Control Supervisor de Sistemas de Eventos Discretos con Retroalimentación de Eventos y Estados

J.F. Sánchez-Blanco
CINVESTAV-IPN Unidad Guadalajara
Prol. López Mateos Sur 590, 45090 Guadalajara Jal. México
e-mail: jsanchez@gdl.cinvestav.mx

Resumen

En este artículo se presenta un esquema general de control supervisor para sistemas de eventos discretos (SED), en el cual se combinan simultáneamente los esquemas de retroalimentación de eventos y de retroalimentación de estados. Bajo este esquema se define el concepto de *gen-observabilidad*, el cual se basa en la fusión de los conceptos de *evento-observabilidad* y *estado-observabilidad* para lenguajes formales; demostrando así, que un *supervisor* existe si y sólo si, el comportamiento requerido (especificación) es *controlable* y *gen-observable*. La síntesis de un supervisor en el esquema aquí propuesto, se realiza a través del *producto síncrono* entre los SED construidos a partir de la *congruencias*, que se derivan de las proyecciones de observación de eventos y estados del sistema a controlar (*planta*).

Índice de términos: ***control supervisor de SED, estado-observabilidad, evento-observabilidad gen-observabilidad.***

1. Motivación

La teoría de control supervisor para sistemas de eventos discretos (SED), formulada inicialmente por Ramadge y Wonham [8][10], ha sido ampliamente utilizada para diseñar, analizar y sintetizar sistemas lógicos de control. Esta teoría propone representar el comportamiento del sistema a controlar (*planta*), con un lenguaje formal que debe ser generado por un SED. Dicho lenguaje debe ser restringido a un comportamiento requerido (*especificación*), mediante la habilitación/deshabilitación de la posible ocurrencia de eventos controlables en la planta, a través de un agente externo denominado *supervisor*.

Según la información que el supervisor obtiene de la planta, dos han sido los esquemas propuestos para el control supervisor de SED: 1) el *esquema de retroalimentación de eventos* (ECSRV) [8][6][3] y 2) el *esquema de retroalimentación de estados* (ECSRS) [7][5][11]. El primero de estos esquemas propone la construcción de un supervisor que "observe" la ocurrencia de los eventos que se susciten en la planta, siendo necesario el uso de sensores que detecten la ocurrencia de dichos eventos, para realizar la implementación de este tipo de esquema. A diferencia del esquema anterior, en el ECSRS el supervisor debe conocer el estado que guarda la planta o en su defecto, determinar en que conjunto de posibles estados puede encontarse ésta. En este esquema son utilizados sensores especializados en medir el estado específico de una parte de la planta, de tal modo que la acción colectiva de todos los sensores, le permita al supervisor emitir una ley de control apropiada.

Cuando el supervisor no puede detectar ni todos los eventos o bien, ni todos los estados de la planta, diversos trabajos han sido publicados en el contexto de control supervisor con retroalimentación de eventos o con retroalimentación de estados. Por ejemplo, en [6] Lin y Wonham abordan el problema de la observabilidad en SED a través de clasificar los eventos de la planta en *eventos observables* y *eventos no-observables*. De esto, una proyección canónica sobre el lenguaje generado por la planta, se utiliza para determinar si el supervisor obtiene suficiente información para llevar a cabo una ley de control. Un resultado similar se presenta en [3]. El problema de control supervisor con retroalimentación de estados ha sido descrito a través de representar la especificación como un conjunto de predicados [9][5]. Posteriormente, en [11] se aborda el problema de construir un supervisor que "observe" estados y en [12] se presenta un resultado para obtener la partición óptima del espacio de estados. El ECSRS puede ser estudiado desde el enfoque de retroalimentación de eventos, tal y como como se propone en [1].

A diferencia de los trabajos anteriores, se presenta el *esquema combinado de control supervisor* (ECCS). Este esquema, propone la construcción de un supervisor que pueda detectar simultáneamente eventos y estados de la planta, facilitando en la práctica, el análisis y la implantación de sistemas lógicos de control. Para establecer las condiciones necesaria y suficientes que determinen la existencia del supervisor en el ECCS, se define el concepto de *gen-*

observabilidad. Este concepto se basa en crear una relación de equivalencia del lenguaje de la planta, que conjunte las clases de equivalencia que se inducen de la proyección canónica de los eventos y de la visualización parcial de los estados y que a su vez, deben ser incluidas en la relación de los activos del lenguaje. Con esto, demostramos que un supervisor existe si y sólo si, la especificación es *controlable* y *gen-observable*.

Este documento está organizado como sigue. En la siguiente sección, se presenta la definición de los *autómatas finitos* (AF), formalismo utilizado en este artículo para modelar SED; además, el concepto de *congruencia* para AF es formalmente descrito. La sección 3 incluye los conceptos generales sobre control supervisor de SED, presentando las características generales de los ECSRV y ECSRS. En la sección 4, se describe el ECCS y se define el concepto de *gen*-observabilidad, el cual sirve como base para determinar las condiciones necesarias y suficientes para la existencia del supervisor en este esquema. Finalmente, se presentan las conclusiones a este trabajo.

