ANEXO B

LA HERRAMIENTA OPTIMAX

B.1.- Objetivo del anexo.

En este anexo se aproximará al lector a la herramienta OPTIMAX un poco más allá de los límites básicos de conocimiento que se establecieron en el capítulo 4 del presente documento. Comenzaremos con una descripción del programa y presentaremos un manual para que el futuro usuario de OPTIMAX disponga de un guión orientativo para su trabajo, esto es, fundamentalmente, para que esté capacitado para proporcionar los datos necesarios para el control multivariable a la herramienta OPTIMAX. Dentro del ámbito de este proyecto se recomienda encarecidamente la lectura de los apartados denominados Ejemplo y manual de usuario.

Para extender la información referente al programa OPTIMAX se remite al lector a la consulta del manual del que disponemos en el laboratorio L1.

B.2.- Descripción del programa.

B.2.1.- Descripción de la aplicación

La aplicación denominada: OPTIMAX consiste en un paquete software desarrollado bajo el lenguaje de programación Visual C++ (versión 6.0). El programa, se basa en una aplicación MDI (Multiple Document Interface), donde se administran múltiples documentos que son gestionados de manera continua en ventanas individuales del área de trabajo de la ventana principal.

En concreto, la aplicación MDI, la constituyen tres plantillas, una de ellas es la principal y las otras dos son hijas de la anterior. Las tres plantillas están asociadas al mismo documento: "CSIMGPCWApp" y en cada una de ellas se implementa el marco o ventana y la vista.

La plantilla principal está formada por las variables de procesos ("CChildFrame"), y las plantillas hijas están constituidas a su vez por los actuadores ("CChFramAct") y el centro de control ("CChFramTabla"). La plantilla de las señales (variable de procesos y actuadores), representan ventanas gráficas donde se muestran la evolución temporal de las señales pertinentes, así como los valores de las restricciones y referencias. En la plantilla del centro de control, se muestran los valores actuales de las señales, variables de proceso, actuadores, referencias, restricciones e históricos de las señales. Desde esta ventana es posible modificar una serie de parámetros sin necesidad de parar el experimento, por ejemplo, el valor de las restricciones, tanto de salida como de entrada y su amplitud. También es posible cambiar el valor del esfuerzo de control (lambda).

La aplicación es multihilo, es decir, contiene varías vías de ejecución paralela, en concreto dos, un hilo referente a la aplicación en sí y un segundo hilo que corresponde al proceso del controlador, que se ejecuta cada tiempo de muestreo. Es necesario establecer un régimen de prioridades para los diferentes hilos del programa. En nuestra aplicación, el hilo con mayor prioridad corresponde al proceso donde se ha implementado el controlador basado en GPC ("mainsimulador").

B.2.2.- Controlador

El controlador implantado en OPTIMAX sigue la teoría del Control Predictivo Generalizado (GPC) al que hemos añadido un Predictor de Smith junto con un filtro paso bajo. Estas estrategias de control se describirán brevemente en los siguientes apartados.

B.2.2.1.- Control Predictivo Generalizado (GPC).

El Control Predictivo Generalizado GPC se ha convertido en uno de los métodos más populares en el ámbito del Control Predictivo tanto en el mundo industrial como en el académico. Se ha empleado con éxito en numerosas aplicaciones industriales, mostrando buenas prestaciones a la vez que un cierto grado de robustez respecto a sobreparametrización o retardos mal conocidos. Puede resolver muchos problemas de control diferentes para un amplio campo de procesos con un número razonable de variables de diseño.

La idea básica del GPC es calcular una secuencia de futuras acciones de control de forma que se minimice una función de coste multipaso. El índice a minimizar es una función cuadrática que representa dos efectos distintos: por un lado la distancia entre la salida predicha para el sistema y una cierta trayectoria de referencia hasta el horizonte de predicción y por otro el esfuerzo de control que es necesario para obtener dicha salida, esto no es más que una medida de cuánto deben variar las acciones de control respecto de su valor actual para llevar el sistema a la referencia. Para profundizar en el concepto de GPC más allá de estas notas se recomienda la bibliografía ([8]).

B.2.2.2.- Formulación del Control Predictivo Generalizado.

La mayoría de los procesos de una sola entrada y una sola salida (single-imput single-output, SISO), al ser considerados alrededor de un determinado punto de trabajo (set point, SP) y ser linealizados, pueden ser descritos de la siguiente forma:

$$A(z^{-1})\, y(t) = z^{-d}\, B(z^{-1})\, u(t-1) + C(z^{-1})\, e(t)$$

Donde: u(t) es la señal de control que se aplica instantáneamente al sistema.

y(t) se corresponde con la señal de salida del proceso.

e(t) es un ruido blanco de media nula.

A, B y C son los siguientes polinomios en el operador de desplazamiento hacia atrás z^{-1}:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{na} z^{-na}$$
$$B(z^{-1}) = b_o + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{nb} z^{-nb}$$
$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \ldots + c_{nc} z^{-nc} \qquad \text{(Ec.B.1)}$$

Donde la variable "d "representa el tiempo muerto del sistema. Esta variable representa el número de períodos de muestreo (tiempo) en que la aplicación de una señal de control se ve reflejado en una variación de la salida y(t).

El modelo caracterizado por estos polinomios es conocido como Autorregresivo de Media Móvil (Controller Auto-Regressive Moving-Average CARMA). En muchas aplicaciones industriales en las que las perturbaciones son no-estacionarias resulta más conveniente el uso de un modelo CARMA integrado, que viene descrito por:

$$A(z^{-1})\, y(t) = B(z^{-1})\, z^{-d}\, u(t-1) + C(z^{-1}) \frac{e(t)}{\Delta} \quad \text{con} \quad \Delta = 1 - z^{-1} \qquad \text{(Ec. B.2)}$$

Por simplicidad, a partir de ahora, el polinomio C se va a tomar igual a 1. Nótese que en el caso de que C^{-1} pueda ser truncado se puede absorber en A y B.

El algoritmo de Control Predictivo Generalizado consiste en aplicar una secuencia de señales de control que minimice una función o índice de costes que se toma con la siguiente estructura general:

$$J(N_1, N_2, N_u) = \sum_{j=N_u}^{N2} \delta(j)\left[\hat{y}(t+j|t) - w(t+j)\right]^2 + \sum_{j=1}^{N_u} \lambda(j)\left[\Delta u(t+j-1)\right]^2 \qquad \text{(Ec. B.3)}$$

Donde: $(t+j|t)$ representará una predicción del valor de la salida. Esta predicción se realiza en el instante t con los datos reales conocidos hasta ese momento y para una salida que tendrá lugar j períodos de muestreo después del instante t.

N1 y N2 son los horizontes mínimo y máximo de coste.

N_u es el horizonte de control.

$\delta(j)$ y $\lambda(j)$ son las secuencias de ponderación. En muchas situaciones se considera $\delta(j)$ igual a 1 y $\lambda(j)$ constante.

$w(t + j)$ es la futura trayectoria de referencia y que se supone conocida en cada instante.

El objetivo será entonces determinar la futura secuencia de control u(t), u(t+l),. . . de tal manera que la salida futura del proceso y(t +j) permanezca lo más próxima a w(t + j) y cumpliendo las condiciones de coste. Esto se logra minimizando J (N1, N_2, N_u).

B.2.2.3.- Predicción óptima.

Con la intención de minimizar la función de coste o índice, se obtendrá previamente la predicción óptima de la salida y (t + j) para j $\geq N_1$ y j $\leq N_2$. Considérese la siguiente ecuación diofántica:

$$1 = E_j(z^{-1})\Delta A + z^{-j} F_j(z^{-1}) \quad \Rightarrow \qquad \text{(Ec. B.4)}$$

$$\Rightarrow \quad 1 = E_j(z^{-1})\tilde{A} + z^{-j} F_j(z^{-1}) \quad \text{con } \tilde{A} = A(1 - z^{-1})$$

Los polinomios Ej y Fj están únicamente definidos con grados j - 1 y na respectivamente. Se pueden obtener dividiendo 1 entre $\tilde{A}$ (z^{-1}) hasta que el resto pueda ser factorizado como z^{-j} Fj (z^{-1}). El cociente de la división es entonces el polinomio $E_j(z^{-1})$.

