

Sistema de Envasado de Líquidos Modelado con Redes de Petri y Simulado con LabVIEW y DSC

Miguel Ángel Trigos Martínez

PhD (c) en Automática y Robótica
Centro de Automática y Robótica Universidad Politécnica de Madrid - España Docente Tiempo Completo,
Investigador Grupo de Aplicaciones Mecatrónicas,
Universidad Santo Tomás, Bucaramanga, Colombia
matrigos@ettsi.upm.es

Jair Leandro Landínez Salazar

Ingeniero Mecatrónico
Universidad Santo Tomás Bucaramanga, Colombia
freeyayo21@gmail.com

Resumen— En las últimas décadas los grupos de investigación, han utilizado las herramientas de simulación para validar sus desarrollos, antes de hacer pruebas reales, los resultados son bastante aceptables y dan o no autorización para continuar en un proyecto. Herramientas de software para simulación existen muchas, pero se busca utilizar LabVIEW y DSC debido a que su ambiente de programación es amigable y muy utilizado en la automática. El trabajo mostrado a continuación, consiste en el desarrollo de una aplicación enfocada hacia el Diagnóstico y Detección de Fallos, mediante la implementación de un algoritmo sistemático, construido con la herramienta de modelado conocida como “Redes de Petri (RdP)”, donde el objetivo principal es realizar un aporte en el ámbito de la seguridad industrial, para evitar pérdidas económicas, evaluadas en daños a los equipos y, lo más importante, preservar la integridad física de los operarios humanos. El contenido muestra e ilustra la simulación de un Proceso de Envasado de Líquidos y la implementación de su Diagnosticador mediante las RdP, con la finalidad de evaluar el comportamiento normal y de fallo ofrecido por el proceso. Se diseña una plataforma e interface de comunicación hombre máquina, en la cual se lleva a cabo la monitorización, control y supervisión On-line del sistema planteado.

Palabras clave— Redes de Petri, Sistema Híbrido, Simulación, Diagnóstico de Fallos, Diagnosticador, LabVIEW, Máquinas de Estado, Diagramas de Bloques.

Abstract— In recent decades, research groups, have been using simulation tools to validate their development, before actual testing, the results are quite acceptable and give permission or not to continue a project. Simulation software tools there are many, but we are using LabVIEW and DSC (National Instruments) because its programming environment is friendly and widely used in the automatic. The work shown below, is the development of an application focused on the fault diagnosis, by implementing a systematic algorithm, built using the modeling tool known as “Petri Nets (MOP)”, which the main target is to make a contribution in the field of industrial safety, avoiding economic losses, estimated in damage to equipment and most important to preserve the physical integrity of human operators. The content illustrates the simulation in LabVIEW of a liquid packaging process and the implementation of their diagnoser using Petri Nets, in order to evaluate the normal and

fault behavior offered by the process. Designing a communication platform and man machine interface, which is carried out monitoring, control and supervising system of way on-line.

Keywords— Petri Nets, Hybrid System, Simulation, Fault diagnosis, Diagnoser, LabVIEW, State Machine, Bloque Diagrams

I. INTRODUCCIÓN

La creación y desarrollo de herramientas de software y plataformas de simulación, han mostrado uno de los grandes avances de la era tecnológica, en donde los conocimientos, planteamientos teóricos y modelos matemáticos, antes de ser llevados al campo real, pueden ser simulados y sometidos a diferentes pruebas para comprobar y definir su alcance, parámetros de fiabilidad, estabilidad y respuesta ante diferentes elementos, que pueden percibirse en un ambiente físico. De esta forma, se crea una predicción de cómo un proceso puede comportarse en condiciones normales o de exigencia, antes de implementarse.

Las fábricas y los grupos de investigación, realizan grandes esfuerzos en hacer que los procesos sean, cada vez, más seguros y confiables, donde la seguridad esté enmarcada entre la parte económica y la seguridad de los operarios. En lo económico es importante reducir los costos por paradas de mantenimiento y las pérdidas que acarrear el incumplimiento en los plazos de entrega, y la seguridad de los operarios.

La simulación aquí expuesta, sirve para validar el algoritmo de diagnóstico de fallos de sistemas híbridos, publicado en [4] [5]. Como herramienta de modelado en la investigación se usan las Redes de Petri (RdP), de las cuales en la actualidad, no se encuentran desarrollos implementados en LabVIEW.

Como herramienta se emplea el software LabVIEW y Datalogging and Supervisory Control (DSC) de National Instruments, en LabVIEW se emula el modelo de una planta de envasado de líquidos. El DSC es utilizado como herramienta de supervisión On-line. Todo es representado en el ambiente gráfico de LabVIEW, el modelo de RdP observa la evolución del proceso en condiciones de operación normal y también, la respuesta del diagnosticador de fallos, ante la presencia de transiciones de fallo, que provoca la generación de alarmas. La aplicación cumple las funciones de un sistema SCADA, es una interfaz hombre máquina, la cual permite al operario interactuar y mantenerse informado de los sucesos ocurridos en el proceso.

