

La presente Guía se ha tomado de la RTE con el fin de facilitar a los estudiantes la “escritura del artículo de su tesis de grado”. Requisito establecido en el Reglamento de Graduación de Pregrado de la ESPOL. Título 2. Del Certificado Único para Graduación. Art. 4, literal c) que concierne al CICYT.
Esta guía contiene las secciones y detalles de presentación de un artículo, normativa que el estudiante debe cumplir. **“Información Adicional”, leer los puntos 11 y 15.**

ACTUALIZACIÓN DE LA BIBLIOTECA DEL SISTEMA LABCON (LABORATORIO REMOTO DE CONTROL AUTOMATICO)

Ing. Cristhian Arroba Rivera ⁽¹⁾

Msc. Dennys Cortez Alvarez ⁽²⁾

Facultad de Ingeniería en Electricidad y Computación (FIEC)

Escuela Superior Politécnica del Litoral (ESPOL)

Campus Gustavo Galindo, Km 30.5 vía Perimetral

Apartado 09-01-5863. Guayaquil-Ecuador

carroba@fiec.espol.edu.ec ⁽¹⁾

dcortez@fiec.espol.edu.ec ⁽²⁾

Resumen

El siguiente trabajo tiene como propósito aumentar la cantidad de bloques que posee actualmente el sistema LabCon de modo que se disponga de una mayor cantidad de elementos para trabajar dentro de una simulación, consecuentemente fue necesario crear un documento tipo tutorial dentro del cual se describen los diferentes pasos a seguir para poder agregar un nuevo bloque (existen dos tipos de bloques: simulink y planta). Además, se modificó la interfase para mejorar el comportamiento con los diferentes bloques y se implementó un nuevo bloque “Fuzzy Logic Controller” para poder trabajar con simulaciones con un controlador difuso.

Para poder confirmar el correcto funcionamiento del bloque “Fuzzy Logic Controller” fue necesario crear un archivo de control con extensión “.fis” que posea toda la lógica difusa y luego crear una simulación en tiempo real usando la planta de tres tanques de agua representada dentro del sistema LabCon con el bloque tipo planta llamada “Planta3T”.

Palabras Claves: matlab, matlab server pages, fuzzy logic controller, bloque, simulación, planta.

Abstract

The puppose of this paper is to increase the number of blocks that actually are in the LabCon System so we can have a large amount of elements to work inside a real-time simulation, so to achieve this objective was require to create a tutorial document with the neccesary steps to add a new block (there are two kinds of block: simulink and plant). Also, we need to modified the interfase to improve the behavior with the different blocks and a new block was created “Fuzzy Logic Controller” to work with real-time simulations with a difuse controller.

To confirm the correct performance from the block “Fuzzy Logic Controller” was necessary to generate an archive with extension “.fis” with the fuzzy logic programation and then create a real-time simulation using the plant of the three water tanks represented in the LabCon System with the block type plant called “Planta3T”.

Keywords: matlab, matlab server pages, fuzzy logic controller, block, simulation, plant.

1. Introducción

Anteriormente para poder realizar alguna práctica con alguna planta dentro del Laboratorio de Control Automático, era necesario que el estudiante pueda usar un computador con acceso directo al Compact

Field Point (CFP), sin embargo permitir que un estudiante se encuentre en contacto directo con este dispositivo junto con la inexperiencia del estudiante y por tratarse de un experimento se desconfiguren los valores dentro del propio Compact Field Point poniendo en riesgo el experimento y en algunos casos

la propia planta. Este fue uno de las principales motivos para realizar la implementación del sistema LabCon de modo que ahora los estudiantes realizar las prácticas con las plantas sin tener la limitante del acceso al Compact Field Point y con la ventaja de poder acceder desde cualquier computador que tenga acceso al sistema LabCon. Así mismo es necesario un documento tipo tutorial especificando los pasos necesarios para la inclusión de nuevos bloques de tipo simulink que se requieran dentro del sistema LabCon.

Una vez que se tenga un proceso adecuadamente documentado que permita actualizar la librería del sistema LabCon, se podrán realizar una mayor cantidad de modelos y prácticas que incluyan nuevas plantas y la complejidad de los mismos aumentaría de forma significativa en beneficio de los estudiantes del Laboratorio de Control Automático.

El siguiente trabajo trata sobre la generación de un procedimiento correctamente documentado de modo que se puedan identificar los pasos a realizar para la inclusión de un nuevo bloque (ya sea categoría simulink o categoría planta) dentro de la librería del sistema LabCon.

La gran expectativa puesta sobre el sistema LabCon usado dentro del laboratorio de Control Automático de la Facultad de Ingeniería en Electricidad y Computación (FIEC), impulsan la necesidad de perfeccionar la librería de bloques, elemento clave donde se albergan todos y cada uno de los elementos (bloques categoría simulink y bloques categoría planta), de modo que el objetivo de este proyecto está dirigido a la actualización de la librería para que se extienda la vida útil del sistema con un proceso sencillo y claro que permita realizar dicha actualización teniendo como consecuencia que los estudiantes puedan incluir nuevas plantas y bloques categoría simulink que permitan realizar modelos más complejos. Adicionalmente, dentro de este trabajo se utiliza el término “**planta**” para hacer referencia a un sistema electrónico ya implementado con anterioridad (como un motor eléctrico o un tanque de presión neumática), que son comúnmente utilizados para realizar las prácticas dentro del Laboratorio de Control Automático. Dentro del trabajo correspondiente a la primera fase del sistema LabCon, realizado por Cristian Idrovo y Humberto Aguilar se usó el término “**maquinaria**”, que será reemplazado dentro de este trabajo por la palabra “**planta**”[2].

