lenguaje de disparo de una st-IPN

Sea σ la secuencia de disparo de transiciones de una st-IPN, (stQ) tal que $\sigma \in \mathcal{L}_F(Q)$, (Definición 13), cuyo disparo genera una secuencia $s = \omega^0, \dots, \omega^k$, tal que $s \in \Omega^*$ a instantes $\tau_0 \leq \dots \leq \tau_k$, siendo $\omega^i = (u_s y_j) \cdot f(t_{r,om})$ donde $u_s \in U_o$, $y_j \in Y$ y $f(t_{r,om}) \in \delta$. El conjunto de todas las secuencias de disparo de transiciones $\sigma \in \mathcal{L}_F(Q)$, cuya probabilidad sea mayor a $(1 - \alpha)$ es el lenguaje de disparo de la st-IPN así:

$$\mathcal{L}_F(stQ) = \{\sigma \in \mathcal{L}_F(Q) : Prob(\sigma|m_0) \geq (1 - \alpha)\} \quad (7)$$

A continuación se relacionan las secuencias de símbolos de E/S temporizados, con el disparo de secuencias de transiciones y las secuencias de marcado generadas, con el objeto de definir el lenguaje de entrada y el lenguaje de salida de una st-IPN.

Definición 17. Lenguaje de entrada y Lenguaje de salida de una st-IPN.

Sea σ la secuencia de disparo de transiciones de una st-IPN, tal que $\sigma \in \mathcal{L}_F(stQ)$, (Definición 7), el lenguaje de entrada se define como la secuencia de las funciones de etiquetado de las transiciones que pertenecen a $\mathcal{L}_F(stQ)$, es decir, $\mathcal{L}_{in}(stQ) = \{\sigma | \sigma = \lambda(tr_1) \dots \lambda(tr_k)\}$ y el lenguaje de salida son las secuencias de marcado alcanzadas $\mathcal{L}_{out}(stQ) = \{\varphi(m(p_0)) \dots \varphi(m(p_k))\}$.

Por lo tanto, $\mathcal{L}(stQ) = \{\omega^0, \dots, \omega^k\}$ con $\omega^i = (u_s y_j) \cdot f(t_{r,om})$, comprende un lenguaje de entrada y un lenguaje de salida.

Aplicando la Definición 2, el lenguaje de entrada se define como: $\mathcal{L}_{in}(stQ) = \{P_{U_o, \delta}(s) | s \in \mathcal{L}(stQ)\}$, entonces $\mathcal{L}_{in}(stQ) = \{u_s^0 \cdot f(t_{om}^0), \dots, u_s^k \cdot f(t_{om}^k)\}$ y el lenguaje de salida como: $\mathcal{L}_{out}(stQ) = \{P_Y(s) | s \in \mathcal{L}(stQ)\}$, entonces $\mathcal{L}_{out}(stQ) = \{y_j^0, \dots, y_j^k\}$.²

Definición 18. Post-lenguaje.

Dada una secuencia de eventos temporizados estocásticos $s = \omega^0, \dots, \omega^k$, el post-lenguaje de s será: $st^* = \mathcal{L}(stQ)/s = \{t \in \Omega^* | s \in \mathcal{L}(stQ)\}$.

4.1. Modelado de Sistemas basado en st-IPN

Como una st-IPN es un generador de lenguaje legal, ésta se construye a partir del lenguaje legal ordenado del sistema $\mathcal{L} = \{\omega^0, \dots, \omega^k\}$.

El modelo inicia con cero transiciones, $ntr = 0$ y con un lugar, $np = 1$, (p_0) . Se toma ω^0 y se calcula y_j como $y_j = P_{CY}(\omega^0)$ y $dy_j = \vec{0}$ como una cadena de ceros de tamaño n (número de salidas); por lo tanto la función de salida en p_0 es $\varphi(m(p_0)) = y_j/dy_j$.

Luego de manera ordenada se toman uno a uno los eventos ω^i de \mathcal{L} , se calcula la función de salida como $y_j/y_j^i - y_j^{i-1}$, se verifica si existe un lugar con esa función de salida; si no existe se genera un nuevo lugar p_{q+1} con una función de salida $\varphi(m(p_{q+1})) = y_j/y_j^i - y_j^{i-1}$ y también se genera una nueva transición $ntr = ntr + 1$ con $\lambda(tr_{ntr}) = P_{CY}(\omega^i)$; si por el contrario existe, se verifica si existe la transición que va hacia el lugar ya modelado, para ello se analizan los vectores \overrightarrow{pre} , \overrightarrow{post} y $\lambda(tr_{ntr})$; si no existe se añade una nueva transición.

Luego se actualizan las matrices Pre y $Post$; este procedimiento conforma el algoritmo 1.

Como resultado del algoritmo se genera la estructura de la red que genera el mismo lenguaje que se pretende modelar.

Proposición 1. Sea $\mathcal{L} = \{\omega^0, \dots, \omega^k\}$ el lenguaje temporizado legal de un sistema a instantes $\tau_0 \leq \dots \leq \tau_k$, si stQ es una st-IPN construida a partir del algoritmo 1, entonces $\mathcal{L} = \mathcal{L}(stQ)$

Demuestra. Dadas s, s_1 dos secuencias de eventos temporizados legales tal que: $s = \omega^0 \dots \omega^k$ y $s_1 = \omega^p$ donde $s, s_1 \in \mathcal{L}$, $s_1 \in \mathcal{L}/s$ y $s \in \mathcal{L}(stQ)$. Asumiendo que $s_1 \notin \mathcal{L}(stQ)$, entonces siendo el estado actual de la st-IPN $\varphi(m(p_k)) = y_j^k / [y_j^k - y_j^{k-1}]$ y dado el evento $\omega^p = (u_s y_j) \cdot f(t_{om}^p)$ al aplicar el algoritmo 1 un estado es alcanzado en $\varphi(m(p_p)) = y_j^p / [y_j^p - y_j^k]$ si $\exists \varphi(m(p_i)) \mid \varphi(m(p_i)) = \varphi(m(p_p))$, $\forall i, i = 1 \dots k$ entonces $p_p \in P$ de lo contrario p_p es un lugar nuevo y se genera una transición con $\lambda(tr_p) = u_s^p \cdot f(t_{om}^p)$, es decir el evento $\omega^p \in \mathcal{L}(stQ)/s$ y se contradice la suposición; si p_p no es nuevo, $p_p = p_i$ siendo i el índice del lugar repetido, se genera una

²Nota: Se adicionan los superíndices con el objeto establecer el orden de los eventos

Algoritmo 1 Construcción de una st-IPN a partir del lenguaje del sistema

Entradas: $\mathcal{L} = \{s \in (U_o, Y)^* \cdot \delta\}$, donde $s = \omega^0, \dots, \omega^k$ en $\tau_0 \leq \dots \leq \tau_k$; $\omega^i = (u_s, y_j) \cdot f(t_{om}^v)$.

Salidas: stQ .

Condiciones Iniciales: $i = 0$; $\omega^0 = (u_s, y_j) \cdot f(t_{om}^0)$, $\varphi(m(p_0)) = y_j/dy_j$; $dy_j = \vec{0}_n$.

Variables: $np = 1$; (Número de lugares). np' : índice temporal de lugar. $ntr = 0$; (Número de transiciones). ntr' : índice temporal de transición $Pre = []$; $Post = []$; $pa = 0$; (lugar actual).