En la parte inicial del artículo, sección 2, se dan a conocer los conceptos fundamentales de las RdP Híbridas, el algoritmo de diagnóstico de fallos y LabVIEW y DSC, los cuales son necesarias en el entendimiento del algoritmo de diagnóstico de fallos. En la sección 3, se describe de forma general el proceso de envasado de líquidos y en la sección 4, se presenta la herramienta de simulación y supervisión, LabVIEW y DSC respectivamente, observándose el modelo de RdP del proceso y el diagnosticador. En la sección 5, se enuncia la conclusión del trabajo realizado, se exalta su gran aporte en la utilización del LabVIEW como una excelente alternativa de representación de las Redes de Petri.

II. CONCEPTOS BÁSICOS

A. Redes de Petri.

Los conceptos de Redes de Petri Híbridas (RdPH) presentados aquí, están basado en los conceptos definidos [1]. Para profundizar los conceptos de RdP consultar [1] y [2] y [3].

B. Diagnóstico de Fallos de Sistemas Híbridos.

En el trabajo previo de [5], se inicia el diagnóstico de fallos de SED's, posteriormente, se usa para el diagnóstico de fallos Intermitentes y se pasa a su implementación en Sistemas Híbridos. A continuación se presenta, el resumen del proceso de construcción del modelo y diagnosticador, el cual puede ser conocido en detalle en [4].

C. Algoritmo de Construcción del Modelo y Diagnosticador

El sistema general está compuesto de varios subsistemas, distribuidos según la naturaleza de las señales que lo integran, sean discretas, continuas o híbridas.

a- Clasificación del sistema general H en subsistemas. $H=H_1 \cup H_2 \cup \dots \cup H_n$, n es el número de subsistemas que componen el sistema.

b- Realizar el modelo de RdP de cada uno de los elementos que componen los diferentes subsistemas. $Q_i = (P_i, T_i, I, O, M_i)$.

c- Realizar la operación de integración; $\tilde{Q} = (\tilde{P}, \tilde{T}, \tilde{I}, \tilde{O}, \tilde{M})$ Representar en un modelo de RdP el comportamiento del SED's,

d- Llevar a cabo el refinamiento del modelo de cada uno de los subsistemas. $Q_i = (P_i, T_i, I, O)$. El refinamiento se realiza basado en la construcción de la tabla de integración de M sensores del sistema, $h_j = \tilde{P} \rightarrow \tilde{Y}_j, j = 1, \dots, M$.

d- Construcción del diagnosticador. Es una RdP Diagnosticadora, $G_d = (P_d, T_d, I, O, p_0, t_0, t_{end})$ Para la construcción del diagnosticador se definen las siguientes funciones:

Función de Asignación de Nivel LA: $P_0 \times \Delta \times T^* \rightarrow \Delta, P \in P_0, l \in \Delta, \text{ y } s \in \ell(Q, p)$. LA Asigna el nivel l sobre s comenzando sobre p y siguiendo la dinámica de Q siguiendo a (1):

$$LA(p, l, s) \begin{cases} \{N\} & \text{si } \forall i [T_{f_i} \notin s] \\ \{F\} & \text{si } \forall i [T_{f_i} \in s] \end{cases} \quad (1)$$

Función de Expansión de Fallo. $EF: R_N \times F_i \rightarrow R_F$, donde R_N es rama de comportamiento normal, y para cada fallo F_i , EF asigna una rama de fallo respectiva. El diagnosticador G_b tendrá tantas ramas como fallos más uno, ver (2).

$$R_{G_d} = R_N + \sum_{i=1}^M R_{f_i} \quad (2)$$

R_{G_d} representa el número total de ramas del diagnosticador. R_N y R_F son las ramas de comportamiento normal y de fallo respectivamente.

Diagnosticabilidad. Una RdP es diagnosticable en relación de la distribución de fallos si se cumple Equ.3:

$$(\sigma \in \ell(Q, M_0), \wedge, \sigma \in T_0), (\forall i, k \in \Pi, f)$$

$$\text{Donde : } \|\sigma\|_{\infty} : M(p_{f_i}) + M(p_{f_k}) \rightarrow M(p_{f_s}) \quad (3)$$

σ es la secuencia de transiciones observables, por lo tanto la RdP que representa el modelo del sistema, es diagnosticable si en un número finito de transiciones observables ésta alcanza un marcado de fallo $M(p_i)$. Solo el $M(p_i)$ o con otra marca de fallo $M(p_{jk})$ se puede identificar un fallo de mayor orden o un fallo crítico. Para profundizar en el algoritmo de diagnóstico de fallos, se puede consultar [5].