Para su información la introducción informa tres elementos muy importantes de la investigación: el **propósito**, la **importancia** y el **conocimiento actual** del tema. Describe el interés que tiene en el tema que se ha escogido, y la importancia del mismo en el contexto científico del momento.

2. Objetivo General

Aumentar la cantidad de bloques que posee el sistema LabCon durante su primera fase y registrar los pasos necesarios dentro de un documento tipo tutorial que sirva de ayuda para futuras mejoras del sistema LabCon.

3. Objetivos Específicos

Incrementar la cantidad de bloques de la librería del sistema LabCon.

Creación de los bloques “Display” y “Fuzzy Logic Controller”.

Modificación del comportamiento e interacción con el usuario para los diferentes bloques del sistema LabCon.

Generación de un manual para la implementación de nuevos bloques categoría planta.

4. Alcances y Limitaciones

Debido a que existen varios bloques tipo simulink existe una limitante sobre los bloques que se pueden agregar al sistema LabCon por la gran cantidad de diferentes parámetros que posee cada diferente bloque y al momento el sistema LabCon soporta un número limitado de parámetros para cada bloque.

Dentro del proyecto se pretende el aumento y creación de nuevos bloques especiales para el sistema LabCon. Sin embargo existen bloques que poseen un mayor grado de dificultad debido a la gran cantidad de parámetros que poseen y el sistema LabCon al momento tiene un número limitado de parámetros por bloque y de sus correspondientes tipos de datos. Con respecto al manejo de la simulación en tiempo real se manejan los eventos de inicio y fin cuando se trabaje con una práctica, sin embargo también es necesario implementar un módulo que permita controlar las sesiones para cada una de las diferentes prácticas que trabajen directamente con algún bloque tipo planta (como por ejemplo el bloque de tres tanques) de modo que el sistema LabCon pueda validar que un usuario pueda utilizar una práctica a la vez y justamente para esto también es necesario poder manejar diferentes horarios para cada usuario de modo que se pueda manejar un tiempo máximo para trabajar dentro del sistema LabCon.

Siempre que se ejecute una simulación, se realizará con un tiempo de muestreo de 0.01 segundos (valor predeterminado dentro del sistema LabCon). Al momento de trabajar con el bloque “Fuzzy Logic Controller”, el tiempo de muestreo debe ser compatible con el valor antes mencionado debido a que el valor del tiempo de muestreo dentro del sistema LabCon no puede ser modificado. Cualquier variación en el tiempo de muestreo afectará directamente al bloque “WebScope”, el cual

fue calibrado para poder trabajar con el tiempo de muestreo de 0.01 segundos.

5. Nuevas Características

1. Modificación del comportamiento del bloque “SimOut”. Anteriormente, sólo se permitía grabar los datos de una simulación si existía el bloque “WebScope” presente en la misma; ahora el bloque “SimOut” funciona de manera independiente, es decir, al momento se puede grabar la información sin importar si el bloque “WebScope” se encuentre presente en la simulación[2].

2. Se eliminó el botón Pause, de modo que ahora para controlar la simulación solo existirán dos botones (estados de la simulación) para la misma: Play y Stop.

3. Se agregó una nueva práctica (Bloque de 3 Tanques), además, se implementó su correspondiente bloque “StopPlanta3T” para esta práctica que se activa durante escenarios especiales a modo de contingencia.

4. Se creó un nuevo bloque “Display”, que permite visualizar hasta 5 señales diferentes mostrando sus valores numéricos de forma independiente.

5. Se pudo identificar y corregir el comportamiento de la simulación mientras se trabaja con alguna práctica en tiempo real (se controla la planta de la práctica de forma remota por medio del Sistema LabCon) de modo que se active el correspondiente bloque Stop para la práctica que se encuentre activa, en estos casos particulares:

Al presionar el botón Stop mientras se ejecuta la simulación que incluya una práctica activa.

Cuando el tiempo de ejecución (es un valor que puede ser modificado por el usuario) de la simulación ha llegado a su límite máximo.

En el momento de salir de presionar el botón “Logout” (salir de sesión), esto puede ocurrir incluso mientras se encuentra trabajando con la práctica.

Para todos los casos mencionados anteriormente, se procederá a activar el correspondiente bloque “Stop” dependiendo de la práctica que el usuario escogió para trabajar. Este bloque “Stop” es un archivo de extensión mdl que realiza un procedimiento interno que permite que la planta real detenga su funcionamiento y luego se apague de forma segura para evitar cualquier daño de la misma.

6. La simulación ahora puede ser ejecutada sin necesidad que se incluya un bloque “WebScope” o un bloque “Display”; anteriormente esto no era posible de modo que la simulación siempre necesitaba que se incluyera un bloque “WebScope” dentro de la misma.

7. Se agregaron las nuevas imágenes para los respectivos nuevos bloques que se crearon, existen algunos nuevos bloques que requieren más de una imagen como por ejemplo: el bloque “Math Function”, en donde dependiendo del valor dentro del

combobox que se elija la imagen se modificará acorde al valor escogido.

8. Anteriormente para los bloques: Discrete Transfer Fcn, Discrete Zero-Pole, Transfer Fcn y Zero Pole[1]; se necesitó cambiar la forma en que presentaba los datos, debido a que anteriormente solo presentaba el valor del numerador o denominador según sea el caso, sin tomar en cuenta que se debía presentar por medio de una fórmula; esto fue corregido para que se presente la fórmula completa con los valores y signos (+/-) que les corresponde.

