

## **5.- MANUALES DE USO**

### **5.1.- Aspen License Manager**

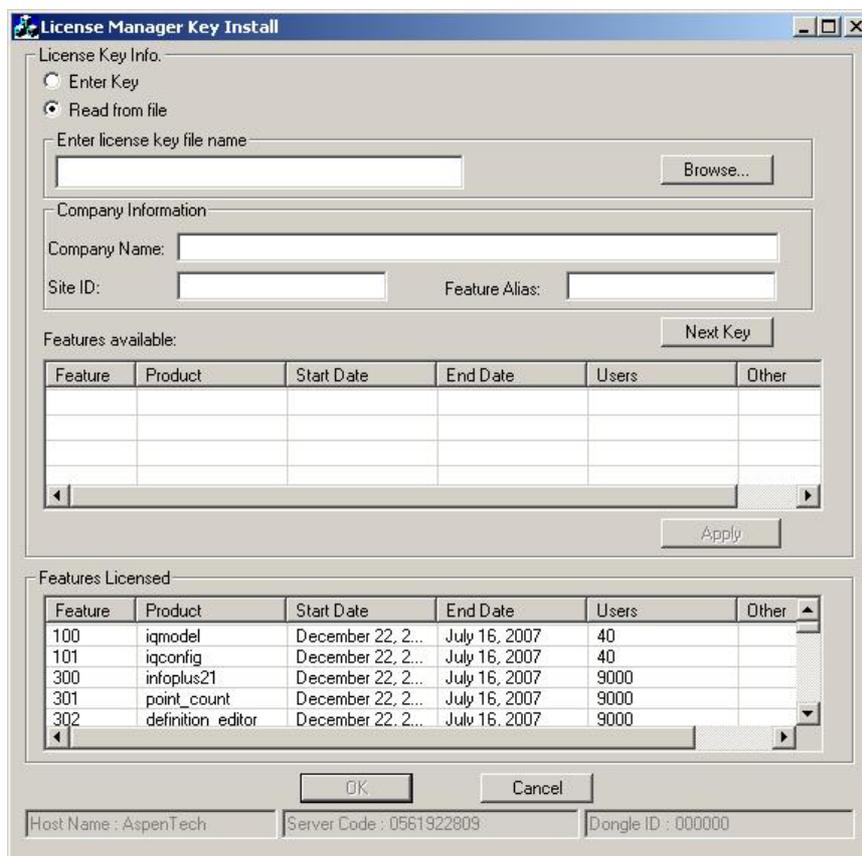
Como ya sabemos, este programa es el encargado de gestionar las licencias del resto de programas de Aspen Manufacturing Suite. Dichas licencias a su vez están asociadas a una llave hardware que es necesario tener acoplada al equipo.

Aspen License Manager (ALM) debe estar activado antes de proceder a utilizar el resto de los programas de la suite, ya que de lo contrario no estarán disponibles las licencias que deben utilizar los programas y éstos no podrán iniciarse.

ALM se compone de varios módulos que nos ayudan en la gestión de las licencias de las que se dispongan. Estos módulos se detallan a continuación.

#### **5.1.1.- License Manager Key Installer**

Este módulo sirve para actualizar, eliminar o añadir licencias durante o después de la instalación de la Suite (ver capítulo 4).



**Figura 5.1.- Ventana de License Manager Key Installer**

El acceso a este módulo se realiza por Inicio> Programas> AspenTech> License Manager 2.2> License Manager Key Installer.

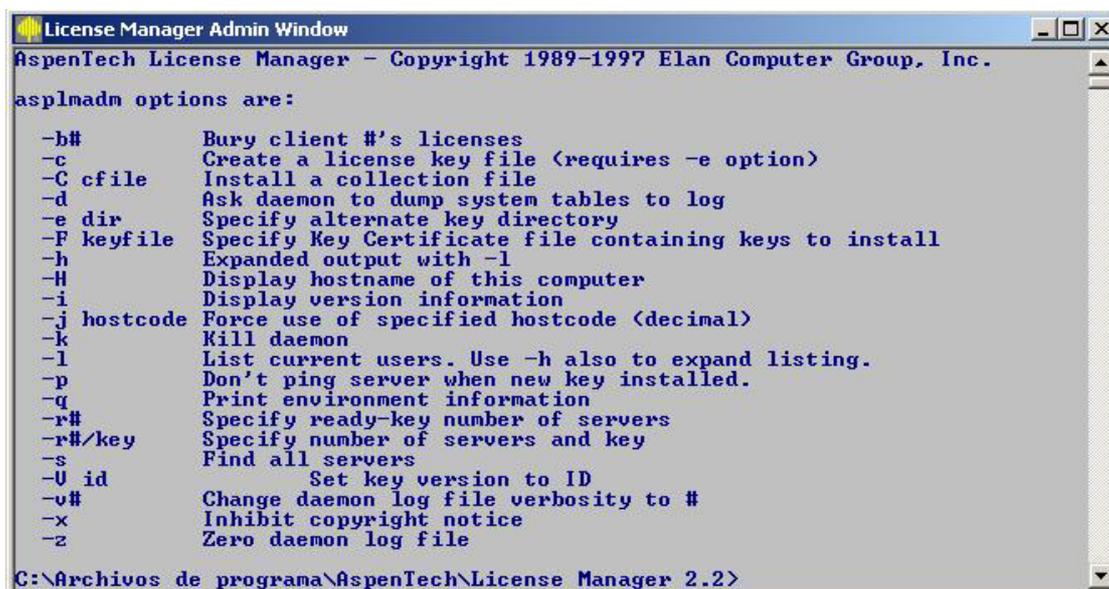
Las licencias podrán ser instaladas de forma manual o de manera automática con un archivo de licencias .lic.