Procedimiento:

1. $i = i + 1$; $\omega^i = (u_s, y_j) \cdot f(t_{om}^v)$; $np' = np$; lugar temporal.
2. $\varphi(m(p_{np'})) = y_j^i / [y_j^i - y_{j-1}^i]$; función de salida en el lugar temporal.
 - a) Si $\varphi(m(p_{np'})) \neq \varphi(m(p_q)) \forall q, q = 0 \dots np - 1$ entonces
 - 1) $ntr = ntr + 1$; nueva transición con $\lambda(tr_{ntr}) = u_s \cdot f(t_{om}^v)$;
 - 2) Calcular vectores pre y $post$ para nueva tr
 - $np' = np$;
 - $\overrightarrow{pre} = ceros(np)$; $pre(pa) = 1$;
 - $\overrightarrow{post} = ceros(np)$; $post(np) = 1$;
 - 3) Actualizar Pre y $Post$
 - $Pre = Pre + \overrightarrow{pre}$; $Post = Post + \overrightarrow{post}$;
 - 4) $\varphi(m(p_{np})) = \varphi(m(p_{np'}))$; $pa = np$;
 - b) De lo contrario, (el lugar ya existe)
 - 1) $pa = k$;
 - 2) $ntr' = ntr + 1$; se genera una transición temporal con $\lambda(tr_{ntr'}) = u_s \cdot f(t_{om}^v)$; $f(t_{om}^v) \subset \delta(ntr, 1, om)$;
 - 3) Calcular los vectores pre y $post$ para la tr temporal
 - $\overrightarrow{pre} = ceros(np)$; $pre(np) = 1$;
 - $\overrightarrow{post} = ceros(np)$; $post(k) = 1$;
 - 4) Para $r = 1 \dots ntr$
 - Si $\overrightarrow{pre} = Pre(:, r) \wedge \overrightarrow{post} = Post(:, r) \wedge \lambda(tr_{ntr'}) = \lambda(tr_r)$, entonces $tr_existe = 1$; $f(t_{om}^v) \subset \delta(ntr, 1, om)$;
 - saltar a paso 5;
 - De lo contrario $tr_existe = 0$;
 - 5) Si $tr_existe = 1$, entonces
 - $Pre = Pre$; $Post = Post$;
 - De lo contrario; $ntr = ntr'$, número de tr temporal.
 - 6) Actualizar Pre y $Post$
 - $Pre = Pre + \overrightarrow{pre}$; $Post = Post + \overrightarrow{post}$;
 - $\lambda(tr_{ntr}) = u_s \cdot f(t_{om}^v)$; $f(t_{om}^v) \subset \delta(ntr, 1, om)$;
 - c) fin condicional
3. ejecutar paso 1

transición con $\lambda(tr_p) = u_s^p \cdot f(t_{om}^p)$, si $\lambda(tr_p) = \lambda(tr_i)$ entonces $\omega^p \in \mathcal{L}(stQ)/s$ y se contradice la suposición, de lo contrario tr_p es una nueva transición con $\lambda(tr_p) = u_s^p \cdot f(t_{om}^p)$ y $\omega^p \in \mathcal{L}(stQ)/s$ y se contradice la suposición. Lo que permite concluir que si $s_1 \in \mathcal{L}$ y \mathcal{L} es legal entonces $s_1 \in \mathcal{L}(stQ)$ por lo tanto $\mathcal{L} = \mathcal{L}(stQ)$. \square

Un modelo representado como una st-IPN, es un generador determinista del lenguaje del sistema a modelar, porque el marcado alcanzado en $\varphi(m(p_k))$ después del disparo de una secuencia de transiciones es único, ya que φ isomórfica.

La asignación isomórfica da lugar a equivalencias entre los símbolos de salida y los marcados alcanzados. Por lo tanto, será inmediato inferir el estado actual de la red a partir de una combinación particular de símbolos de salida.

4.2. Observabilidad

La observabilidad implica la posibilidad de detectar si un evento no-observable ha ocurrido, analizando la secuencia de eventos generados. El conjunto de eventos de un sistema está compuesto por eventos observables y eventos no-observables: $\Omega = \Omega_o \cup \Omega_{uo}$. Un evento temporizado es de la forma: $\omega^i = (u_s, y_j) \cdot f(t_{om}^v)$. La proyección basada en la Definición 1, $P_{\Omega_o}(\Omega)$ elimina los Ω_{uo} de una secuencia de eventos. Un evento no-observable se genera cuando existe una entrada no-observable a la planta.

Definición 19. Un evento ω es n-observable con respecto a una proyección $P(\Omega)$, si $\forall t \in \omega t^*$, donde $|t| > n$, $P(swt) \neq P(st)$.

Es decir un evento ω es n-observable si puede ser detectado con certeza dentro de un número, n , de eventos observables después de su ocurrencia. Esto implica que un evento ω es observable si para cada traza s de eventos que finalicen en ω , existe una traza t de tamaño n , tal que la secuencia st no genere el mismo registro de eventos observables conteniendo a ω .

Para realizar test de observabilidad en sistemas estocásticos con base en su lenguaje, a continuación se propone una definición de observabilidad temporal.

Definición 20. Observabilidad temporal.

Sea s una secuencia de eventos $s \in \mathcal{L}(stQ)$ y $st^* = \mathcal{L}(stQ)/s$. Dado un evento temporizado $\omega^p = (u_s, y_j) \cdot f(t_{om}^p)$ con $\mathcal{L}_{in}(\omega^p) = u_s^p \cdot t_{om}^p$ y $\mathcal{L}_{out}(\omega^p) = y_j^p$; ω^p es observable si $Pt_{U_o}(s\omega^p t) \neq Pt_{U_o}(st) \text{ o } Pt_Y(s\omega^p t) \neq Pt_Y(st)$, en n pasos después de su ocurrencia ($n \leq |t|$).

Donde Pt_{U_o} y Pt_Y se realizan con base en la Definición 7.

Esta definición permite inferir respecto a las condiciones para que un evento no-observable sea detectado en una st-IPN, lo cual se demuestra en la Proposición 2.

Proposición 2. Sea \mathcal{L} el lenguaje de un sistema y stQ una st-IPN que representa a \mathcal{L}^o . Dada una secuencia s de eventos temporizados tal que $s \in \mathcal{L}(stQ)$ y sea $\omega^k \in \Omega_{uo}$, $\omega^p = (u_s^p, y_j^p) \cdot (t_{om}^p)^3$ tal que $\omega^p \in \mathcal{L}/s$; ω^p es observable si $u_s^p \in$

³Nota: Se ha añadido el superíndice en los símbolos del lenguaje, para identificar el evento al que pertenecen

$\mathcal{L}_{in}(stQ)$ y $y_j^p \notin \mathcal{L}_{out}(stQ)$ o si $u_s^p \notin \mathcal{L}_{in}(stQ)$ y $y_j^p \in \mathcal{L}_{out}(stQ)$ o si $u_s^p \in \mathcal{L}_{in}(stQ)$, $y_j^p \in \mathcal{L}_{out}(stQ)$ y $t_{om}^p \notin [a, b]$ donde $\int_a^b f(t^a) \geq (1 - \alpha)$.

