

S.E.P.

S.E.I.T.

D.G.I.T.

**CENTRO NACIONAL DE INVESTIGACIÓN
Y DESARROLLO TECNOLÓGICO**

cenidet

***REGULACIÓN DE UNA PLANTA TERMOELÉCTRICA
POR MEDIO DE CONTROLES ADAPTABLES***

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS EN
INGENIERÍA ELECTRÓNICA**

PRESENTA:

ING. ROBERTO GALINDO DEL VALLE

DIRECTOR DE TESIS:

DR. ENRIQUE QUINTERO-MÁRMOL M.

Cuernavaca, Morelos, México. Noviembre de 2001

Índice

Lista de tablas	ix
Lista de figuras	xi
Lista de abreviaturas y acrónimos	xv

Resumen	xvii
----------------------	-------------

1 Introducción

1.1 Antecedentes	1
1.1.1 Origen del problema	1
1.1.2 Aplicación a la planta termoeléctrica	2
1.2 Estado del arte	2
1.2.1 Teoría del control adaptable	2
1.2.2 Aplicaciones	5
1.3 Organización del documento	6

2 Control por colocación de polos

2.1 Elección de un método de control adaptable	9
2.2 Descripción del método de colocación de polos de Aström.....	11
2.2.1 Método recursivo de mínimos cuadrados (<i>RLS</i>).....	12
2.2.2 Controladores lineales generales (<i>CLG's</i>).....	14
2.2.2.1 Enfoque de Chen: Método algebraico lineal.....	15
2.2.2.2 Primer enfoque de Aström: Ecuaciones lineales.....	16
2.2.2.3 Segundo enfoque de Aström: Solución general.....	17
2.2.2.4 Eliminación de errores de offset: Acción reset.....	17
2.2.2.5 Comparación de los tres enfoques de diseño de <i>CLG's</i>	19

3 Descripción de la planta termoeléctrica (PTE)

3.1 Descripción general de la PTE.....	21
3.2 Sistemas de control considerados.....	22
3.2.1 Control de velocidad de la Turbo bomba (TB).....	23
3.2.2 Control de Presión de vapor (PV).....	24
3.2.3 Control de temperatura en el sobrecalentador secundario(TSC).....	24

4	Diseño del controlador	
4.1	Diseño del Controlador Adaptable	27
4.1.1	Componentes del <i>CACP</i>	27
4.1.1.1	Identificador	27
	Filtros de datos	28
	Estimador	29
	Selector de modelo	29
4.1.1.2	Controlador	31
	PID discreto	31
	CLG discreto	31
4.1.1.3	Supervisor	34
4.1.2	Modos de operación	35
4.1.2.1	Selección de Modelo (<i>MS</i>)	35
4.1.2.2	Autosintonización (<i>STR</i>)	37
4.1.2.3	CLG con parámetros fijos (<i>CLG</i>)	37
4.1.2.4	PID con sintonización fija (<i>PID</i>)	37
4.1.2.5	Manual	37
4.2	Implementación en <i>LabWindows</i> [®]	38
4.3	Comentarios importantes	40
4.4	Aplicación del <i>CACP</i> a tres lazos de la PTE	40
4.4.1	Inicialización de los <i>CACP</i> 's	42
4.4.2	Prueba 1: de distintos modelos de primer y segundo orden	43
4.4.3	Prueba 2: de ubicación de polos	48
4.4.4	Prueba 3: de elección de filtros de datos	53
5	Evaluación de desempeño del <i>CACP</i>	
5.1	Pruebas de evaluación de desempeño	57
5.1.1	Prueba 1: de Regulación	58
5.1.1.1	Decremento en rampa de potencia generada del 100% al 77.5% ...	59
5.1.1.2	Decremento en rampa de potencia generada del 77.5% al 50%	63
5.1.2	Prueba 2: de operación en modo 1 (<i>MS</i>)	66
5.1.3	Prueba 3: de operación con ($NB > NA$)	68
6	Conclusiones y Recomendaciones	
6.1	Conclusiones	71
6.2	Recomendaciones	75
Anexo 1		
	Regulador Autosintonizable de Varianza Mínima (<i>MV-STR</i>)	79

Anexo 2Control Predictivo Generalizado (*GPC*) 85**Anexo 3**Diseño de Controladores Lineales Generales (*CLG's*) 93**Anexo 4**

Filtrado de señales 97

Anexo 5

Programa de simulación 101

Referencias Bibliográficas 129

Anexo 5

Programa de simulación

Este Anexo se dedica por completo al programa de simulación desarrollado en el lenguaje C de *LabWindows*[®] para evaluar el comportamiento del *CACP*. Inicialmente se describe el programa desde una perspectiva "externa", para conocer el funcionamiento durante una corrida típica. A continuación se describe el programa desde una perspectiva "interna", considerando las diferentes subrutinas desarrolladas, sus variables asociadas y sus interacciones correspondientes. Para terminar, se describe la forma de reproducir las pruebas presentadas en el Capítulo 3 del documento de tesis.

I. Programa de Simulación de la Planta Termoeléctrica: *TermoCACP2.prj*

Este programa simula el comportamiento de una planta termoeléctrica de 600 MW, usando el modelo propuesto por [Usoro, 1977].

Para que el usuario observe el comportamiento simulado, al ejecutar el programa se despliega un panel principal, véase la figura A5.1, en el que aparece un diagrama esquemático de la planta con diversos indicadores numéricos y seis botones de control. A su vez, los indicadores numéricos despliegan las magnitudes de algunas variables de la simulación, mientras que los botones sirven para mostrar los diferentes paneles de control relacionados con el mismo número de sistemas de la planta. De esta forma, cuando se desea observar un panel de control específico el usuario debe oprimir el botón asociado.

Menú del Panel Principal

El panel principal cuenta con un menú que presenta las alternativas listadas y descritas a continuación:

Inicio: Sirve para iniciar la simulación.

Pausa: Detiene la simulación. Para continuarla es necesario escoger de nuevo este encabezado del menú principal.

Ayuda: Muestra el cuadro de ayuda del programa, el cual describe brevemente el funcionamiento del mismo.

Evaluación: Despliega el Panel de Evaluación, mostrado en la figura A5.2a. Este panel permite observar el comportamiento de la salida de cada sistema controlado con *CACP*, así como la magnitud del criterio de evaluación J usado en las pruebas de regulación de la planta termoeléctrica. Una de las dos formas de ocultar el panel mencionado es elegir de

nuevo esta opción del menú principal. La otra forma consiste en oprimir el botón correspondiente sobre dicho panel. Este panel sólo se usa en las pruebas de regulación.

Opciones: Esta alternativa del menú principal presenta las Opciones de la Simulación, que son: Operación en estado estable al 100% de capacidad, Primer Prueba de Regulación, Operación en estado estable al 77.5% de capacidad y Segunda Prueba de Regulación. En las figuras A5.2b y A5.2c se muestran los submenús correspondientes a estas opciones de simulación.

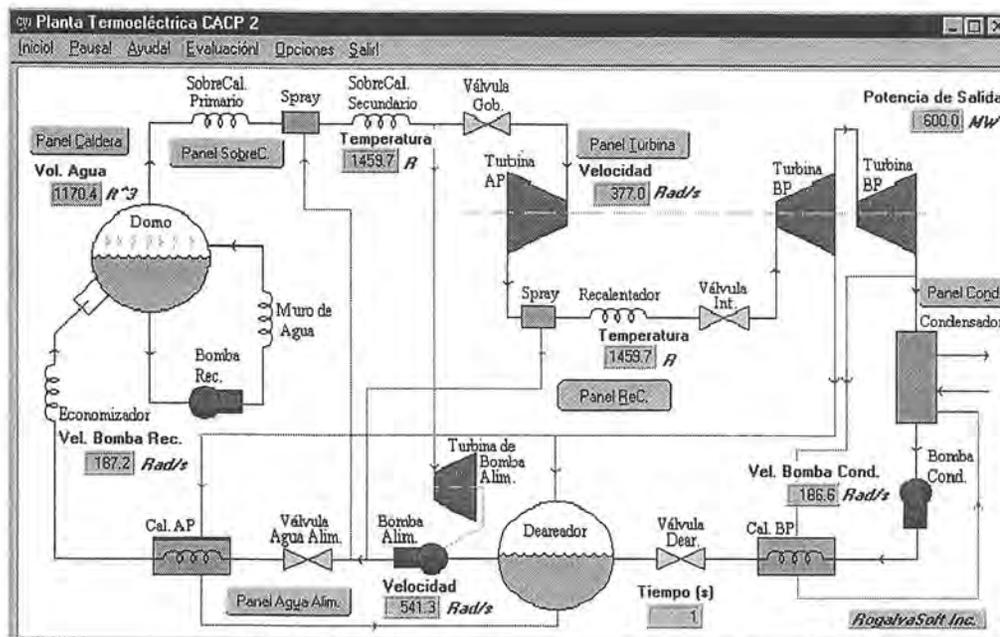
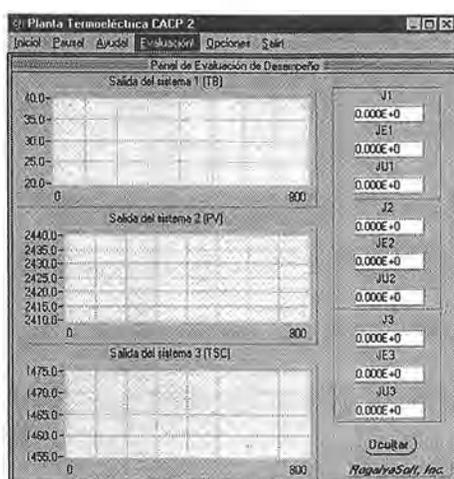


Figura A5.1. Panel Principal del programa *TermoCACP2.prj*.



(a)



(b)



(c)

Figura A5.2. (a) Panel de Evaluación; (b) Submenú de inicio de operación al 100 % de capacidad: para simular estado estable o primer prueba de regulación; (c) Submenú de inicio de operación al 77.5 % de capacidad: para simular estado estable o segunda prueba de regulación.

IMPORTANTE: El usuario debe proponer las condiciones de operación que desea simular *antes* de iniciar la simulación (o sea, antes de oprimir **Inicio**).

Salir: Provoca el fin de la simulación y termina la ejecución del programa.

Paneles de Control

En la tabla A5.1 se mencionan los paneles de control existentes en el programa, así como los controladores relacionados con cada uno de ellos. En esta tabla se especifica si el controlador utilizado es *PID* convencional o *CACP*. Además, los controladores configurados como esclavos se han marcado con una *E*.