Al insertar una licencia nos aparecerá el o los productos para los cuales es válida la licencia en el apartado Features Available. Entonces elegiremos aquellos programas que nos interesen o todos los que aparecen, y tras pulsar el botón Apply, los programas seleccionados pasarán a engrosar la lista de programas con licencias disponibles que se puede consultar en el apartado Features Licensed.

### **5.1.2.- License Manager Admin Window**

Accedemos a este módulo a través de Inicio> Programas> AspenTech> License Manager 2.2> License Manager Admin Window.

Con él podremos realizar un seguimiento de las licencias que está en uso, su tiempo de utilización, restricción en el uso y demás tareas típicas de gestión.



```

License Manager Admin Window
AspenTech License Manager - Copyright 1989-1997 Elan Computer Group, Inc.
asplmadm options are:
-b#       Bury client #'s licenses
-c        Create a license key file (requires -e option)
-C cfile  Install a collection file
-d        Ask daemon to dump system tables to log
-e dir    Specify alternate key directory
-F keyfile Specify Key Certificate file containing keys to install
-h        Expanded output with -l
-H        Display hostname of this computer
-i        Display version information
-j hostcode Force use of specified hostcode (decimal)
-k        Kill daemon
-l        List current users. Use -h also to expand listing.
-p        Don't ping server when new key installed.
-q        Print environment information
-r#       Specify ready-key number of servers
-r#/key   Specify number of servers and key
-s        Find all servers
-U id     Set key version to ID
-v#       Change daemon log file verbosity to #
-x        Inhibit copyright notice
-z        Zero daemon log file

C:\Archivos de programa\AspenTech\License Manager 2.2>

```

Figura 5.1.- Menú principal de License Manager Admin Window

Para trabajar con esta herramienta hemos de escribir *asplmadm* seguido de una de las opciones que se nos muestran en la figura 5.2, algunas de las cuales son:

- b num ⇒ quita las licencias que en ese momento estuviera utilizando el cliente con ID igual a num.
- l ⇒ muestra la actividad de las licencias de todos los servidores de licencias en la red local..

- i ⇒ nos genera la versión de asplmadm.
- k ⇒ detiene (realiza un 'kill') el servidor de licencias.
- H ⇒ muestra nuestro nombre de host.
- q ⇒ muestra información del entorno de trabajo para depurar fallos.
- s [num] ⇒ muestra los servidores de licencias en la red. Si se especifica un número sólo se mostrará el servidor con dicha ID.
- v [niv] ⇒ cambia el nivel al valor dado por niv (entre 1 y 9).
- z ⇒ trunca el archivo log de control de la administración del servidor de licencias.

### **5.1.3.- License Server Control Program**

Esta herramienta nos sirve para controlar el funcionamiento del servidor de licencias.

Si abrimos este módulo siguiendo el camino Inicio> Programas> AspenTech> License Manager 2.2> License Server Control Program, nos aparecerá una ventana como la de la figura 5.3.