LabVIEW y DSC

LabVIEW es un lenguaje de programación gráfico “lenguaje G”, tiene diversas aplicaciones en la academia y la industria, sin conocer previamente el uso de LabVIEW con las RdP. Si se busca profundizar los conceptos en LabVIEW, consultar [9].

Datalogging and Supervisory Control (DSC): Ofrece herramientas y características para el desarrollo de sistemas distribuidos de monitoreo y control. Este módulo permite la construcción de aplicaciones de registro de datos y creación de alarmas. Facilita la creación de las interfaces hombre-máquina y acciones supervisoras o de control, sistemas SCADA y visión de tendencias en tiempo real e históricos de datos además de opciones de seguridad para las aplicaciones. Para mejorar los conceptos en DSC ver [10].

III. SIMULACIÓN DEL PROCESO DE ENVASADO DE LÍQUIDOS EN LABVIEW

A continuación se describe el Proceso de Envasado de Líquidos y la dinámica que define su comportamiento. Posteriormente se detalla la técnica de programación utilizada para el desarrollo de la herramienta de simulación y el diagnosticador con RdP, construidos en LabVIEW.

Basado en el enfoque de [4], el sistema híbrido es separado en las dinámica continua y discreta, con esto se puede interpretar mejor el funcionamiento del proceso.

Dinámica Continua

Está compuesta de tres variables fundamentales del sistema global, las cuales están divididas en subsistemas: El Subsistema de Control de temperatura, Subsistema de Control de pH y Subsistema

de Control de Nivel. La parte continua del proceso cuenta con un tanque, alimentado por dos líquidos (A) y (B) respectivamente, el llenado del tanque corresponde a establecer un nivel de mezcla indicado y conseguir las características de pH deseables, dicha mezcla es calentada por un sistema de inyección de vapor, hasta lograr las condiciones de temperatura apropiadas del líquido a envasar.

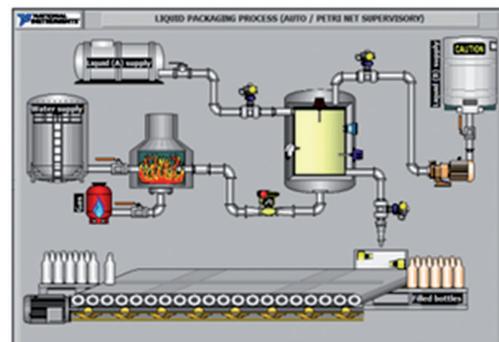
Para las acciones de control, se cuenta con dos válvulas alimentadoras de líquido y una válvula de inyección de vapor. Para medir las variables se utiliza un sensor de temperatura, un sensor de nivel y un sensor de pH. Cada variable es controlada de manera independiente por su respectivo controlador.

Dinámica Discreta

Consta de los siguientes componentes: Banda transportadora, la cual es movida por un motor M, válvula de llenado de líquido VII, sensor de llegada de botella para envasado BP, Sensor de llenado de botellas LL y controlador C. En el funcionamiento del sistema, asumimos que el líquido del tanque está en óptimas condiciones y que siempre existen botellas en la parte inicial de la banda transportadora.

En el inicio del sistema de envasado el motor de la banda hace que las botellas se desplacen hasta el lugar de envasado, el sensor BP detecta la botella y por consiguiente el motor de la banda se detiene. La válvula VII de llenado se abre, cuando la botella está llena el sensor de llenado LL informa este suceso y se cierra la válvula de llenado VII, una vez esta secuencia de funcionamiento normal se haya cumplido, se puede retirar la botella para que sea almacenada, como se puede observar se trata de un proceso secuencial y repetitivo que actúa indefinidamente. En la Fig. 1 se ilustra un esquema general que comprende el sistema de envasado.