2. Automatas y congruencias dinámicas

En la actualidad, una gran diversidad de formalismos han sido definidos para modelar el comportamiento de los SED [2]. Los *autómatas finitos* (AF) han sido el modelo más utilizado en el estudio del control supervisor. Formalmente, un AF es una 5-tupla $G = (Q, q_0, \Sigma, \delta, Q_m)$ donde: Q es un conjunto finito de *estados*; $q_0 \in Q$ es el llamado *estado inicial*; Σ es un conjunto de símbolos denominado *alfabeto*, tal que $\sigma \in \Sigma$ representa un *evento*; $\delta : Q \times \Sigma \to Q$ es la *función de transición* que determina el cambio de estado a través de la ocurrencia de un evento; y $Q_m \subseteq Q$ es el denominado conjunto de *estados marcados*. Una *cadena* generada por un AF, es una secuencia de eventos $\tau = \sigma_r \cdots \sigma_s$ tal que $\delta(q_0, \sigma_r) = q, \ldots, \delta(q', \sigma_s) = q''$ con $\sigma_r, \ldots, \sigma_s \in \Sigma$ y $q_0, q, q', \ldots, q'' \in Q$. Denotamos por Σ^* al conjunto de todas las posibles cadenas de tamaño finito que pueden realizarse con el alfabeto Σ, entonces la función de transición puede extenderse de manera natural a $\delta : Q \times \Sigma^* \to Q$, tal que para $w \in \Sigma^*$ y $\sigma \in \Sigma$ se tiene que $\delta(q_0, w\sigma) = \delta(\delta(q_0, w), \sigma)$ y $\delta(q_0, \varepsilon) = q_0$, siendo $\varepsilon \in \Sigma^*$ la *cadena vacía*.

Nota: En este trabajo, suponemos que todos los estados de un AF son alcanzables a partir de q_0; es decir, $\exists w \in \Sigma^*$ tal que $\delta(q_0, w) = q, \ \forall q \in Q$.

Un *lenguaje* K sobre Σ, es todo $K \subseteq \Sigma^*$. Dos son los lenguajes que determinan el comportamiento de un AF $G = (Q, q_0, \Sigma, \delta, Q_m)$: 1) el *lenguaje cerrado* definido como $L(G) = \{w \in \Sigma^* \mid \delta(q_0, w) = q, \ q \in Q\}$ y 2) el *lenguaje marcado* definido como $L_m(G) = \{w \in L(G) \mid \delta(q_0, w) = q, \ q \in Q_m\}$. De estas definiciones se sigue obviamente que $L_m(G) \subseteq L(G)$. Para una secuencia $w \in L(G)$, $w' \in L(G)$ es un *prefijo* de w, si existe una cadena $w'' \in \Sigma^*$ tal que $w = w'w''$. Sea $\overline{K}$ el conjunto de todos los prefijos de un lenguaje K, se dice que K es *prefijo cerrado* si $K = \overline{K}$. Sea L un lenguaje, entonces un lenguaje K es *L-cerrado* si $\overline{K} \cap L = K$. Un lenguaje $L(G)$ se dice ser *no-bloquente* si $L(G) = \overline{L_m(G)}$.

Sean G_1 y G_2 dos AF, con $L(G_1)$ y $L(G_2)$ como sus respectivos comportamientos. Podemos construir un AF G tal que $L(G) = L(G_1) \cap L(G_2)$ utilizando el *producto síncrono* entre AF, operación que se define a continuación.

Definición 1 *Sean* $G_1 = \{Q_1, q_0^1, \Sigma_1, \delta_1, Q_{m_2}\}$ *y* $G_2 = \{Q_2, q_0^2, \Sigma_2, \delta_2, Q_{m_2}\}$ *dos AF. El producto síncrono entre* G_1 *y* G_2*, denotado por* $G_1||G_2$*, se define como el automata* $G = G_1||G_2 = \left(Q_1 \times Q_2, (q_0^1, q_0^2), \Sigma_1 \cap \Sigma_2, \delta, Q_{m_1} \times Q_{m_2}\right)$ *donde:*

$$\delta\left((q_1, q_2), \sigma\right) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) & \text{si } \sigma \in \Sigma_1 \cap \Sigma_2 \\ \text{no definida} & \text{otro caso} \end{cases}$$