9. Aumento en el número de bloques del archivo “librería.mdl”. En total tenemos un aumento de 25 nuevos bloques de simulink para la librería del sistema LabCon. A continuación el detalle de nuevos bloques simulink:

- Planta 3T (Bloque especial, corresponde a una nueva práctica).
- Planta de tanque de presión (Bloque especial, corresponde a una nueva práctica).
- Display (Bloque especial).
- Trigonometric Function.
- Subtract.
- Math Function.
- MinMax.
- Abs.
- Sine Wave Function.
- Sign.
- Complex to Real-Imag.
- Complex to Magnitude-Angle.
- Product.
- Divide.
- Product of Elements.
- Backlash.
- Relay
- Rate Limiter.
- Variable Transport Delay.
- Variable Time Delay.
- Transport Delay.
- Quantizer.
- Hit Crossing.
- Dead Zone.
- Fuzzy Logic Controller.

6. Modificación del bloque “SimOut”

Anteriormente al trabajar dentro de una simulación que incluye el bloque “SimOut” existía una dependencia con el bloque “WebScope” debido a que si se ejecutaba la simulación con un bloque “SimOut” pero sin incluir un bloque “WebScope” se mostraba un mensaje de alerta informando al usuario que debía agregar a la simulación un bloque “WebScope”. En vista de este inconveniente se procedió a modificar el archivo fuente “panel fla” que se encuentra dentro de la ruta C:\MATLABServerPages\webapps\ROOT\Panel,

escena MenuBar; específicamente el método agregarMenuBar().

7. Eliminación de la opción Pausa

Inicialmente, dentro de la primera versión del sistema LabCon, existían tres controles principales para el manejo de la simulación: iniciar (play), detener (stop) y pausa (pause). Obviamente al tratarse de una simulación en donde se involucra una planta real de modo que fue necesario tener solamente dos controles (iniciar y detener), debido a que la simulación tiene un inicio y un fin pero experimentalmente no se requiere y así mismo las plantas usadas dentro del sistema LabCon no poseen ninguna funcionalidad de pausa (poseen encendido y apagado) de modo que este control en particular se vuelve ineficaz al no poseer mayor uso dentro de las simulaciones en general especialmente dentro de las simulaciones que involucran una planta. Para llevar a cabo este cambio se debió modificar el archivo fuente “panel.fla” que se encuentra dentro de la ruta C:\MATLABServerPages\webapps\ROOT\Panel. Por último fue necesario modificar el método onRollOver del botón “boton_pause” para que aparezca el texto “Stop”.

8. Mejora en la ejecución de la simulación

Al momento de ejecutar una simulación correspondiente a una práctica que involucre una planta (para poder controlar la planta de forma remota en tiempo real por medio del sistema LabCon), estas mejoras están relacionadas directamente con el control detener (stop) mientras se está ejecutando la práctica dentro de los siguientes escenarios:

- Al presionar el botón Stop mientras se ejecuta la simulación que incluya una práctica activa.
- Cuando el tiempo de ejecución (es un valor que puede ser modificado por el usuario) de la simulación ha llegado a su límite máximo.
- En el momento de salir de presionar el botón “LogOut”, esto puede ocurrir incluso mientras se encuentra trabajando con la práctica.

9. Mejora en la presentación de las fórmulas dentro de la simulación

Anteriormente para los bloques: Discrete Transfer Fcn, Discrete Zero-Pole, Transfer fcn y Zero Pole; se necesitó cambiar la forma en que presentaban los datos, debido a que inicialmente solo se presentaba el valor del numerador o denominador según sea el caso, sin tomar en cuenta que se debía presentar por medio de una fórmula; esto fue corregido para que se

presente la fórmula completa con los valores y signos (+/-) que les corresponde.

A continuación se presentará las imágenes de los parámetros presentados por cada uno de los bloques mencionados:

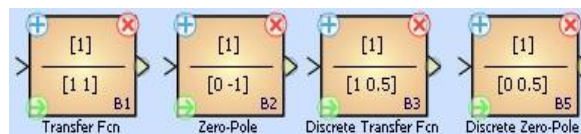


Figura 1. Bloques antes de realizar la modificación

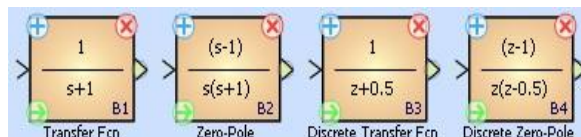


Figura 2. Bloques después de realizar la modificación

Para poder realizar este cambio fue necesario modificar el archivo dentro de la ruta C:\MATLABServerPages\webapps\ROOT\Panel\panel.fla, dentro de la clase “datos” se editó la función “modificarEP”.

10. Proceso para agregar un nuevo bloque categoría Simulink dentro de la librería del sistema LabCon

Inicialmente se debe modificar el archivo “grupos_bloques.xml” que se encuentra en la ruta C:\MATLABServerPages\webapps\ROOT\Panel\datos. Para esto es necesario tener en cuenta el siguiente código ejemplo:

```
<block name="Rate Limiter" npIN="1" npOUT="1" sWIM="0.6" desc="Limit rising and falling rates of signal." enable="1" >
<start name="stNameBlock" values="" />
<param name="TextInput" label="Rising slew rate:" varname="RisingSlewLimit" def="1"/>
<param name="TextInput" label="Falling slew rate:" varname="FallingSlewLimit" def="-1"/>
<param name="ComboBox" label="Sample type mode:" varname="SampleTimeMode" def="inherited,|_|,continuous" validation="inherited,|_|,continuous" image="" ports="" />
<param name="TextInput" label="Initial condition:" varname="InitialCondition" def="0" />
<param name="CheckBox" label="Treat as gain when linearizing" varname="LinearizeAsGain" def="on" validation="" image="" ports="" />
</block>
```

Figura 3. Código de ejemplo de un bloque dentro del archivo “grupos_bloques.xml”

Dentro del código de ejemplo podemos observar cuatro parámetros que procederemos a explicar a continuación:

- **Param name:** Se refiere al tipo de parámetro de entrada (TextInput es una caja de texto, ComboBox para elegir varias opciones, CheckBox para elegir si se selecciona o no).
- **Label:** Es el nombre propio que aparece en la ventana de parámetros.