Si se multiplica la ecuación A.2 por $Ej(z^{-1})\, z^{j} A$

$$\tilde{A}(z^{-1}) \cdot E_j(z^{-1}) \cdot y(t+j) = E_j(z^{-1}) \cdot B(z^{-1}) \cdot \Delta u(t+j-d-1) + E_j(z^{-1}) \cdot e(t+j) \quad \text{Ec.B.5)}$$

teniendo en cuenta Ec. A.4, la ecuación A.5 queda:

$$\left[1 - z^{-j} \cdot F_j(z^{-1})\right] \cdot y(t+j) = E_j(z^{-1}) \cdot B(z^{-1}) \cdot \Delta u(t+j-d-1) + E_j(z^{-1}) \cdot e(t+j)$$

la cual se puede escribir como:

$$y(t+j) = F_j \cdot y(t) + E_j(z^{-1}) \cdot B(z^{-1}) \cdot \Delta u(t+j-d-1) + E_j(z^{-1}) \cdot e(t+j) \qquad \text{(Ec.B.6)}$$

Al ser el grado del polinomio Ej (z-1) igual a j - 1 los términos del ruido en la ecuación A.6 están todos en el futuro. La mejor predicción de y(t+j) será por consiguiente:

$$G(t+j|t) = G_j(z^{-1}) \cdot \Delta u(t+j-d-1) + F_j(z^{-1}) \cdot y(t)$$

donde:

$$G_j(z^{-1}) = E_j(z^{-1}) \cdot B(z^{-1})$$

Resulta simple demostrar que los polinomios E_j y F_j se pueden obtener recursivamente, de forma que los nuevos valores en el paso j+1 (E_{j+1} y F_{j+1}) sean función de los del paso j. A continuación se muestra una demostración simple de la recursividad de la ecuación diofántica. Existen otras formulaciones del GPC que no están basadas en la recursividad de esta ecuación.

Considérense que los polinomios E_j y F_j se han obtenido dividiendo 1 entre $\tilde{A}(z^{-1})$ hasta que el resto haya sido factorizado como $z^{-j} F_j(z^{-1})$.

Con:

$$F_j(z^{-1}) = f_{j,0} + f_{j,1} \cdot z^{-1} + K + f_{j,na} \cdot z^{-na}$$

$$E_j(z^{-1}) = e_{j,0} + e_{j,1} \cdot z^{-1} + K + e_{j,j-1} \cdot z^{-(j-1)}$$

Supóngase que se utiliza el mismo procedimiento para obtener E_{j+1} y F_{j+1}, es decir, dividir 1 entre $\tilde{A}(z^{-1})$ hasta que el resto se pueda factorizar como $z^{-(j+1)} F_{j+1}(z^{-1})$ con:

$$F_{j+1}(z^{-1}) = f_{j+1,0} + f_{j+1,1} \cdot z^{-1} + K + f_{j+1,na} \cdot z^{-na}$$

Está claro que solamente es necesario dar un paso más en la división para obtener los polinomios E_{j+1} y F_{j+1}. Al ser E_{j+1} el nuevo cociente de la división, será igual al cociente que había hasta el momento (E_j) más un nuevo término, que será el $f_{j,0}$ pues el divisor ($\tilde{A}$) es mónico. Por tanto:

$$E_{j+1}(z^{-1}) = E_j(z^{-1}) + e_{j+1,1} \cdot z^{-1} \qquad \text{con} \quad e_{j+1,1} = f_{j,0}$$

Teniendo en cuenta que el nuevo resto será el resto anterior menos el producto del cociente por el divisor, los coeficientes del polinomio F_{j+1} se pueden expresar como:

$$f_{j+1,j} = f_{j,i+1} - f_{j,0} \cdot \tilde{a}_{j+1} \qquad \text{con } i = 0, \ldots, na$$

En resumen, la forma de obtener los polinomios E_j y F_j es la siguiente:

1. Comenzar con $E_j = 1$, $F_1 = z(1 - \tilde{A})$
2. Ir añadiendo nuevos términos a E_j con $e_{j+1,j} = f_{j,0}$
3. Calcular $f_{j+1,i} = f_{j,i+1}, - f_{j,0}\tilde{a}_{i+1}$ para i = 0, ... na (siendo $f_{j,na+1} = 0$).

El polinomio G_{j+1} puede ser obtenido recursivamente como sigue:

$$G_{j+1} = E_{j+1} \cdot B = (E_{j+1} + f_{j+0} \cdot z^{-j}) \cdot B = G_j + f_{j+0} \cdot z^{-j} \cdot B$$

Es decir, los primeros j coeficientes de G_{j+1} serán idénticos a los de G_j mientras que el resto viene dado por:

$$g_{j+1,j+i} = g_{j+j,j+i} + f_{j,0} \cdot b_i \qquad \text{para } i = 0, \ldots, nb$$

Como hemos fijado anteriormente, para implementar el GPC es necesario obtener el conjunto de señales de control u(t), u(t+1), . . .,u(t + N) que minimizan el índice de la ecuación A.3. Al tener el proceso un retardo de d períodos de muestreo, la salida sólo se verá influenciada por la señal de control u(t) después del instante d + 1. Los valores N_1, N_2 y N_u que marcan los horizontes pueden ser definidos como $N_1 = d + 1$, $N_2 = d + N$ y $N_u = N$. No tiene sentido hacer $N_1 < d + 1$ ya que los términos de ec. B.3 sólo dependerán de las señales de control pasadas.

Por otro lado, tomando $N_1 > d + 1$ los primeros puntos de la secuencia de salida, que serán los mejor estimados, no se tendrán en cuenta.

El conjunto de las j predicciones óptimas:

$$\hat{y}(t+d+1|t) = G_{d+1} \cdot \Delta u(t) + F_{d+1} \cdot y(t)$$

$$\hat{y}(t+d+2|t) = G_{d+2} \cdot \Delta u(t+1) + F_{d+2} \cdot y(t)$$

$$\vdots$$

$$\hat{y}(t+d+N|t) = G_{d+N} \cdot \Delta u(t+N-1) + F_{d+N} \cdot y(t)$$

Puede ser escrito en forma matricial como:

$$y = G \cdot u + F(z^{-1}) \cdot y(t) + G'(z^{-1}) \cdot \Delta u(t-1) \qquad \text{(Ec. B.7)}$$

donde:

$$y = \begin{bmatrix} y(t+d+1|t) \\ y(t+d+2|t) \\ \vdots \\ y(t+d+N|t) \end{bmatrix} \qquad u = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix}$$

$$G = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix}$$

$$G'(z^{-1}) = \begin{bmatrix} z\left(G_{d+1}(z^{-1}) - g_0\right) \\ z\left(G_{d+2}(z^{-1}) - g_0 - g_1 \cdot z^{-1}\right) \\ \vdots \\ z\left(G_{d+N}(z^{-1}) - g_0 - g_1 \cdot z^{-1} - \cdots - g_{N-1} \cdot z^{-(N-1)}\right) \end{bmatrix}$$

$$F(z^{-1}) = \begin{bmatrix} F_{d+1}(z^{-1}) \\ F_{d+2}(z^{-1}) \\ \vdots \\ F_{d+N}(z^{-1}) \end{bmatrix}$$

Al depender los últimos términos de la ecuación B.7 sólo del pasado, pueden agruparse en f, dando lugar a:

$$Y = G \cdot u + f \qquad \text{(Ec. B.8)}$$

B.2.2.4.- Obtención de la ley de control.

Entonces la ecuación del índice podrá escribirse como:

$$J = (G \cdot u + f - w)^T \cdot (G \cdot u + f - w) + \lambda \cdot u^T \cdot u \qquad \text{(Ec. B.9)}$$

donde:

$$w = [w(t+d+1) \;\; w(t+d+2) \;\; \ldots \;\; w(t+d+N)]^T \qquad \text{(Ec. B.10)}$$

la ecuación A.9 se puede poner como:

$$J = \frac{1}{2} u^T H u + bu + f_0 \qquad \text{(Ec. B.11)}$$

donde:

$$H = 2 \cdot (G^T \cdot G + \lambda \cdot I)$$
$$B = 2 \cdot (f - w)^T \cdot G$$
$$F_0 = (f - w)^T \cdot (f - w)$$

El mínimo de J, siempre que no existan restricciones en la señal de control, puede ser calculado igualando a cero el gradiente de J, lo cual conduce a:

$$U = - H^{-1} \cdot b^T \qquad \text{(Ec. B.12)}$$

Debido al uso de la estrategia deslizante, solo aplicaremos el primer elemento del vector u, repitiendo de nuevo el mismo procedimiento al siguiente instante de muestreo. La solución propuesta involucra la inversión (o al menos la triangularización) de una matriz de dimensión N x N, lo cual conlleva a una gran carga de cálculo. El conocido concepto de horizonte de control se emplea con la finalidad de reducir la cantidad de cálculo, asumiendo que las señales de control permanecerán en un valor constante a partir del intervalo $N_u < N$. Por tanto la dimensión de la matriz que hay que invertir queda reducida a N_u x N_u , quedando la carga de cálculo reducida (en el caso límite de $N_u = 1$, se reduce al caso escalar) aunque restringiendo el carácter óptimo de la solución encontrada.

