ECI 2009
Modelado y Simulación de Sistemas Dinámicos
Trabajo Final

Ernesto Kofman – kofman@fceia.unr.edu.ar

7 de agosto de 2009

El objetivo de este trabajo es el de modelar y simular un sistema *híbrido* de cierta complejidad, aplicando las metodologías y herramientas vistas a lo largo del curso.

Comenzaremos con una descripción global del sistema que queremos modelar (y de los objetivos por los cuales queremos realizar el modelo) y luego iremos desglosando el mismo en distintos subsistemas de los que obtendremos modelos de manera más sencillas.

Descripción del Sistema

Un ascensor es impulsado por un *motor de corriente continua* comandado por un *sistema de control* muy elemental. Este sistema recibe pedidos del tipo "ir al piso j" y eventos provenientes de sensores que indican "el ascensor pasa por tal lugar" o bien "ascensor completamente detenido". En base a estos eventos, el sistema de control activa al motor en un sentido o en otro de forma de cumplir con los pedidos que recibe, o activa un freno tipo traba (sólo puede activarse con el ascensor detenido) a los fines de que este permanezca en la posición alcanzada.

La Figura 1 muestra un esquema del sistema completo, dividido en los distintos subsistemas que iremos modelando y simulando.

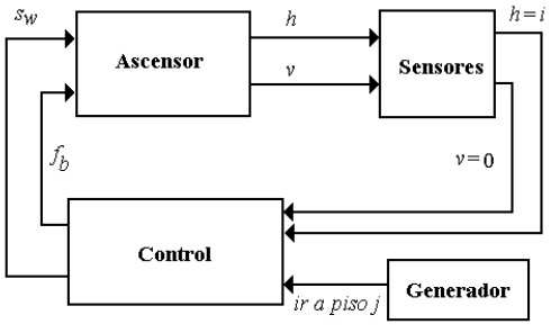

Figura 1: Sistema de control de un ascensor

El objetivo que nos proponemos para realizar el modelo es el de diseñar el sistema de control del ascensor y verificar el correcto funcionamiento del mismo.

Subsistema Electro–Mecánico (Motor – Ascensor)

La Figura 2 muestra un esquema del sistema físico compuesto por el motor y el ascensor.

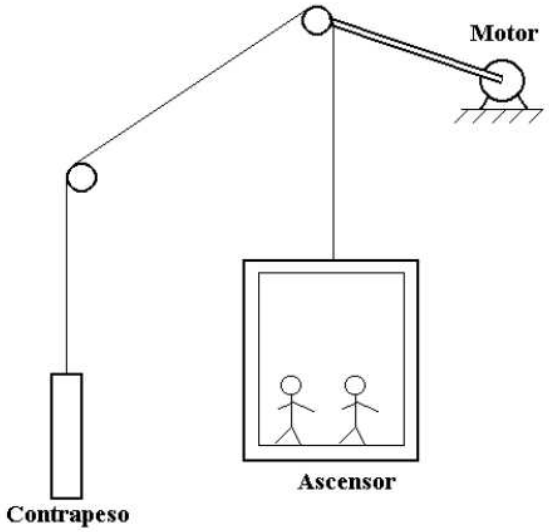

Figura 2: Esquema físico del ascensor

El siguiente sistema de ecuaciones de estado puede modelar la dinámica del subsistema:

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \frac{1}{m_a + m}(-b \cdot x_2 + \frac{k_m}{r} x_3 - m \cdot g) \cdot f_b \\
\dot{x}_3 &= \frac{1}{L_a}(-\frac{k_m}{r} x_2 - R_a \cdot x_3 + x_4) \\
\dot{x}_4 &= \frac{1}{\tau}(-x_4 + U_m \cdot S_w)
\end{aligned} \tag{1}$$

Las variables x_1 y x_2 representan la altura y velocidad del ascensor, respectivamente; x_3 y x_4 son la corriente y tensión en la armadura del motor. S_w representa la posición de la llave que conecta el motor (una entrada para este subsistema), pudiendo tomar los valores 1, -1 y 0, en tanto que f_b representa el estado del freno (otra entrada), siendo 1 cuando está desactivado y 0 cuando el sistema está frenado.