FIG. 1. SISTEMA DE ENVASADO DE LÍQUIDOS

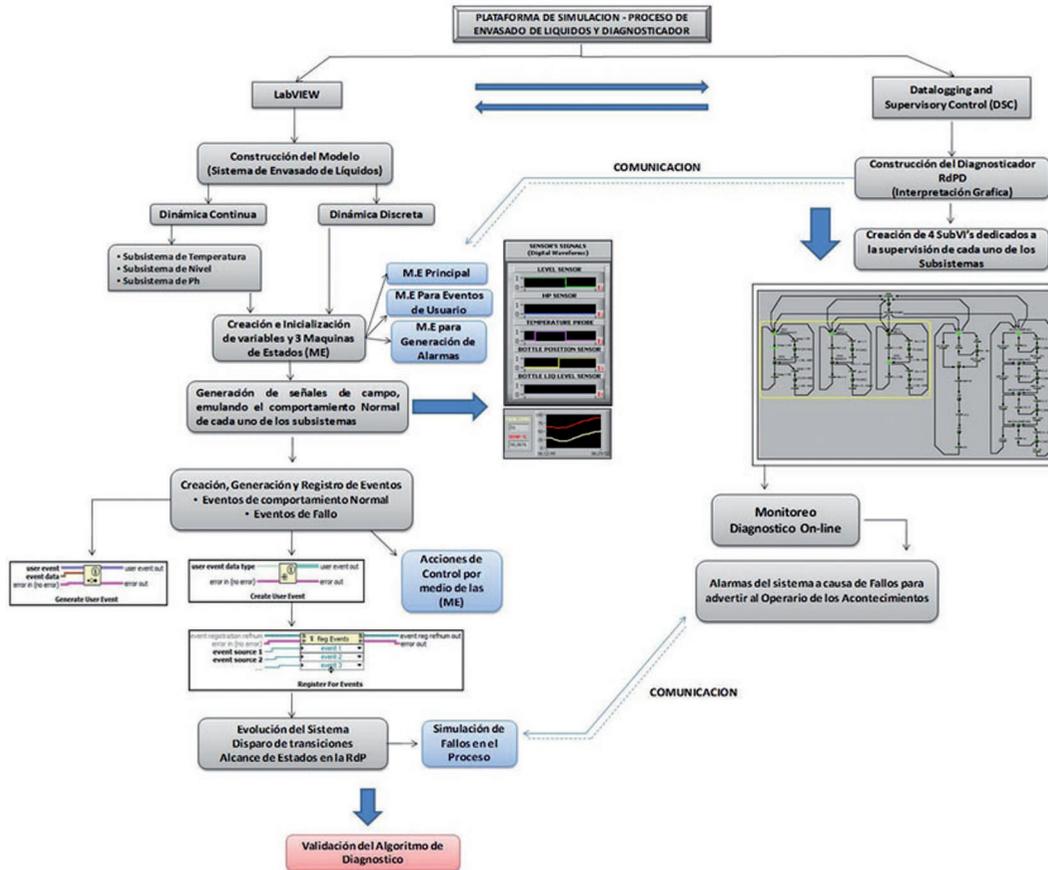


Fuente: Autor del proyecto

La construcción de la plataforma, se aborda de tal forma que se pueda observar a manera general cuáles fueron los pasos y metodología propuesta para la implementación

de las RdP en el entorno LabVIEW. La Fig. 2, ilustra el diagrama secuencial que sigue la estructura de programación concentrándose en la interfaz que contiene el Diagnosticador.

FIG. 2. DIAGRAMA SECUENCIAL Y ESTRUCTURA DE PROGRAMACIÓN



Fuente: Autor del proyecto

IV. SIMULACIÓN LABVIEW

La plataforma cuenta con un panel principal, que permite al usuario tener acceso a la aplicación. La simulación se lleva a cabo de manera cíclica, en donde el proceso de envasado transcurre de manera automática y el operario sólo puede intervenir cuando desee simular el conjunto de fallos, para evaluar el desempeño del Diagnosticador; la plataforma “Liquid Packaging Process Sim.VI” es la estructura principal, de la cual se desprenden las demás ventanas (interfaces) de simulación, ver Fig. 3.

Previo a la metodología implementada, se muestra un panorama global de la relación entre los VI’s elaborados, si se observa el árbol de

jerarquía se aprecia la comunicación de cada subVI y el flujo de datos, entre las diferentes interfaces que permite su interacción, ver Fig. 4.

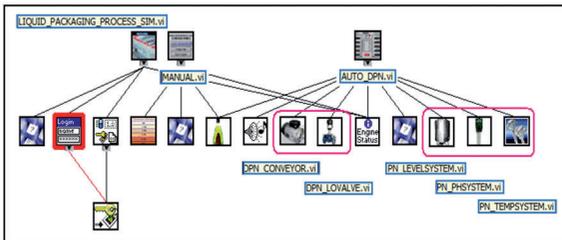
Con el objetivo de dar un enfoque hacia la implementación de la RdP en LabVIEW, se considerará sólo la ventana de programación (diagrama de bloques). Esta interfaz de usuario recibe el nombre de Auto_DPN.VI, en ella se observa y analiza el comportamiento normal y de fallo del sistema, mediante el monitoreo de las señales de campo (continuas/discretas), que representan a cada elemento que interviene en el proceso. El diagnosticador realiza la supervisión on-line de los estados de cada uno de los subsistemas de la planta.