*Demuestra*o. Sea $\mathcal{L} = s\omega^p t$ el lenguaje del sistema, donde $s = \omega^0 \cdots \omega^k \omega^p$, $t = \omega^{p+1} \wedge \omega^p = (u_s^p, y_j^p)$. t_{om}^p es un evento inobservable, tal que $s \in \mathcal{L}(stQ)$ y $t \in \mathcal{L}(stQ) / s$; entonces $Pt_{U_o}(st) = u_s^0 \cdot t_{om}^0 \cdots u_s^k \cdot t_{om}^k \cdot u_s^{p+1} \cdot t_{om}^{p+1}$ y $Pt_Y(st) = y_j^0 \cdots y_j^k \cdot y_j^{p+1}$; dado $\omega^p \in \mathcal{L}$ si:

- i) u_s^p es observable y y_j^p es no-observable, entonces $Pt_{U_o}(\omega^p) = u_s^p \cdot t_{om}^p$ y $Pt_Y(\omega^p) = \varepsilon$ entonces $Pt_{U_o}(st) = Pt_{U_o}(s\omega^p t)$ y $Pt_Y(st) \neq Pt_Y(s\omega^p t)$ de acuerdo a la Definición 20, ω^p es observable.
- ii) u_s^p es no-observable y y_j^p es observable, entonces $Pt_{U_o}(\omega^p) = \varepsilon$ y $Pt_Y(\omega^p) = y_j^p$ entonces $Pt_{U_o}(st) \neq Pt_{U_o}(s\omega^p t)$ y $Pt_Y(st) = Pt_Y(s\omega^p t)$ de acuerdo a la Definición 20, ω^p es observable.
- iii) u_s^p es no-observable y y_j^p es no-observable, entonces $Pt_{U_o}(\omega^p) = \varepsilon$; $Pt_Y(\omega^p) = \varepsilon$; si $t_{om}^p \notin [a, b]$ ω^p es observable; de lo contrario ω^p es no-observable. Por lo tanto la proposición queda demostrada. \square

A partir de las proposiciones 1 y 2, se resuelven las preguntas planteadas en el enfoque del trabajo, puesto que se comprueba que una st-IPN modela el lenguaje legal de un SED y su lenguaje permite detectar algunos eventos no-observables legales y eventos de fallo; por lo tanto, en un sistema modelado como una st-IPN, se puede realizar diagnóstico de fallos.

5. Sistemas de gran escala

Los sistemas a nivel industrial, generalmente son sistemas de gran dimensión, compuestos por una gran cantidad de dispositivos distribuidos, independientes o no, que generan concurrencia de estados a partir de la ocurrencia de eventos de manera asíncrona. La generación de un modelo único en estos sistemas ha llevado a que se presente una explosión en el número de estados, lo cual conlleva una serie de inconvenientes al realizar análisis sobre ellos como test de observabilidad, o para detectar y localizar comportamientos no normales. Para reducir la complejidad, muchos enfoques de modelado han propuesto la división del sistema en partes más pequeñas, (Peres et al., 2011).

En esta propuesta, el modelado de este tipo de sistemas se realiza a partir de la división en subsistemas, donde un subsistema es una parte del sistema que posee un comportamiento particular.

La estructura a considerar, de un SED de gran escala, se muestra en la Figura 2.

El conjunto de entradas a un SED de gran escala se clasifican como: $Entradas = \{EExC, EExP, EInP\}$; donde $EExC$ y $EExP$ tienen el mismo significado que para sistemas sencillos.

Para la aplicación del método de modelado a sistemas de gran escala, se han dividido las señales de entrada internas a la planta ($EInP$), en dos subgrupos: $EInP = \{GCC, LCC\}$; GCC está compuesto por órdenes de control globales que afectan a más de un subsistema, donde $GCC = \{GCC_1, \dots, GCC_{n_{gc}}\}$, n_{gc} es

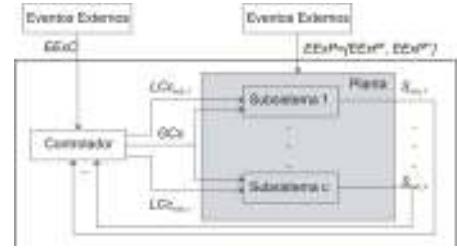


Figura 2: Esquema de un SED de gran escala.

el número de órdenes de control globales y el conjunto $LCC = \cup LCC_l$, donde LCC_l está compuesto por órdenes de control que solo afectan el subsistema l , con $l = 1 \cdots c$, c es el número de subsistemas. Entonces, $LCC_l = \{CC_{l,1}, \dots, CC_{l,n_{lc}}\}$, n_{lc} es el número de órdenes de control locales en el subsistema l .

El conjunto de salidas es: $Salidas = Y = \cup S_l$, donde $S_l = \{SR_{l,n_l}, \dots, SR_{l,n_{ls}}\}$, es decir, contiene las lecturas sensoriales asignadas al subsistema l y n_{ls} es el número de lecturas sensoriales en el subsistema l .

El conjunto de entradas es $U \subset Entradas$ donde, $U = \{EExP^{uo}, EExP^o, GCC, LCC\}$, $GCC, LCC, EInP, EExP^o$ constituyen las entradas observables y $EExP^{uo}$ las entradas no-observables.

Las entradas a cada subsistema l son $U_l \subset U$ donde $U_l = \{EExP^{uo}, EExP^o, GCC, LCC_l\}$, entonces U_l es el conjunto de todas las combinaciones posibles de entradas al subsistema l . El número total de entradas, no-observables y observables en el subsistema l es: $m_l = m_{l,uo} + m_{l,o}$ donde; $m_{l,o} = n_{eop} + n_{gc} + n_{lc}$. Una entrada a un subsistema l en particular, $u_{l,s}$, es una cadena de valores binarios, tal que:

$$u_{l,s} = [euop_1 \cdots euop_{n_{eop}} eop_1 \cdots eop_{n_{eop}} g_{cc1} \cdots g_{cc_{n_{gc}}} g_{cc_{l,1}} \cdots g_{cc_{l,n_{cc}}}] \quad (8)$$

siendo s la representación decimal de la cadena.

Las salidas de cada subsistema $Y_l \subset Salidas$, donde $Y_l = \{SR_l\}$; entonces Y_l es el conjunto de todas las combinaciones posibles de salidas del subsistema l . Una salida de un subsistema l en particular, $y_{l,j}$, es una cadena de valores binarios, tal que:

$$y_{l,j} = [sr_{l,1} \cdots sr_{l,n_l}] \quad (9)$$

siendo j la representación decimal de la cadena.

La división en subsistemas se realiza con base en las relaciones de funcionamiento de los dispositivos y las señales de E/S, coherentes con ese funcionamiento en cada subsistema.

Esta clasificación de señales de E/S permite generar modelos en cada subsistema, lo cual evita la explosión combinacional de estados en sistemas de gran escala. Esta descomposición modular presenta ventajas, puesto que se modelan comportamientos de estados concurrentes en diferentes subsistemas, de recursos compartidos por los subsistemas o funcionamiento colaborativos entre sistemas.

Esta es una ventaja importante cuando el objetivo del modelo es realizar diagnóstico de fallos, como en la presente propuesta.

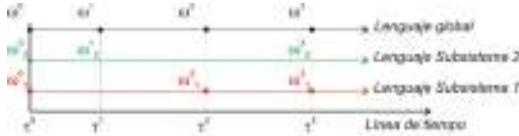


Figura 3: Lenguajes Temporizados

5.1. Modelo del SED de Gran Escala

El modelo global (modelo del sistema) es un conjunto de modelos locales (modelos de los subsistemas) que se representan a partir de st-IPNs y éstas se construyen a partir de su lenguaje observable con base en el algoritmo 1.

Definición 21. st-IPN para un Sistema de gran escala

Cuando se modela un subsistema se agrega el subíndice l , a la st-IPN definida en la definición 15, correspondiente al subsistema a modelar, es decir: $stQ_l = (Q_l, \Omega_l, \delta_l, OM)$. Con $Q_l = (N_l, U_{ol}, Y_l, \lambda_l, \varphi_l)$ donde $U_l = \{u_{l,0}, \dots, u_{l,2^{m_l}-1}\}$, $Y_l = \{y_{l,0}, \dots, y_{l,2^{m_l}-1}\}$, $\lambda_l : TR_l \rightarrow U_l \times \delta_l$ es la función de etiquetado en cada transición del subsistema l y φ_l es la función de salida definida como $\varphi_l : R(Q_l, M_{l,0}) \rightarrow y_{l,j}/dy_{l,j}$, φ_l es isomórfica sobre $y_{l,j}/dy_{l,j}$. $\Omega_l = U_l \times Y_l \times \delta_l$ es el alfabeto del subsistema. $\delta_l := TR_l \times OM \rightarrow f_{(TR_l \times OM)}$ es una función de asignación de las funciones de densidad del tiempo de disparo de cada transición por cada modo de operación OM , en el subsistema l .⁴. $OM = \{o_0, \dots, o_{2^{n_ec}-1}\}$ es el conjunto de modos de operación, n_ec es el número de entradas externas al controlador.