- **Varname:** Es el nombre de la variable con que se identificará dentro del sistema es muy importante que este nombre sea exactamente el mismo al momento de ingresarlo dentro del archivo “grupos bloques.xml”.
- **Def:** Se refiere a los valores predefinidos que deberá tener un parámetro determinado. Para un parámetro de tipo ComboBox se debe ingresar cada valor separado por estos caracteres “,|,” como este ejemplo: Valor1,|,Valor2,|,Valor3... etc.
- **Validation:** Se refiere a los valores que requieren validar los parámetros tipo ComboBox y se usa exactamente el mismo formato para los valores.
Ejemplo: validation = “Valor1,|,Valor2,|,Valor3”.

Es importante tener claro los nombres de los parámetros que deben agregarse al nuevo bloque dentro del archivo “grupos_bloques.xml”, para esto es necesario obtener los nombres originales del bloque en Matlab creando un modelo sencillo, esto se realiza buscando el bloque dentro del menú de simulink y se escoge el bloque “Rate Limiter” y se agrega ese bloque a una simulación nueva de Matlab y se graba el archivo “ratelimiter.mdl”. Luego de esto abrimos el archivo “ratelimiter.mdl” mediante el programa “notepad” y buscamos la palabra clave “BlockParameterDefaults” como podemos observar en la imagen a continuación:

```
BlockParameterDefaults {
  Block {
    BlockType           RateLimiter
    1 RisingSlewLimit    "1"
    2 FallingSlewLimit  "-1"
    3 SampleTimeMode    "continuous"
    4 InitialCondition  "0"
    5 LinearizeAsGain   on
  }
}
```

Figura 4. Los nombres de las variables del bloque “Rate Limiter”.

Si realizamos una comparación entre los nombres de los parámetros del archivo “grupos_bloques.xml” y el modelo del bloque en Matlab en la siguiente imagen para el bloque “Rate Limiter”:

	BlockParameterDefaults	Variables en grupos_bloques.xml
Número 1	RisingSlewLimit	varname="RisingSlewLimit"
Número 2	FallingSlewLimit	varname="FallingSlewLimit"
Número 3	SampleTimeMode	varname="SampleTimeMode"
Número 4	InitialCondition	varname="InitialCondition"
Número 5	LinearizeAsGain	varname="LinearizeAsGain"

Figura 5. Parámetros del bloque en Matlab (a la izquierda). Los nombres del archivo “grupos_bloques.xml” (a la derecha).

Una vez que se pudo confirmar que los nombres de los parámetros están correctos, procedemos a agregar el archivo “.mdl” dentro del archivo “libreria.mdl” que se encuentra dentro de la ruta C:\MATLABServerPages\webapps\ROOT\Code\Libra

ry y se procede a editar el archivo “libreria.mdl” usando Matlab y agregamos el archivo “ratelimiter.mdl”.

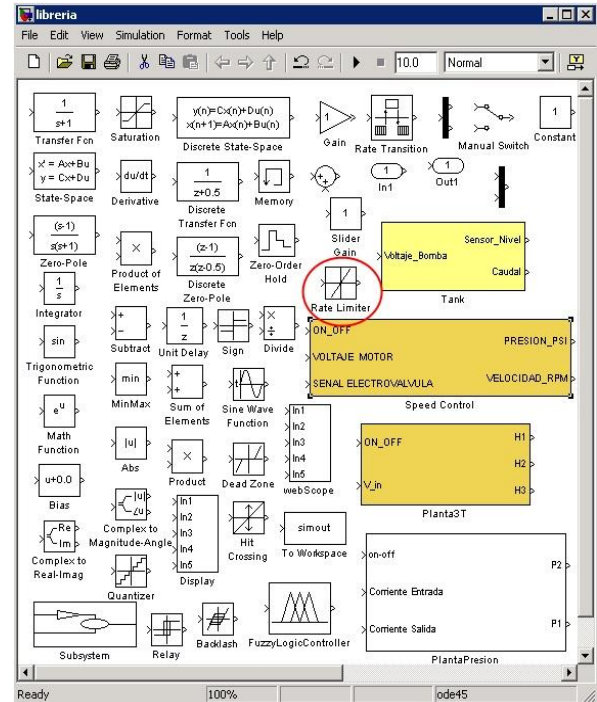


Figura 6. El Bloque “Rate Limiter” (resaltado en rojo) que fue agregado recientemente.

Finalmente solo falta agregar la imagen que se mostrará en el menú de LabCon, para esto nos dirigimos a la ruta C:\MATLABServerPages\webapps\ROOT\Panel\imagenes_panel\Simulink y escoger la carpeta del grupo donde se encuentra el nuevo bloque y agregar la imagen con extensión “.png” (único formato permitido y con 55x55 pixeles).

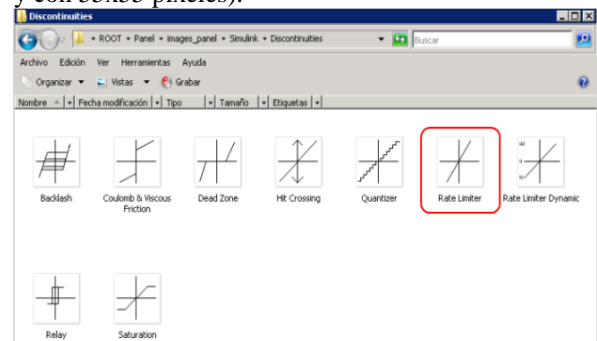


Figura 7. Archivo con extensión .png del bloque “Rate Limiter”.