B.2.2.5.- El Predictor de Smith.

En la figura B.1 se muestra el esquema clásico de control en bucle cerrado de un proceso. El problema fundamental que presenta la regulación de sistemas con retardo, es que el bucle cerrado mantiene el mismo tiempo muerto que tenga el sistema en bucle abierto. De esta manera, cualquier variación en la referencia no modificará el error que emplea el controlador hasta que transcurre un tiempo igual al tiempo muerto.

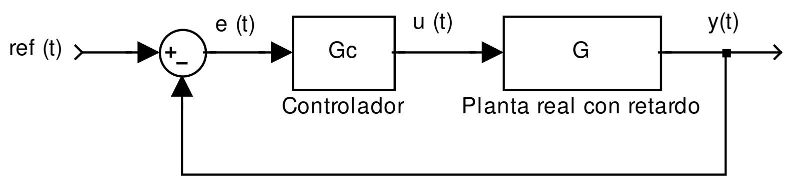


Figura B. 1 - Esquema de control en bucle cerrado.

Si Tm es el tiempo muerto de la planta, la idea del Predictor de Smith consiste en utilizar en la realimentación la señal **y(t+Tm)** en lugar de y(t). Para el cálculo de esta señal “retardada” se empleará un modelo; de esta manera el retraso saldría del bucle y los problemas que introduce en el bucle cerrado (malas características dinámicas, inestabilidades) dejarían de existir.

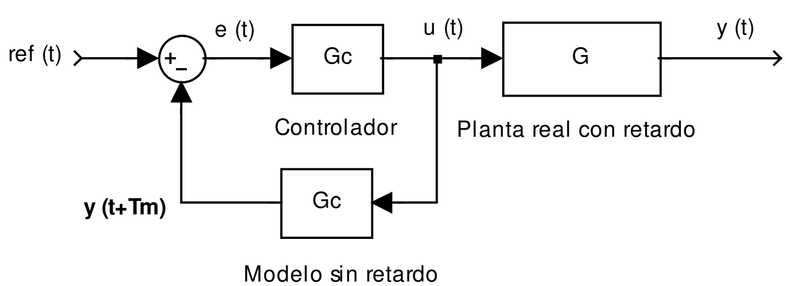


Figura B. 2 - Realimentación de la señal y(t+Tm).

La figura B.2 representa el efecto de realimentar la señal "retardada", es fácil deducir que con esta disposición, estamos controlando la salida de la planta en bucle abierto, por lo que, ante cualquier ruido o cambio en la referencia, los errores se verían amplificados. Para compensar el efecto de estas perturbaciones, se compara la señal de salida que proporciona el sistema con la que daría el modelo y se realimenta la señal del error, con el fin de corregir las diferencias entre ambas.

El esquema final que resulta es el típico del Predictor de Smith:

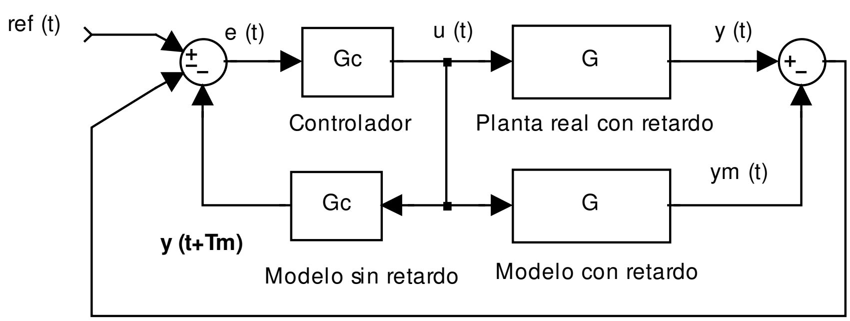


Figura B. 3 - Esquema del Predictor de Smith.

B.3.2.6.- Filtro paso bajo.

El filtro paso bajo puede ser definido como:

$$F(z) = \left[\frac{(1-\beta)z}{z-\beta} \frac{z-\beta\gamma}{1-\beta\gamma} \right]^m \qquad \text{(Ec. B.13)}$$

donde el cero es introducido para atenuar las altas frecuencias de la señal de salida y β y m se pueden cambiar para alcanzar una forma característica del filtro paso bajo.

B.3.2.7.- Esquema general de control implementado en OPTIMAX.

De forma sencilla el esquema de control que utiliza OPTIMAX combinando las técnicas descritas con anterioridad (GPC, Predictor de Smith y filtro paso) bajo puede representarse como:

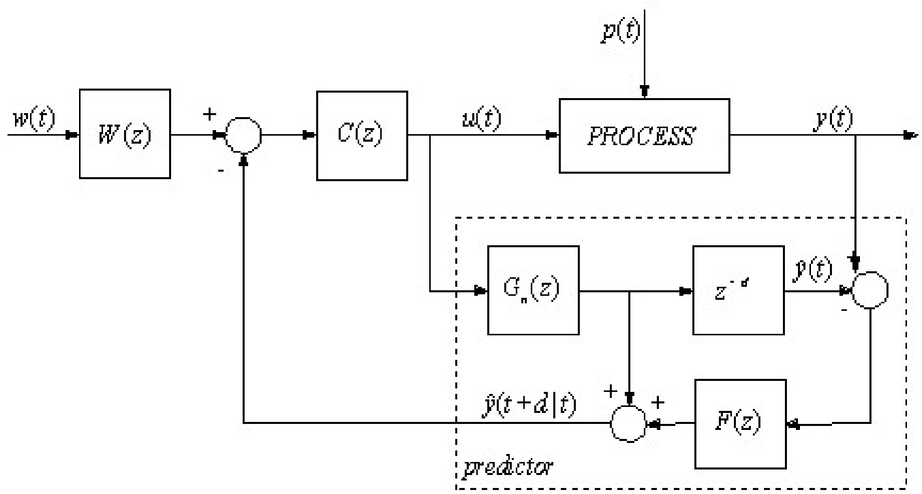


Figura B.4 Esquema de control empleado por OPTIMAX.

B.3.4.- Analizador léxico y analizador sintáctico

Continuando con la descripción del programa OPTIMAX pasaremos ahora a describir el conjunto de aplicaciones software que permiten la elaboración de programas y su uso en distintas aplicaciones de control.

Para introducir los datos de la planta a controlar, los parámetros de sintonización del control y algunos datos necesarios para la comunicación del OPTIMAX con el SCADA, se ha implementado un analizador léxico y un analizador sintáctico.

Los datos anteriormente mencionados, se describirán en un fichero con extensión GPC, respetando el léxico y la semántica desarrollada.

A continuación se describirán el analizador léxico y el sintáctico.

ANALIZADOR LÉXICO

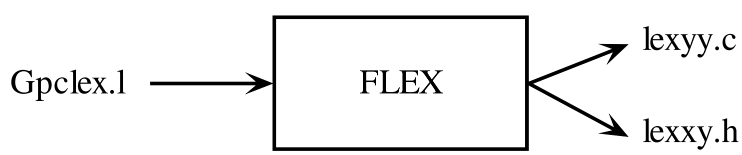
El analizador léxico lee los caracteres uno a uno desde la entrada y va formando grupos de caracteres con alguna relación entre sí (tokens), que constituirán la entrada para la siguiente etapa del compilador. Cada token representa una secuencia de caracteres que son tratados como una única entidad. Por ejemplo, en C un token es la palabra reservada while.

Las principales funciones que realiza son:

- Identificar los símbolos.
- Eliminar los blancos, caracteres de fin de línea, etc...
- Eliminar los comentarios que acompañan al programa fuente.
- Crear unos símbolos intermedios llamados tokens.
- Avisar de los errores que detecte.

La descripción del analizador léxico se describe en un fichero con extensión l, y se utiliza un traductor denominado FLEX, para la generación del mismo.

Ejemplo: Con la siguiente sentencia: LEX Gpclex.l -8 –h –l , se genera los dos ficheros lexyy.c y lexxy.h, que representan el analizador léxico.



DESCRIPCIÓN DEL ANALIZADOR LÉXICO

Se introducirá los conceptos básicos del lenguaje del modelo, destacando los aspectos léxicos más relevantes y la estructura de dicho modelo.

- COMENTARIOS

La especificación de comentarios en un modelo se realiza mediante los caracteres: --.Cualquier línea que comience con --, se considera comentario.

- CADENA DE CARACTERES

Las cadenas de caracteres se representan entre comillas simples, y están formadas por cualquier carácter, excepto fin de línea: /n y las comillas simples: ‘.