2.1. Congruencias

Sea X un conjunto. Una *relación de equivalencia* π sobre X, es una relación binaria $\pi \subseteq X \times X$ que cumple con las siguientes propiedades: 1) *reflexividad* i.e. $\forall x \in X \quad (x, x) \in \pi$, 2) *simetría* i.e. $\forall x, x' \in X \quad (x, x') \in \pi \Rightarrow (x', x) \in \pi$, y 3) *transitividad* i.e. $\forall x, x', x'' \in X \quad (x, x') \in \pi \wedge (x', x'') \in \pi \Rightarrow (x, x'') \in \pi$. Denotamos $x \equiv x' mod\, \pi$ si $x \in X$ es equivalente a $x' \in X$ a través de la relación π. Una *clase de equivalencia* $[x]$, es el conjunto de todos los elementos $x' \in X$ tal que $x \equiv x' mod\, \pi$. El conjunto de todas las clases de equivalencia inducidas por π es denotado por X/π. Una relación de equivalencia π sobre un conjunto X, induce una partición en él; es decir, $\forall [x_i], [x_j] \in X/\pi \quad [x_i] \neq [x_j] \Rightarrow [x_i] \cap [x_j] = \emptyset$ y $\bigcup_i [x_i] = X$.

Sean π y π' dos relaciones de equivalencia. Denotamos por $\pi \preceq \pi'$ si dado que $x \equiv x' mod\, \pi$ entonces $x \equiv x' mod\, \pi'$ con $x, x' \in X$. Además, la operación de *conjunción* entre π y π', denotada por $\pi \wedge \pi'$, es una relación de equivalencia definida como $x \equiv x' mod(\pi \wedge \pi')$, ssi $x \equiv x' mod\, \pi$ y $x \equiv x' mod\, \pi'$. Claramente, $(\pi \wedge \pi') \preceq \pi$ y $(\pi \wedge \pi') \preceq \pi'$.

Una *proyección canónica* es una función $f_\pi : X \to X/\pi$ que relaciona cada $x \in X$ con su clase de equivalencia $[x] \in X/\pi$, i.e. $x \mapsto [x]$. Sea $f : X \to X'$ una función, el *kernel* de f, denotado por $\ker f$, es una relación de equivalencia definida como $x \equiv x' mod(\ker f)$ ssi $f(x) = f(x') \ \forall x, x' \in X$.

Sea $G = (Q, q_0, \Sigma, \delta, Q_m)$ un AF representando a un SED. Una *congruencia* para la dinámica de G, representada por la función de transición δ, es una relación de equivalencia π sobre Q tal que la función $\delta_\pi : Q/\pi \times \Sigma \to Q/\pi$, definida como $\delta_\pi([q],\sigma) = [q']$ si $\delta(q,\sigma) = q'$, respeta consistentemente la evolución del comportamiento de G; lo cual se expresa formalmente en la siguiente definición.

Definición 2 *Sea $\delta : Q \times \Sigma \to Q$. Sea π una relación de equivalencia con una proyección canónica $f_\pi : Q \to Q/\pi$. Decimos que π es una congruencia para δ si existe una relación $\delta_\pi : Q/\pi \times \Sigma \to Q/\pi$ tal que:*

$$\delta_\pi \circ g_\pi = f_\pi \circ \delta, \tag{1}$$

siendo $g_\pi : Q \times \Sigma \to Q/\pi \times \Sigma$ con $g_\pi(q,\sigma) = (f_\pi(q),\sigma)$; es decir, el siguiente diagrama conmuta:

$$\begin{tikzcd}
Q \times \Sigma \arrow[r, "\delta"] \arrow[d, "g_\pi"'] & Q \arrow[d, "f_\pi"] \\
Q/\pi \times \Sigma \arrow[r, "\delta_\pi"] & Q/\pi
\end{tikzcd}$$

Sea π una congruencia para la función $\delta : Q \times \Sigma \to Q$. Entonces, es posible construir un autómata G_π cuya dinámica sea consistente con π, tal y como se enuncia en la siguiente definición.

Definición 3 *Sea $G = (Q, q_0, \Sigma, \delta, Q_m)$ un AF. Sea π una congruencia para δ. Entonces π induce el autómata $G_\pi = (Q/\pi, [q_0], \delta_\pi, \Sigma, Q_m/\pi)$ tal que, el conjunto de clases de equivalencia Q/π representa al conjunto de estados, $f_\pi(q_0) = [q_0]$ es la clase de equivalencia inicial, $\delta_\pi : Q/\pi \times \Sigma \to Q/\pi$ definida como $\delta_\pi([q],\sigma) = [q']$ si $\delta(q,\sigma) = q'$ y $Q_m/\pi \subseteq Q/\pi$ está dado por $[q] \in Q_m/\pi$ si $q \in Q_m$.*

3. Control supervisor

Análogamente a la teoría de control moderno, un sistema de control supervisor para SED se compone de tres elementos: 1) el sistema que será controlado, comúnmente llamado *planta*, 2) el comportamiento requerido para la planta, al cual se le denomina *especificación*, y 3) un agente externo, denominado *supervisor*, el cual manipula la posible ocurrencia de eventos de la planta, para que ésta se comporte conforme a la especificación establecida.