Luego de esto se debe aparecer el nuevo bloque dentro del menú del sistema LabCon como podemos observar en la imagen a continuación:

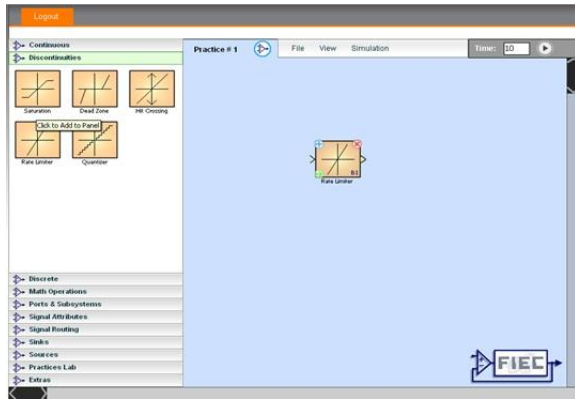


Figura 8. Bloque “Rate Limiter” dentro del menú del sistema LabCon.

11. Proceso para agregar un nuevo bloque categoría Planta dentro de la librería del sistema LabCon

Los pasos anteriormente descritos dentro del “Proceso para agregar un nuevo bloque categoría Simulink dentro de la librería del sistema LabCon” deben ser aplicados junto con otros que se explicarán a continuación:

Al momento de agregar el bloque dentro del archivo “libreria.mdl” que se encuentra dentro de la ruta C:\MATLABServerPages\webapps\ROOT\Code\Library. Sin embargo para un bloque de categoría “Planta” es necesario un segundo archivo que se utilizara para poder apagar la planta de modo que debe ser agregado en la misma ruta donde se encuentra el archivo “libreria.mdl”.

Nombre	Fecha de modifica...	Tipo	Tamaño
jdelpozo_p3_accel_rtwt	20/03/2013 14:27	Carpeta de archivos	
slprj	20/03/2013 14:27	Carpeta de archivos	
Display.m	02/05/2011 11:27	Archivo M	5 KB
index	05/04/2009 9:29	Opera Web Docu...	1 KB
jdelpozo_p3_acc_mexw32	10/05/2011 14:19	Archivo MEXW32	26 KB
libreria.mdl	26/07/2011 13:43	Archivo MDL	96 KB
stopMotor.mdl	04/03/2010 16:04	Archivo MDL	35 KB
stopPlanta3T.mdl	10/05/2011 14:28	Archivo MDL	37 KB
stopTanq.mdl	18/04/2011 15:09	Archivo MDL	28 KB
webScope.m	02/05/2011 11:23	Archivo M	5 KB

Figura 9. Se pueden ver el archivo “StopMotor.mdl” que permite apagar la planta.

Ahora para poder invocar el archivo que permite apagar la planta es necesario modificar los archivos “ProcesarSesionPanel.jsp” y “GenerarModelo.jsp” los cuales se encuentran dentro de la ruta C:\MATLABServerPages\webapps\ROOT\Panel\Funciones.

El archivo “ProcesarSesionPanel.jsp” es utilizado al momento de ingresar a una práctica siendo usado para enviar información (número de práctica y nombre del usuario) hacia la interfase en flash. Al momento de salir de la práctica como precaución debe ejecutarse el bloque correspondiente que sea encargado de apagar la planta. Para esto cuando se agrega una nuevo bloque de categoría planta es necesario modificar el código dentro del archivo antes

mencionado. Nos dirigimos a la opción 11 (línea de código 108) y agregamos la porción de código como la que se muestra en la imagen:

```
if(numeroPractica.equals("numero correspondiente a la practica que se desea agregar"))
{
}
<matlab:Command cmd="<%= "open_system('Library/stopNombredelBloque.mdl')" %>"/>
<matlab:Command cmd="<%= "set_param('stopNombredelBloque', 'SimulationCommand','Start')" %>"/>
<%
}
```

Figura 10. Código de ejemplo con los nombres del bloque Stop del nuevo bloque tipo planta.

Como vemos dentro del código se están ejecutando comandos propios de matlab, entre ellos tenemos:

- **Open System:** que permite abrir un archivo de extensión “mdl” y como parámetro recibe la ruta en donde se encuentra el modelo.
- **Set param:** permite ejecutar un comando que permite establecer un valor para un bloque simulink específico. Ejemplo: set param(‘stopPlanta3T’,SimulationCommand,’ Start’) para iniciar el bloque stopPlanta3T.

Luego de esto debemos configurar la relación entre el nuevo bloque de la planta con la práctica esto se hace dentro de la línea de código 210, en la opción 20 (case 20) dentro de la siguiente porción de código:

```
case 20: //Retorna el bloque de practica que debe ser activado (lo busca en el archivo Panel\Datos\grupos_bloques.xml)
switch( Integer.parseInt( request.getParameter("numPRACT") ) )
{
case 4:
out.print( "&practice=PlantaPresion&" );
break;
case 3:
out.print( "&practice=Planta3T&" );
break;
case 2:
out.print( "&practice=Tank&" );
break;
case 1:
out.print( "&practice=Speed Control&" );
break;
default:
out.print( "&practice=ERROR&" );
break;
}
break;
default:
out.print( "&estado=ERROR_CASE&" );
break;
}
```

Figura 11. Código donde se identifica cual bloque de planta debe ser activado para la práctica.