Ej: ‘a’,’ ‘, ‘Pilar’, etc.

- NÚMERO ENTERO

Los números enteros son números sin parte fraccionaria que pueden ser positivos, negativos o cero. El signo antepuesto al número es opcional.

Ej: 0, 9, -567, +44, +7, 7

- NÚMERO ENTERO CIENTÍFICO

Consiste en la representación científica de un número entero, que se compone de una base (número entero), seguido de un exponente (número entero), donde el número expresado sería: base * 10 ^{exponente}. Ambas partes quedan separadas por el símbolo E.

Ejemplos de esta notación son los siguientes: +99E-557, -777E54, 87E+2

- NÚMERO REAL

Los números reales se pueden representar de tres modos distintos.

El primero de ellos consiste en dos secuencias de dígitos separados por un punto, representando la parte entera y la parte fraccionaria respectivamente. Ej: +45.777 –4.5 88.77

El segundo modo consiste en que la secuencia que corresponde a la parte entera esté vacía. Ej: +.33, .907, .77

Y por último, se puede representar un número real, prescindiendo de la parte fraccionaria pero conservando el carácter “.” Ej: +5., 5., –5.

- NÚMERO REAL CIENTÍFICO

Consiste en la representación científica de un número real, que se compone de una base (número entero o número real), seguido de un exponente (número entero), donde el número expresado sería: $base * 10^{exponente}$. Ambas partes quedan separadas por el símbolo E.

Ejemplos de esta notación son los siguientes: +99E-557, –777E54, 87E+2, 9.77E-4, .78E5

- IDENTIFICADOR.

A la hora de escribir un modelo, es necesario dar nombre a ciertos elementos que en él se definen. Estos nombres, que son creados por el programador se denominan identificadores. Un identificador comienza siempre por una letra seguida de una secuencia de dígitos y letras.

Ejemplos: aux1, TT5_PVT, LT1_PVT, RESISTENCIA.

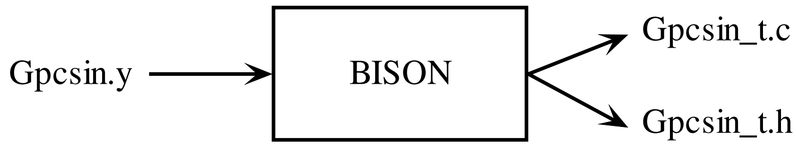
<u>NOTA</u>: El lenguaje es sensible a mayúsculas, es decir, las letras mayúsculas son distintas a las correspondientes en letras minúsculas.

<u>ANALIZADOR SINTÁCTICO</u>

El analizador sintáctico, también llamado *parser*, recibe como entrada los *tokens* que le pasa el Analizador Léxico (el analizador sintáctico no maneja directamente caracteres) y comprueba si esos *tokens* van llegando en el orden correcto (orden permitido por el lenguaje). La salida "teórica" de la fase de análisis sintáctico sería un árbol sintáctico.

Para generar el analizador sintáctico se utiliza el traductor denominado BISON, y las especificaciones requeridas para el mismo se describen en un fichero con extensión y.

Ejemplo: Con la siguiente sentencia: BISON Gpcsin.y –d –l , se genera los dos ficheros Gpcsin_t.c y Gpcsin_t.h.



<u>DESCRIPCIÓN DEL ANALIZADOR SINTÁCTICO</u>

Para la descripción del modelo de la planta se utiliza la siguiente estructura:

1. Todos los modelos comienzan con la palabra: LMFD (Left Matriz Fraction Description).

2. Se indica el tiempo de muestreo, en milisegundos, de la siguiente forma:

   TM = número entero;

3. Descripción de la dimensión de los vectores de entrada y salida, con el siguiente orden:

INPUT = número entero;	Indica el número de entradas del modelo.
OUTPUT = número entero;	Indica el número de salidas del modelo.
DISTURBANCE = número entero;	Indica el número de perturbaciones.
NA = número entero;	Indica el máximo grado del polinomio A.
NB = número entero;	Indica el máximo grado del polinomio B.
NAD = número entero;	Indica el máximo grado del polinomio A de las perturbaciones.
NB = número entero;	Indica el máximo grado del polinomio B de las perturbaciones.

4. Valor de LAMBDA.

   LAMBDA = [lista_números_reales_o_números_enteros];

5. Definición de matriz A.

```
A = [ lista_números_reales_o_números_enteros;
      lista_números_reales_o_números_enteros;
      ⋮                                       ;
      ];
```

6. Definición de matriz B.

```
B = [ lista_números_reales_o_números_enteros;
      lista_números_reales_o_números_enteros;
      ⋮                                      ;
      ];
```

7. Vector de retrasos. Indica los retrasos de la matriz B.

```
D = [numero _ entero];
```

8. Definición de matriz A de las perturbaciones.

```
AD = [ lista_números_reales_o_números_enteros;
       lista_números_reales_o_números_enteros;
       ⋮                                      ;
       ];
```

9. Definición de matriz B de las perturbaciones.

```
BD = [ lista_números_reales_o_números_enteros;
       lista_números_reales_o_números_enteros;
       ⋮                                      ;
       ];
```

10. Vector de retrasos de las perturbaciones. Indica los retrasos de la matriz BD.

```
DD = [número_entero];
```

11. Horizonte de predicción.

```
NY = [lista_de_números_enteros];
```

12. Horizonte de control.

```
NU = [lista_de_números_enteros];
```

13. Vector de referencias.

```
REF = [lista_de_números_enteros_o_reales];
```

14. Vector de restricciones: YMAX, YMIN, uMAX, uMIN, UMAX, UMIN.

```
YMAX = [lista_de_números_enteros_o_reales];    Restr. de entrada
YMIN = [lista_de_números_enteros_o_reales];    Restr. de entrada
```

uMAX = [lista_de_números_enteros_o_reales]; Restr. de salida en incremento
uMIN = [lista_de_números_enteros_o_reales]; Restr. de salida en incremento
UMAX = [lista_de_números_enteros_o_reales]; Restr. de salida en amplitud
UMIN = [lista_de_números_enteros_o_reales]; Restr. de salida en amplitud

15. Configuración de la planta: Nombre de la zona y el nombre de la unidad.

CONFIGURACION_PLANTA

Cadena_de_caracteres Indica el nombre de la zona, sobre la que trabajaremos.

Cadena_de_caracteres Indica el nombre de la unidad, sobre la que trabajaremos.

16. Configuración de Variables de Procesos (PVS)

CONFIGURACIÓN_PVS

Lista_de_cadena_de_caracteres Indican el nombre de las variables de proceso.

17. Configuración de Variables de Salida (OUTS)

Lista_de_cadena_de_caracteres Indican el nombre de las variables de salida.

18. Configuración de la perturbaciones

CONFIGURACION_DIST

Lista_de_cadena_de_caracteres →

→ Indican el nombre de las variables perturbadoras.

¿CÓMO REALIZAR MODIFICACIONES EN EL ANALIZADOR LÉXICO?

LEX -8 -h -l

1) Archivo .h

Al principio del todo:

```
#define YY_<nombre escáner>_CHAR  unsigned char   /*permite caracteres de
                                                    8 bits*/
#ifndef YY_USE_CLASS
#define YY_USE_CLASS
#endif
```

2) Archivo .c → renombrarlo a .CPP

Al principio del todo:

```
#define YY_CHAR   unsigned char
#ifndef YY_USE_CLASS
#define YY_USE_CLASS
#endif
```

Borrar:

```
#ifndef _MSDOS
        #include <osfcn.h>
#endif
```

BISON -d -l

1) Archivo .h

Antes de todo:

```
#ifndef YY_USE_CLASS
#define YY_USE_CLASS
#endif
```

2) Archivo .c → renombrarlo a .CPP

Buscar y borrar:

```
/*Added lex prototype */
#ifndef YY_ParserGpc_PURE
```

⋮
#else
⋮
#endif

Buscar la definición de la matriz YY+name y quitar la primera línea con # por encima y la primera línea por debajo. Concretamente: #if YY_ParseGPC_DEBUG !=0 y #endif

Después de hacer todos los cambios, los ficheros hay que copiarlos al directorio del proyecto.

A continuación se ofrece un ejemplo de cómo incluir los datos de un control tipo GPC de forma que sea comprensible para OPTIMAX.

EJEMPLO

-- Modelo de archivo para introducir los datos --

-- OPTIMAX --

-- Todas las líneas que no comiencen con los dos guiones serán parte de un ejemplo
-- para saber la forma que tienen los datos. Todos los archivos hay que iniciarlos con la
-- secuencia LMFD debido a que la programación impone esto para saber que el
--archivo está programado para OPTIMAX

LMFD

-- Se introduce el tiempo de muestreo en mseg. Este tiempo de muestreo es con el que
-- trabajará OPTIMAX y debe de ser con el que hemos obtenido la matriz de
-- funciones de transferencia de nuestra planta (ensayos). Con el que discretizamos la
--característica dinámica continua.