En el contexto del control supervisor de SED, la planta se representa mediante un formalismo, que en este trabajo es el AF $G = (Q, q_0, \Sigma, \delta, Q_m)$, cuyo comportamiento queda determinado por los lenguajes $L(G)$ y $L_m(G)$; mientras que la especificación, generalmente está dada por un lenguaje $K \subseteq L_m(G) \subseteq L(G)$. Para llevar a cabo control sobre la planta, el alfabeto de eventos Σ se particiona en dos subconjuntos, el conjunto de los *eventos controlables* Σ_c, cuya ocurrencia puede ser habilitada/deshabilitada por el supervisor, y el conjunto de los *eventos incontrolables* $\Sigma_u = \Sigma - \Sigma_c$, los cuales no poseen restricción en su ocurrencia; es decir, siempre están habilitados. Entonces, un lenguaje $L(G)$ se dice ser controlable con respecto a $L(G)$ si $\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K}$ [8].

En general, un supervisor es un mapeo que relaciona la información obtenida de la evolución de la planta (estados o eventos), con un subconjunto de eventos $\gamma \subseteq \Sigma$, de manera tal que γ incluya todos los eventos que pueden ocurrir en la planta, por lo que $\Sigma_u \subseteq \gamma$. La acción conjunta entre un supervisor S y la planta G, es el sistema a lazo cerrado denotado por S/G, cuyo comportamiento está dado por los lenguajes $L(S/G)$ (lenguaje cerrado) y por $L_m(S/G) = L(S/G) \cap L_m(G)$ (lenguaje marcado).

Asociamos a cada cadena generada por la planta $w \in L(G)$ dos subconjuntos de eventos como sigue. Sea $K \subseteq \Sigma^*$ un lenguaje arbitrario, para $w \in \Sigma^*$ definimos el conjunto de sus *eventos activos* como:

$$A_K(w) = \begin{cases} \{\sigma \in \Sigma \,|\, w\sigma \in \overline{K}\} & w \in \overline{K} \\ \emptyset & \text{otro caso} \end{cases}$$

y el conjunto de los *eventos inactivos* como:

$$I_K(w) = \begin{cases} \{\sigma \in \Sigma \,|\, w\sigma \in L(G) - \overline{K}\} & w \in \overline{K} \\ \emptyset & \text{otro caso} \end{cases}$$

Esto es, $A_K(w)$ consiste únicamente de aquellos eventos cuya ocurrencia posterior a un prefijo de K, mantiene al comportamiento dentro de $\overline{K}$, mientras que los eventos en $I_K(k)$ podrían ocurrir en G, pero "desvían" el comportamiento fuera de $\overline{K}$.

Podemos entonces definir una relación binaria entre cadenas, denominada act_K, tal que $(w, w') \in act_K$ si: 1) $A_K(w) \cap I_K(w') = A_K(w') \cap I_K(w) = \emptyset$ y 2) $w \in K \cap L_m(G)$ y $w' \in K \cap L_m(G) \Rightarrow (w \in K \Leftrightarrow w' \in K)$. Equivalentemente, se tiene la siguiente definición para la relación act_K.

Definición 4 *Sean $w, w' \in \Sigma^*$. Entonces $(w, w') \in act_K$ ssi se cumple lo siguiente:*

1. *$\forall \sigma \in \Sigma \; w\sigma \in \overline{K}, w' \in \overline{K}, w'\sigma \in L(G) \Rightarrow w'\sigma \in \overline{K}$,*
2. *$w \in K \cap L_m(G), w' \in \overline{K} \cap L_m(G) \Rightarrow w' \in K$, y*
3. *las condiciones anteriores se cumplen intercambiando w y w'.*

Note que $(w, w') \in act_K$, si tanto w como w' no pertenecen a $\overline{K}$, debido a que si $w \notin \overline{K}$ entonces $A_K(w) = I_K(w) = \emptyset$. En otro caso $(w, w') \in act_K$, significa que los prefijos w y w' de K tienen idéntica continuación a la ocurrencia del siguiente evento, con respecto a su inclusión en $\overline{K}$ y si cada una de éstas, está en $L_m(G)$ y una ya pertenece a K, entonces la otra también tiene que pertenecer.

Nota: Si K es prefijo cerrado o $L_m(G)$-cerrado, la condición (2) cumple automáticamente. Además, puede fácilmente observarse que act_K no es una relación de equivalencia, ya que no cumple con la propiedad de transitividad; pero es posible compararla con una relación de equivalencia π sobre un lenguaje K, de manera tal que $\pi \preceq act_K$ ssi $w \equiv w' mod\, \pi \Rightarrow (w, w') \in act_K$.