5.1.1. Evento temporizado en subsistemas

Un evento temporizado en un subsistema l es de la forma: $\omega_l^i = (u_{l,s}y_{l,j}) \cdot t_{l,om}^ev$, i identifica el instante de tiempo en el cual se genera el evento, l es el número del subsistema, $u_{l,s}$ y $y_{l,j}$ se calculan a partir de la (8) y la (9) y $t_{l,om}^ev$ es el tiempo transcurrido entre los eventos ev y $ev-l$ en el subsistema l , para el modo de operación om . La Figura 3 explica la relación del tiempo en los subsistemas frente al sistema global.

Como se observa en la Figura 3, en la línea en negro se presenta una secuencia de eventos $\omega^0 \omega^1 \omega^2 \omega^3$ del sistema global, en instantes $\tau_0 \leq \dots \leq \tau_3$, en la línea roja la secuencia de eventos $\omega_1^0 \omega_1^1 \omega_1^2$ para el subsistema 1 y en verde una secuencia de eventos $\omega_2^0 \omega_2^1 \omega_2^2$ para el subsistema 2.

El tiempo de los eventos en los subsistemas es: $t_{l,om}^ev = |\tau_{ev_actual}| - |\tau_{ev_anterior}|$. Por ejemplo para el evento $\omega_2^3 = (u_{2,s}y_{2,j}) \cdot t_{2,om}^2$, el tiempo asociado es $t_{2,om}^2 = |\tau_3| - |\tau_1|$.

5.2. Operaciones entre Modelo Global y Modelos Locales

El modelo de un sistema de gran escala, representado como un conjunto de st-IPNs, con diversos comportamientos o modos de funcionamiento, tiene ventajas frente a un modelado del sistema global, puesto que son más fáciles de interpretar debido su estructura más reducida y porque modelan solo lenguajes activos; por lo tanto permiten establecer en un instante de tiempo cuál subsistema funciona.

⁴ $tr_{l,r}$, indica la transición r en el subsistema l .
 $pl_{l,q}$, indica el lugar q en el subsistema l .

5.2.1. Lenguaje local a partir del lenguaje global

El lenguaje de los subsistemas se puede hallar a partir de la operación de proyección de lenguajes presentada en la Definición 6.

Dado el lenguaje de un sistema $\mathcal{L} = \omega^0 \dots \omega^k$, con $\omega^i = (u_{s,y,j}) \cdot t_{om}^ev$, ordenando las señales de E/S como $\omega^i = (u_{1,s} \dots u_{c,s}) \cdot t_{om}^ev$, la $P_{\Omega_l}(\omega^i) = (u_{l,s}y_{l,j}) \cdot t_{l,om}^ev$, si ha existido un evento temporizado en l , basado en Definición 5; de lo contrario $P_{\Omega_l}(\omega^i) = \varepsilon \cdot t_{l,om}^ev$ (Definición 7). $u_{l,s}$ y $y_{l,j}$ se organizan como en las ecuaciones (8) y (9), $t_{l,om}^ev$ se halla como en la Definición 7.

Por lo tanto el lenguaje de cada subsistema se puede hallar a partir del lenguaje global.

El lenguaje del subsistema 1 representado en la Figura 3, a partir de la proyección sobre lenguaje global es:

$P_{\Omega_1}(\omega_1^0 \omega_1^1 \omega_1^2) = (u_{1,s}y_{1,j}) \cdot t_{1,om}^0 \varepsilon \cdot t_{1,om}^1 (u_{1,s}y_{1,j}) \cdot t_{1,om}^2 (u_{1,s}y_{1,j}) \cdot t_{1,om}^3$, donde $\omega_1^0 = (u_{1,s}y_{1,j}) \cdot t_{1,om}^0$, $\omega_1^1 = (u_{1,s}y_{1,j}) \cdot t_{1,om}^1$ con $t_{1,om}^1 = t_{1,om}^1 + t_{1,om}^2$ y $\omega_1^2 = (u_{1,s}y_{1,j}) \cdot t_{1,om}^2$.

5.2.2. Lenguaje global a partir de lenguajes locales

Un evento del sistema global ω^i en el instante τ_i , se obtiene a partir de la operación de splice sincronizado temporizado, $\omega^i = \omega_1^i \oplus \dots \oplus \omega_c^i$ (ver Muñoz et al. (2014)), donde $\omega^i = (u_{s,y,j}) \cdot t_{om}^ev$ y $t_{om}^ev = \min(t_{1,om}^h \dots t_{c,om}^h)$.

Por lo tanto el lenguaje del sistema global \mathcal{L} , se construye por la sincronización temporal de los lenguajes de los subsistemas \mathcal{L}_l que está compuesto.

Nota: El lenguaje del sistema global se puede reconstruir si y solo si se realiza el splice sincronizado temporizado sobre todos los subsistemas que integran el sistema global.

5.2.3. st-IPN global a partir de st-IPN locales

El modelo del sistema global representado como una st-IPN se puede hallar a partir del Producto Síncrono de st-IPNs. (Muñoz et al. (2014)) o del lenguaje global construido por la sincronización temporal de los lenguajes de sus subsistemas.

Al dividir un sistema en subsistemas, cada modelo conserva las propiedades en cuanto a representatividad y a observabilidad; es decir, el modelo de un subsistema representado como una st-IPN es un generador de lenguaje determinístico, capaz de modelar comportamientos estocásticos y de detectar algunos eventos no-observables, bajo las condiciones expuestas en la Proposición 2.

Teorema 1. Sea $\mathcal{L}(stQ)$ el lenguaje global compuesto por c lenguajes locales $\mathcal{L}(stQ_l)$. Un evento $\omega^i \in \mathcal{L}(stQ)$ es observable en el sistema global si $\omega_l^i \in \mathcal{L}(stQ_l)$ es observable en el algún subsistema l , con $l = 1 : c$.

Demostración. Condición necesaria: si un evento ω^i es observable en el sistema global, entonces el evento ω_l^i es observable en el subsistema l .

Suponiendo que ω^i es observable pero $\exists p \in [1 \dots c] \mid \omega_p^i$ es inobservable. Como $\omega^i = (u_{s,y,j}) \cdot f(t_{om}^ev) = (u_{1,s} \dots u_{p,s} \dots u_{c,s}) (y_{1,s} \dots y_{p,s} \dots y_{c,s}) \cdot f(t_{om}^ev)$ es observable; $\exists u_{p,s} \wedge y_{p,j}$ tal que

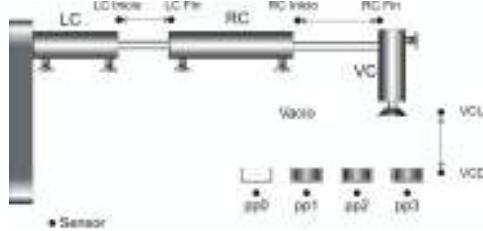


Figura 4: Brazo Robótico (Patil et al., 2012);

$P_{\Omega_l}(\omega_l^i) = (u_{p,s}, y_{p,j})$ porque el lenguaje del sistema global se reconstruye a partir de todos los lenguajes locales, entonces $\exists, l, | P_{\omega_l^i}(\omega^i) \neq \varepsilon$. Por lo tanto ω_p^i es observable.