TM= 4000; -- 40 segundos.

-- Introducimos el número de entradas y salidas que tiene nuestra planta.

INPUT= 3; --número de entradas en sistema controlador.

OUTPUT=1; --número de salidas controladas.

-- Número de perturbaciones que vamos a tratar. Poner a 1 si no hay perturbaciones

DISTURBANCE=1;

-- Para introducir la matriz de funciones de transferencia de nuestro sistema se realiza
-- lo que se denomina Matrix Fraction Description de forma que tenemos que una
-- matriz G será descompuesta en una matriz A y en otra matriz B de forma:
-- G(z^-1)=[A(z^-1)^-1]*B(z^-1), de orden (n*m), resultando A una matriz diagonal, de
-- orden (n*n), donde cada polinomio de la diagonal sale de obtener el mínimo común
-- múltiplo de los denominadores de cada fila de la matriz G. Así B, de orden (n*m), se
-- obtendrá realizando las operaciones pertinentes. NA será el mayor orden de los
-- polinomios que forman A, así si tenemos por ejemplo:
-- $1-0.987z^{-1}+1.114z^{-2}+0.876z^{-3}$
-- NA será igual a 3. Los polinomios deben tener todos los mismos órdenes de forma
--que si alguno de ellos es de orden menor se le añadirán ceros a la cola del vector
-- para tener todos los polinomios con igual dimensión. Uno de los objetivos de este
–-proyecto es desarrollar una función matlab que proporcione estas matrices
-- características del sistema según esta regla.

NA=3;

-- NB será lo mismo pero para los polinomios que conforman la matriz B que se
--obtiene también a partir de la factorización LFMD.

NB=13;

-- NAD y NBD son los correspondientes a las perturbaciones. Poner a 1 si no hay
-- perturbaciones

NAD=1;

NBD=1;

-- Factores de corrección lambda, deben existir tantos como número de entradas
--(INPUT).

LAMBDA= [100 1 10];

-- Matriz A de la Matrix Fraction Description donde los polinomios se escriben en
-- orden descendente de z^{-i} empezando por z^{0}. El polinomio que se corresponde con el
– ejemplo es:
-- $1-2.2586z^{-1}+1.6731z^{-2}-0.4046z^{-3}$
-- Los polinomios deben ser todos del mismo orden de forma que si alguno de ellos es
--de orden menor (potencia más negativa menor) se le añadirán ceros
-- por detrás para que todos los polinomios tengan la misma longitud.

A =[1.0   -2.2586   1.6731   -0.4046];

-- Matriz B de la Matrix Fraction Description donde los polinomios se escriben según el --mismo orden antes descrito en orden. Una vez más, todos deben tener la misma --longitud.

B =[0 0 0 0 0 0 0 0 0 0 0.5438 -0.7830 0.2688 0 0 0 0 0 0.249 -0.0345

0.01154 0 0 0 0 0 0 0 0 0 0 0.2176 -0.3686 0.1559 0 0 0 0 0];

-- Mínimo retraso de cada fila de la matriz B. Coincide con el número mínimo de ceros
-- que tenemos al empezar los polinomios de la matriz B.

D= [5];

-- Matrices A, B y D correspondientes a las perturbaciones. Poner de la siguiente forma
-- si no hay perturbaciones.

--DISTURBANCE

AD =[1.0 0.0];

BD =[0 0];

DD= [0];

-- NY se aconseja que sea el horizonte de predicción menos el horizonte de control y
-- NU el retraso más 1

NY= [30];

NU= [11 6 6];

-- Referencia a conseguir en la variable/es controlada/as.

REF= [700];

-- Rango en el que se va a mover las variables que estamos controlando

YMAX= [50000];

YMIN= [-50000];

-- Restricciones en velocidad de cambio (slew-rate), es el salto máximo que puede --proporcionar en la salida el actuador que estamos manipulando.

uMAX=[1 50 25];

uMIN=[-1 -50 -25];

-- Restricciones en amplitud, valores máximos y mínimos permitidos para la salida del
-- controlador

UMAX= [36 4500 500];

UMIN= [30 2000 50];

-- Metemos la configuración de la planta de forma que primero aparece el nombre de la
-- zona que estamos tratando y después el nombre de la unidad (PMC) donde se
-- encuentran nuestras variables a tratar.

CONFIGURACIÓN _ PLANTA

ALMAZARA

EXTRACCION

-- Introducimos el nombre de las variables de entrada de nuestro control que serán las
-- salidas en nuestra planta real. Son las variables que queremos controlar.

CONFIGURACION_PVS

FTL1155_PVC2

-- Introducimos el nombre de las variables de entrada a la planta que serán las salidas de
-- nuestro controlador. |C| -outs- |P| - pvs -

CONFIGURACION_OUTS

TICL1111_CSSP

FICL1138_CSSP

FTL1139_CORMAN

-- Introducimos el nombre de las perturbaciones. Si no existen perturbaciones, se mete
-- una cualquiera

CONFIGURACION_DIST

TTl0112_PV

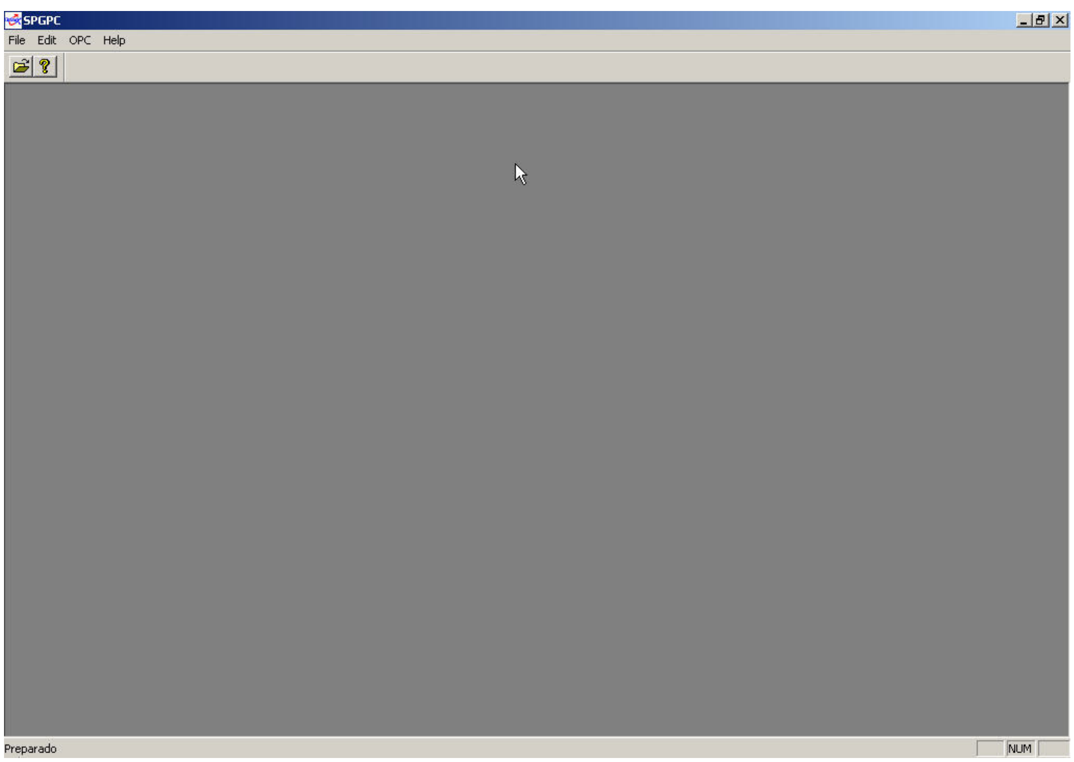
B.4.- Manual de usuario

La finalidad de incluir este manual es presentar cómo deben incluirse los datos de nuestra planta, determinados mediante ensayos, para que estos sean comprensibles para OPTIMAX de forma que podamos experimentar distintas estrategias de control sobre la planta.

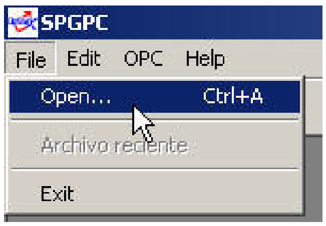
Se va a desarrollar un ejemplo monovariable, pero todo lo que se explica es extensible al caso multivariable.