Panel de Control	Controladores Asociados	Tipo de Ctrl.
De la <i>Caldera</i> .	1. Controlador de Presión de Vapor. 2. Controlador de Aire. 3. Controlador de Combustible.	1. <i>CACP</i> . 2. <i>PID, E</i> . 3. <i>PID, E</i> .
Del Sobrecalentador. (<i>SobreC</i>)	1. Controlador de Temperatura en el SobreC. 2. Control de Presión en el hogar.	1. <i>CACP</i> . 2. <i>PID</i> .
De la <i>Turbina</i> .	1. Unidad de Control de Carga. 2. Unidad de Control de Velocidad.	1. <i>PID</i> . 2. <i>PID</i> .
Del Recalentador. (<i>ReC</i>)	1. Controlador de Temperatura en el ReC. 2. Control de Recirculación de Gases.	1. <i>PID</i> . 2. <i>PID</i> .
Del Condensador. (<i>Cond</i>)	1. Control de Nivel en el Deareador. 2. Control de Flujo de Condensado.	1. <i>PID</i> . 2. <i>PID, E</i> .
Del Agua de Alimentación. (<i>Agua Alim</i>)	1. Control de Nivel en el Domo. 2. Controlador del Agua de Alimentación. 3. Control de la Turbo Bomba del sistema de alimentación de agua.	1. <i>PID</i> . 2. <i>PID, E</i> . 3. <i>CACP</i> .

Tabla A5.1. Paneles de control del programa *TermoCACP2.prj*.

Por otro lado, en la figura A5.3 es posible observar la apariencia de un panel de control típico. En particular se muestra el Panel del Sistema de Alimentación de Agua, el cual es desplegado al oprimir el botón *Panel Agua Alim*. Otra forma de desplegar los paneles de control es pulsar la combinación *Alt+ letra subrayada en el botón*. En el caso mostrado en la figura A5.3 la combinación es *Alt+u*. Cada panel cuenta con diversos indicadores numéricos, entre los que se encuentra el que despliega el tiempo simulado, y con algunos controladores asociados con el sistema específico. Además, en cada panel de control se tienen 4 gráficas de despliegue continuo que muestran el comportamiento de diversas variables de la simulación que se relacionan con el sistema de interés. También, se tiene un botón *Ocultar*, el cual permite regresar al panel principal. Otra forma de hacer esto es mediante la combinación *Alt+l*. Nótese que en la parte superior de la figura sigue apareciendo el menú principal.

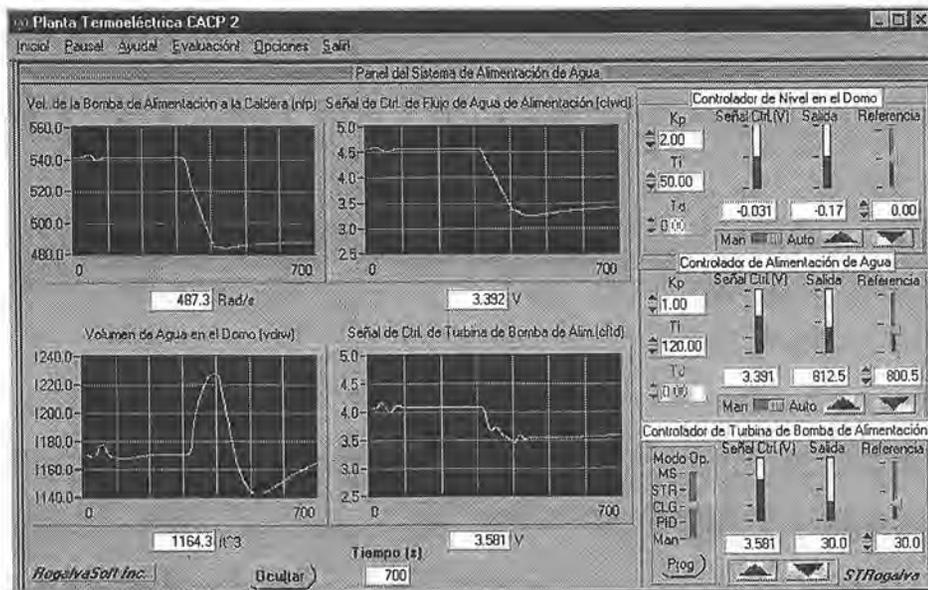


Figura A5.3. Apariencia de un panel de control típico del programa *TermoCACP2.prj*.

En lo que se refiere a los controladores mostrados en cada panel, éstos pueden ser *PID's* convencionales (continuos o analógicos) o *CACP's* (discretos), dependiendo del sistema controlado (sólo se usan *CACP's* en sistemas de TB, de PV y de TSC).

Controladores *PID* convencionales

Los *PID's* convencionales permiten modificar la sintonización del controlador, así como el punto de ajuste (señal de referencia) del mismo, a la vez que muestran los valores de la salida real del sistema controlado y de la acción de control tomada. En cada controlador aparecen atenuados los parámetros que no se utilizan; por ejemplo: *en el controlador de Nivel en el Domo*, del panel mostrado en la figura A5.3, *se tiene atenuado* el parámetro que se relaciona con la acción derivativa de control, lo cual indica que la ley de control utilizada es de tipo *PI*. Además, cada *PID* convencional tiene implementada una estación Manual/Automático sencilla, la cual no cuenta aún con una transferencia sin saltos de manual a automático. Cuando el controlador está en modo manual, el usuario debe usar las teclas de flecha arriba o abajo para incrementar o disminuir la magnitud de la señal de control, respectivamente. En modo automático estas teclas están inhabilitadas. También, se debe decir que *no es posible modificar el punto de ajuste* de aquellos controladores que están definidos como *esclavos* en una configuración *cascada*, ni de la Unidad de Control de Carga, cuyo punto de ajuste está definido por la señal de control demanda de carga *ldc*.

Controladores Adaptables Por Colocación de Polos (*CACP's*)

Por su parte, los *CACP's* permiten modificar el punto de ajuste y observar la magnitud tanto de la salida real del sistema controlado, como de la acción de control tomada (en el intervalo de 1-5 Volts o de 2.5-5 Volts), mediante tres indicadores numérico-visuales dispuestos para tal fin. En la figura A5.3 el controlador de la turbina de la bomba de alimentación (TB) es un *CACP* típico. Además, los *CACP's* cuentan con un conmutador o selector de modo de operación

(marcado como *Modo Op.*), dos botones de flecha arriba/abajo para la manipulación de la señal de control en modo manual y un botón de programación (*Prog*). Este último, al ser oprimido, despliega un panel en el que se pueden modificar los diversos parámetros del controlador. Este panel simula ser el menú de programación del *CACP* y contiene tres graficadores de despliegue continuo que permiten conocer la salida del sistema, la evolución de la señal de control y de algunos parámetros del estimador. Este panel en adelante se denominará *Panel de Programación*. La única forma de ocultar el panel de programación es volver a oprimir el botón *Prog* correspondiente. De esta forma, si el usuario quiere volver al panel principal o ir a otro panel de control, primero debe ocultar el panel de programación, después el panel de control que está activo y finalmente desplegar el panel de control deseado, oprimiendo el botón correspondiente sobre el panel principal.

En la figura A5.4 se muestra el Panel de Programación del *CACP*, así como los paneles de Parámetros de Respuesta deseada y de Parámetros del modelo, que serán descritos más adelante.

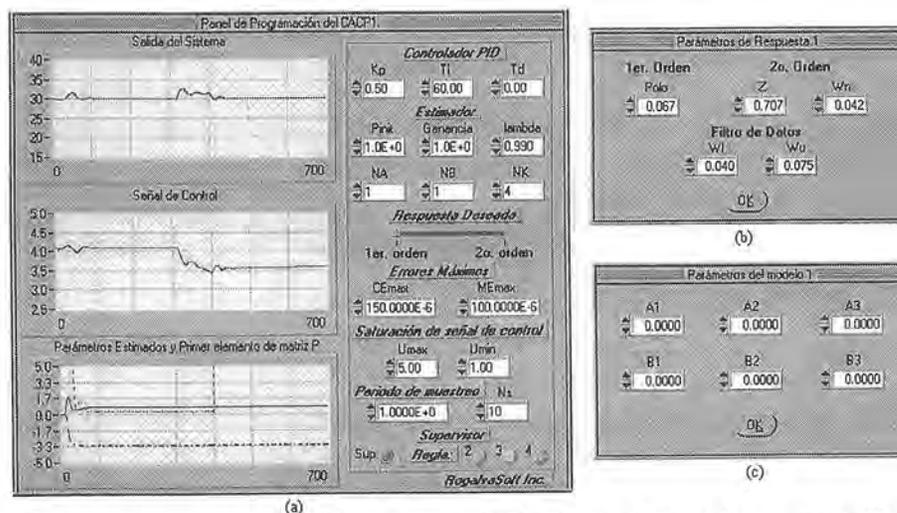


Figura A5.4. (a) Panel de programación del *CACP*. Este panel simula ser el menú de programación del controlador y permite modificar los parámetros más importantes del mismo. (b) Panel de Parámetros de Respuesta deseada; este panel es mostrado cuando el usuario utiliza el selector de *Respuesta Deseada* del Panel de Programación. (c) Panel de Parámetros del modelo. Este panel se despliega cuando el usuario solicita la operación en modo 2 (*STR*) sin haber usado antes el modo 1 (*MS*) y permite proponer la inicialización de hasta seis parámetros del modelo que será utilizado.

En la parte superior del Panel de Programación de la figura A5.4a es posible modificar las ganancias del controlador *PID* (K_p , τ_i y τ_d). A continuación, se muestran las variables del estimador: al modificar *Pinit*, internamente se reinicia la matriz *P* como *Pinit* veces la matriz identidad; también, es posible proponer los valores de la ganancia del regresor, del factor de olvido (λ) y de las características del modelo a utilizar (*NA*, *NB* y *NK*). Después, aparece el selector de Respuesta Deseada, el cual despliega el panel de parámetros de respuesta deseada mostrado en la figura A5.4b, donde es posible modificar la *magnitud* del polo propuesto (cuando se solicita una respuesta de primer orden) o de la relación de amortiguamiento ζ y de la frecuencia natural no amortiguada ω_n (cuando se solicita una respuesta de segundo orden), así como de las características del filtro de datos a usar (ω_l y ω_u). Más abajo, es posible modificar los

errores de modelado (ME) y de control (CE) máximos permisibles antes de detener la autosintonización. A continuación, se pueden establecer los límites inferior y superior de la señal de control (U_{\min} y U_{\max}). Después, es posible modificar tanto el periodo de muestreo (h) en segundos, como la variable NS . Esta variable se usa para simular el proceso de muestreo de las señales continuas y su magnitud debe ser elegida igual al número de veces enteras que el paso de integración ($dt=0.1s$) es contenido por el periodo de muestreo; en otras palabras, NS sólo se utiliza para los fines de la simulación, debido a lo cual no aparecería como un parámetro más de la versión real del $CACP$. Finalmente, aparece un conjunto de cuatro LED's indicadores que permiten observar si el supervisor está habilitado (el primero de ellos, marcado Sup) y cuál de las reglas (2, 3 ó 4) ha sido activada por los sucesos presentes (sólo reglas que detienen la autosintonización). Además, es posible habilitar o inhabilitar el supervisor del $CACP$ seleccionando el LED correspondiente con el botón izquierdo del *mouse*.