Condición suficiente: si ω_l^i es un evento observable en el subsistema l en algún $l = 1 : c$, entonces ω^i es observable en el sistema global.

Asumiendo que ω_p^i es observable, para algún $p \in [1 \dots c]$, pero en ω^i no. $\omega^i = \omega_1^i \oplus \dots \oplus \omega_p^i \oplus \dots \oplus \omega_c^i$ y $\omega_p^i \subset \omega^i$ por lo tanto ω_p^i es observable en ω^i , lo cual contradice la suposición y el teorema queda demostrado. \square

6. Aplicación a un Sistema Robótico

El sistema a considerar, para aplicar la presente propuesta de modelado, es el sistema presentado en (Patil et al., 2012), el cual no es un sistema de gran escala; pero sí es un sistema que permite mostrar las ventajas de la propuesta.

El sistema es un brazo robótico para recoger piezas en las posiciones $pp1, pp2$ o $pp3$ y colocarlas en $pp0$. Está compuesto por tres cilindros, dos horizontales y uno vertical; los cilindros horizontales se extienden o retraen, dependiendo de la orden del controlador para ubicar la pieza. El cilindro vertical se desplaza y recoge las piezas por succión, a partir de una sola orden de control. Los sensores son: uno para cada posición de los cilindros horizontales, otro incorporado en la unidad de succión, sensores de posición de las piezas y otro para detección de la pieza recogida. En la Figura 4 se presentan diferentes configuraciones para la interrelación de los componentes mecatrónicos.

Las señales externas al controlador $EExC$, que definen los modos de operación, son las señales de presencia de pieza en las posiciones 1, 2, y 3; es decir, $EExC = \{pp1, pp2, pp3\}$, donde $pp_i = \{1, 0\}$ (presencia o no presencia en posición $i = 1 : 3$). El brazo robótico comienza a trabajar cuando se detecta al menos una pieza; por lo tanto los modos de funcionamiento son 7, $OM = \{o_1, \dots, o_7\}$. La combinación o_0 no es posible porque no existiría funcionamiento. Por ejemplo $o_1 = [001]$ es el comportamiento del sistema en el cual hay una pieza en la posición 3 ($pp_3 = 1$).

Aunque el sistema no es un sistema de gran escala, con el objeto de aplicar el método propuesto, se ha subdividido con base en el número de dispositivos mecatrónicos presentes, así: El subsistema 1 corresponde al cilindro horizontal de la izquierda, el cual tiene dos sensores de posición $sr_{1,1} = Lc_{inicio}$ y $sr_{1,2} = Lc_{fin}$ y dos órdenes de control $cc_{1,1} = Ext_{LC}$ y $cc_{1,2} = Ret_{LC}$. El

subsistema 2 corresponde al cilindro horizontal de la derecha, tiene dos sensores de posición $sr_{2,1} = Rc_{inicio}$ y $sr_{2,2} = Rc_{fin}$ y dos órdenes de control $cc_{2,1} = Ext_{RC}$ y $cc_{2,2} = Ret_{RC}$. El subsistema 3 corresponde al cilindro vertical, el cual tiene un sensor que indica que la pieza ha sido colocada $sr_{3,1} = pp0$, dos sensores de posición $sr_{3,2} = Vc_{up}$ y $sr_{3,3} = Vc_{down}$, un sensor que indica succión o no $sr_{3,4} = vacío$ y dos órdenes de control, una para que el cilindro baje $cc_{3,1} = Ext_{VC}$ y $cc_{3,2} = Vac$ que es la orden para vacío del succionador; para recoger no es necesario orden de control puesto que cuando $cc_{3,1}$ no está activa, el cilindro se recoge por la compresión de un resorte.

Se desea modelar el comportamiento del sistema cuando está en modo de operación o_1 , los símbolos de E/S se presentan en la tabla 1:

Los símbolos se ordenan como $u_{1,s} = [cc_{1,1}cc_{1,2}]$, $u_{2,s} = [cc_{2,1}cc_{2,2}]$, $u_{3,s} = [cc_{3,1}cc_{3,2}]$ y $y_{1,j} = [sr_{1,1}sr_{1,2}]$, $y_{2,j} = [sr_{2,1}sr_{2,2}]$, $y_{3,j} = [sr_{3,1}sr_{3,2}sr_{3,3}sr_{3,4}]$.

Los lenguajes en cada subsistema \mathcal{L}_l , para el modo de operación o_1 son:

$\mathcal{L}_1 = (u_{1,0}, y_{1,0}) . t_{1,o_1}^0 (u_{1,2}, y_{1,2}) . t_{1,o_1}^1 (u_{1,2}, y_{1,3}) . t_{1,o_1}^2 (u_{1,1}, y_{1,2}) . t_{1,o_1}^3 (u_{1,1}, y_{1,0}) . t_{1,o_1}^4 (u_{1,0}, y_{1,0}) . t_{1,o_1}^5$ en $\tau_0, \tau_1, \tau_2, \tau_{12}, \tau_{13}, \tau_{18}$ instantes.

$\mathcal{L}_2 = (u_{2,0}, y_{2,0}) . t_{2,o_1}^0 (u_{2,2}, y_{2,2}) . t_{2,o_1}^1 (u_{2,2}, y_{2,3}) . t_{2,o_1}^2 (u_{2,1}, y_{2,2}) . t_{2,o_1}^3 (u_{2,1}, y_{2,0}) . t_{2,o_1}^4 (u_{2,0}, y_{2,0}) . t_{2,o_1}^5$ en $\tau_0, \tau_3, \tau_4, \tau_{10}, \tau_{11}, \tau_{18}$ instantes.

$\mathcal{L}_3 = (u_{3,0}, y_{3,0}) . t_{3,o_1}^0 (u_{3,2}, y_{3,4}) . t_{3,o_1}^1 (u_{3,2}, y_{3,6}) . t_{3,o_1}^2 (u_{3,3}, y_{3,7}) . t_{3,o_1}^3 (u_{3,1}, y_{3,5}) . t_{3,o_1}^4 (u_{3,1}, y_{3,1}) . t_{3,o_1}^5 (u_{3,3}, y_{3,5}) . t_{3,o_1}^6 (u_{3,3}, y_{3,7}) . t_{3,o_1}^7 (u_{3,2}, y_{3,14}) . t_{3,o_1}^8 (u_{3,0}, y_{3,4}) . t_{3,o_1}^9 (u_{3,0}, y_{3,0}) . t_{3,o_1}^{10}$ en $\tau_0, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9, \tau_{14}, \tau_{15}, \tau_{16}, \tau_{17}, \tau_{18}$ instantes.

Estos lenguajes representan el comportamiento del sistema. En el instante τ_0 , los tres subsistemas están en reposo (el valor de las lecturas de los sensores es de cero y no hay órdenes de control), en el instante τ_1 se genera un evento en el subsistema 1, de extender el cilindro horizontal izquierdo y la respuesta del sensor $sr_{1,1}$ es inmediata, por lo tanto el evento en τ_1 es: $(u_{1,2}, y_{1,2}) . t_{1,o_1}^1$; al cabo de un tiempo en el instante τ_2 se genera un evento en el subsistema 1 denotado como $(u_{1,2}, y_{1,3}) . t_{1,o_1}^2$, luego se genera un evento en el subsistema 2 denotado como $(u_{2,2}, y_{2,2}) . t_{2,o_1}^1$ y así sucesivamente.