Para arrancar el programa hay que ejecutar el archivo Spgpc.exe, el cual se puede encontrar con el siguiente icono en el escritorio:

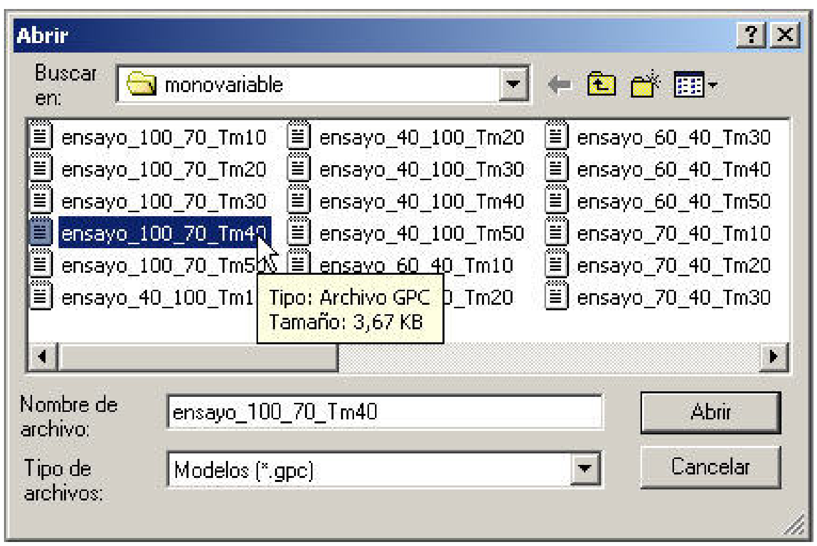
Una vez ejecutado este archivo sale la pantalla inicial del programa donde la primera acción a realizar consistirá en elegir como extensión del archivo la “gpc” si vamos a desarrollar un nuevo programa de control.



También es posible ejecutar un programa elaborado con anterioridad, para elegir el archivo a ejecutar, abriremos el comando "*File*" y elegiremos la opción "*Open*" o daremos con el botón izquierdo del ratón sobre la carpeta amarilla, debajo del comando "*File*" como se indica en las siguientes figuras:



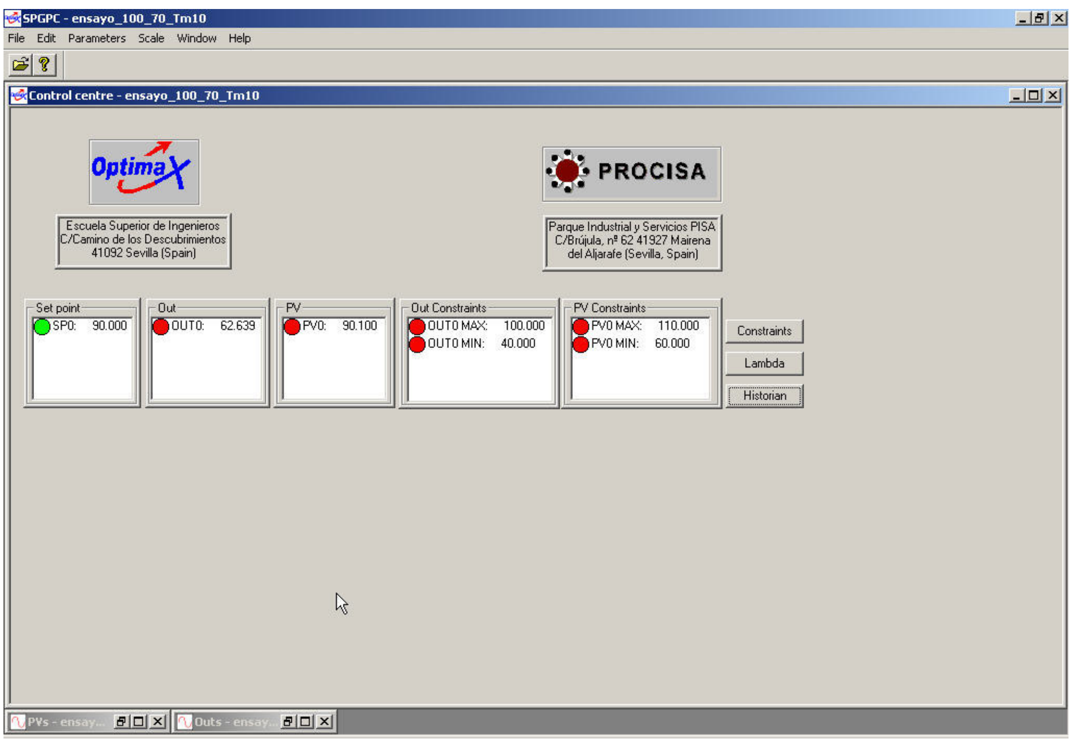
A partir de este instante se buscará la carpeta donde ha sido creado el archivo "gpc" y seleccionaremos el que se vaya a utilizar. En nuestro ejemplo un programa de control monovariable tal y como aparece en la siguiente figura.



Una vez abierto el archivo deseado, lo siguiente que realiza OPTIMAX es la conexión con el servidor de CUBE. Cuando esto ocurra nos avisará con la siguiente pantalla:

Debemos de aceptar para continuar, y nos saldrá una nueva pantalla avisando de que el OPTIMAX se ha activado:

Una vez que se escuche un pitido fino todo estará listo para trabajar con OPTIMAX. La pantalla que se tendrá en este momento será de la siguiente forma:

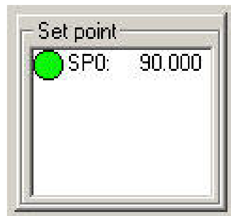


Una vez llegado a este punto, se describirán las variables que deben estar en la base de datos de la planta a controlar para que la herramienta OPTIMAX funcione correctamente:

- SPTAG00, SPTAG01, SPTAG02, ...

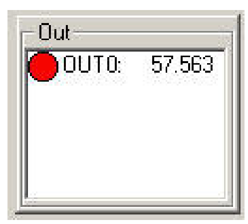
Variables reales. Habrá tantas variables del tipo SPTAG0X, como set-points se implanten en el control. Por ejemplo, si se tienen 3 set-points se necesitará SPATG00, SPTAG01 y SPTAG02 correspondiendo, respectivamente, cada variable a los set-points definidos en el archivo "gpc".

Para cambiar el set-point, se manipulará directamente cada variable, ya que en la aplicación OPTIMAX sólo se visualiza el set-point que se desea y éste no se puede manipular.



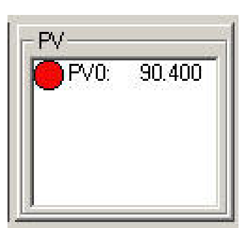
- OUT_MPC00, OUT_MPC01, OUT_MPC02, …

Variables reales. En estas variables es donde escribirá OPTIMAX las acciones de control, por lo tanto habrá que copiarlas a las variables manipulables. Los valores que toman quedan reflejados en la siguiente pantalla:



- PVTAG00, PVTAG01, PVTAG02, ...

Variables reales. Son las variables que utiliza OPTIMAX para guardar las variables de proceso que se están controlando. Los valores que toman quedan reflejados en el recuadro "*PV*".



- PREDICTOR_SMITH

Variable lógica. Se activa desde Cube. La activa el usuario pero hacerlo sólo cuando hay retardo.

- NF_FILTRO_MPC

Variable real. Orden del filtro paso bajo. En la fórmula corresponde a "m". Se modifica en Cube. Ponerlo normalmente a 1 y aumentarlo sólo en caso que el control no sea efectivo. El orden del filtro será el mismo para todas las salidas.

- AF_FILTRO_MPC00, AF_FILTRO_MPC01, AF_FILTRO_MPC02, ...

Variable real. Polo del filtro para la salida 0...n. En la fórmula corresponde a "γ". Se modifica en Cube. En este caso el polo del filtro puede ser distinto para cada una de las salidas.

- CTE_FILTRO_MPC00, CTE_FILTRO_MPC01, CTE_FILTRO_MPC02, ...

Variable real. Factor del cero del filtro para la salida 0...n. En la fórmula corresponde a "β". En este caso también la constante del filtro puede ser distinta para cada una de las salidas.

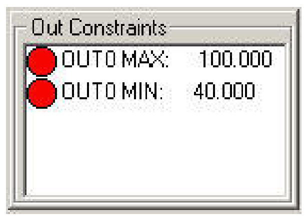
- OPTIMAX AUTOMATIC

Variable lógica. Variable creada para activar el control de la planta a través de una secuencia de Cube. Es posible crear secuencias que actúen al verificarse una condición lógica, como por ejemplo:

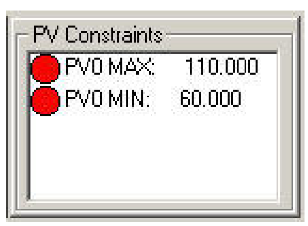
```
IF (OPTIMAX_AUTOMATIC) THEN
    VM110001=OUT_MPC00
ENDIF
```

Una vez descritas las variables, continuemos describiendo la pantalla inicial de la aplicación acercándonos para ello a los otros bloques de los cuáles aún no hemos comentado nada.