3.1. Control supervisor con retroalimentación de eventos

Se dice que un sistema de control supervisor está en un ECSRV, si el supervisor únicamente puede detectar la ocurrencia de eventos que se susciten en la planta. Debido a que en general el supervisor no puede detectar la ocurrencia de todos los eventos de la planta, el conjunto de eventos Σ se particiona en dos subconjuntos, el conjunto de *eventos observables* Σ_o, cuya ocurrencia es detectada por el supervisor, y el conjunto de *eventos no-observables* $\Sigma_d = \Sigma - \Sigma_o$, que son transparentes al supervisor. Cuando un supervisor puede detectar todos los eventos que ocurren en la planta ($\Sigma_o = \Sigma$), entonces el sistema a lazo cerrado S/G se comporta conforme a una especificación dada $K \subseteq L(G)$, i.e. $L(S/G) = K$, ssi K es prefijo cerrado y es controlable [8]. Más aún, existe un supervisor no bloqueante S tal que $L_m(S/G) = K$, ssi K es prefijo cerrado, controlable y $L_m(G)$-cerrado [8].

En el caso de que no todos los eventos de la planta sean observables ($\Sigma_o \subset \Sigma$), entonces definimos la proyección de eventos de la planta como un mapeo $P_E : \Sigma^* \to \Sigma_o^*$, tal que $P_E(\varepsilon) = \varepsilon$, $P_E(\sigma) = \sigma$ si $\sigma \in \Sigma_o$ ó $P_E(\sigma) = \varepsilon$ si $\sigma \notin \Sigma_o$, y $P_E(w\sigma) = P_E(w)P_E(\sigma)$ con $w \in \Sigma^*$ y $\sigma \in \Sigma$. La siguiente definición formaliza el concepto de un supervisor bajo el ECSRV.

Definición 5 *Sea $L(G) \subseteq \Sigma^*$ el comportamiento de una planta. Sea $P_E : \Sigma^* \to \Sigma_o^*$ una proyección de eventos. Un supervisor de observación parcial de eventos, es un mapa $\Gamma : P_E(L(G)) \to 2^\Sigma$, tal que para $w \in L(G)$ se tiene que $\Gamma(P_E(w)) = \gamma$ con $\Sigma_u \subseteq \gamma \subseteq \Sigma$.*

Para determinar las condiciones de existencia del supervisor, debemos decidir si la información que obtiene el supervisor es suficiente para tomar una decisión sobre la acción que debe realizar la planta, para poder cumplir con la especificación requerida. De esto, se define a continuación el concepto de *evento-observabilidad* o simplemente P_E-observabilidad [6].

Definición 6 *Sea $K \subseteq L(G)$ un lenguaje no vacío. Sea $P_E : \Sigma^* \to \Sigma_o^*$ una proyección. Decimos que K es evento-observable si:*

$$\ker P_E \preceq act_L.$$

Esta definición establece que la relación de equivalencia $\ker P_E$ es más "fina" que la relación act_K, por lo que P_E preserva la información mínima requerida para decidir consistentemente si el comportamiento del sistema continua perteneciendo a $\overline{K}$ después de la ocurrencia hipotética de un evento σ o en su defecto, decidir si pertenece a K cuando pertenece a $\overline{K} \cap L_m(G)$ después de un evento conocido. Es decir, si dos cadenas tienen la misma proyección, entonces una regla desición que aplique a una, puede ser usada para la otra. La siguiente proposición, establecida en [6], determina la existencia de un supervisor, cuando detecta parcialmente la ocurrencia de eventos de la planta.

Proposición 1 *Sea $K \subseteq L_m(G)$ un lenguaje no vacío. Existe un supervisor no-bloqueante S tal que $L_m(S/G) = K$, ssi K es controlable y evento-observable.*

■

Ejemplo 1 *Sea la planta representada por el AF que se muestra en la figura (1). Sea $\Sigma = \{\alpha, \beta, \lambda, \mu\}$, $\Sigma_c = \{\alpha, \beta\}$ y $\Sigma_o = \{\alpha, \beta\}$. De esto, $L(G) = \{\varepsilon, \lambda, \mu, \lambda\alpha, \mu\alpha, \mu\beta\}$ y $L_m(G) = \{\varepsilon, \lambda, \lambda\alpha, \mu\alpha, \mu\beta\}$. Sea la especificación $K = \{\varepsilon, \lambda, \mu, \lambda\alpha, \mu\beta\}$, por lo que claramente K es prefijo cerrado, controlable y $L_m(G)$-cerrado. Entonces, la siguiente*

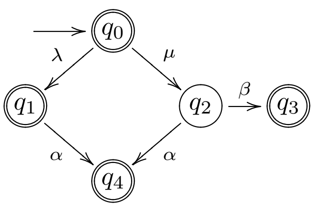

Figura 1: AF que representa la planta del ejemplo (1).

tabla muestra los conjuntos $A_K(w)$ y $I_K(w)$ para todas las cadenas $w \in L(G)$, además de las proyecciones correspondientes sobre los eventos observables.