FIG. 3. PLATAFORMA DE SIMULACIÓN



Fuente: Autor del proyecto

FIG. 4. ÁRBOL DE JERARQUÍA



Fuente: Autor del proyecto

Descripción General. Para la construcción del panel principal Auto_DPN.VI, se debe emular las señales de campo que representan el comportamiento del proceso, para ello se emplea una estructura de programación enfocada hacia la creación, registro y generación de eventos, mediante una máquina de estados principal, dos secundarias para el control de eventos y la manipulación de alarmas del sistema. Para esta tarea se emplean los “User Events” , esto permite ejecutar eventos en paralelo, pero da prioridad única a un evento en particular. Para controlar la comunicación y flujo de datos en las máquinas de estado, se usa un semáforo dentro del bucle principal. Este semáforo recibe el nombre de “Busy” , representando un retardo en milisegundos en el evento anterior. Una vez el evento principal se ejecuta el siguiente evento es disparado, tomando el lugar de evento principal. Este comportamiento emula el funcionamiento de una RdP, que realiza la activación y/o disparo de eventos, que ocurre cuando ciertas condiciones son ciertas. Los pasos a seguir son:

Creación e Inicialización de Variables y Máquinas de Estado

En primer lugar se plantean las señales de campo de cada subsistema (representado como altos

y bajos), para monitorear las lecturas de los sensores, luego se crean las señales de control sobre los actuadores del sistema, que definen las condiciones de operación.

En una secuencia “Stacked Sequence”  se realiza en su estado “0” la creación e inicialización de las variables, que el sistema controla y manipula en la simulación, en el estado “1”, el cual contiene una estructura “While” , se desarrolla la máquina de estados principal y secuencial cuya programación es basada en condicionales “Case Structures” , comparadores, compuertas lógicas y la intervención del semáforo “Busy”. Aquí se generan y disparan los eventos. Adicionalmente a la máquina de estados principal, se implementan dos máquinas de estados adicionales para registro y librería de eventos. Una ejecuta acciones de control y la otra es dedicada a identificar las señales de alarma, presentes en el sistema de comunicación con el DSC. La máquina de estados principal y la auxiliar interactúan continuamente e intercambian información, según el flujo de datos demarcado por el semáforo, que provoca la evolución del algoritmo.

Eventos del Sistema Normal y de Fallo

En la programación se definen dos subconjuntos para la creación de eventos de usuario “Create User Event” , destinados a condiciones normales y a condiciones de simulación de fallos, para validar el algoritmo de diagnóstico en los subsistemas continuos y discreto. Para que un evento se cumpla, debe ser creado y registrado en un módulo dinámico “Register for Events” , y adicionalmente cumplir con las condicionales establecidos para su generación en “Generate User Events” . Los eventos son ejecutados en la segunda máquina de estados, elaborada mediante una “Event Structure” , donde se determina la acción que se debe tomar en la simulación.

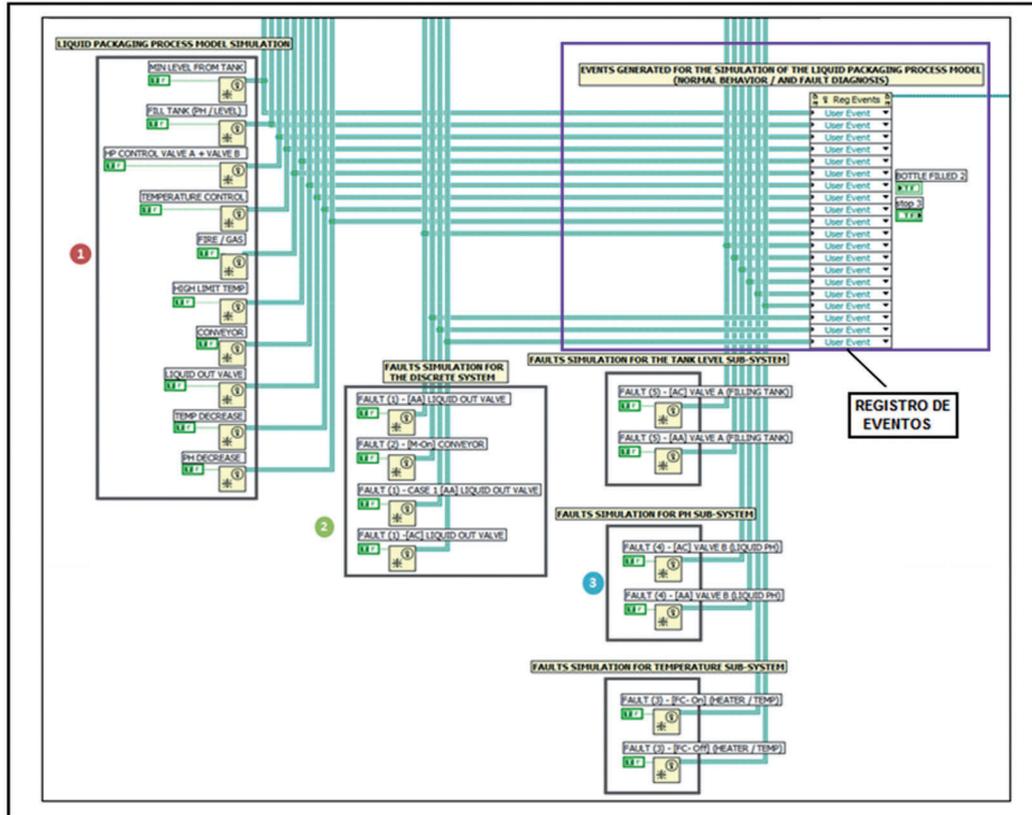
En la Fig. 5, se observa la distribución de la creación de eventos de usuario, se tiene un conjunto de diez eventos para la construcción de la simulación del proceso con su comportamiento normal, atendiendo a cada uno de los subsistemas donde se controla el incremento y decremento de los valores de Temperatura, Ph, Nivel y los efectos para la acción de la banda transportadora, la

detección de botellas y el llenado de las mismas; este grupo de eventos fue denominado como “*Liquid Packaging Process Simulation Model*” (Índice 1 de la Fig. 5).