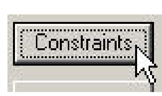
En el recuadro "*OUT Constraints*" (4º por la izqda.) aparecen representados los valores límite (tanto máximos como mínimos) que pueden tomar cada una de las variables de salida incluidas en el recuadro "*Out*". Representan las restricciones impuestas a las variables manipulables, éstas vienen relacionadas con las acciones de mando.



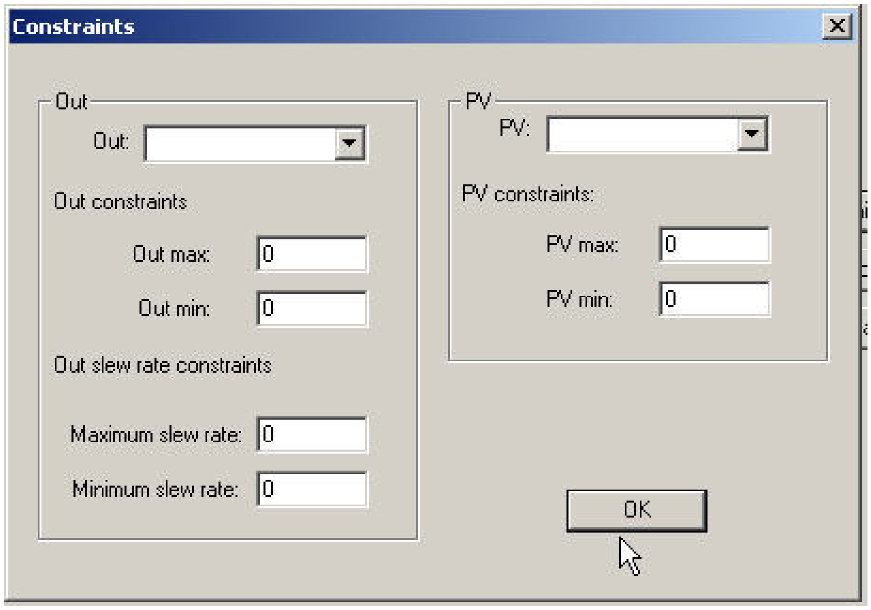
En último lugar analicemos el significado de las variables que aparecen en "*PV Constraints*" (5º izqda.).En este cuadro se recogen las restricciones a las que están sometidas las variables de proceso que estamos controlando, esto es los valores límites que pueden llegar a tomar en cada caso.



Los dos tipos de restricciones anteriores se pueden modificar, con OPTIMAX en funcionamiento, pulsando sobre el botón "Constraints" en el menú de la derecha de la pantalla inicial.

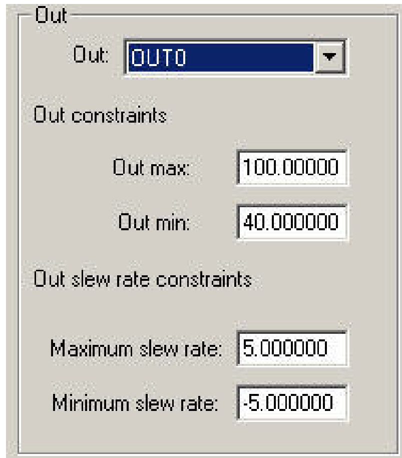


Una vez pulsado saltará la siguiente pantalla:



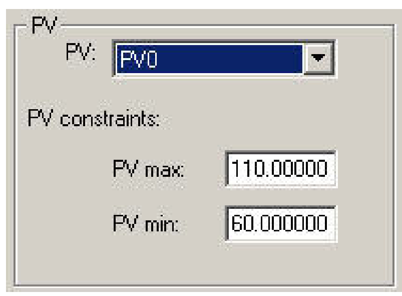
Lo primero que se debe de hacer para cambiar alguna restricción es elegir la variable deseada, ya sea de salida, “*Out*”, o de entrada, “*PV*” sobre la cual vamos a imponer la restricción. Así, si desea cambiar, por ejemplo las restricciones de una de las variables de salida, habrá que pulsar sobre la flecha en la parte de “*Out*” y elegir la variable de la lista que se despliega (aparecerá más de una señal en el caso multivariable)

Esto hará que se muestren los valores asociados a la variable elegida.

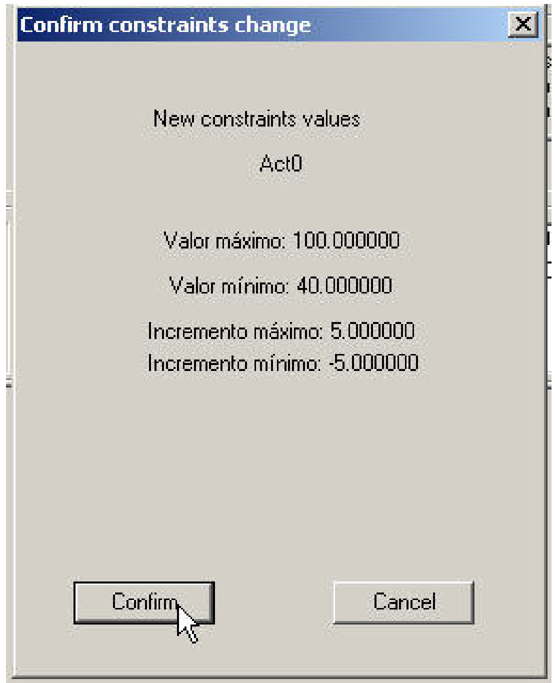


Una vez mostrado los valores actuales, se pueden modificar directamente los nuevos valores en los cuadros correspondientes según queramos acotar los valores extremos de la salida y/o la velocidad de cambio permitida a la misma.

De forma análoga se puede proceder para modificar las restricciones de las variables de proceso.



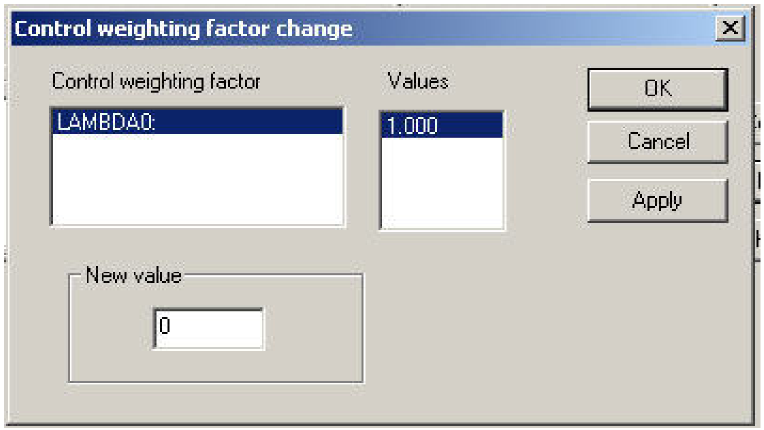
Una vez escritos todos los nuevos valores se pulsará el botón "*OK*".Si todo ha ido correctamente aparecerá una pantalla de confirmación.



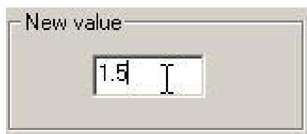
En esta pantalla se confirmarán o se cancelarán los cambios anteriormente introducidos.

Otra de las opciones de este menú en la pantalla principal viene representada por "*Lambda*" que permite modificar los valores del esfuerzo de control que considera OPTIMAX para el cálculo del control GPC.

Una vez pulsado, se puede ver en un nuevo recuadro el valor que toma λ en el instante actual. Dicho cuadro será como sigue:



Para cambiar este parámetro habrá que escribir el valor deseado para el esfuerzo de control en el recuadro "*New value*" tal como indica la figura.



Posteriormente pulsaremos el botón "Apply".

Si no se pulsa este botón antes de salir, el cambio no se hará efectivo. Una vez pulsado "Apply" cerramos el recuadro bien confirmando los cambios con el botón "*OK*" o cancelando con el botón "*Cancel*".

A las posibilidades que nos dan los dos botones anteriores, "*Constraints*" y "*Lambda*", podemos acceder también a través de la barra de comandos que se encuentra en la parte superior de la pantalla inicial de la aplicación. Si seleccionamos la opción "*Parameters*" se despliega un menú que nos permite acceder bien sobre el valor de lambda bien sobre el valor de las restricciones.