Los graficadores de despliegue continuo solamente son usados en la simulación para observar la forma en que evolucionan la salida del sistema, la señal de control, los dos primeros parámetros de cada polinomio del modelo (A_1 en color azul o línea segmentada y B_1 en color rojo o línea continua) y el primer elemento de la matriz P ($P[0,0]$ en color verde o línea punteada). Debe comentarse que, con el fin de poder desplegar el comportamiento de estos tres últimos parámetros en una misma gráfica, se han escalado todas las magnitudes, de tal forma que en realidad se despliega $5B_1$, $3A_1$ y $5P[0,0]/Pinit$. Ninguno de estos graficadores aparecería en una versión real del $CACP$, a menos de que a éste se le agregara una pantalla de cristal líquido lo suficientemente grande.

El programa está desarrollado en forma tal que sólo pueden modificarse en cualquier tiempo de una corrida: el modo de operación de los controladores, la magnitud de la señal de control (cuando el controlador está en modo manual), el punto de ajuste o referencia, las ganancias de los controladores PID (continuos y discretos), la matriz P , el factor de olvido (λ), la ganancia del regresor y los errores máximos permisibles antes de detener la autosintonización (CE_{max} y ME_{max}). Sin embargo, ninguno de los controladores cuenta con un módulo de transferencia sin saltos, por lo que podrían existir variaciones en la señal de control al modificar alguno de estos parámetros. Todos los demás parámetros sólo pueden ser modificados antes de iniciar la simulación, o sea en $t=0s$. En otras palabras, si el usuario desea modificar la magnitud de cualquier otro parámetro y ya ha elegido la opción **Inicio** del menú, entonces deberá hacerlo en otra corrida del programa. En especial, cuando el usuario solicita *por primera vez* la operación en modo 2 (STR), sin haber usado antes el modo 1 (MS), el $CACP$ le ofrece la posibilidad de proponer las magnitudes iniciales de hasta seis parámetros del modelo a utilizar. Para ello se despliega el panel de Parámetros del modelo, mostrado en la figura A5.4c.

II. Organización del Programa

En esta sección se describirá la forma en que está organizado internamente el programa *TermoCACP2.prj*, describiendo las subrutinas que lo componen y sus interacciones respectivas. Dado el tamaño del programa, la explicación será breve en la mayoría de los casos y solamente se extenderá para las subrutinas del $CACP$ y para tratar las modificaciones que sería necesario realizar con el fin de sustituir un PID con un $CACP$ en algún otro sistema de la planta. Para ello, se asume que el lector posee, en general, conocimientos básicos de programación en lenguaje C y, en particular, de la utilización del ambiente de desarrollo *LabWindows*®. Si esto último se cumple, entonces en esta sección se encontrará todo lo necesario para entender el código fuente

del programa y poder modificarlo en el futuro. Si no es este el objetivo del lector, se recomienda evitar la lectura de esta sección y pasar a la de "Reproducción de las pruebas realizadas al CACP".

Antes de tratar el funcionamiento y organización interna del programa, se mencionarán las distintas funciones que son utilizadas. En adelante, y en este contexto, se usará la palabra *subrutina* como un sinónimo de *función*. Las subrutinas pueden ser de alguno de los siguientes tres tipos:

Funciones de Biblioteca: Son aquellas que están incluidas en las distintas bibliotecas con las que cuenta el *LabWindows/CVI* (User Interface, Advanced Analysis, ANSI C, etc).

Funciones de Llamado :Son aquellas que están asociadas con algún objeto de la interface gráfica del usuario (opciones del menú, botones, temporizadores o *timers*, etc.), de tal forma que cuando se realiza algún evento en o sobre dicho objeto, éste *llama* a su función correspondiente.

Funciones de Usuario : Son las que el programador ha elaborado y que contienen la parte fundamental de la aplicación considerada.

Funciones de Biblioteca usadas

Add1D	: Suma elemento a elemento dos arreglos unidimensionales de datos.
calloc	: Asigna espacio de memoria para un arreglo de objetos con tamaño y número especificado. Todos los bits son inicializados en cero.
Clear1D	: Hace que todos los elementos de un arreglo sean iguales a cero.
Convolve	: Realiza la convolución entre dos arreglos de datos.
Copy1D	: Copia el contenido de un arreglo a otro.
DisplayPanel	: Despliega un panel en la pantalla.
exp	: Calcula la función exponencial del argumento especificado.
free	: Causa que el espacio apuntado por un apuntador de bloque de memoria sea liberado.
GetCtrlVal	: Obtiene el valor actual de un control.
HidePanel	: Oculta un panel de la pantalla pero lo deja en memoria.
InitCVIRTE	: Inicializa el CVI RTE (<i>CVI Run-Time Engine</i>) de <i>LabWindows®</i> .
InvMatrix	: Determina la inversa de una matriz cuadrada.
LoadPanel	: Carga un panel de memoria desde el archivo de recursos de la interface del usuario (*.uir).
MatrixMul	: Multiplica dos matrices.
MessagePopup	: Despliega un mensaje en la pantalla en una caja de dialogo y espera hasta que el usuario oprime el botón 'OK'.
PlotStripChart	: Adiciona uno o más puntos a cada trazo en una gráfica de despliegue continuo (<i>StripChart</i>).
pow	: Calcula la potencia indicada de un número.
QuitUserInterface	: Termina la ejecución de RunUserInterface.
RunUserInterface	: Corre la interface del usuario y envía eventos a las funciones de llamado.
SetCtrlVal	: Fija el valor de un control al valor especificado.
SetInputMode	: Determina que tanto es reconocida o no la entrada de un usuario. Es decir, habilita o inhabilita un control de la interface gráfica, para aceptar o no entradas del usuario.

Shift	: Desplaza los elementos de un arreglo un número especificado de lugares.
sqrt	: Calcula la raíz cuadrada no negativa del argumento dado.
Sub1D	: Resta elemento a elemento dos arreglo unidimensionales de datos.
Sum1D	: Encuentra la suma de todos los elementos de un arreglo.
Transpose	: Determina la matriz transpuesta.
WhiteNoise	: Genera un arreglo de números aleatorios uniformemente distribuidos entre -amplitude y +amplitude.

Funciones de Usuario

Las funciones de usuario utilizan las siguientes variables, algunas de las cuales son parámetros del controlador, mientras que otras sólo se usan para los fines de la simulación:

thetaA1	: Almacena todos los vectores de parámetros de los modelos a comparar.
PhiA1	: Almacena todos los vectores regresores de los modelos a comparar.
PA1	: Almacena todas las matrices de covarianzas de los modelos a comparar.
data1	: Arreglo para guardar <i>MSS</i> y <i>MSR</i> de cada uno de los modelos a comparar.
naMax1	: Número máximo de parámetros a considerar en los polinomios <i>A</i> y <i>B</i> durante la comparación de modelos.
nkMax1	: Retardo máximo considerado en los modelos a comparar.
SC1	: contiene el valor numérico del criterio 2.6 de cada modelo a comparar.
min1, imin1	: Variables usadas para determinar el modelo que minimiza el criterio 2.6.
ValStart1	: Indica el tiempo de inicio de la validación-comparación de los 33 modelos.
ModSelEnd1	: Indica el tiempo de finalizar la validación-comparación de los 33 modelos.
theta1	: Vector de parámetros del modelo usado en operación normal del <i>CACP</i> .
Phi1	: Vector regresor del modelo usado en operación normal del <i>CACP</i> .
P1	: Matriz de covarianzas (<i>P</i>) del modelo usado en operación normal del <i>CACP</i> .
Pinit1	: Usada para inicializar la matriz de covarianzas: $P = Pinit * I$.
lambda1	: Factor de olvido usado para estimar los parámetros del modelo.
gain1	: Ganancia del regresor (ver expresión 2.5).
NA1	: Número de parámetros del polinomio <i>A</i> .
NB1	: Número de parámetros del polinomio <i>B</i> .
NK1	: Retardo del modelo usado en operación normal del <i>CACP</i> .
h1	: Intervalo de muestreo usado por el <i>CACP</i> .
u1	: Señal de control (Arreglo de datos).
ufd1	: Señal de control filtrada (Arreglo de datos).
y1	: Salida del sistema (Arreglo de datos).
yfd1	: Salida del sistema filtrada (Arreglo de datos).
y_sim1	: Salida simulada por el modelo estimado.
PredErr	: Error de predicción del modelo (ver expresiones 2.4).
uc1	: Señal de referencia o valor deseado (Arreglo de datos).
e1	: Error en el sistema, definido como $e(kh) = y(kh) - u_c(kh)$ (Arreglo de datos).
ua1	: Cálculo anticipado de la señal de control (ver expresiones 2.7** y 2.26).
w11	: Frecuencia de corte inferior del filtro de datos.
wu1	: Frecuencia de corte superior del filtro de datos.
Nfd1, Dfd1	: Arreglos que contienen a los coeficientes del filtro de datos discretizado.
Kp1, Ti1, Td1	: Ganancias del controlador <i>PID</i> , antes de ser discretizado.
E1	: Coeficientes del <i>PID</i> discreto correspondiente.

- umin1 y umax1 : Valores mínimo y máximo de la señal de control, respectivamente.
 model : Modo de operación del *CACP*.
 PRBS1 : Señal Binaria Pseudo Aleatoria usada por el *CACP*.
 A1, B1 : Polinomios del modelo estimado.
 sB : ganancia en estado estable del polinomio B1 estimado.
 n1 : orden real del sistema (expresión 2.8**).
 R1, S1, T1 : Polinomios del *CLG* (ver expresión 2.23).
 nr1 : Grado de los polinomios del *CLG*.
 sR : Ganancia en estado estable del polinomio R1.
 ord1 : Orden de la respuesta deseada por el usuario (primero o segundo: 1 ó 2).
 pole1,z1,wn1: Parámetros de la respuesta deseada.
 Am1 : Polinomio característico de la respuesta deseada (ver expresión 2.11).
 sAm1 : Ganancia en estado estable del polinomio Am1.
 Bm1 : Constante para asegurar ganancia unitaria en lazo cerrado.
 M1 : Polinomio para agregar acción integral (ver la expresión 2.20).
 sM1 : ganancia en estado estable de M1.
 N1 : Polinomio para agregar acción integral (ver expresión 2.21).
 Ao1 : Polinomio observador (ver expresión 2.13*).
 Ac1 : Se le llama así al producto $A_m(q) \cdot A_0(q)$ (ver expresión 2.15).
 X1 : Arreglo (vector) que contiene los coeficientes de R y S (al resolver el sistema de ecuaciones generado por la expresión 2.15).
 NX1 : Tamaño del vector X1.
 Mc1 y mc1 : Matrices de coeficientes del sistema de ecuaciones generado por la expresión 2.15.
 Tt : Polinomio (T temporal) usado en la función de diseño del *CLG*.
 Evaluate1 : Bandera que indica si se desea evaluar el comportamiento del *CACP* (usada en pruebas de regulación).
 J1, JE1, JU1 : Variables para almacenar el valor numérico del criterio 3.1 del *CACP* (usadas en pruebas de regulación).
 k1 : Variable usada para contar el número de muestreos que han transcurrido desde el inicio del periodo de selección o de adaptación inicial.
 sup1 : Indica si el supervisor está activado o no.
 CtrlErr1 : Error de control del sistema (definido en la expresión 2.27).
 CEmax1 : Error de control máximo permisible antes de detener la autosintonización.
 ModErr1 : Error de modelado (definido en la expresión 2.27).
 MEmax1 : Error de modelado máximo permisible antes de detener la autosintonización.
 R11, R21, R31: Variables que indican cuál regla del supervisor se ha activado.
 j1, m1, Pant1: Variables asociadas con el funcionamiento del supervisor.
 Ts1,Ns1 : Variables usadas para simular el muestreo de las señales continuas.
 validate1 : Indica si el *CACP* está en modo 1(*MS*), o si está en el periodo de adaptación inicial, en el modo 2(*STR*).
 firstime1 : Indica si el *CACP* ya ha usado o no el modo 2(*STR*) de operación.
 flag1 : Usada para desplegar en la interface gráfica el modelo elegido al concluir el periodo de selección. O sea, para desplegar NA1, NB1 y NK1 en la interfase.