Se considera un sistema estocástico por lo tanto $t_{l,o_{om}}^i \sim f(t)$. Por ejemplo: $\omega_3^4 = (u_{3,2}, y_{3,6}) . t_{3,o_1}^2, t_{3,o_1}^2 \sim N(2, 0, 1)$ y para $1 - \alpha = 0,95$, el intervalo de confianza es $[1,804 - 2,196]$.

Para representar este comportamiento se construyen las STIPNs para cada subsistema, aplicando el algoritmo 1. Cada subsistema inicia con un lugar, cuya función de salida está representada por las lecturas de los sensores, y_j , en τ_0 y el diferencial dy_j como cadenas de ceros; es decir $y_{1,0}/00$, lo cual se muestra en la Figura 5.



Figura 5: Proceso de modelado del brazo robótico a partir del lenguaje: estados iniciales

Tabla 1: Símbolos de E/S: Brazo Robótico

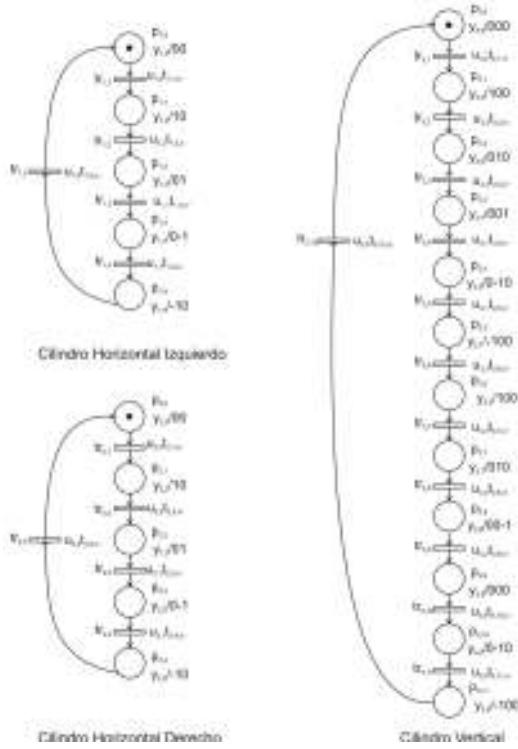
$f(t_{om}^v)$	Órdenes de Control (Entradas)						Lecturas Sensoriales (Salidas)								
	Ext_{LC}	Ret_{LC}	Ext_{RC}	Ret_{RC}	Ext_{VC}	Vac	Lc_{inicio}	Lc_{fin}	Rc_{inicio}	Rc_{fin}	pp_0	vc_{up}	vc_{down}	$vacio$	pp_3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1
$N(1,5,0,12)$	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1
0	1	0	1	0	0	0	1	1	1	0	0	0	0	0	1
$N(2,0,1)$	1	0	1	0	0	0	1	1	1	1	0	0	0	0	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮



Figura 6: Proceso de modelado del brazo robótico a partir del lenguaje: paso 1

En τ_1 el subsistema 1, evoluciona a otro estado; para esta evolución se añade una transición con $\lambda(tr_{1,1}) = u_{1,2}.t_{1,o_1}^1$ y se alcanza un nuevo lugar con función de salida $y_{1,2}/10$, ($dy_j = 10 - 00$); los otros subsistemas no evolucionan en este instante, este comportamiento se puede observar en la Figura 6.

Al final, las st-IPNs que modelan el comportamiento de los tres subsistemas se muestran en la Figura 7.



$\tau_8, \tau_9, \tau_{14}, \tau_{15}, \tau_{16}, \tau_{17}, \tau_{18}$ instantes; es decir $\mathcal{L}_3 = \omega_3^0 \omega_3^5 \omega_3^6 \omega_3^7 \omega_3^8 \omega_3^9 \omega_3^{14} \omega_3^{15} \omega_3^{16} \omega_3^{17} \omega_3^{18}$.

Es de aclarar que superíndice de t , indica la secuencia de eventos en el subsistema 3 y el superíndice de ω , es la secuencia en los eventos de todo el sistema.

Considerando una entrada externa a la planta como un fallo en la unidad del succionador; es decir $EExP = \{f_{suc}\}$, el símbolo de entrada al subsistema 3 será: $u_{3,s} = [f_{suc}cc3,1cc3,2]$ y el de salida $y_{3,j} = [sr3,1sr3,2sr3,3sr3,4]$.

Sea s una secuencia de eventos normales en el subsistema 3, tal que: $s = (u_{3,0}, y_{3,0}) \cdot t_{3,o_1}^0 (u_{3,2}, y_{3,4}) \cdot t_{3,o_1}^1 (u_{3,2}, y_{3,6}) \cdot t_{3,o_1}^2 (u_{3,3}, y_{3,7}) \cdot t_{3,o_1}^3 (u_{3,1}, y_{3,5}) \cdot t_{3,o_1}^4 (u_{3,1}, y_{3,1}) \cdot t_{3,o_1}^5 (u_{3,0}, y_{3,0}) \cdot t_{3,o_1}^6 (u_{3,3}, y_{3,4}) \cdot t_{3,o_1}^7 (u_{3,3}, y_{3,6}) \cdot t_{3,o_1}^8 (u_{3,2}, y_{3,14}) \cdot t_{3,o_1}^9 (u_{3,0}, y_{3,4}) \cdot t_{3,o_1}^{10}$ en $\tau_0, \tau_5, \tau_6, \tau_7, \tau_8$ y sea t una secuencia de eventos observados tal que $t \in \mathcal{L}_3/s$.

Ahora, asumiendo que en el instante τ_9 se genera el fallo en el succionador; es decir $f_{suc} = 1$, el símbolo de entrada no cambia puesto que el fallo es no-observable; pero las lecturas sensoriales podrían cambiar después de uno o más eventos; es decir en la secuencia t . Entonces, si la $Pt_{U_o}(\mathcal{L}_3) \neq Pt_{U_o}(st)$ y/o la proyección $Pt_Y(\mathcal{L}_3) \neq P_Y(st)$; entonces el evento de fallo es n-observable, donde n es el número de eventos que suceden hasta que el fallo es detectado. Por ejemplo si st es:

$st = (u_{3,0}, y_{3,0}) \cdot t_{3,o_1}^0 (u_{3,2}, y_{3,4}) \cdot t_{3,o_1}^1 (u_{3,2}, y_{3,6}) \cdot t_{3,o_1}^2 (u_{3,3}, y_{3,7}) \cdot t_{3,o_1}^3 (u_{3,1}, y_{3,5}) \cdot t_{3,o_1}^4 (u_{3,1}, y_{3,1}) \cdot t_{3,o_1}^5 (u_{3,0}, y_{3,0}) \cdot t_{3,o_1}^6 (u_{3,3}, y_{3,4}) \cdot t_{3,o_1}^7 (u_{3,3}, y_{3,6}) \cdot t_{3,o_1}^8 (u_{3,2}, y_{3,14}) \cdot t_{3,o_1}^9 (u_{3,0}, y_{3,4}) \cdot t_{3,o_1}^{10}$ en $\tau_0, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9, \tau_{14}, \tau_{15}, \tau_{16}, \tau_{17}, \tau_{18}$ instantes.

$Pt_Y(\mathcal{L}_3) = Pt_Y(\omega_3^0) Pt_Y(\omega_3^5) \cdots Pt_Y(\omega_3^{18})$, donde, $Pt_Y(\omega_3^0) = Pt_Y((u_{3,0}, y_{3,0}) \cdot t_{3,o_1}^0) = P_{CY}(u_{3,0}, y_{3,0}) P(t_{3,o_1}^0)$, ver Definiciones 2 y 7; entonces $Pt_Y(\omega_3^0) = y_{3,0}$.

Por lo tanto $Pt_Y(\mathcal{L}_3) = y_{3,0}y_{3,4}y_{3,6}y_{3,7}y_{3,5}y_{3,1}y_{3,5}y_{3,7}y_{3,14}y_{3,4}y_{3,0}$.