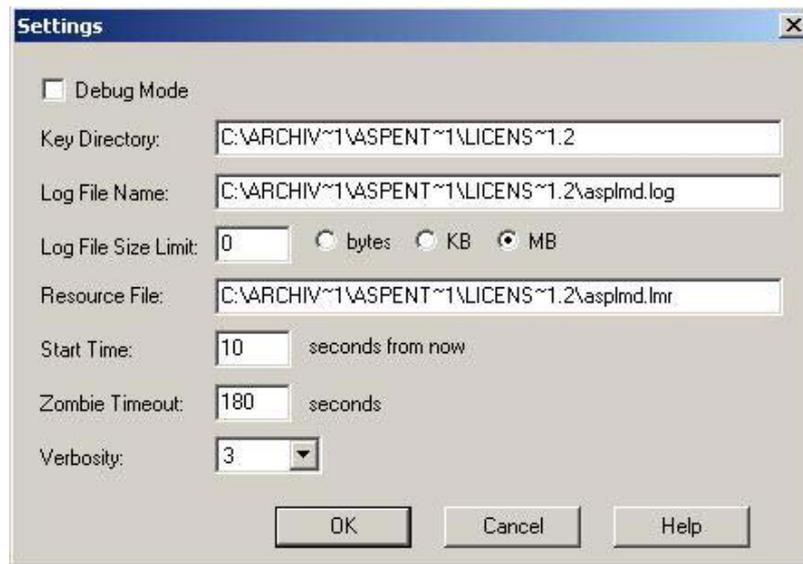
**Figura 5.2.- License Server Control**

En la ventana anterior podemos configurar el arranque del servidor de licencias de tal forma que arranque de forma automática al iniciar el ordenador, que lo haga de forma manual (debiendo entonces arrancar pulsando el botón Launch/Stop) o bien que no arranque nunca.

También existe la posibilidad de configurar una serie de características del servidor. Para ello detenemos primero el servidor y pulsamos el botón Settings. Entonces se abrirá una nueva ventana en la que podemos especificar:

- Debug Mode: hace que el servidor de licencias escriba los mensajes que genere en el

archivo log correspondiente. Es útil cuando encontramos problemas con el servicio de licencias, ya que de otra forma lo único que conseguimos es ralentizar el funcionamiento del servidor. Se recomienda dejarlo sin validar.



**Figura 5.3.- Características configurables**

- Key Directory: dirección completa para localizar el archivo de licencias.
- Log File Name: localización del archivo log.
- Log File Size Limit: tamaño máximo en bytes, Kbytes o Mytes del archivo log. Cuando el archivo alcanza dicho tamaño se trunca automáticamente. Puesto a cero no se trunca nunca, aunque debe tenerse cuidado para no hacer un archivo excesivamente grande.
- Resource File: dirección del archivo de recursos del ALM.
- Start Time: tiempo en segundos para arrancar e servidor de licencias.
- Zombie Timeout: debe permanecer en 180 a no ser que Aspen Tech nos indique otro valor.
- Log File Vervosity: nivel para el archivo log. Según el nivel se mostrarán unos mensajes u otros.

## **5.2.- Collect y Extract**

En este apartado trataremos la toma de datos previa para el modelado del proceso que queremos controlar.

Vamos a utilizar las herramientas Collect y Extract. Éstas son herramientas que trabajan en entorno MS-DOS y se utilizan para obtener los datos de la planta mientras realizamos los ensayos correspondientes y formatearlos de tal manera que DMCplus Model pueda ser capaz de trabajar con ellos.

Un resumen del uso de estas utilidades es el siguiente:

- Mediante Collect hacemos una recopilación de datos en formato binario. Estos muestreos pueden ser configurados según nuestras preferencias.
- Una vez que tenemos los datos de la planta, haciendo uso de Extract les damos el formato apropiado para poder exportar los datos al Model.

### **5.2.1.- Collect**

Para hacer uso de Collect ha de estar configurada y activa la conexión con el servidor OPC de la planta tal y como se explicó en capítulos anteriores, es decir, el CIM-IO for OPC interfase debe estar activo.

1. El primer paso para obtener datos de la planta es crear un archivo sin extensión mediante el cual configuraremos el muestreo de los datos, llamado Input File. Este archivo se muestra en la figura 5.5.

Como podemos observar, este archivo de configuración está dividido en dos secciones separadas por símbolos de igual. En la primera de estas secciones se introducen los datos referentes a la recogida de datos como frecuencia de muestreo, número de muestras y demás. En la sección de etiquetas (tags en inglés) lo que especificamos son los nombres de las variables del sistema que queremos obtener.

2. Una vez configurado el archivo de recopilación de datos es importante realizar una validación de dicho archivo para evitar futuros errores. Para ello basta abrir una sesión MS-DOS y una vez situados en el directorio en el que se encuentra el archivo ejecutamos `dmcpcollect -v [nombre de archivo]`, con lo cual obtendremos un resumen de las características de la recogida de datos que vamos a obtener.