NOTA: En particular, se han mostrado las variables correspondientes al *CACP* de TB. Las correspondientes al *CACP* de PV se obtienen sustituyendo el "1" al final de las variables por un "2". Las del *CACP* de TSC usan un "3".

A continuación se muestran las funciones de usuario desarrolladas:

- c_ftCACP1** : Realiza el algoritmo *CACP*, con las operaciones indicadas en la figura 2.2a, para controlar el sistema de TB. En particular, determina la señal de control usando las expresiones (2.25 y 2.26) en modos 2(*STR*) y 3(*CLG*) o las expresiones (2.7* y 2.7**) en modos 1(*MS*) y 4(*PID*). Durante su ejecución usa las funciones de usuario *DataFilter*, *SelectModel1*, *Identifier*, *PP_Design1* y *UpdateCriterionValue*. Además, usa las funciones de biblioteca *pow*, *Shift* y *MatrixMul*.
- c_brCACP2** : Realiza el algoritmo *CACP*, con las operaciones indicadas en la figura 2.2a, para controlar el sistema de PV. Prácticamente se trata del mismo código que en la función *c_ftCACP1*, pero adecuado a las variables y funciones del *CACP2*. La diferencia más notable radica en el código de adquisición de datos.
- c_shCACP3** : Realiza el algoritmo *CACP*, con las operaciones indicadas en la figura 2.2a, para controlar el sistema de TSC. Prácticamente se trata del mismo código que en la función *c_ftCACP1*, pero adecuado a las variables y funciones del *CACP3*. La diferencia más notable radica en el código de adquisición de datos.
- DataFilter** : Realiza el filtrado de los datos a usarse en la estimación paramétrica, con la expresión (2.2). Usa la función de biblioteca *MatrixMul*.
- DiscretizeDataFilter** : Discretiza el filtro de datos propuesto por el usuario (ver expresión 2.1), determinando la magnitud de los coeficientes del filtro discreto, dados por la expresión (2.2).
- DiscretizePID** : Realiza la discretización del controlador PID determinado por las ganancias impuestas por el usuario, de acuerdo con las expresiones (2.7).
- Identifier** : Actualiza el vector regresor, calcula el error de predicción y actualiza el modelo del sistema. Utiliza las funciones de usuario *RecursiveLS* y *MultEsc*; además de las funciones de biblioteca *MatrixMul* y *Shift*.
- InitializeCACP1** : Asigna memoria en forma dinámica, con la función de biblioteca *calloc*, para todos los arreglos o vectores de datos a usarse con el algoritmo del *CACP* de TB: *A1*, *B1*, *Am1*, *Ao1*, *Ac1*, *R1*, *S1*, *T1*, *X1*, *Mc1* y *mc1*. Además, determina el orden real del sistema (*n1*), el tamaño del vector *X1* (*NX1*), los polinomios *Am1* y *M1*, la ganancia en estado estable de *Am1* (*sAm1*) y la ganancia en estado estable de *M1* (*sM1*).
- InitializeCACP2** : Igual que anterior pero para el *CACP* de PV. En esta caso las variables son: *A2*, *B2*, *Am2*, *Ao2*, *Ac2*, *R2*, *S2*, *T2*, *X2*, *Mc2*, *mc2*, *n2*, *NX2*, *Am2*, *M2*, *sAm2* y *sM2*. Prácticamente se ha repetido todo el código de *InitializeCACP1* con el fin de evitar el uso de una función con gran cantidad de parámetros o de una programación más compleja.
- InitializeCACP3** : Igual que anterior pero para el *CACP* de TSC. En esta caso las variables son: *A3*, *B3*, *Am3*, *Ao3*, *Ac3*, *R3*, *S3*, *T3*, *X3*, *Mc3*, *mc3*, *n3*, *NX3*, *Am3*, *M3*, *sAm3* y *sM3*.
- InitSt77** : Asigna memoria para los vectores de datos que guardarán la información de entrada (*u1*, *u2* y *u3*) y salida (*y1*, *y2* y *y3*) del sistema, así como de la señal de referencia (*uc1*, *uc2* y *uc3*) de los tres *CACP*'s.

	Además, inicializa estos vectores y todos los estados continuos de la planta termoeléctrica con sus valores correspondientes al 77.5% de capacidad. Estos últimos con los nombres asignados de acuerdo a la nomenclatura establecida por Usoro en su documento original.
InitSt100	: Igual que la anterior, pero ahora se inicializan todos los estados de la planta con sus valores correspondientes al 100% de capacidad.
MultEsc	: Multiplica por una constante dada a todos los elementos de un arreglo unidimensional de datos.
PP_Design1	: Realiza el diseño de un <i>CLG</i> mediante las expresiones (2.14 - 2.22), para el <i>CACP</i> de TB. Para esto, utiliza las funciones de biblioteca <i>calloc</i> , <i>Copy1D</i> , <i>Sum1D</i> , <i>Sub1D</i> , <i>Convolve</i> , <i>Transpose</i> , <i>InvMatrix</i> , <i>MatrixMul</i> , <i>Clear1D</i> , <i>Add1D</i> y <i>free</i> . Además, usa la función de usuario <i>MultEsc</i> .
PP_Design2	: Igual que la anterior, pero para el <i>CACP</i> de PV. Prácticamente es el mismo código, pero el sufijo numérico "1" de todas las variables se ha cambiado por "2". Esto se ha hecho por simplicidad, con el fin de evitar el uso de una función con gran cantidad de parámetros o de una programación más compleja.
PP_Design3	: Igual que la anterior, pero para el <i>CACP</i> de TSC.
RecursiveLS	: Realiza la estimación paramétrica de acuerdo con las expresiones (2.4). Además, determina la salida simulada por el nuevo modelo. Usa las funciones de biblioteca <i>calloc</i> , <i>MatrixMul</i> , <i>Sub1D</i> , <i>Add1D</i> y <i>free</i> ; además de la función de usuario <i>MultEsc</i> .
SelectModel1	: Es usada transitoriamente en el modo 1 (<i>MS</i>), para realizar la selección del modelo de acuerdo con el algoritmo propuesto para el <i>CACP</i> de TB. Durante el periodo de selección, cada instante de muestreo actualiza los 33 modelos a comparar y determina recursivamente <i>MSR</i> y <i>MSS</i> para cada uno de ellos. Después, elige el modelo que minimiza el criterio (2.6). Usa las funciones de biblioteca <i>calloc</i> , <i>pow</i> , <i>Copy1D</i> y <i>free</i> . Además, llama a las funciones de usuario <i>ToMakeVectors_Matrix</i> , <i>Identifier</i> , <i>InitializeCACP1</i> y <i>PP_Design1</i> .
SelectModel2	: Igual que anterior, pero para el <i>CACP</i> de PV. En este caso, las funciones de usuario llamadas son <i>ToMakeVectors_Matrix</i> , <i>Identifier</i> , <i>InitializeCACP2</i> y <i>PP_Design2</i> .
SelectModel3	: Igual que anterior, pero para el <i>CACP</i> de TSC. En este caso, las funciones de usuario llamadas son <i>ToMakeVectors_Matrix</i> , <i>Identifier</i> , <i>InitializeCACP3</i> y <i>PP_Design3</i> .
signo	: Regresa un valor +amplitud o -amplitud, dependiendo si el argumento es positivo o negativo.
ToMakeVectors_Matrix	: Asigna memoria en forma dinámica para los vectores regresor ($\varphi = \text{Phi}$) y de parámetros ($\theta = \text{theta}$), así como para la matriz P de un modelo dado. Además, inicializa la matriz P como <i>Pinit</i> veces la matriz identidad. Usa la función de biblioteca <i>calloc</i> .
UpdateCACP1Panel	: Actualiza las gráficas mostradas en el panel de programación del <i>CACP</i> de TB. Usa las funciones de biblioteca <i>PlotStripChart</i> y <i>SetCtrlVal</i> .

- UpdateCACP2Panel** : Igual que la anterior, pero a las variables que aparecen en su interior se les ha puesto el sufijo numérico 2, para denotar que pertenecen al sistema de PV.
- UpdateCACP3Panel** : Igual que la anterior, pero a las variables que aparecen en su interior se les ha puesto el sufijo numérico 3, para denotar que pertenecen al sistema de TSC.
- UpDateCriterionValue** : Determina el valor numérico del criterio de evaluación (3.1), usado en la comparación de los *CACP's* con los *PID's* en las pruebas de regulación.
- UpdateGraph** : Actualiza la información en los indicadores numéricos y en las gráficas de despliegue continuo del panel principal y de los distintos paneles de control. Usa las funciones de biblioteca *SetCtrlVal* y *PlotStripChart*.

Funcionamiento del Programa

En la figura A5.5 se muestra la organización interna de las diferentes funciones del programa. Por simplicidad, en el diagrama se utiliza una combinación informal de nombres de subrutinas y de bloques operacionales. Estos últimos, a su vez, representan grupos de funciones relacionadas.