w	$A_K(w)$	$I_k(w)$	$P_E(w)$	w	$A_K(w)$	$I_K(w)$	$P_E(w)$
ε	$\{\lambda, \mu\}$	$\emptyset$	ε	$\lambda\alpha$	$\emptyset$	$\emptyset$	α
λ	$\{\alpha\}$	$\emptyset$	ε	$\mu\alpha$	$\emptyset$	$\emptyset$	α
μ	$\{\beta\}$	$\{\alpha\}$	ε	$\mu\beta$	$\emptyset$	$\emptyset$	β

De esto tenemos que, $(\varepsilon, \lambda), (\varepsilon, \mu) \in act_K$ pero $(\lambda, \mu) \notin act_K$. Como $P_E(\lambda) = P_E(\mu)$, entonces $\lambda \equiv \mu \bmod \ker P_E \not\Rightarrow (\lambda, \mu) \in act_K$; por lo que K no *es evento-observable.*

3.2. Control supervisor con retroalimentación de estados

A diferencia del ECSRV, en el ECSRS el supervisor obtiene información de la planta a través de conocer el estado que guarda la misma [9][7]. Tradicionalmente, en este esquema la especificación no está dada por un lenguaje, sino por un conjunto de proposiciones lógicas asignadas a cada estado de la planta, con la finalidad de alcanzar un posible conjunto de estados finales [5]. Sin embargo, en este trabajo seguimos suponiendo que la especificación está dada por un lenguaje.

Una primera solución al problema de control supervisor con retroalimentación de eventos, consiste en suponer que el supervisor puede conocer todos los estados de la planta, por lo que el supervisor es formalmente un mapeo $\Gamma : Q \to 2^\Sigma$, que siempre habilita la ocurrencia de los eventos incontrolables de la planta. Si definimos para un AF G

la relación de equivalencia $eq(G)$ sobre $L(G)$, denominada *relación de equirespuesta* [4], tal que para $w, w' \in L(G)$ se tiene que $w \equiv w' mod\,(eq(G))$ ssi $\delta(w, q_0) = \delta(w', q_0)$, entonces una variante del concepto de evento-observabilidad, denominada *estado-observabilidad* es definida como sigue:

Definición 7 *Sea G un AF. Sea $K \subseteq L(G)$ un lenguaje no vacío. Decimos que K es (total) estado-observable si:*

$$eq(G) \preceq act_K$$

Usando la definición anterior, obtenemos la siguiente proposición que determina las condiciones para la existencia de un supervisor para el ECSRS, cuando éste puede detectar todos los estados de la planta.

Proposición 2 *Sea $K \subseteq L_m(G)$ un lenguaje no vacío. Existe un supervisor S tal que $L_m(S/G) = K$ ssi K es controlable y (total) estado-observable.* ■

La proposición anterior supone que la planta tiene los sensores suficientes para detectar cada estado en particular de ella misma. En la práctica, no es posible afirmar esto siempre. Un sensor de estado puede conceptualizarse desde dos enfoques: 1) cuando el sensor detecta exactamente un estado en particular de la planta, y 2) cuando el sensor detecta que la planta se puede encontrar en un posible conjunto de estados. Con base en esto, definimos $\widetilde{Q} = Q \cup \{q_\varepsilon\}$ como el conjunto de estados de la planta Q más un estado especial q_ε denominado *estado no detectable*, el cual representa a todos aquellos estados que no pueden detectarse mediante algún tipo de sensor. Si representamos por $H : Q \to 2^{\widetilde{Q}}$ la acción de los sensores de la planta, entonces dada la suposición de que cada sensor pertenece a un único subconjunto de $\widetilde{Q}$, podemos definir la relación de equivalencia π_S sobre $\widetilde{Q}$, tal que dos estados son equivalentes si son detectados por el mismo sensor ó no pueden ser detectados. Definimos entonces formalmente un supervisor bajo el esquema de retroalimentación de estados.

Definición 8 *Sea $G = (Q, q_0, \Sigma, \delta, Q_m)$ una planta. Formalmente, un supervisor de retroalimentación de estados es un mapeo $\Gamma : H(Q) \to 2^\Sigma$, tal que $\forall q \in Q \quad \Gamma(H(q)) = \gamma$ con $\Sigma_u \subseteq \gamma$.*

Sea Q/π_S la partición del conjunto de estados de la planta generada por la acción de los sensores de estados. Decimos que un supervisor $\Gamma : H(Q) \to 2^\Sigma$ es Q/π_S*-compatible* si para cualquier $[q] \in Q/\pi_S$, se cumple que $\forall q, q'$ tal que $[q] = [q'], \Gamma(H(q)) = \Gamma(H(q'))$. Esta condición requiere que el mismo patrón sea asignado a los estados que se encuentran en la misma clase de equivalencia. Si un supervisor es Q/π_S-compatible, entonces es suficiente para los sensores distinguir entre los elementos de Q/π_S y no de Q; por lo tanto, cada partición es considerada un espacio de observación. De esto, definimos una proyección $P_S : \Sigma^* \to \Sigma^*$ tal que para un AF G, $\forall w, w' \in L(G) \quad P_S(w) = P_S(w')$ ssi $H(\delta(q_0, w)) = H(\delta(q_0, w'))$. Definimos entonces el concepto general de estado-observabilidad.