Una vez que se realizaron estos cambios procedemos a editar el archivo “GenerarModelo.jsp” nos dirigimos a la línea de código 165 donde podemos ver la siguiente porción de código a continuación:

```
case 3:
<%
<!-- DETENER -->
<matlab:SessionGet name="<%= modeloNombre %>"/>
<matlab:Command cmd="<%= "set_param(' + modeloNombre + ', 'SimulationCommand', 'Stop')" %>"/>
<matlab:Command cmd="<%= "pause(2)" %>"/>
<%
//Abre el modelo stop motor para que apague la planta si pone stop sin apagar.....
//dependiendo del numero de practica en este caso la 2 para el motor y la 3 para el tanque
//
if(numeroPractica.equals("1")){
<%
<matlab:Command cmd="<%= "open_system('Library/stopMotor.mdl')" %>"/>
<matlab:Command cmd="<%= "set_param('stopMotor', 'SimulationCommand', 'Start')" %>"/>
<%
}
if(numeroPractica.equals("2")){
<%
<matlab:Command cmd="<%= "open_system('Library/stopTanq.mdl')" %>"/>
<matlab:Command cmd="<%= "set_param('stopTanq', 'SimulationCommand', 'Start')" %>"/>
<%
}
if(numeroPractica.equals("3")){
<%
<matlab:Command cmd="<%= "open_system('Library/stopPlanta3T.mdl')" %>"/>
<matlab:Command cmd="<%= "set_param('stopPlanta3T', 'SimulationCommand', 'Start')" %>"/>
<%
}
```

Figura 12. Código donde se identifica cual bloque de planta debe ser activado para la práctica. Parte I

```

}
if(numeroPractica.equals("4")){
%>
<matlab:Command cmd="<%= "open_system('Library/stopPlantaPresion.mdl')"%>" />
<matlab:Command cmd="<%= "set_param('stopPlantaPresion','SimulationCommand',
'Start')"%>" />
%<
}
out.print( "%estado=Stopped%" );
%>
%<
break;
}

```

Figura 13. Código donde se identifica cual bloque de planta debe ser activado para la práctica. Parte II

Nos ubicamos después de la práctica que posee el número más alto (en la imagen 13 podemos observar que el número más alto es la práctica número cuatro) una nueva planta debemos adicionar un nuevo elemento condicional if con este formato de ejemplo en la imagen:

```

%>
<!-- DETENER -->
<matlab:SessionSet name="<%= modeloNombre %>" />
<matlab:Command cmd="<%= "set_param('" + modeloNombre + "', 'SimulationCommand',
'Stop')"%>" />
<matlab:Command cmd="<%= "pause(2)" %>" />
%<
//Abre el modelo stop motor para que apague la planta si pone stop sin apagar....
//dependiendo del numero de practica en este caso la 2 para el motor y la 3 para el
tanque
if(numeroPractica.equals("1")){
%>
<matlab:Command cmd="<%= "open_system('Library/stopMotor.mdl')"%>" />
<matlab:Command cmd="<%= "set_param('stopMotor','SimulationCommand',
'Start')"%>" />
%<
}
if(numeroPractica.equals("2")){
%>
<matlab:Command cmd="<%= "open_system('Library/stopTanq.mdl')"%>" />
<matlab:Command cmd="<%= "set_param('stopTanq','SimulationCommand',
'Start')"%>" />
%<
}
if(numeroPractica.equals("3")){
%>
<matlab:Command cmd="<%= "open_system('Library/stopPlanta3T.mdl')"%>" />
<matlab:Command cmd="<%= "set_param('stopPlanta3T','SimulationCommand',
'Start')"%>" />
%<
}
if(numeroPractica.equals("Numero de la nueva practica")){
%>
<matlab:Command cmd="<%= "open_system('Library/NombredelnuevoBloque.mdl')"%>" />
<matlab:Command cmd="<%= "set_param(NombredelnuevoBloque,'SimulationCommand',
'Start')"%>" />
%<
}
out.print( "%estado=Stopped%" );
%>
%<
break;
}

```

Figura 14. Se indica el lugar en que se debe colocar el nuevo código del bloque tipo planta que se desea agregar

Toda la porción de código final que se debe agregar corresponde al bloque donde el comando “numeroPractica” tiene como valor “Numero de la nueva practica”. Finalmente luego de realizados estos cambios procedemos a grabar el archivo “generarModelo.jsp”.

12. Creación del nuevo bloque Fuzzy Logic Controller

Debido a la necesidad de realizar prácticas con un grado mayor de complejidad se tuvo que crear el bloque “Fuzzy Logic Controller”, como ya se ha visto anteriormente se han seguido los mismos pasos para agregar un nuevo bloque al sistema LabCon, sin embargo, para el correcto funcionamiento existe un parámetro llamado “fis” en donde se especifica el nombre del archivo tipo fis que se requiere para la configuración del respectivo bloque. Para agregar el bloque Simulink “Fuzzy Logic Controller” al sistema LabCon debemos seguir los pasos especificados en el punto 10 PROCESO PARA AGREGAR UN NUEVO

BLOQUE A LA LIBRERIA DEL SISTEMA LABCON.