$Pt_Y(st) = y_{3,0}y_{3,4}y_{3,6}y_{3,7}y_{3,5}y_{3,1}y_{3,0}y_{3,4}y_{3,6}y_{3,14}y_{3,4}$. De donde se concluye que $Pt_Y(\mathcal{L}_3) \neq Pt_Y(st)$ y el evento de fallo se puede detectar después de 2 eventos de su ocurrencia.

Pero si la entrada no es un fallo, sino un cambio en el tiempo que tarda el cilindro horizontal en recoger el brazo, es decir, $EExP = \{e_{uo_1}\}$, el símbolo de entrada al subsistema 2 no cambia puesto que es un evento no-observable; las lecturas sensoriales tampoco se verían afectadas; entonces si $Pt_{U_o}(\mathcal{L}_2) \neq Pt_{U_o}(st)$ el evento no-observable puede ser detectado.

El lenguaje en comportamiento normal para el subsistema 2 es $\mathcal{L}_2 = (u_{2,0}, y_{2,0}) \cdot t_{2,o_1}^0 (u_{2,2}, y_{2,2}) \cdot t_{2,o_1}^1 (u_{2,2}, y_{2,3}) \cdot t_{2,o_1}^2 (u_{2,1}, y_{2,2}) \cdot t_{2,o_1}^3 (u_{2,1}, y_{2,0}) \cdot t_{2,o_1}^4 (u_{2,0}, y_{2,0}) \cdot t_{2,o_1}^5$, en $\tau_0, \tau_3, \tau_4, \tau_{10}, \tau_{11}, \tau_{18}$ instantes, es decir $\mathcal{L}_2 = \omega_2^0 \omega_2^3 \omega_2^4 \omega_2^{10} \omega_2^{11} \omega_2^{18}$.

Por ejemplo si st es:

$st = (u_{2,0}, y_{2,0}) \cdot t_{2,o_1}^0 (u_{2,2}, y_{2,2}) \cdot t_{2,o_1}^1 (u_{2,2}, y_{2,3}) \cdot t_{2,o_1}^2 (u_{2,1}, y_{2,2}) \cdot t_{2,o_1}^3 (u_{2,1}, y_{2,0}) \cdot t_{2,o_1}^4 (u_{2,0}, y_{2,0}) \cdot t_{2,o_1}^5$.

$Pt_{U_o}(\mathcal{L}_2) = Pt_{U_o}(\omega_2^0) Pt_{U_o}(\omega_2^3) \cdots Pt_{U_o}(\omega_2^{18})$, donde $Pt_{U_o}(\omega_2^0) = Pt_{U_o}((u_{2,0}, y_{2,0}) \cdot t_{2,o_1}^0) = P_{CU_o}(u_{2,0}, y_{2,0}) P(t_{2,o_1}^0) = u_{2,0} \cdot t_{2,o_1}^0$; entonces:

$Pt_{U_o}(\mathcal{L}_2) = u_{2,0} \cdot t_{2,o_1}^0 u_{2,2} \cdot t_{2,o_1}^1 u_{2,2} \cdot t_{2,o_1}^2 u_{2,1} \cdot t_{2,o_1}^{*3} u_{2,1} \cdot t_{2,o_1}^4 u_{2,0} \cdot t_{2,o_1}^5$.

Si $a \leq t_{2,o_1}^{*3} \leq b$, donde $[a, b]$ es el intervalo de confianza para el tiempo t_{2,o_1}^3 , se considera que la variación del tiempo del evento obedece a una variabilidad propia del comportamiento estocástico del sistema, de lo contrario se ha detectado un comportamiento que no es normal.

Lo anterior comprueba la importancia de la proyección de eventos temporizados definida en este artículo, para detectar comportamientos no normales a partir de eventos no-observables, en sistemas estocásticos.

7. Conclusiones

Este artículo presenta un generador de lenguaje regular que permite modelar el comportamiento observado de SED estocásticos, representado bajo una estructura de red de Petri denominada st-IPN; la cual es una extensión de las redes de Petri interpretadas, con características relevantes en cuanto a que es determinística, su lenguaje es el mismo lenguaje legal del sistema y permite modelar sistemas con eventos estocásticos. A su vez, se define la estructura de lenguaje regular con base en la organización de las señales de entrada / salida y con base en los modos de funcionamiento. La evolución de la red se genera a partir de eventos temporizados y el método propuesto puede ser aplicado a sistemas de gran escala a partir de la subdivisión en sistemas más sencillos. Además se presenta un nuevo concepto de observabilidad que se ha denominado observabilidad temporal, que permite realizar test de observabilidad en sistemas estocásticos.

Los resultados de este artículo permiten generar nuevos temas de investigación, relacionados con la subdivisión óptima de sistemas, la generación de eventos temporizados estocásticos y la aplicación en problemas de diagnóstico de fallos.

English Summary

Deterministic Generation of Regular Languages in Discrete Event Systems

Abstract

In this paper is proposed a stochastic interpreted Petri net, (st-IPN) as a model to represent the regular language derived from the combination of input signals in a Discrete Event System (DES) in closed loop. The input signals are external signals affecting the system and the control commands issued by the controller to the plant and the output signals are the responses of the sensors to the control commands. The st-IPN proposed is a deterministic generator of the system legal language able to represent sequences of stochastic timed events. The proposed model can be applied to large-scale systems, from the division of the system into subsystems, since the global model can be a composition of the subsystems models.

Keywords:

Modelling of Discrete Event Systems, Petri Nets, Timed Observability.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias a la comisión académica financiada por la Universidad del Cauca, referencia 2.3-31.2/05 2011.