Definición 9 *Sea G un AF. Sea $K \subseteq L(G)$ un lenguaje no vacío. Decimos que G es estado-observable si:*

$$\ker P_S \preceq act_K$$

Nota: En el esquema de control supervisor con retroalimentación de estados, la representación de un AF es muy importante. Si para un lenguaje $L(G)$ construimos un AF con mayor cantidad de estados, obtendremos mayor información de la planta y por el contrario, si construimos un AF con la mínima cantidad de estados, la información se verá disminuida.

Proposición 3 *Sea $K \subseteq L_m(G)$ un lenguaje no vacío. Existe un supervisor no-bloqueante S tal que $L_m(S/G) = K$, ssi K es controlable y estado-observable.* ■

Ejemplo 2 *Consideremos la planta del ejemplo (1). Supongamos que tenemos sensores de estado tal que $H(q_0) = q_0, H(q_1) = H(q_3) = q_\varepsilon$ y $H(q_2) = H(q_4) = \{q_2, q_4\}$. Para este caso tenemos que $Q/\pi_S = \{\{q_0\}, \{q_1, q_3\}, \{q_2, q_4\}\}$. En este caso tenemos que $(\lambda, \mu) \in \ker P_S$, ya que $H(\delta(q_0, \lambda)) = H(q_0, \mu)$. Como $(\lambda, \mu) \notin act_K$, entonces K* no *es estado-observable.*

4. Esquema combinado de control supervisor

En la práctica, la implementación de sistemas de control para SED puede utilizar sensores que detecten la ocurrencia de eventos en la planta y/o sensores que detecten el estado que guarda la misma. En la figura (2) se muestra el esquema de control supervisor propuesto (ECCS). En este esquema puede apreciarse que el supervisor se compone de dos sistemas, uno que utiliza la información obtenida de la proyección de eventos (P_E) y la otra, de la proyección de estados (P_S).

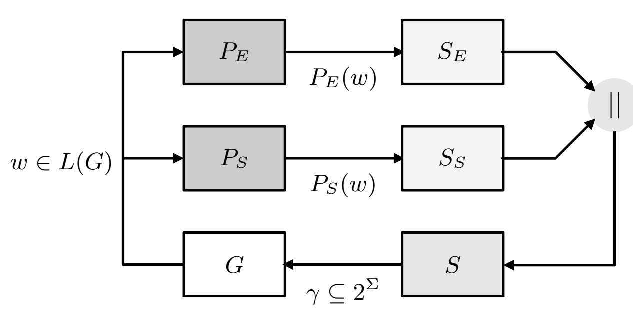

Figura 2: Esquema Combinado de Control Supervisor.

Tanto S_E como S_S se representan por los AF G_E y G_S respectivamente, teniendo que el supervisor S puede entenderse como el producto síncrono de ambos. En base al esquema propuesto se formula el siguiente problema que define al *control supervisor de esquema combinado.*

Problema 1 *Sea G un AF que representa una planta y cuyo comportamiento está definido por el lenguaje $L(G)$. Sea $K \subseteq L(G)$ una especificación dada. Sea $P_E : \Sigma^* \to \Sigma_o^*$ y $P_S : \Sigma^* \to \Sigma^*$ las proyecciones de eventos y estados, respectivamente. Construir un supervisor no-bloqueante S de esquema combinado tal que $L_m(S/G) = K$.*

Para resolver el problema anterior, tomamos ventaja de la operación de conjunción entre relaciones de equivalencia y proponemos la siguiente definición de *gen*-observabilidad, la cual combina simultáneamente la detección de eventos y estados.

Definición 10 *Un lenguaje $K \subseteq L(G)$ es gen-observable, si* $(\ker P_E \wedge \ker P_S) \preceq act_K$.

Utilizando la definición anterior, se obtiene el siguiente resultado.

Teorema 1 *Sea $K \subseteq L_m(G)$ un lenguaje no vacío. Existe un supervisor no-bloqueante S, tal que $L_m(S/G) = K$ ssi K es controlable y gen-observable.*

Proof: Similar a la demostración del lema 2.1 de [9], pero se sustituye $\ker P$ por $\ker P \wedge \ker P_S$.