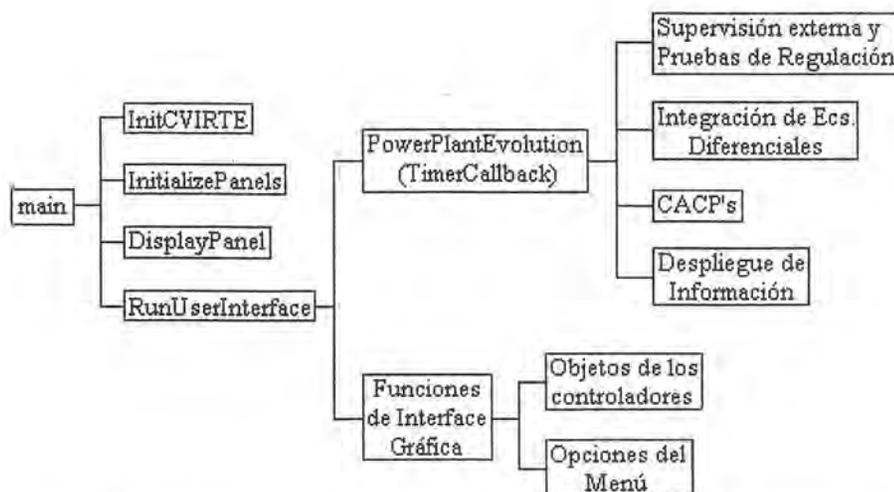


Figura A5.5. Organización de las subrutinas del programa *TermoCACP2.prj*.

Descripción de funciones y bloques operacionales

InitCVIRTE: Esta es una función de biblioteca que inicializa el CVI RTE (*CVI Run-Time Engine*) de *LabWindows®*.

InitializePanels: Esta es una función de usuario que inicializa los paneles usados en la interface gráfica. Para ello utiliza la función de biblioteca *LoadPanel*.

DisplayPanel: Es una función de biblioteca que se usa para desplegar en la pantalla el panel principal.

RunUserInterface: Función de biblioteca que corre la interface del usuario y envía eventos a las funciones de llamado.

PowerPlantEvolution: Es una función de llamado, activada por un temporizador (*timer*). Realiza en forma cíclica las acciones siguientes:

Supervisión Externa y Pruebas de Regulación: es un código programado que realiza la supervisión externa de los *CACP's* durante las pruebas de regulación y que, de hecho, calcula el decremento en rampa de la potencia generada, manipulando para ello el valor de la señal de demanda de carga *ldc*. Esta porción de código utiliza las variables *TCACP1*, *TCACP2* y *TCACP3* para determinar cual *CACP* desea supervisar el usuario (como si la supervisión fuera realizada externamente por la computadora de demanda de carga), o si no desea supervisar ninguno. Además, usa las variables *test1*, *test2*, *Tstart* y *Tend*, para determinar si el usuario desea simular la primer o segunda prueba de regulación o ninguna, así como su tiempo de inicio y de fin.

Integración de las ecuaciones diferenciales: Esta parte de código realiza la integración de las ecuaciones de la planta con el método de Runge-Kutta de cuarto orden y un paso de integración de 0.1 segundos. En esta parte del programa se llama a las siguientes funciones de usuario que contienen el modelo de cada sistema de la planta y de los controladores convencionales correspondientes:

- air2* : Caldera gases.
- shtr2* : Sobrecalentador.
- hpt2* : Turbina de alta presión.
- rhtr2* : Recalentador.
- ilpt2* : Turbina de intermedia y baja presión.
- genr2* : Generador.
- cndr2* : Condensador.
- boil2* : Caldera agua.
- fwtr2* : Agua de alimentación.
- c_sh2* : Prealimentación de las señales de presión en la primera etapa y de inclinación de los quemadores para el control de TSC. Incluye actuador.
- c_br2* : Controladores de flujo de Aire y Combustible. Incluye actuadores.
- c_tr2* : Control de la turbina (Unidades de control de carga y velocidad). Incluye actuador.
- c_rh2* : Controladores de Temperatura en el recalentador y de Recirculación de gases. Incluye actuadores.
- c_cn* : Controladores de Nivel en el deareador y de Flujo de condensado. Incluye actuadores.
- c_fw* : Controladores de Nivel en el domo y del Agua de alimentación. Incluye actuadores.
- c_ft* : Actuador del *CACP* de TB.
- c_fn* : Controlador de Presión en el hogar. Incluye actuador.

CACP's: En esta porción del programa se llama a las funciones de usuario `c_ftCACP1`, `c_brCACP2` y `c_shCACP3`, que realizan el algoritmo del *CACP* para el control de cada uno de los tres sistemas considerados. Para simular el proceso de muestreo se utilizan las variables `Ts1`, `Ts2`, `Ts3`, `Ns1`, `Ns2` y `Ns3`. Además, se llama a las funciones de usuario `UpdateCACP1Panel`, `UpdateCACP2Panel` y `UpdateCACP3Panel` que actualizan las gráficas mostradas en los paneles de programación de los *CACP's*.

Despliegue de Información: Se llama a la función de usuario `UpdateGraph`, que actualiza todas las gráficas y las cantidades mostradas en los indicadores del panel principal y de los diferentes paneles de control.

Funciones de Interface Gráfica: Este es un bloque de funciones de llamado que están asociadas con los objetos de la interface gráfica del programa. En particular se trata de las funciones relacionadas con:

Objetos de los controladores: Este grupo de subrutinas de llamado son usadas para modificar los diferentes parámetros de los controladores *PID's* y *CACP's*. Además, algunas se usan para mostrar y ocultar paneles asociados con los *CACP's* (de Parámetros del modelo, de Parámetros de respuesta y de Programación del *CACP*). Las funciones que están asociadas a los objetos gráficos (botones, conmutadores, etc.) de la interface y que se usan para modificar los parámetros de los controladores son:

En todos los controladores (*PID's* o *CACP's*).

<code>ChangeMode_</code>	: Se usa para cambiar el modo de operación del controlador.
<code>ModKp_</code>	: Para modificar la ganancia proporcional del algoritmo <i>PID</i> .
<code>ModTi_</code>	: Para modificar la constante de tiempo integral del algoritmo <i>PID</i> .
<code>ModTd_</code>	: Para modificar constante de tiempo derivativo del algoritmo <i>PID</i> .
<code>ModSP_</code>	: Para modificar el punto de ajuste o referencia del controlador.
<code>Up_</code>	: Para incrementar la señal de control en modo manual.
<code>Down_</code>	: Para disminuir la señal de control en modo manual.
	Sólo en <i>CACP's</i> .
<code>ModPinit_</code>	: Para reiniciar la matriz $\mathbf{P} = \mathbf{Pinit} * \mathbf{I}$.
<code>ModGain_</code>	: Para modificar la ganancia del regresor.
<code>ModFF_</code>	: Para modificar el factor de olvido (λ).
<code>ModNA_</code>	: Para modificar el número de parámetros del polinomio <i>A</i> .
<code>ModNB_</code>	: Para modificar el número de parámetros del polinomio <i>B</i> .
<code>ModNK_</code>	: Para modificar el retardo del modelo.
<code>ModDesResp_</code>	: Para establecer el tipo de respuesta deseada (1 ^{er} o 2 ^o orden).
<code>ModPole_</code>	: Para establecer la magnitud deseada del polo dominante.
<code>ModZ_</code>	: Para establecer la relación de amortiguamiento deseada.
<code>ModWn_</code>	: Para establecer la frecuencia natural no amortiguada. }
<code>ModWL_</code>	: Para modificar la frecuencia de corte inferior del filtro de datos.
<code>ModWU_</code>	: Para modificar la frecuencia de corte superior del filtro de datos.
<code>ModCEmax_</code>	: Para establecer el máximo error de control.
<code>ModMEMax_</code>	: Para establecer el máximo error de modelado.
<code>ModUmax_</code>	: Para establecer el valor máximo de la señal de control.
<code>ModUmin_</code>	: Para establecer el valor mínimo de la señal de control.
<code>Modh_</code>	: Para establecer el periodo de muestreo del <i>CACP</i> .
<code>ModNs_</code>	: Para modificar el valor de la variable <i>NS</i> .

ActSup_	: Para activar o desactivar el supervisor del <i>CACP</i> .
ModA1_	: Para inicializar el primer parámetro de <i>A</i> .
ModA2_	: Para inicializar el segundo parámetro de <i>A</i> .
ModA3_	: Para inicializar el tercer parámetro de <i>A</i> .
ModB1_	: Para inicializar el primer parámetro de <i>B</i> .
ModB2_	: Para inicializar el segundo parámetro de <i>B</i> .
ModB3_	: Para inicializar el tercer parámetro de <i>B</i> .

NOTA: para determinar a que sistema pertenece el controlador, se agrega el siguiente sufijo a las funciones recién mencionadas:

DrumLevelCtrl	: Para el controlador de nivel en el domo.
FeedWtrCtrl	: Para el controlador del agua de alimentación.
FeedPmpTurbCtrl	: Para el controlador de la turbo bomba.
PresCtrl	: Para el controlador de presión de vapor.
AirCtrl	: Para el controlador de flujo de aire.
FuelCtrl	: Para el controlador de flujo de combustible.
SupHtrTempCtrl	: Para el controlador de Temperatura en el sobrecalentador.
FncPressCtrl	: Para el controlador de presión en el hogar.
LoadCtrlUnit	: Para la unidad de control de carga.
SpeedCtrlUnit	: Para la unidad de control de velocidad de la turbina.
ReHtrTempCtrl	: Para el controlador de temperatura en el recalentador.
GasRecircCtrl	: Para el controlador del sistema de recirculación de gases.
DtrLevelCtrl	: Para el controlador de nivel en el deareador.
CondFlowCtrl	: Para el controlador de flujo de condensado.

Por ejemplo: La subrutina llamada *ModPole_PresCtrl* se usa para establecer la magnitud del polo dominante en el *CACP* del sistema de presión de vapor. O bien, la función: *ChangeMode_CondFlowCtrl* se usa para cambiar el modo de operación (manual o automático) del controlador *PID* de flujo de condensado

Además, se tienen las siguientes funciones asociadas con botones en los distintos paneles de los *CACP's*:

ShowCACP1Panel: Despliega u oculta el *Panel de Programación* del *CACP* del sistema de TB.

ShowCACP2Panel: Igual que anterior, pero para el *CACP* de PV.

ShowCACP3Panel: Igual que anterior, pero para el *CACP* de TSC.

HideRespPanel1: Oculta el panel de Parámetros de Respuesta del *CACP* de TB.

HideRespPanel2: Oculta el panel de Parámetros de Respuesta del *CACP* de PV.

HideRespPanel3: Oculta el panel de Parámetros de Respuesta del *CACP* de TSC.

HideParPanel1 : Oculta el panel de Parámetros del Modelo del *CACP* de TB.

HideParPanel2 : Oculta el panel de Parámetros del Modelo del *CACP* de PV.

HideParPanel3 : Oculta el panel de Parámetros del Modelo del *CACP* de TSC.

Adicionalmente, las subrutinas anteriores pueden utilizar una o varias de las siguientes funciones de biblioteca: *GetCtrlVal*, *SetInputMode*, *SetCtrlVal*, *DisplayPanel*, *HidePanel*, *MessagePopup*, *ClearID*, *free*, *MatrixMul*, y de las funciones de usuario *DiscretizePID*, *ToMakeVectors_Matrix* y *InitializeCACP*.