Para la simulación de uno de los fallos en el mismo, se construyen dos grupos de eventos, cada uno representa un tipo de dinámica en particular. Se tiene un conjunto de cuatro eventos denominado “*Fault Simulation for Discrete System*”

(Índice 2 de la Fig. 5), quien determina los eventos de fallo para el subsistema discreto y un conjunto para la ejecución de fallos para los subsistemas continuos, representador por: “*Fault Simulation for the Tank Level Sub-System*”, “*Fault Simulation for Ph Sub-System*” y “*Fault Simulation for Temperature Sub-System*” con seis eventos de fallo. (Índice 3 de la Fig. 5).

FIG. 5. CREACIÓN Y REGISTRO DE EVENTOS



Fuente: Autor del proyecto

C. Diagnosticador y Comunicación con el Módulo DSC

En el diagnóstico On-line, el Auto_DPN.VI interactúa con diferentes SubVI's creados en DSC de LabVIEW, estos representan de manera gráfica el modelo de la RdPD de cada subsistema que se desea analizar; los subVI's creados son: Subsistemas continuos *PN_TempSystem.VI*, *PN_PhSystem.VI*, *PN_LeveISystem.VI*, y *DPN_Conveyor.VI* y para la dinámica discreta

DPN_Loalve.VI. Los SubVI's efectúan comunicación constante con las máquinas de estado, para realizar la tarea de: actualizar los valores en la red, identificar las transiciones disparadas, lugares actuales de la simulación de la RdP Diagnosticadora y activar en la red las ramas de fallo del diagnosticador, en el caso donde el operario efectuó la inducción de un fallo al sistema, por medio del disparo de las transiciones de fallo y el alcance de lugares de fallo. Toda esta labor interactiva muestra instantáneamente las inciden-

cias ocurridas en el proceso mediante alarmas que indican la detección y aislamiento del fallo. En la aplicación se observa el funcionamiento y evolución del diagnosticador, interactuando con los subsistemas que lo componen, y al mismo tiempo es posible lanzar de manera individual cada red, para una visualización específica de cada sistema.

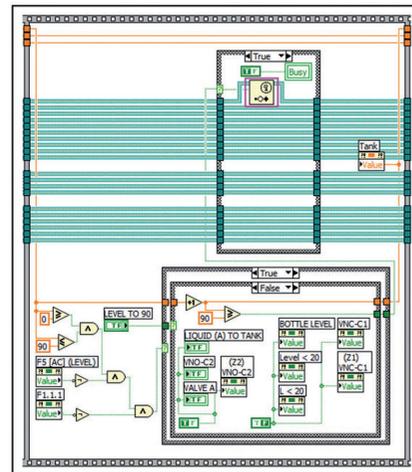
Los siguientes fallos son evaluados en el diagnosticador: En el subsistema discreto: La rama de fallo de atascamiento en abierto o atascamiento en cerrado de la válvula de salida de líquido, {F1 - [AA],[AC]}, la rama de fallo de encendido o apagado del motor de la banda transportadora, {F2 - [MOn],[MOff]}. En el subsistema continuo: Rama de fallo de encendido y apagado de la caldera para el control de Temperatura, {F3 - [FC-Off],[FC-On]}, rama de fallo de atascamiento en abierto o en cerrado de la válvula de suministro de líquido (B) para el control de Ph {F4 - [AA],[AC]}, y rama de fallo de atascamiento en abierto o cerrado de la válvula del paso del líquido (A) y el control de nivel en el tanque F5 - [AA],[AC]

D. Diagnosticador y Comunicación con el Módulo DSC

Para dar una mejor ilustración de la metodología, se presenta la creación de un evento del conjunto de "User Events". El evento "Min Level from Tank" es el evento encargado de realizar el llenado del tanque principal con el líquido (A), a través de abrir la válvula de nivel, hasta alcanzar el set-point (90Lts).