El proceso de carga del archivo de configuración tipo fis, se hace mediante una ventana emergente (tipo popup) dentro de la opción “Fuzzy” que aparecerá en el menú, justo debajo del listado de prácticas como se podrá observar en la siguiente imagen:

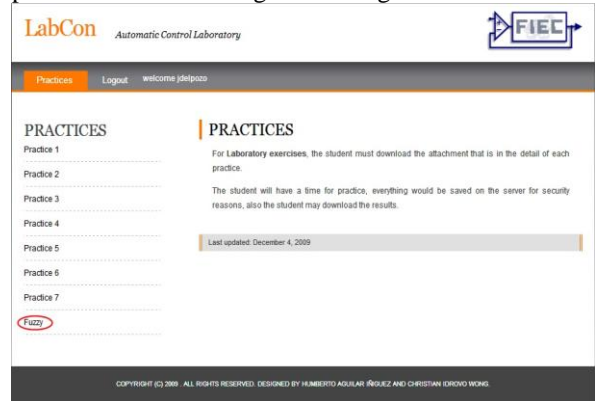


Figura 15. Opción de menú “Fuzzy” para cargar el archivo .fis

Al momento de escoger esta opción debe aparecer un navegador para que podamos elegir el archivo con extensión .fis para cargar dentro del sistema LabCon luego de esto debemos observar el mensaje “El archivo .fis ha podido ser subido correctamente” como se muestran en las imágenes siguientes:

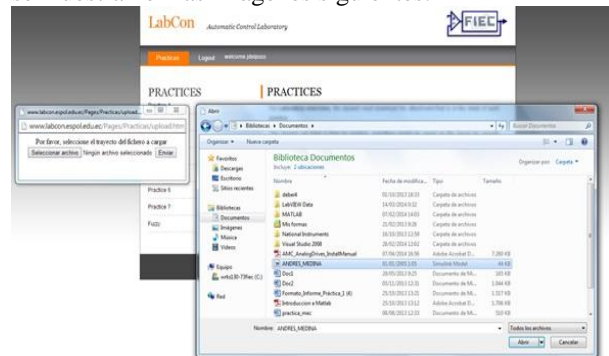


Figura 16. Navegador para buscar el archivo .fis para cargar al sistema LabCon



Figura 17. Mensaje que se muestra después de cargar el archivo .fis

Este proceso se lleva a cabo usando un servlet [3] que se encuentra dentro del archivo “uploadFichero.java” y su archivo compilado “uploadFichero.class” ubicado dentro de la ruta C:\MATLABServerPages\webapps\ROOT\WEB-INF\Classes. Es importante tomar en cuenta que solo

se puede cargar un archivo por usuario y el nombre del mismo tendrá el siguiente formato “nombreusuario.fis” dentro de la ruta C:\MATLABServerPages\webapps\ROOT\Panel\Archivos_Config.

El archivo .fis se carga dentro del bloque “Fuzzy Logic Controller” dentro del cual se encuentran las reglas de control dependiendo de la práctica que se desee utilizar dentro del sistema LabCon.

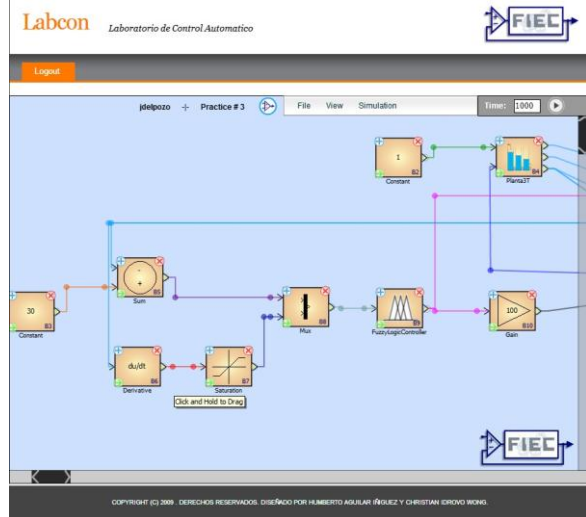


Figura 18. Bloque “Fuzzy Logic Controller” dentro de una simulación en LabCon.

Para realizar esta prueba fue necesaria crear una simulación utilizando la planta de tres tanques de agua junto con el bloque “Fuzzy Logic Controller” con un archivo de control configurado con las reglas para controlar la mencionada planta [4]. A continuación se muestran las reglas de control usadas para esta prueba:

1. If (Error is NG) and (Error is NG) then (output1 is NG)(1).
2. If (Error is NG) and (Error is NP) then (output1 is NG)(1).
3. If (Error is NG) and (Error is C) then (output1 is NG)(1).
4. If (Error is NG) and (Error is PP) then (output1 is NP)(1).
5. If (Error is NG) and (Error is PG) then (output1 is C)(1).
6. If (Error is NP) and (Error is NG) then (output1 is NG)(1).
7. If (Error is NP) and (Error is NP) then (output1 is NP)(1).
8. If (Error is NP) and (Error is C) then (output1 is NP)(1).
9. If (Error is NP) and (Error is PP) then (output1 is C)(1).
10. If (Error is NP) and (Error is PG) then (output1 is PP)(1).
11. If (Error is C) and (Error is NG) then (output1 is NG)(1).
12. If (Error is C) and (Error is NP) then (output1 is NP)(1).
13. If (Error is C) and (Error is C) then (output1 is C)(1).
14. If (Error is C) and (Error is PP) then (output1 is PP)(1).
15. If (Error is C) and (Error is PG) then (output1 is PG)(1).
16. If (Error is PP) and (Error is NG) then (output1 is NP)(1).
17. If (Error is PP) and (Error is NP) then (output1 is C)(1).
18. If (Error is PP) and (Error is C) then (output1 is PP)(1).
19. If (Error is PP) and (Error is PG) then (output1 is PG)(1).
20. If (Error is PP) and (Error is PG) then (output1 is PG)(1).
21. If (Error is PG) and (Error is NG) then (output1 is C)(1).
22. If (Error is PG) and (Error is NP) then (output1 is PP)(1).
23. If (Error is PG) and (Error is C) then (output1 is PG)(1).
24. If (Error is PG) and (Error is PP) then (output1 is PG)(1).
25. If (Error is PG) and (Error is PG) then (output1 is PG)(1).

Figura 19. Reglas de control dentro del bloque “Fuzzy Logic Controller”.