Opciones del Menú: Este grupo de subrutinas de llamado están asociadas con las opciones del menú:

- StartFun** : Inicia la simulación. Para ello, inicializa los filtros de datos a usar (con la función de usuario InitializeDataFilter), asigna dinámicamente memoria (con la función de biblioteca calloc) para los arreglos de datos a usar con los filtros y con la estimación paramétrica (yfd1, ufd1, yfd2, ufd2, yfd3 y ufd3). Genera la PRBS a usar por el CACP (con la función de biblioteca WhiteNoise y la de usuario signo). También, realiza la discretización de los controladores PID, para el modo 4 de operación del CACP, de acuerdo a las ganancias proporcionadas por el usuario (con la función de usuario DiscretizePID).
- PauseFun** : Para detener o continuar la simulación. Conmuta el estado lógico de la variable pause.
- HelpFun** : Muestra el cuadro de ayuda del programa. Utiliza la función de biblioteca MessagePopup.
- EvalFun** : Muestra u oculta el Panel de Evaluación del programa. Utiliza las funciones de biblioteca DisplayPanel y HidePanel.
- QuitFun** : Provoca el fin de la simulación y el término de la corrida del programa. Usa la función de biblioteca QuitUserInterface. Antes de esto, con la función de biblioteca free, libera la memoria asignada previamente para todos los arreglos de datos.
- StdySt100** : Inicializa los estados de la planta para la simulación al 100% de capacidad en estado estable (con la función de usuario InitSt100).
- Test1100** : Inicializa los estados de la planta para la simulación al 100% de capacidad (con la función de usuario InitSt100), Activa la variable test1, que sirve de bandera para indicar que se simulará la primer prueba de regulación. También, inicializa apropiadamente las variables Tend, Tadic y LDCo que permiten calcular el decremento en la señal de demanda de carga ldc. Finalmente, despliega el panel de *Parámetros de la Simulación*, que solicita el tiempo de inicio de la perturbación y permite solicitar supervisión externa para uno o varios CACP's durante la prueba realizada.
- StdySt77** : Inicializa los estados de la planta para la simulación al 77.5% de capacidad en estado estable (con la función de usuario InitSt77).
- Test277** : Inicializa los estados de la planta para la simulación al 77.5% de capacidad (con la función de usuario InitSt77), Activa la variable test2, que sirve de bandera para indicar que se simulará la segunda prueba de regulación. También, inicializa apropiadamente las variables Tend, Tadic y LDCo que permiten calcular el decremento en la señal de demanda de carga ldc. Finalmente, despliega el panel de *Parámetros de la Simulación*, que solicita el tiempo de inicio de la perturbación y permite solicitar supervisión externa para uno o varios CACP's durante la prueba realizada.
- HideEvalPanel** : Función relacionada con las opciones del menú. Oculta el panel de evaluación. Usa la función de biblioteca HidePanel.

- ModTstart : Función relacionada con las opciones del menú. Modifica la variable de tiempo de inicio de la perturbación Tstart. Utiliza la función de biblioteca GetCtrlVal.
- ModCACP1 : Función relacionada con las opciones del menú. Modifica la variable TCACP1, que indica si se desea supervisar externamente o no al CACP de TB. Utiliza la función de biblioteca GetCtrlVal.
- ModCACP2 : Igual que la anterior, pero para el CACP de PV. La variable a modificar es TCACP2.
- ModCACP3 : Igual que la anterior, pero para el CACP de TSC. La variable a modificar es TCACP3.
- HideParPanel : Función relacionada con el menú. Oculta el Panel de *Parámetros de la Simulación*. Usa la función de biblioteca HidePanel.

Para Sustituir un PID con un CACP en al Programa de Simulación

Para realizar la sustitución de un controlador PID convencional con un CACP pueden ser útiles las siguientes recomendaciones:

Sustitución en el archivo de recursos de la interface del usuario (*Termo2Tst.uir*).

1. Sustituir el controlador en la interface gráfica.
 - (a) Copiar (*Ctrl+c*) uno de los CACP's ya existentes.
 - (b) Crear un nuevo panel y en él pegar (*Ctrl+v*) el nuevo CACP.
 - (c) Cortar (*Ctrl+x*) el controlador PID que se desea sustituir, de su panel de control correspondiente, y pegarlo en el panel creado en (b).
 - (d) Editar los objetos del nuevo CACP. Primero, cambie la función de llamado correspondiente. Para ello, modifique todos los sufijos que identifican a que sistema pertenece el controlador (DrumLevelCtrl, FeedWtrCtrl, etc), sustituyéndolos por el sufijo del sistema en el que se desea sustituir el PID con el CACP. La excepción será el botón **Prog** del CACP, cuya función de llamado deberá tener un nombre tal como ShowCACP4Panel, para estar de acuerdo con los nombres usados con los anteriores CACP's. Después, cambie el identificador (*Constant Name*) del objeto que se esté editando: en particular, intercambie los identificadores de las teclas de flecha arriba/abajo, así como de los indicadores numérico visuales de las señales de referencia, salida y señal de control entre los controladores PID y CACP (para seguir usando el mismo código ya programado para dichos objetos). Los identificadores del selector de modo de operación y del botón de programación deben ser cambiados para adecuarlos al nuevo CACP (Usando, por ejemplo, los identificadores: MODE_CACP4 y PROG_CACP4, respectivamente. De hecho el número "4" puede ser cambiado por "5" ó "6", dependiendo de cuantos CACP's estén ya en uso).
 - (e) Cortar el CACP recién modificado en el panel nuevo y pegarlo en el espacio ocupado anteriormente por el PID que se eliminó en (c).
 - (f) Borre (corte o elimine) el controlador PID que se encuentra en el panel nuevo.
 - (g) En este punto ya debe ser posible ir del archivo de recursos (*Termo2Tst.uir*) al de código de programa (*TermoPl2.c*), y viceversa, con el botón derecho del

mouse, usando las opciones *View Control Callback* y *Find UI Object*, respectivamente. Compruebe que esto ocurre en las teclas de flecha arriba/abajo, en el selector de modo de operación y en el indicador numérico visual de la señal de referencia del nuevo *CACP*. Si no es así, busque y corrija los errores existentes.

- (h) Generar el código de la función de llamado asociada con el botón de programación del nuevo *CACP*. (Sugerencia: use el botón derecho del *mouse*).
 - (i) Grabar las modificaciones realizadas.
2. Agregar un Panel de Programación para el *CACP* recién creado.
- (a) Modificar el tamaño del panel creado anteriormente, para igualarlo al tamaño de un Panel de Programación típico. Si es necesario, utilice la opción *View>>Bring Panel to Front* del menú principal, para comparar el panel nuevo con un panel de programación ya existente (el Panel de Programación del *CACP1*, por ejemplo).
 - (b) Editar el panel nuevo, nombrándolo como "Panel de Programación del *CACP4*", por ejemplo. Además, cambie el identificador del panel (*Constant Name*) por *CACP4PANEL*, por ejemplo (puede usarse cualquier otro nombre).
 - (c) Copiar todos los objetos contenidos en un panel de programación de los ya existentes en el archivo (el Panel de Programación del *CACP1*, por ejemplo).
 - (d) Pegar en el nuevo Panel de Programación todos los objetos copiados en (c).
 - (e) Editar los objetos del nuevo Panel de Programación, cambiando la función de llamado correspondiente. Para ello, modifique todos los sufijos que identifican a que sistema pertenece el controlador (*DrumLevelCtrl*, *FeedWtrCtrl*, etc), sustituyéndolos por el sufijo del sistema en el que se desea sustituir el *PID* con el *CACP*. En aquellos casos en que resulte necesario, modifique los identificadores de los objetos. Use identificadores que sean significativos.
 - (f) Generar el código de las funciones de llamado asociadas con los diferentes objetos del nuevo Panel de Programación. Nótese que las ganancias del controlador *PID* (*Kp*, *Ti* y *Td*) ya tendrán sus funciones de llamado en el código fuente.
 - (g) Grabar las modificaciones realizadas.
3. Agregar un Panel de Parámetros de Respuesta Deseada para el *CACP* recién creado.
- (a) Crear un nuevo panel.
 - (b) Modificar el tamaño del panel creado para igualarlo al tamaño de un Panel de Parámetros de Respuesta típico (Parámetros de Respuesta 1, por ejemplo).
 - (c) Editar el panel nuevo, nombrándolo como "Parámetros de Respuesta 4", por ejemplo. Además, cambie el identificador del panel (*Constant Name*) por *RESP_CACP4*, por ejemplo (puede usarse cualquier otro nombre).
 - (d) Copiar todos los objetos contenidos en un panel de parámetros de respuesta de los ya existentes en el archivo (Parámetros de Respuesta 1, por ejemplo).
 - (e) Pegar en el nuevo Panel de parámetros de respuesta todos los objetos copiados en (d).
 - (f) Editar los objetos del nuevo Panel de parámetros de respuesta, cambiando la función de llamada correspondiente. Para ello, modifique todos los sufijos que

identifican a que sistema pertenece el controlador (DrumLevelCtrl, FeedWtrCtrl, etc), sustituyéndolos por el sufijo del sistema en el que se desea sustituir el *PID* con un *CACP*. En el botón 'OK' se debería llamar a la función HideRespPanel4, para estar de acuerdo con los otros *CACP's* implementados. En aquellos casos en que resulte necesario, modifique los identificadores de los objetos. Use identificadores que sean significativos.

- (g) Generar el código de las funciones de llamado asociadas con los diferentes objetos del nuevo Panel de parámetros de respuesta.
- (h) Grabar las modificaciones realizadas.

4. Agregar un nuevo Panel de Parámetros del Modelo para el *CACP* recién creado.

- (a) Crear un nuevo panel.
- (b) Modificar el tamaño del panel creado para igualarlo al tamaño de un Panel de Parámetros del modelo típico (Parámetros del modelo 1, por ejemplo).
- (c) Editar el panel nuevo, nombrándolo como "Parámetros del modelo 4", por ejemplo. Además, cambie el identificador del panel (*Constant Name*) por MOD_CACP4, por ejemplo (puede usarse cualquier otro nombre).
- (d) Copiar todos los objetos contenidos en un panel de parámetros del modelo de los ya existentes en el archivo (Parámetros del modelo 1, por ejemplo).
- (e) Pegar en el nuevo Panel de parámetros del modelo todos los objetos copiados en (d).
- (f) Editar los objetos del nuevo Panel de parámetros del modelo, cambiando la función de llamada correspondiente. Para ello, modifique todos los sufijos que identifican a que sistema pertenece el controlador (DrumLevelCtrl, FeedWtrCtrl, etc), sustituyéndolos por el sufijo del sistema en el que se desea sustituir el *PID* con un *CACP*. En el botón 'OK' se debería llamar a la función HideParPanel4, para estar de acuerdo con los otros *CACP's* implementados. En aquellos casos en que resulte necesario, modifique los identificadores de los objetos. Use identificadores que sean significativos.
- (g) Generar el código de las funciones de llamado asociadas con los diferentes objetos del nuevo Panel de parámetros del modelo.
- (h) Grabar las modificaciones realizadas.

Sustitución en el código fuente.