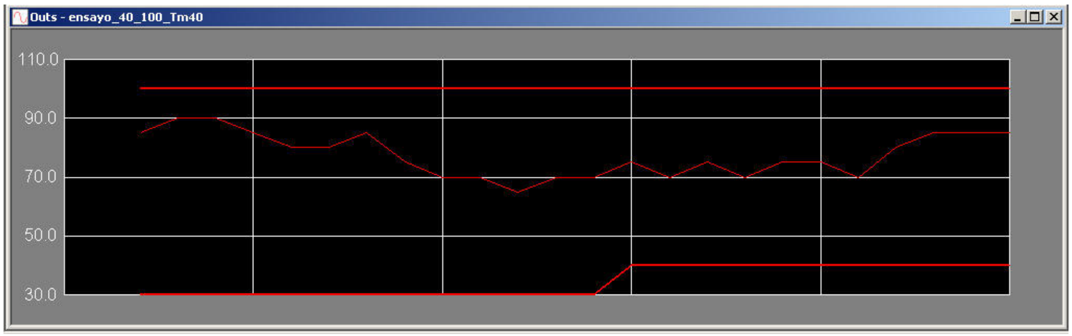
Una opción más del menú de la derecha es “*Historian*”. Este botón activa el siguiente recuadro:

Como el título del recuadro indica, aquí se puede ver el histórico de algunas de las variables que se están tratando ya sea de salida, de entrada o un set-point. Para hacer la consulta, sólo se tiene que pulsar, con el ratón, sobre la variable deseada y pulsar el botón seleccionar.

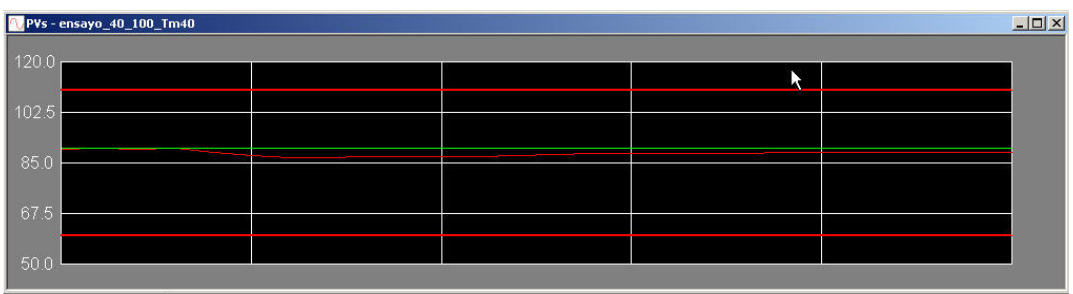
Una vez pulsado aparecerá una lista con los últimos 10 valores tomados por esa variable en los últimos 10 tiempos de muestreos. El tiempo de muestreo es el que hallamos fijado en el archivo con extensión “gpc”.

En la parte inferior de la pantalla, encima de la barra de tareas, hay dos ventanas minimizadas. Estas representan dos pantallas gráficas donde podemos seguir la evolución de nuestro control.

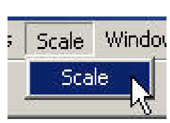
Si se restaura la pantalla “*Outs*” se podrá visualizar la evolución de las variables manipulables, además de dos líneas, como en la gráfica siguiente, que representan el valor superior e inferior de las restricciones.



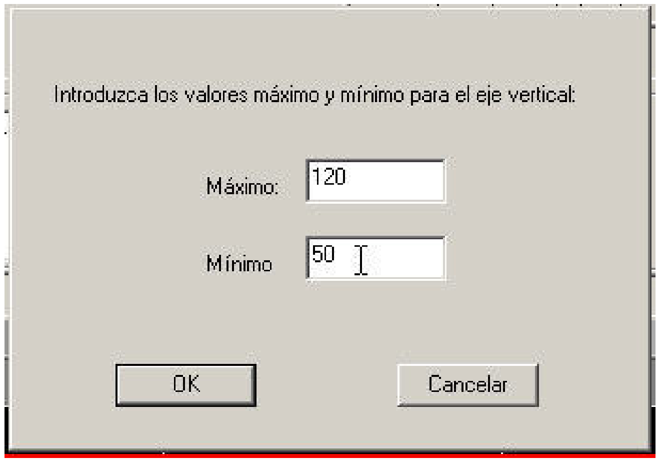
Si se restaura la pantalla "*PVs*" se podrá visualizar la evolución de las variables de proceso, de los set-points, además de dos líneas que representan el valor superior e inferior de las restricciones asociadas a dichas variables de proceso.



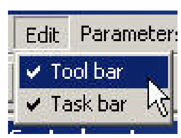
Al arrancar la aplicación OPTIMAX se establecen unos valores predefinidos para la escala de las gráficas. Si queremos modificar la escala debemos tener la ventana visible y restaurada. Pulsamos a continuación sobre el comando "Scale" situado en la barra de herramientas de la parte superior de la pantalla inicial y será posible la modificación.



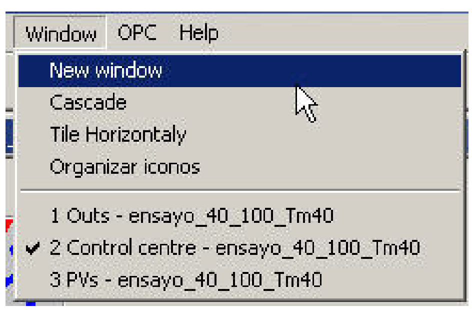
En la pantalla que se activa podremos poner los nuevos límites.



Otro de los comandos que se tienen en la barra superior de herramientas es "Edit" donde se podrá activar o desactivar la barra de herramientas, "Tool bar", y la barra de tareas del OPTIMAX, "Task bar".



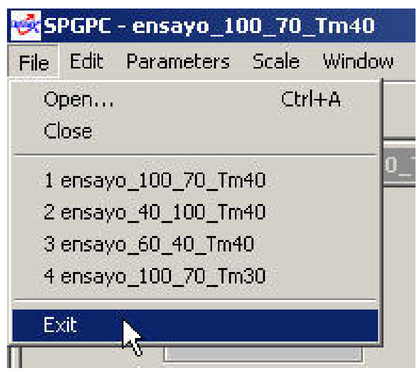
En el comando "*Windows*" podemos organizar las ventanas que tenemos en el OPTIMAX, la principal y las dos ventanas gráficas.



Por último, el comando "*Help*" muestra información sobre la versión del OPTIMAX con la que se está trabajando.

También se puede acceder a esta información pulsando sobre el interrogante amarillo que aparece en la barra de herramientas.

Para cerrar el OPTIMAX se pulsará sobre la opción "*Exit*" del comando "*File*".

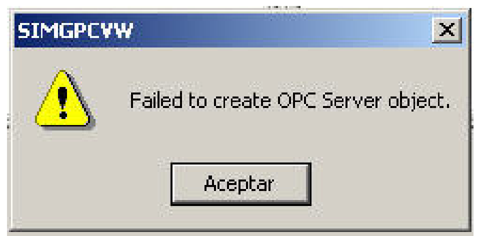


Una vez pulsada se cerrará la pantalla con la que se ha trabajado, y pasado un poco de tiempo debe de aparecer, una primera ventana que te avisa que estabas conectado.

Y una vez que se pulsa el botón "*Aceptar*", se activa una nueva ventana para desconectar definitivamente.

- **ERRORES QUE PUEDEN SURGIR**

A continuación mostraremos las pantallas que saltan para avisar de que existe algún error.



Esta pantalla es debido a que no se ha podido establecer la comunicación OPC, bien debido a que no hay servidor, bien porque no se ha especificado correctamente el nombre del ordenador al que se desea conectar o bien debido a que hay algún problema en la comunicación, ya sea por problema de red o cualquier otra causa.

Se puede solucionar ejecutando manualmente el archivo *CubeOPCSvr.exe* que se encuentra ubicado en *C:\ICUBESYS\BIN* .

Una vez establecida la conexión, el siguiente error que puede aparecer es:

Este error es debido a que no se pueden añadir los ítems(valores) en la base de datos de nuestra planta. Esto puede ser debido a que la base de datos está abierta (semejante a lo que pasa con Word cuando quiero cambiar el nombre de un archivo que estoy utilizando) o bien porque las variables anteriores que hemos descrito, SPTAG, PVTAG, etc, no han sido añadidas a la base de datos y son irreconocibles.

Después de tener estos errores, normalmente, hay que abrir el Administrador de tareas de Windows y terminar el proceso Spgpc.exe, ya que se queda ocupando casi todo el uso de la CPU comportamiento totalmente indeseable.