Referencias

- Ashley, J., 2004. Diagnosis of condition systems. Ph.D. thesis, University of Kentucky.
- Ashley, J., Holloway, L., 2004. Qualitative diagnosis of condition systems. *Discrete Event Dynamic Systems* 14 (4), 395–412.
- Basile, F., Chiacchio, P., Coppola, J., De Tommasi, G., June 2011. Identification of petri nets using timing information. En: Dependable Control of Discrete Systems (DCDS), 2011 3rd International Workshop on. pp. 154–161.
- Berthomieu, B., Diaz, M., Mar 1991. Modeling and verification of time dependent systems using time petri nets. *Software Engineering, IEEE Transactions on* 17 (3), 259–273.
- Berthomieu, B., Peres, F., Vernadat, F., 2006. Bridging the gap between timed automata and bounded time petri nets. En: Asarin, E., Bouyer, P. (Eds.), *Formal Modeling and Analysis of Timed Systems*. Vol. 4202 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 82–97.
- Berthomieu, B., Ribet, P.-O., Vernadat, F., 2004. The tool tina - construction of abstract state spaces for petri nets and time petri nets. *International Journal of Production Research* 42 (14), 2741–2756.
- Boucheneb, H., Hadjidj, R., 2006. Model checking for time petri nets. *Theoretical Computer Science* 353 (13), 208–227.
- Cabasino, M., Giua, A., Poccetti, M., Seatzu, C., 2011. Discrete event diagnosis using labeled petri nets. an application to manufacturing systems. *Control Engineering Practice* 19 (9), 989–1001.
- Cassandras, C. G., Lafontaine, S., 2008. Introduction to Discrete Event Systems.
- Cassez, F., Tripakis, S., may 2008. Fault diagnosis with dynamic observers. En: *Discrete Event Systems*, 2008. WODES 2008. 9th International Workshop on. pp. 212–217.
- Correcher, A., 2005. Diagnóstico de fallos intermitentes en procesos industriales basado en modelos de eventos discretos. Ph.D. thesis, Universidad Politécnica de Valencia.
- Desel, J., July 2013. On cyclic behaviour of unbounded petri nets. En: Application of Concurrency to System Design (ACSD), 2013 13th International Conference on. pp. 110–119.
- Dotoli, M., Fanti, M., Mangini, A., Ukovich, W., may 2008a. On-line identification of petri nets with unobservable transitions. En: *Discrete Event Systems*, 2008. WODES 2008. 9th International Workshop on. pp. 449–454.
- Dotoli, M., Fanti, M., Mangini, A. M., May 2008b. Real time identification of discrete event systems using petri nets. Vol. 44. Pergamon Press, Inc., Tarrytown, NY, USA, pp. 1209–1219.
DOI: 10.1016/j.automatika.2007.10.014
- Estrada-Vargas, A.-P., Lesage, J.-J., López-Mellado, E., Jun 2012. Identification of industrial automation systems: Building compact and expressive petri net models from observable behavior. En: 2012 American Control Conference (ACC'12). Canada, pp. 6095–6101.
- Fanti, M. P., Mangini, A. M., Ukovich, W., 2012. Fault detection by labeled petri nets in centralized and distributed approaches. *Automation Science and Engineering, IEEE Transactions on* PP (99), 1.
- Gardey, G., Lime, D., Magnin, M., (h. Roux, O., 2005. Roméo: A tool for analyzing time petri nets. En: In Proc. CAV'05, vol. 3576 of LNCS. Springer, pp. 418–423.
- Gaubert, S., Giua, A., Dec 1996. Deterministic weak-and-marked petri net languages are regular. *Automatic Control, IEEE Transactions on* 41 (12), 1802–1803.
- Girault, C., Valk, R., 2003. Petri nets for systems engineering - a guide to modeling, verification, and applications. Springer.
DOI: <http://www.springer.com/computer/swe/book/978-3-540-41217-5>
- Giua, A., 2013. Supervisory control of petri nets with language specifications. En: Seatzu, C., Silva, M., van Schuppen, J. H. (Eds.), *Control of Discrete-Event Systems*. Vol. 433 of *Lecture Notes in Control and Information Sciences*. Springer London, pp. 235–255.
- González-Miranda, O., Cerrada-Lozada, M., 2014. Diagnóstico de sistemas de eventos discretos controlados: Un enfoque basado en crónicas y análisis modular usando modelos de autómatas. *Revista Iberoamericana de Automática e Informática Industrial (RIAII)* 11 (2), 191–201.
DOI: <http://dx.doi.org/10.1016/j.riaai.2014.02.003>
- Guasch, A., Piera, M., Casanovas, J., Figueras, J., 2005. Modelado y simulación. Alfaomega.
- Hernández, K., Meda-Campana, M., July 2012. Fault diagnosis using petri nets. a case study. *Proceedings of the 10th Latin American and Caribbean Conference for Engineering and Technology*.
- Holloway, L. E., Guan, X., Sundaravadivelu, R., Ashley, Jr., J., Oct. 2000. Automated synthesis and composition of taskblocks for control of manufacturing systems. *Trans. Sys. Man Cyber. Part B* 30 (5), 696–712.
- Hu, H., Zhou, M. C., Li, Z., Tang, Y., 2012. An optimization approach to improved petri net controller design for automated manufacturing systems. *Automation Science and Engineering, IEEE Transactions on* PP (99), 1.
- Ichikawa, A., Hiraishi, K., 1988. Analysis and control of discrete event systems represented by petri nets. En: Varaiya, P., Kurzhanski, A. (Eds.), *Discrete Event Systems: Models and Applications*. Vol. 103 of *Lecture Notes in Control and Information Sciences*. Springer Berlin Heidelberg, pp. 115–134.
- Kumar, R., Holloway, L., Feb 1996. Supervisory control of deterministic petri nets with regular specification languages. *Automatic Control, IEEE Transactions on* 41 (2), 245–249.
- Lunze, J., 1998. Qualitative modelling of dynamical systems: Motivation, methods, and prospective applications. *Mathematics and Computers in Simulation* 46 (6), 465–483.
- Merlin, P., Farber, D. J., Sep 1976. Recoverability of communication protocols -implications of a theoretical study. *Communications, IEEE Transactions on* 24 (9), 1036–1043.
- Muñoz, D. M., Correcher, A., García, E., Morant, F., 2014. Identification of stochastic timed discrete event systems with st-ipn. *Mathematical Problems in Engineering* 2014 (00), 21.
DOI: 10.1155/2014/85312
- Murata, T., apr 1989. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* 77 (4), 541–580.
DOI: 10.1109/5.24143
- Nakamura, D., Takeda, Y., Murakoshi, H., Funakubo, N., Dohi, Y., Aug 1998. A modeling language for petri net based factory automation systems. En: *Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE*. Vol. 1. pp. 120–125.
- Patil, S., Vyatkin, V., Sorouri, M., 2012. Formal verification of intelligent mechatronic systems with decentralized control logic. En: *Emerging Technologies Factory Automation (ETFA)*, 2012 IEEE 17th Conference on. pp. 1–7.
- Peres, F., Berthomieu, B., Vernadat, F., Sep. 2011. On the composition of time petri nets. *Discrete Event Dynamic Systems* 21 (3), 395–424.
- Piera, M., Music, G., 2011. Coloured petri net scheduling models: Timed state space exploration shortages. *Mathematics and Computers in Simulation* 82 (3), 428–441, 6th Vienna International Conference on Mathematical Modelling.
DOI: <http://dx.doi.org/10.1016/j.matcom.2010.10.014>
- Ramadge, P., Wonham, W., Jan 1989. The control of discrete event systems. Vol. 77. pp. 81–98.
- Ramírez-Treviño, A., Ruiz-Beltran, E., Aramburo-Lizarraga, J., Lopez-Mellado, E., March 2012. Structural diagnosability of des and design of reduced petri net diagnosers. Vol. 42. pp. 416–429.
DOI: 10.1109/TS MCA.2011.2169950
- Ramírez-Treviño, A., Ruiz-Beltran, E., Rivera-Rangel, I., Lopez-Mellado, E., Jan 2007. Online fault diagnosis of discrete event systems. a petri net-based approach. Vol. 4. pp. 31–39.
DOI: 10.1109/TASE.2006.872120
- Salum, L., 2008. Petri nets and time modelling. *The International Journal of Advanced Manufacturing Technology* 38 (3–4), 377–382.
- Sampath, M., Sengupta, R., Lafontaine, S., Sinnamohideen, K., Teneketzis, D., Mar 1996. Failure diagnosis using discrete-event models. Vol. 4. pp. 105–124.
- Silva, M., 1993. Introducing petri nets. En: *Practice of Petri Nets in Manufacturing*. Springer Netherlands, pp. 1–62.
DOI: 10.1007/978-94-011-6955-4-1
- Silva, M., Recalde, L., 2007. Redes de petri continuas: Expresividad, análisis y control de una clase de sistemas lineales comutados. *Revista Iberoamericana de Automática e Informática Industrial* 04 (03), 5–33.
- Sreenivas, R., Aug 1993. Deterministic lambda-free petri net languages and their application to the supervisory control of discrete event dynamic systems. En: *Circuits and Systems*, 1993., Proceedings of the 36th Midwest Symposium on. pp. 340–343.
- Sreenivas, R., May 2006. On minimal representations of petri net languages. *Automatic Control, IEEE Transactions on* 51 (5), 799–804.
- Sreenivas, R., Krogh, B., 1991. Petri net based models for condition/event systems. En: *American Control Conference*, 1991. pp. 2899–2904.
- Valk, R., Vidal-Naquet, G., 1981. Petri nets and regular languages. *Journal of Computer and System Sciences* 23 (3), 299–325.