1. Inicialmente, cortar el código de las funciones de llamado añadidas anteriormente (que se encuentra al final de programa) y pegarlo en la porción que le corresponde al sistema en el que se realiza el cambio de controlador. Grabar las modificaciones hechas.
2. Definir las variables (*Handlers*) asociadas con los distintos paneles recién creados.
 - (a) En el archivo "*Vars.h*" definir las variables (*Handlers*) que servirán para manipular los paneles agregados anteriormente al archivo de recursos de la interfase de usuario. Pueden usarse los nombres: CACP4Handle, RespHandle4 y ThPHandle4, para los paneles de Programación, de parámetros de Respuesta y de parámetros del modelo, respectivamente. El número a usar ("4" en este caso) cambiará dependiendo el número de *CACP's* que ya estén en funcionamiento en la planta termoeléctrica. Grabar las modificaciones hechas.

3. Agregar el código necesario para poder desplegar los paneles nuevos.
 - (a) En la función de usuario InitializePanels, copiar el código usado para cargar los paneles de Programación, de parámetros de Respuesta deseada y de Parámetros del modelo de cualquier otro *CACP*, de los que ya se encuentran implementados.
 - (b) Anexar el código copiado en (a) al final de la función ya mencionada.
 - (c) Modificar el código anexado, sustituyendo por un número "4" los números que se encuentren en las instrucciones copiadas en (a). De esta forma el código será adecuado a los paneles recién añadidos al programa.
 - (d) Grabar las modificaciones realizadas.
4. Definir todas las variables necesarias para el funcionamiento del nuevo *CACP*.
 - (a) En el archivo "*CtrlVars.h*", agregar las siguientes variables:

```

//*****      VARIABLES DEL CACP 4      *****
double      *ufd4,*yfd4,Nfd4[2],Dfd4[3],wl4,wu4; // Info para Filtro discreto.
int          Evaluate4,sup4;
int          mode4,firstime4,validate4,flag4,ValStart4,ModSelEnd4,
            R14,R24,R34,Ts4,Ns4,k4,j4,imin4,m4,
            NA4,NB4,naMax4,nkMax4,NK4,
            NX4,nr4,n4,ord4; //NOTA: MAX_MODEL=33 (cte. definida arriba)
double      lambda4,Pinit4,gain4,*theta4,*P4,*Phi4,PRBS4[2000],Pant4,
            *thetaA4[MAX_MODEL],*PA4[MAX_MODEL],*PhiA4[MAX_MODEL],
            *data4[MAX_MODEL],SC4[MAX_MODEL],min4,
            CtrlErr4,ModErr4,CEmax4,MEmax4,
            h4,*u4,*y4,*uc4,y_sim4,e4[3],ua4,
            umin4,umax4,E4[3],Kp4,Ti4,Td4,
            sAm4,*R4,*S4,*T4,*Ao4,*A4,*B4,*Am4,*Ac4,M4[2],sM4,
            N4,*X4,*Mc4,*mc4,pole4,z4,wn4,
            J4,JE4,JU4;

```

o bien, cópiese el código de definición de las variables de alguno de los *CACP's* ya implementados y cámbiense todos los sufijos numéricos que indican a que controlador se asocia la variable. Es importante anexar las nuevas variables al final de la porción de código que pertenece al controlador que se está sustituyendo. Grabar las modificaciones hechas.

- (b) Agregar las mismas variables en el archivo "*CtrlVarsInic.h*". En este caso es necesario inicializar algunas variables (ver el código correspondiente a las variables de los otros *CACP's*). En lo que se refiere a los parámetros de respuesta (*pole4*, *wn4* y *z4*) y del filtro de datos (*wl4* y *wu4*), la inicialización debe ser hecha con base en el mejor conocimiento que se tenga del sistema que se quiere controlar. Después, cuando la sustitución haya concluido, estos valores pueden ser refinados. En el caso de las ganancias del algoritmo *PID* (*Kp4*, *Ti4* y *Td4*), sus valores pueden ser iguales a los usados originalmente por Usoro, siempre y cuando el periodo de muestreo sea elegido suficientemente pequeño (*h4* =1s, por ejemplo). Los valores de las restantes variables que necesitan inicialización, pueden ser copiados de los otros *CACP's* o de la tabla 4.2. Grabar las modificaciones hechas.

5. Anexar el código de operación de las funciones de llamado creadas anteriormente.
 - (a) Esto puede hacerse copiando el código que corresponde a las funciones de llamado de los otros *CACP's* ya implementados, para después pegarlo en las funciones de llamado equivalentes del nuevo *CACP*.
 - (b) En el código copiado, sustituir los nombres de las variables del *CACP* que se relacionan con las funciones de llamado, con los manipuladores (*Handlers*) de los paneles y con los identificadores (*Constant Names*) de los objetos gráficos que producen la llamada, adecuándolos a las variables e identificadores del nuevo *CACP*. No olvide modificar las subrutinas asociadas con las ganancias del algoritmo *PID*, con la señal de referencia, con el selector de modo de operación y con las teclas de flecha arriba/abajo. En especial, al modificar la subrutina asociada con el botón de programación, será necesario definir un variable entera: *showcontrol* apropiada (que se usa para determinar si el panel está oculto o desplegado), mientras que en la función asociada con el selector de respuesta deseada será necesario definir la variable: *Resp4* (para saber si se desea una respuesta de primer o segundo orden). Ambas variables se deberán definir en el archivo "*Vars.h*".
 - (c) En todo caso, es necesario cerciorarse de que las funciones de llamado modifican o usan a las variables apropiadas del nuevo *CACP* y que incluyen los identificadores correctos, de los objetos en la interfase que se asocian con dichas variables o parámetros. Esto último puede verificarse con el botón derecho del *mouse*, usando las opciones *Find UI Object* (para ir a la interfase gráfica) y *View Control Callback* (para volver al código fuente).
 - (d) Grabar las modificaciones hechas.

6. Eliminar la(s) variable(s) de estado relacionada(s) con el controlador *PID* que se quiere sustituir.
 - (a) Eliminar la definición de todas las variables de estado y sus derivadas respectivas en los archivos "*Termo2_e.h*", "*Termo2_d.h*", "*Main2_d.h*", "*Main2_e.h*", "*Termo2_y.h*", "*Term2_dy.h*". También, eliminar la inicialización respectiva de las funciones de usuario *InitSt77* e *InitSt100*.
 - (b) Disminuir el tamaño de las variables *ya[]* y *dya[]*, en el archivo "*Vars.h*", de tal forma que su tamaño final sea suficiente para los estados restantes.
 - (c) En el archivo *TermoPI2.c*, disminuir la cantidad de iteraciones que realizan los distintos ciclos *for* en los que se integran las ecuaciones diferenciales de la planta, de tal forma que el número de iteraciones final se iguale con el número de estados restantes en la planta.
 - (d) En el archivo de usuario correspondiente al *PID* eliminado, cancelar (no eliminar) todo el código relacionado con el mismo (medición de la salida, determinación de la señal de referencia y actualización de la señal de control, etc.).
 - (e) Cancelar la inicialización de todas las variables que se relacionan con el controlador *PID* eliminado y que ya no serán utilizadas.
 - (f) Grabar las modificaciones realizadas.

7. Crear archivo con el código del algoritmo *CACP* para sustituir al *PID* eliminado.
 - (a) Grabar con otro nombre el archivo de usuario correspondiente al *PID* eliminado. Use un nombre significativo (se puede agregar simplemente *CACP4* al final del primer nombre de dicho archivo, tal como fue hecho en los otros tres casos). Usar el mismo nombre para la función definida dentro del archivo.
 - (b) Dentro de la función de usuario renombrada anteriormente, eliminar todo el código, excepto el que se relaciona con la medición de la señal de salida y de determinación de la señal de referencia (que fue cancelado anteriormente). Habilite este código.
 - (c) Copiar todo el código, a partir de la adquisición de datos, de cualquier otro archivo *CACP* ya implementado (incluyendo las funciones *PP_design*, *SelectModel* e *InitializeCACCP* correspondientes).
 - (d) Anexar el código anterior en el archivo del *CACP* recién creado en (a).
 - (e) Tomando como modelo el código de adquisición de datos del *CACP* ya implementado, modificar el código correspondiente del nuevo *CACP* para asegurar que a la variable *y4[0]* le sea asignada la variable medida y a la variable *uc4[0]* le sea asignada la señal de referencia. Después de esto, elimine el código de adquisición de datos correspondiente al *CACP* ya implementado.
 - (f) Adecuar todo el código restante a las variables y funciones del nuevo *CACP*. Solo es necesario cambiar los sufijos numéricos que sirven para identificar a cuál *CACP* pertenecen las variables y funciones.
 - (g) Es importante asegurarse de que, en las parte marcadas como "Conversión D/A" de la señal de control, el valor de la variable *u4[0]* sea asignado a la variable de salida del controlador *PID* eliminado (que constituye la entrada del actuador del mismo). Si es necesario tómense como modelos los otros *CACP*'s ya implementados.
 - (h) En la parte superior del archivo, incluir el encabezado `#include "FuncGlob.h"`, para poder utilizar las funciones de usuario de estimación paramétrica y filtrado de datos, entre otras.
 - (i) Grabar las modificaciones hechas.
 - (j) Anexar el prototipo de la función del nuevo *CACP* en el archivo "Main2_1.h".
 - (k) Abrir el archivo "Func1.h" que contiene los prototipos de funciones locales del *CACP* de TB. Grabe el archivo con otro nombre ("Func4.h", por ejemplo).
 - (l) Cambie los nombre de las funciones que aparecen en los prototipos (sustituyendo el sufijo numérico que identifica a que controlador pertenecen las subrutinas: es decir, cambie el "1" por un "4").
 - (m) Agregar el encabezado `#include "Func4.h"` en el archivo del nuevo *CACP* y en el archivo *TermoPl2.c*.
 - (n) Grabar las modificaciones hechas.
8. Agregar llamada a la nueva subrutina de *CACP* en la función *PowerPlantEvolution*.
 - (a) En la función *PowerPlantEvolution*, del archivo *Termo2Pl.c*, agregar una llamada a la nueva subrutina *CACP*. Tome como modelo las llamadas a los *CACP*'s ya implementados.