Los parámetros, en tanto, son los siguientes:

- $m_a = 1000$ kg : masa conjunta del ascensor y contrapeso.
- $g = 9.8$ m/seg2 : aceleración de la gravedad.
- $m = 100$ Kg: masa de la carga en el ascensor (utilizaremos luego distintos valores).
- $L_a = 0.003$ Hy : inductancia de armadura del motor.
- $b = 1.1$ N $\cdot$ m $\cdot$ seg : coeficiente de rozamiento.
- $R_a = 0.05\ \Omega$: resistencia de armadura del motor.
- $km = 6.4$ V $\cdot$ seg/rad : constante del motor.
- $\tau = 1/3$ seg. : constante de tiempo de los actuadores.
- $r = 0.02$ m : radio equivalente de la polea.
- $U_m = 460$ V : tensión continua de alimentación (utilizaremos luego distintos valores).

Las unidades están todas en S.I. por lo que no es necesario hacer ninguna conversión.

Notar que este subsistema, si bien es esencialmente continuo, tiene ciertas características híbridas debido a la naturaleza de las señales de entrada S_w y f_b.

Subsistema de sensores

Los sensores los representaremos como un sistema que recibe dos señales continuas (altura y velocidad), y que en función de las mismas produce dos tipos de señales de eventos discretos. Consideraremos que cada vez que la altura alcance un valor entero (es decir, cada un metro) un sensor detectará el paso del ascensor y provocará un evento indicando el número del piso. Por otro lado, cada vez que la velocidad alcance el valor cero, un sensor provocará un evento indicando esta situación.

Este subsistema es claramente híbrido, ya que recibe señales continuas y provoca señales de eventos discretos.

Subsistema generador

El generador será el encargado de enviar la secuencia de pedidos al ascensor. Estos pedidos estarán caracterizados por números enteros que indicarán el piso al que hay que dirigirse.

Este subsistema es puramente de eventos discretos.

Subsistema de control

El control del ascensor es el encargado de recibir los pedidos del generador y las señales de los sensores y en función de estos producir las señales S_w y f_b para cambiar la alimentación del motor y para liberar o activar el freno.

Consideraremos que el control sólo atiende un pedido por vez ignorando todos los pedidos que puedan llegar entre tanto. Al recibir el pedido, el control

debe liberar el freno (enviando $f_b = 0$) y cambiar la señal S_w al valor 1 o -1 según deba enviar el ascensor hacia arriba o hacia abajo.

Supondremos que cada piso tiene tres metros de altura, por lo que habrá dos sensores intermedios entre piso y piso. Cuando el ascensor llegue al último sensor previo al del piso al que se dirige, el control deberá esperar cierto tiempo (que luego determinaremos) y luego invertir la señal S_w para forzar el frenado. Una vez recibida la señal del sensor que detecta que la velocidad es nula, se debe aplicar el freno enviando la señal $f_b = 0$, cortando también la alimentación del motor haciendo $S_w = 0$. A partir de ese momento, el control queda a la espera de otro pedido.

El subsistema de control es también puramente de eventos discretos.

1. Desarrollo del Trabajo

El trabajo consistirá en el modelado y simulación del sistema, partiendo de la modelización y simulación de cada uno de los subsistemas.

1.1. Modelado y Simulación del Subsistema Electromecánico

Comenzaremos el trabajo simulando en PowerDEVS el modelo del subsistema electromecánico, dado por el sistema de ecuaciones de estado (1).

Para esto, deberemos transformar el sistema de ecuaciones en un Diagrama de Bloques, y luego contruir el mismo en PowerDEVS utilizando los elementos de su librería. Si bien puede utilizarse otro método, sugerimos usar QSS3 con un quantum del orden de 1×10^{-3}. Sugerimos también utilizar de manera simbólica los parámetros m y U_m, definiéndolos en Scilab, para poder cambiarlos más fácilmente en las simulaciones que haremos al final del trabajo.

Para analizar y corroborar el correcto funcionamiento de este subsistema, podemos simularlo conectado a fuentes (de la librería de PowerDEVS) que generen escalones o pulsos para las variables S_w y f_b.

Además, al modelo obtenido lo "encapsularemos" como un único modelo acoplado con dos entradas (S_w y f_b) y dos salidas (x_1 y x_2).

Una vez realizado esto, cambiaremos el quantum de los integradores y analizaremos como varía el costo computacional en función del mismo y como mejora o se degrada la solución.

1.2. Modelado y Simulación de los Sensores

Aquí podremos utilizar también componentes de la librería de bloques híbridos de PowerDEVS, con los cuales crearemos un modelo encapsulado con dos entradas (altura y velocidad del motor) y provocará dos salidas diferentes indicando cuando la altura alcanza un valor entero y cuando la velocidad se hace cero.

Para analizar y corroborar el funcionamiento de los sensores, lo simularemos conectado al modelo del motor, al cual excitaremos de la misma forma que en el punto anterior.