Listado de nombre de las funciones de control:

- **NG:** Negativo grande.
- **NP:** Negativo pequeño.
- **C:** Cero.
- **PG:** Positivo grande.
- **PP:** Positivo pequeño.

13. Resultados

Una vez que conocemos de forma general la estructura del archivo de control procedemos a ejecutar una simulación que fue creada usando la interfase de LabCon como se pudo observar anteriormente para que pueda realizar los siguientes pasos para controlar la altura en el primer tanque con un tiempo total de 900 segundos para nuestra simulación en tiempo real, detallados a continuación:

1. De 0 segundos a 299 segundos, la altura debe alcanzar y permanecer a un nivel de 40 centímetros.
2. De 300 segundos a 599 segundos, la altura debe alcanzar y permanecer a un nivel en 15 centímetros.
3. De 600 segundos a 900 segundos, la altura debe alcanzar y permanecer a un nivel de 30 centímetros.

Luego de ejecutar la simulación podemos observar dentro de la siguiente grafica el resultado usando el bloque “WebScope” (Osciloscopio):



Como podemos observar dentro del osciloscopio existen cinco señales diferentes que describiremos a continuación:

1. La señal de color rojo corresponde a la altura del primer tanque.
2. La señal de color verde corresponde a la altura del segundo tanque.
3. La señal de color naranja corresponde a la altura del tercer tanque.
4. La señal de color celeste corresponde a la señal de referencia del bloque de control.
5. La señal de color amarilla corresponde a la señal del error.

Finalmente como se puede apreciar en la gráfica con respecto a la señal del primer tanque podemos notar que existe un desfase respecto a cada una de las alturas que debería tener el primer tanque (40 centímetros durante los primeros 300 segundos, luego 15 centímetros y luego 30 centímetros), cuya explicación se refiere a la falta de calibración de la planta, sin embargo debido a que el propósito de este trabajo se trata de la posibilidad del uso del bloque “Fuzzy Logic Controller” para controlar de forma

remota la planta de los tres tanques de agua mediante el sistema LabCon, y como podemos observar los datos que se encuentran en la gráfica podemos confirmar el uso del bloque “Fuzzy Logic Controller” de forma correcta dentro del sistema LabCon.

14. Conclusiones

Para poder agregar un nuevo bloque (categoría simulink o categoría planta) al sistema LabCon se documentaron los pasos a seguir. Sin embargo, se recomienda que antes de realizar algún cambio dentro del sistema LabCon se realice una copia de seguridad de todo el sistema para que al momento de presentarse un error se pueda regresar a la última versión funcional del sistema.

Si el tiempo de procesamiento para la señal de control es mayor al tiempo de muestreo del sistema LabCon (0.01 segundos) se generaba un error durante la simulación causando que no se pueda apagar la planta de forma automática.

Se recomienda que el valor del tiempo de muestreo del sistema LabCon pueda ser modificado como un parámetro adicional dentro de una simulación. Sin embargo se debe tomar en cuenta que cada usuario puede trabajar con tiempos de muestreo distintos de modo que al momento de realizar una modificación de este tiempo este factor debe tomarse en cuenta dentro del análisis que se debe realizar para poder implementar esta mejora.

Durante las pruebas al momento de terminar el tiempo de ejecución de forma automática se llama a un bloque Stop (este bloque tiene como tarea enviar las señales pertinentes para el correcto apagado de una planta), sin embargo esto no ocurre si el bloque “WebScope” (osciloscopio) no se encuentra presente dentro del modelo construido para la simulación. Se recomienda modificar esta dependencia que actualmente existe con el bloque “WebScope” mientras se trabaja con un bloque categoría planta.

Para poder agregar un bloque especial como es el caso del “Fuzzy Logic Controller”, es necesario que lo haga una persona calificada en el área de computación con experiencia en manejo de Java, debido a que este proceso requiere la modificación de los archivos de configuración (webconfig) de todo el sistema LabCon, y el uso de un servlet (un programa implementado en Java que se ejecuta dentro del servidor del sistema LabCon) para realizarla carga del archivo de configuración al servidor en una ruta preestablecida (revisar el punto creación del nuevo bloque “Fuzzy Logic Controller”) así como para cualquier bloque que requiera algún tipo de funcionalidad adicional.

Se recomienda crear un programa que permita monitorear la comunicación entre las plantas y el sistema LabCon y emita una alerta detallada al usuario administrador cuando se pierda la comunicación entre ellos.

Es aconsejable agregar un módulo que permita crear un horario de trabajo con las diferentes plantas y así evitar que varios usuarios puedan trabajar con una misma planta a la vez.

Para una mayor usabilidad en la interfase del bloque “WebScope” se recomienda poder modificar los nombres de las diferentes señales y aumentar la cantidad de bloques “WebScope” que se pueden incluir dentro de una simulación.

Se sugiere asignar a una persona administradora del sistema quien se encargará de realizar un monitoreo constante, organice los horarios de trabajo para las diferentes plantas.

15. Agradecimientos

Gracias a la ayuda incondicional del Ing. Franklin Kuonqui y al MSC. Juan del Pozo para la colaboración en la realización de este trabajo.

16. Referencias

- [1] The Mathworks Inc., Simulink Documentation, <http://www.mathworks.com/help/simulink/>.
- [2] Idrovo Cristhian, Aguilar Humberto, “Laboratorios Remotos: Comunicación cliente servidor y ejecución remota para las prácticas del laboratorio de control automático de la facultad de Ingeniería en Electricidad y Computación (FIEC)” - Tesis de grado.
- [3] Alegsa Leandro, Alegsa.com.ar, <http://www.alegsa.com.ar/Dic/servlet.php>.
- [4] Passino M. Kevin y Yurkovich Stephen, Fuzzy Control, Addison-Wesley.