- (b) Crear una función que actualice las gráficas en el Panel de Programación del nuevo CACP: UpdateCACP4Panel. Tome como modelo una de las funciones equivalentes ya implementadas, sólo es necesario adecuar el sufijo numérico de las variables utilizadas.
- (c) Agregar el prototipo de la función UpdateCACP4Panel en el archivo "Main2_1.h".
- (d) Es importante cerciorarse de que en la función UpdateCACP4Panel se usan las variables apropiadas del nuevo CACP y que se incluyen los identificadores correctos, de los objetos en la interfase que se asocian con dichas variables o parámetros.
- (e) Finalmente, debe asegurarse de que en las funciones de usuario StartFun, InitSt100 e InitSt77, se asigna memoria para los arreglos de datos a usarse con el nuevo CACP y que se inicializan apropiadamente sus variables. Tómese como ejemplo el código existente

9. Depurar el programa resultante.

III. Reproducción de las pruebas realizadas al CACP

Prueba 1: de distintos modelos de primer y segundo orden

Para reproducir esta prueba:

- (1) Elija *Opciones* >> 100% >> *Estado Estable*, en el menú principal.
- (2) Al inicio de la corrida todos los CACP's están en modo *PID*. Oprima el botón del panel donde se encuentra el controlador que se desea probar (*Panel Agua Alim.*, *Panel Caldera* o *Panel SobreC.*).
- (3) Oprima el botón de programación (*Prog*) del CACP a probar. Aparecerá el Panel de Programación del mismo.
- (4) Modifique las características del modelo a utilizar (el número de parámetros: *NA*, *NB* y el retardo supuesto del sistema: *NK*).
- (5) Elija el tipo de respuesta deseada: de primer o segundo orden (recuerde que para elegir una respuesta de segundo orden es necesario que $NA \geq 2$), aparecerá el panel de Parámetros de Respuesta.
- (6) Si es necesario, modifique los parámetros de la respuesta solicitada y del filtro de datos. Para ocultar este panel oprima el botón *OK*.
- (7) Finalice la programación del CACP oprimiendo de nuevo el botón *Prog*.
- (8) Solicite el modo 2 (*STR*) de operación del CACP. Aparecerá el panel de Parámetros del Modelo.
- (9) Proporcione la inicialización de los parámetros del modelo a utilizar (cero en todos los parámetros). Para ocultar este panel oprima el botón *OK*.
- (10) Inicie la simulación, eligiendo la opción *Inicio* del menú principal.
- (11) [opcional] Si desea observar la evolución de la salida del sistema controlado, de la señal de control y de los parámetros estimados oprima de nuevo el botón *Prog*.
- (12) [Opcional] En cualquier momento se puede detener la simulación eligiendo la opción *Pausa* del menú principal. Para continuar, basta con oprimir nuevamente la misma opción.

- (13) Durante el periodo de adaptación inicial (*pai*, de 60 muestreos) el sistema es controlado utilizando el algoritmo *PID* discreto, mientras se obtiene un modelo adecuado para aproximar al sistema real. Después de esto, puede intentarse un cambio en el valor de referencia para examinar el comportamiento del controlador recién adaptado.
- (14) Para finalizar la simulación oprima Salir en el menú principal.

Prueba 2: de ubicación de polos

Para reproducir esta prueba:

- (1) Elija Opciones>>100%>>Estado Estable, en el menú principal.
- (2) Oprima el botón del panel donde se encuentra el controlador que se desea probar (*Panel Agua Alim.*, *Panel Caldera* o *Panel SobreC.*).
- (3) Oprima el botón de programación (*Prog*) del *CACP* a probar. Aparecerá el Panel de Programación del mismo.
- (4) Elija el tipo de respuesta deseada: de primer o segundo orden (recuerde que para elegir una respuesta de segundo orden es necesario que $NA \geq 2$), aparecerá el panel de Parámetros de Respuesta.
- (5) Modifique los parámetros de la respuesta solicitada y del filtro de datos. Para ocultar este panel oprima el botón OK.
- (6) Finalice la programación del *CACP* oprimiendo de nuevo el botón *Prog*.
- (7) Solicite el modo 2 (*STR*) de operación del *CACP*. Aparecerá el panel de Parámetros del Modelo.
- (8) Proporcione la inicialización de los parámetros del modelo a utilizar (cero en todos los parámetros). Para ocultar este panel oprima el botón OK.
- (9) Inicie la simulación, eligiendo la opción Inicio del menú principal.
- (10) Después del *pai*, realice un cambio en el valor de referencia para examinar el comportamiento del controlador recién adaptado.
- (11) Para finalizar la simulación oprima Salir en el menú principal.

Prueba 3: de elección de filtros de datos

Para reproducir esta prueba:

- (1) Elija Opciones>>100%>>Estado Estable, en el menú principal.
- (2) Oprima el botón del panel donde se encuentra el controlador que se desea probar (*Panel Agua Alim.*, *Panel Caldera* o *Panel SobreC.*).
- (3) Oprima el botón de programación (*Prog*) del *CACP* a probar. Aparecerá el Panel de Programación del mismo.
- (4) Elija una respuesta deseada de primer orden, aparecerá el panel de Parámetros de Respuesta.
- (5) Modifique los parámetros del filtro de datos. Para ocultar este panel oprima el botón OK.
- (6) Finalice la programación del *CACP* oprimiendo de nuevo el botón *Prog*.
- (7) Solicite el modo 2 (*STR*) de operación del *CACP*. Aparecerá el panel de Parámetros del Modelo.
- (8) Proporcione la inicialización de los parámetros del modelo a utilizar (cero en todos los parámetros). Para ocultar este panel oprima el botón OK.
- (9) Inicie la simulación, eligiendo la opción Inicio del menú principal.

- (10) Después del *pai*, puede intentarse un cambio en el valor de referencia para examinar el comportamiento del controlador recién adaptado.
- (11) Para finalizar la simulación oprima **Salir** en el menú principal.

Prueba 4: de Regulación

Para reproducir esta prueba:

- (1) Elija en el menú principal *Opciones>>100%>>1er Prueba de Regulación* u *Opciones>>77.5%>>2a Prueba de Regulación*, según se desee. Aparecerá el panel de Parámetros de Simulación de la prueba, mostrado en la figura A5.6.

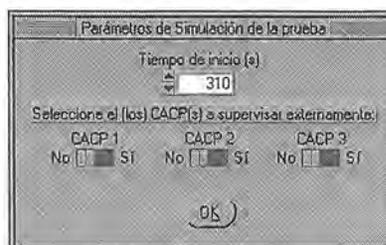


Figura A5.6. Panel de Parámetros de la simulación.

- (2) Proporcione el tiempo de inicio -medido en segundos- de la perturbación (decremento en rampa de la potencia generada). Seleccione el (los) *CACP(s)* que se quiere sea(n) supervisado(s) externamente por la computadora de demanda de carga durante la perturbación. Si no se desea utilizar supervisión externa, entonces sólo proporcione tiempo de inicio de la perturbación. Para ocultar este panel oprima el botón **OK**.
- (3) Oprima el botón del panel donde se encuentra el controlador que se desea probar (*Panel Agua Alim.*, *Panel Caldera* o *Panel SobreC.*).
- (4) Solicite el modo 2 (*STR*) de operación del *CACP* a probar. Aparecerá el panel de Parámetros del Modelo.
- (5) Proporcione la inicialización de los parámetros del modelo a utilizar. En caso de que la inicialización dada sea diferente de cero, se recomienda dar un valor pequeño a *Pinit* (0.200 ó 0.100), lo cual indica que se tiene poca incertidumbre acerca de la magnitud de dichos parámetros. De esta manera se logra que el modelo no cambie bruscamente durante el *pai* (el cual se reduce a 20 muestreos en este caso). Para ocultar este panel oprima el botón **OK**.
- (6) Si desea probar simultáneamente otro *CACP*, oculte el panel de control activo y vuelva al inciso (3).
- (7) Inicie la simulación, eligiendo la opción **Inicio** del menú principal.
- (8) Detenga la simulación, oprimiendo **Pausa**, una vez concluido el periodo de prueba (800 muestreos para la primer prueba y 1500 para la segunda).
- (9) Para finalizar la simulación oprima **Salir** en el menú principal.

Prueba 5: de operación en modo 1 (MS)

Para reproducir esta prueba:

- (1) Elija *Opciones*>>100%>>Estado Estable, en el menú principal.
- (2) Oprima el botón del panel donde se encuentra el controlador que se desea probar (*Panel Agua Alim.*, *Panel Caldera* o *Panel SobreC.*).
- (3) Solicite el modo 1 (MS) de operación del CACP.
- (4) Inicie la simulación, eligiendo la opción **Inicio** del menú principal.
- (5) Después del periodo de selección, puede intentarse un cambio en el valor de referencia para examinar el comportamiento del controlador con el modelo seleccionado.
- (6) Para finalizar la simulación oprima **Salir** en el menú principal.

También, se pueden seguir la siguientes instrucciones:

- (1) Elija *Opciones*>>100%>>1er Prueba de Regulación, en el menú principal. Aparecerá el panel de Parámetros de Simulación de la prueba.
- (2) Proporcione el tiempo de inicio de la perturbación (310s, por ejemplo). Oprima el botón **OK**.
- (3) Oprima el botón del panel donde se encuentra el controlador que se desea probar (*Panel Agua Alim.*, *Panel Caldera* o *Panel SobreC.*).
- (4) Solicite el modo 1 (MS) de operación del CACP.¹
- (5) Inicie la simulación, eligiendo la opción **Inicio** del menú principal.
- (6) Detenga la simulación después de transcurrido el periodo de prueba (800 muestreos después del inicio de la perturbación).
- (7) Para finalizar la simulación oprima **Salir** en el menú principal.

Prueba 6: de operación con NB>NA.

Para reproducir esta prueba:

- (1) Elija *Opciones*>>100%>>Estado Estable, en el menú principal.
- (2) Oprima el botón del panel donde se encuentra el controlador que se desea probar (*Panel Agua Alim.*, *Panel Caldera* o *Panel SobreC.*).
- (3) Oprima el botón de programación (*Prog*) del CACP a probar. Aparecerá el Panel de Programación del mismo.
- (4) Modifique las características del modelo a utilizar (el número de parámetros: NA, NB y el retardo supuesto del sistema: NK).
- (5) Puede ser necesario modificar la inicialización de la matriz P, la ganancia del regresor o el factor de olvido (Ver descripción de esta prueba en Capítulo 3).
- (6) Finalice la programación del CACP oprimiendo de nuevo el botón *Prog*.
- (7) Solicite el modo 2 (STR) de operación del CACP. Aparecerá el panel de Parámetros del Modelo.
- (8) Proporcione la inicialización de los parámetros del modelo a utilizar (cero en todos los parámetros). Para ocultar este panel oprima el botón **OK**.
- (9) Inicie la simulación, eligiendo la opción **Inicio** del menú principal.
- (10) Después del *pai*, puede intentarse un cambio en el valor de referencia para examinar el comportamiento del controlador recién adaptado. También en este momento puede

¹ Recuérdese que el CACP no es capaz de elegir un modelo adecuado en el sistema de PV.

ser necesaria alguna de las siguientes acciones: reinicializar matriz **P**, modificar la ganancia del regresor o, en algunos casos, desactivar la autosintonización, solicitando el modo 3 (*CLG*) de operación.

(11) Para finalizar la simulación oprima **S**alir en el menú principal.

IV. Referencias Bibliográficas

- [Usoro, 1977] Usoro, P.B., *Modeling and Simulation of a Drum Boiler-Turbine Power Plant Under Emergency State Control*, M.Sc. Thesis, MIT, 1977, United States of America.