1.3. Modelado y Simulación del Generador

En este caso tendremos que crear un nuevo modelo atómico DEVS, encargado de generar la secuencia de pisos a los que deberá ir el ascensor.

En este modelo incorporaremos un comportamiento estocástico, de forma tal que tanto el piso, como el intervalo entre pedidos sucesivos sea elegido al azar. Para esto, tendremos permitido utilizar en las funciones de transición la función RND[1] para calcular el tiempo de avance y/o el valor de algún estado que determine el piso del próximo pedido.

Supondremos además que el ascensor tiene 10 pisos, y que el máximo tiempo entre pedidos sucesivos es de 60 segundos.

Si bien los eventos de salida de este modelo serán números enteros, utilizaremos la clase *double[10]* para su representación (es decir, un arreglo de 10 elementos tipo double, donde y[0] contiene el valor del evento y los restantes componentes deben ser nulos) ya que de esta forma tendremos compatibilidad con las librerías de QSS de PowerDEVS.

Una vez construido este modelo en PowerDEVS, corroboraremos su correcto funcionamiento mediante simulación, observando los eventos que produce a la salida.

1.4. Diseño, Modelado y Simulación del Control

Para implementar el control, primero deberemos ajustar el tiempo transcurrido entre la recepción de la última señal del sensor y el instante al que hay que invertir la alimentación del motor (mediante la inversión de la señal S_w).

Una manera de ajustar este tiempo es experimentando directamente sobre el modelo del motor. Utilizando las fuentes de la librería de PowerDEVS podemos comenzar impulsando el motor hacia arriba y luego de cierto tiempo, invertir la alimentación hasta que comience a bajar. Analizando luego la gráfica de la altura en función del tiempo puede llegar a determinarse cuanto tiempo debe transcurrir desde que el motor está 1 metro por debajo del objetivo hasta que se deba invertir la alimentación.

Por supuesto, podemos utilizar prueba y error también hasta encontrar el retardo correcto.

Consideraremos que el control funciona bien si el error cometido al frenar no supera los 3 cm.

Una vez determinado el retardo correcto, procederemos a obtener el modelo DEVS del Control. Dicho modelo tendrá tres puertos de entrada (uno para los pedidos del generador, uno para las señales de los sensores de altura y otro para el sensor de velocidad nula). Consideraremos que los valores de los eventos son todos punteros a la clase double[10], y el valor está en el primer elemento del arreglo.

Además el modelo tendrá 2 puertos de salida, uno para la señal S_w y otro para f_b. Para guardar compatibilidad con las librerías, utilizaremos también punteros a un arreglo de diez doubles, con el valor en el primer componente (los restantes componentes deben ser nulos).

[1] DEVS en principio no permite definir sistemas estocásticos, pero recientemente se desarrolló un formalismo que extiende la noción de DEVS a los modelos estocásticos y garantiza, entre otras cosas, la correctitud de utilizar funciones tipo RND dentro de las funciones de transición [1].

Para corroborar el funcionamiento de este modelo deberemos simular el sistema completo.

1.5. Simulación y Análisis del Sistema Completo

Comenzaremos en primer lugar simulando el sistema completo y observando como evoluciona la altura pedida (o sea, el piso pedido multiplicado por 3) y la altura del ascensor (x_1), verificando que la diferencia al alcanzar un piso y frenar el ascensor no supere los 3 cm que nos propusimos.

Luego, corroborado esto, cambiaremos la masa de la carga (m) y analizaremos que sucede con el error final, buscando determinar a partir de que valores este error supera la consigna de 3 cm.

Tras esto, cambiaremos el valor de tensión de alimentación U_m y repetiremos el análisis, concluyendo como las variaciones en la tensión de alimentación afectan el funcionamiento de nuestro sistema.

Finalmente, una limitación física del motor utilizado es que la corriente de armadura (x_3) no debería superar el valor de 300 Amp (a partir de este límite se puede destruir por calentamiento de los bobinados). Verificar que esto no ocurra (analizando también si esto impone un límite a la carga que se puede transportar).

2. Informe

El informe del trabajo deberá incluir la definición de los modelos realizados en cada punto y el análisis de los resultados de simulación correspondientes.

Además, se deberán adjuntar los archivos de PowerDEVS utilizados para las simulaciones (modelos .pdm y el código c++ (.cpp y .h) de los modelos atómicos desarrollados).

Referencias

[1] R. Castro, E. Kofman, and G. Wainer. A Formal Framework for Stochastic DEVS Modeling and Simulation. *Simulation: Transactions of the Society for Modeling and Simulation International*, 2009. In press.