La acción es programada mediante un nodo de sumatoria del valor de nivel, el cual se actualiza en cada instante y está sometido a ciertos condicionales, que mediante una combinación de compuertas lógicas, cambia de estado "true" a "false" de la estructura, que genera un evento, ver Fig. 6. Cuando el evento es disparado y el semáforo "Busy" controla el tráfico en ejecución, el flujo de programación pasa a la segunda máquina de estados, donde la acción del evento es ejecutada y desarrolla "Event Structure", así mismo, recurre al bloque de registro y verifica qué evento del registro es ejecutado, de esta forma se ejecuta y da lugar a la secuencia programada que corresponde al llenado del tanque. (Comportamiento normal del subsistema de nivel).

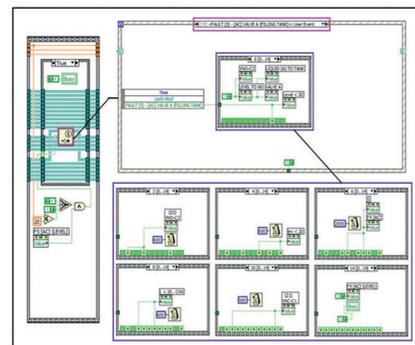
FIG. 6. MIN LEVEL FROM TANK (USER EVENT)



Fuente: Autor del proyecto

Al continuar con el ejemplo, se presenta la creación y ejecución de uno de los eventos de fallo, que se evalúan en el subsistema de nivel "Fault [AC], Valve A/ Filling Tank", el fallo representa el atascamiento en cerrado de la válvula del líquido (A), que controla el nivel tanque. Los rangos de nivel máximo y mínimo del tanque son: 90L y 20L. El evento de fallo de atascamiento en cerrado, ocurre cuando el nivel de líquido se encuentra por debajo de 20L y la acción de control para la activación de la válvula no se ejecuta; el evento es también relacionado con el llenado de las botellas, esto se debe a que el nivel del tanque decrece hasta su valor mínimo, a medida que el tanque se vacía con el llenar de las botellas. La acción es realizada en el nodo de decremento que disminuye el valor del nivel por debajo del valor mínimo permisible y genera el fallo. Cuando el fallo ocurre, la secuencia registrada en la máquina de estados auxiliar, dispara la transición de fallo del diagnosticador, ver Fig. 7.

FIG. 7. FAULT [AC], VALVE A/ FILLING TANK

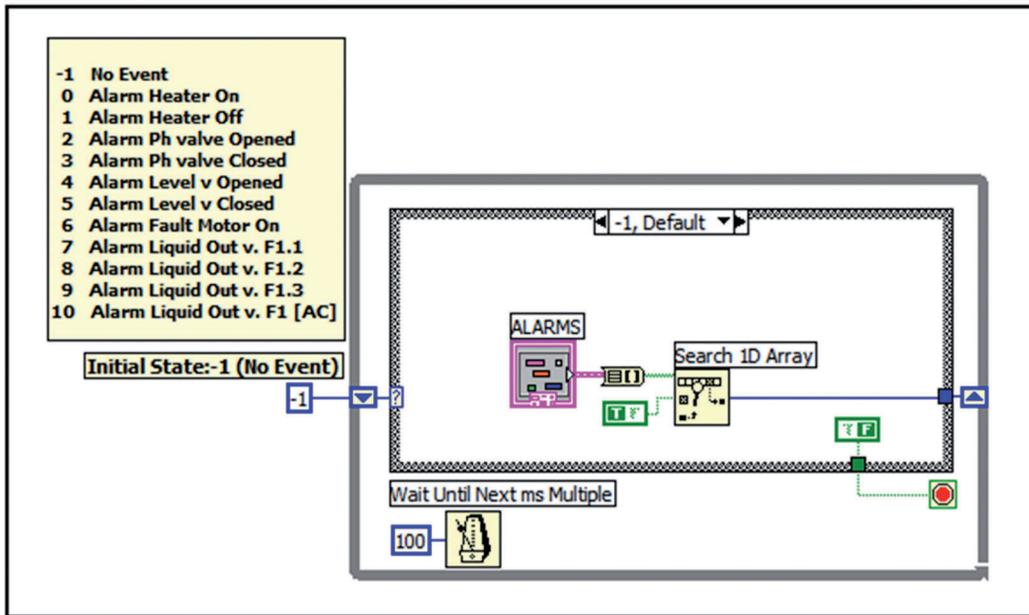


Fuente: Autor del proyecto

Cuando el evento de fallo alcanza el estado "enable", la interfaz comunica con el correspondiente VI creado en el módulo DSC, la red evoluciona y entra en la tercera máquina de control de alarmas, ver Fig. 8, que identifica en un "Clúster" el síntoma producido y después de