∎

Ejemplo 3 *Retomando los ejemplos anteriores, se tiene que* $\ker P_E = \{\{\varepsilon, \lambda, \mu\}, \{\lambda\alpha, \mu\alpha\}, \{\mu\beta\}\}$ *y* $\ker P_S = \{\{\varepsilon\}, \{\lambda, \mu\beta\}, \{\mu, \lambda\alpha, \mu\alpha\}\}$*, por lo que* $(\ker P_E \wedge \ker P_S) = \{\{\varepsilon\}, \{\lambda\}, \{\mu\}, \{\mu\beta\}, \{\lambda\alpha, \mu\alpha\}\}$*. Entonces tenemos que $\forall w, w' \in L(G)$, $w \equiv mod\,(\ker P_E \wedge \ker P_S) \Rightarrow (w, w') \in act_K$, por lo que K es gen-observable.*

4.1. Síntesis del supervisor

Sea $G = (Q, q_0, \Sigma, \delta, Q_m)$ un AF representando a una planta. Sea $K \subseteq L_m(G)$ una especificación dada. Para llevar a cabo la construcción del supervisor S, construyamos los siguientes autómatas: $S_E = (Q_E, q_0, \Sigma, \delta_E, Q_m)$ y $S_S = (Q_S, q_0, \Sigma, \delta_S, Q_m)$. El primer AF se construye a partir de reducir el AF que genera K, quitando los eventos no-observables. El segundo de estos AF, se induce a partir de la relación de equivalencia $\gamma_E = \ker P_E$.

Construyamos ahora el autómata $S = (Q, q_0, \Sigma, \delta, Q_m)$, tal que $L(S) = L(S_E) \cap L(S_S)$. Entonces, proponemos construir S como el producto síncrono entre S_E y S_S.

Proposición 4 *Sea $K \subseteq L_m(G)$ un lenguaje no vacío, controlable y G-observable. Sean P_E y P_S las proyecciones de estados y eventos respectivamente. Sean S_E y S_S AF inducidos por P_E y P_S respectivamente. Entonces, $S = S_E || S_S$.*

∎

Conclusiones

Un esquema general de control supervisor, en donde se combinan simultáneamente el ECSRE y el ECSRS, ha sido presentado. Se define el concepto de *gen*-observabilidad, el cual se deriva a partir de los conceptos ya establecidos de evento-controlabilidad y estado-observabilidad. Un supervisor bajo este esquema existe, si y sólo si la especificación, expresada como un lenguaje formal K, es controlable y *gen*-observable; realizando la síntesis de dicho supervisor, por medio del producto síncrono entre los AF que se obtienen de las congruencias inducidas a partir de las relaciones de equivalencia obtenidas de las proyecciones de eventos y estados.

Referencias

[1] C. Cao, F. Lin y Z.H. Lin, "Why Event Observation: Observability Revisited", *Discrete Event Dynamic Systems: Theory and Applications*, no. 7, Kluwer Academic Publishers, Boston 1997, pp. 127-149

[2] C.G. Cassandras y S. Lafortune, "Introduction to Discrete Event Systems", *Kluwer Academic Publisher*, 1999

[3] R. Cieslak, C. Desclaux, A.S. Fawaz y P. Varaiya, "Supervisory Control of Discrete-Event Processes with Partial Observations", *IEEE Trans. on Autom. Contr.*, vol. 33, no. 3, Marzo 1988, pp. 249-260

[4] M.A. Harrison, "Introduction to Switching and Automata Theory", *McGraw Hill*, New York, 1965

[5] R. Kumar, V.K. Garg y S.I. Marcus, "Predicates and Predicate Transformers for Supervisory Control of Discrete-Event Dynamical Systems", *IEEE Trans. on Autom. Contr.*, vol. 38, Febrero 1993, pp. 232-247

[6] F. Lin y W.M. Wonham, "On Observability of Discrete Event Systems", *Inf. Sciences*, vol. 44, no. 4, 1988, pp. 173-198

[7] C.M. Özveram y A.S. Willsky, "Observability of Discrete-Event Dynamical Systems", *IEEE Trans. on Autom. Contr.*, vol. 35, no. 7, Julio 1990, pp. 797-806

[8] P.J. Ramadge y W.M. Wonham, "Supervisory Control of a Class of Discrete Event Processes", *SIAM J. on Cont. and Optim.*, vol. 25, no. 1, Enero 1987, pp. 206-230

[9] P.J. Ramadge y W.M. Wonham, "Modular Feedback Logic for Discrete Event Systems", *SIAM J. on Contr. and Optim.*, vol. 25, no. 5, Septiembre 1987, pp. 1202-1218

[10] P.J.G. Ramadge y W.M. Wonham, "The Control of Discrete Event Systems", *Proc. of the IEEE*, vol. 77, no. 1, Enero 1989, pp. 81-89

[11] S. Takai, T. Ushio y S. Kodama, "Static-State Feedback control of Discrete-Event Systems Under Partial Observation", *IEEE Trans. on Autom. Contr.*, vol. 40, Noviembre 1995, pp. 1950-1954

[12] S. Takai, "Optimal State-Space Partition for Control of Discrete-Event Systems with Static Specifications", *IEEE Trans. on Autom. Contr.*, vol. 43, no. 7, Julio 1998, pp. 1013-1016