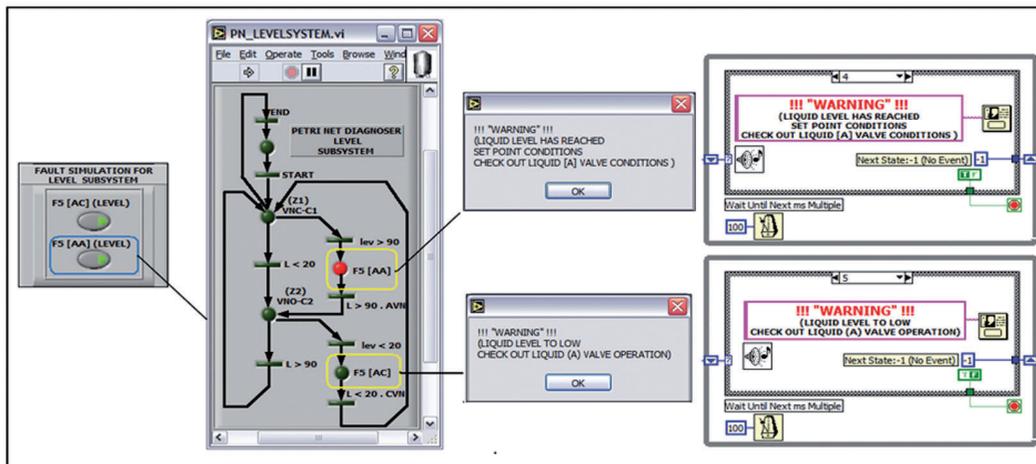
una conversión en "Array" envía un mensaje de advertencia en pantalla. El cuadro de alarmas para eventos de fallo en el Subsistema de nivel para la rama F5, se presenta junto con la RdPD del subsistema, ver Fig. 9.

FIG. 8. MÁQUINA DE ALARMAS



Fuente: Autor del proyecto

FIG. 9. RDPD SUBSISTEMA DE NIVEL (ESTADO DE FALLO)



Fuente: Autor del proyecto

V. CONCLUSIÓN

En este trabajo se ha abordado el tema de simulación de Redes de Petri mediante LabVIEW. Se presenta el proceso de simulación de un modelo y

diagnosticador, aplicado a un proceso de envasado de líquidos. Se puede visualizar en el ambiente gráfico, el comportamiento normal y de fallo del proceso. Esta aplicación en LabVIEW es un resultado

innovador, debido a que se puede construir las Redes de Petri, mediante una estructura programática que emula las señales de campo del proceso y la utilización de disparo de eventos en LabVIEW, Este proceso es metodológico y paso a paso, se indica como interactuar simulación con una herramienta tipo SCADA, que da lugar a un nuevo uso de las plataformas en automatización con LabVIEW. El buen abordaje de la herramienta en LabVIEW, sirvió para desarrollar la aplicación mostrada en [6].

so Latinoamericano de Control Automático ACCA 2010, Santiago – Chile.

- [9] National Instruments, 'User manual LabVIEW 9.0', Austin- Texas. USA, Disponible en: <http://digital.ni.com/worldwide/latam.nsf/main?readform>.
- [10] National Instruments, 'User manual DSC 9.0', Austin- Texas. USA, Disponible en: <http://digital.ni.com/worldwide/latam.nsf/main?readform>

REFERENCIAS

- [1] R. David and H. Alla, "Discrete, Continuous and Hybrid Petri Nets" . Springer, Verlag Berlin Heidelberg.. 2005.
- [2] T. Murata, J. Yim, H. Yin and O. Wolfson, "Petri-Net Model and Minimum Cycle Time for Updating Moving Objects Database," International Journal of Computer Systems Science & Engineering, Vol.21, No.3, pp.211 - 217, May 2006
- [3] Silva M., "Unforced continuous Petri Nets and positive systems" IEEE Control Syst Soc" ANIPLA, Springer-Verlag Berlin - 2003.
- [4] Trigos M., Barrientos, A., Del Cerro, J. and López, H., "Modelling and Fault Diagnosis by means of Petri Nets. Unmanned Aerial Vehicle Application", Book Petri Nets, Theory and Applications, pp 353-378 Editorial IN-TECH Austria. Disponible en "<http://www.sciyo.com/books/show/title/petri-nets-applications>", 2009.
- [5] Trigos M., García, E., "Faults Diagnosis and Modeling of the Liquid Packaging Process. A Research Based on Petri Nets" Proceeding from the 10th International Conference of Robotics & Automation IEEE, Hanoi – Vietnam, 2008.
- [6] Trigos, M., García, E., Modelado y Diagnóstico de Fallos por Medio de Redes de Petri de un Sistema de Envasado de Líquidos, Congreso Latinoamericano de Control Automático 2008 – Mérida Venezuela. 2008
- [7] Trigos, M., Garcia, E., "Petri Nets Used for Intermittent Faults Diagnosis and Modeling of Discrete Event Systems", SAFEPROCESS 09, Barcelona - Spain
- [8] Trigos, M., Méndez, D., García, J., Barrientos, A., Del Cerro, J., "Diseño e Implementación de la Instrumentación para la Lectura de Variables Críticas en Aeronaves no Tripuladas (UAS)", Congre-