- Una vez verificada la ausencia de errores en el archivo procedemos a recoger los datos de la planta. Utilizando otra vez una sesión MS-DOS como en el caso anterior aunque en este caso ejecutamos *dmcpcollect [nombre de archivo]*. Entonces veremos por pantalla un mensaje en el que se nos confirma el comienzo de la recolección de datos, mensajes que seguirán apareciéndonos según se haya configurado la mensajería.

```

!Version del programa que estamos usando
VERSIONNUM=6.0
!Descripcion del archivo
IDLINE=Recoleccion de datos de la planta piloto v.1
!Informacion sobre la conexión al proceso: service name,unit,format
CIMIOINFO=CIOOPCCUBE1,1,UT
!Lista de datos aextraer: con -1 se hace automático
CIMIOLIST=-1,-1,-1
!Tiempo de muestreo que vamos a usar
SAMPLEPER=2s
!Numero de muestras que tomaremos. Con 5400=3h
MAXSAMP=5400
!Offset para el comienzo de la recoleccion de datos desde el comienzo de un minuto
OFFSET=0
!Cuando se hagan todos los muestreos se para la recolección
EOFCHOICE=0
!NOMSGS=0 muestra mensajes en cada muestreo de las variables
!WATCHDOGTAG="CIOOPCCUBE1":1:"PLANTA\PERRO":REAL:UT escribe en este tag cuando ha
habido alguna parada en el muestreo de datos
!NUMSAMTAG="CIOOPCCUBE1":1:"PLANTA\NUMEROTAG":REAL:UT escribe en este tag el
numero de muestreos que van realizándose
!STATUSTAG="CIOOPCCUBE1":1:"PLANTA\STATUSTAG":REAL:UT escribe en este tag si fue
bien la cosa
=====
"": "PLANTA\TT1":REAL:
"": "PLANTA\FT1":REAL:
"": "PLANTA\V4":REAL:
"": "PLANTA\TT2":REAL:
"": "PLANTA\FT2":REAL:
"": "PLANTA\V5":REAL:
"": "PLANTA\TT4":REAL:
"": "PLANTA\FT4":REAL:
"": "PLANTA\V8":REAL:
"": "PLANTA\TT5":REAL:
"": "PLANTA\LT1":REAL:
!Tag de encendido de la resistencia del depósito
"": "PLANTA\RESIST_DIG":REAL:
!Tag del porcentaje de resistencia utilizado
"": "PLANTA\RESISTENCIA":REAL:
!Tag de activación de alguna de las alarmas programadas
!"": "PLANTA\ALARMA_R":REAL:
    
```

**Figura 5.5.- Archivo de configuración para el muestreo del proceso**

- Al comenzar la recolección de datos se crea un archivo con el mismo nombre que el de la configuración con la extensión *.cle* que se utilizará luego con el Extract, y el

archivo .bin con los datos que se van obteniendo del proceso en formato binario.

```
Y~~~TT1~~~""::"PLANTA\TT1":REAL:~~~~~  
Y~~~FT1~~~""::"PLANTA\FT1":REAL:~~~~~  
Y~~~V4~~~""::"PLANTA\V4":REAL:~~~~~  
Y~~~TT2~~~""::"PLANTA\TT2":REAL:~~~~~  
Y~~~FT2~~~""::"PLANTA\FT2":REAL:~~~~~  
Y~~~V5~~~""::"PLANTA\V5":REAL:~~~~~  
Y~~~TT4~~~""::"PLANTA\TT4":REAL:~~~~~  
Y~~~FT4~~~""::"PLANTA\FT4":REAL:~~~~~  
Y~~~V8~~~""::"PLANTA\V8":REAL:~~~~~  
Y~~~TT5~~~""::"PLANTA\TT5":REAL:~~~~~  
Y~~~LT1~~~""::"PLANTA\LT1":REAL:~~~~~  
Y~~~RES_D~~~""::"PLANTA\RESIST_DIG":REAL:~~~~~  
Y~~~RES~~~""::"PLANTA\RESISTENCIA":REAL:~~~~~
```

**Figura 5.6.- Archivo de extensión .cle**

5. La obtención de datos termina según se haya señalado en el archivo de configuración o bien si utilizamos la combinación de teclas Ctrl+C.

#### PARÁMETROS DEL ARCHIVO DE CONFIGURACIÓN:

A continuación se da una breve descripción de los parámetros necesarios para la configuración del muestreo del proceso, parámetros que han de ir en la sección principal del archivo.

- VERSIONNUM: sólo indica la versión de DMCplus que se está utilizando.
- IDLINE: texto de identificación del archivo en uso.
- CIMIOINFO: especifica el dispositivo CIM-IO que se va a usar, el número de unidad y el formato de los datos.
- CIMIOLIST: describe el tamaño de la lista CIM-IO, la longitud de la descripción del dato y la longitud de las unidades ingenieriles. Podemos obviarlo poniendo todo a -1.
- SAMPLEPER: período de muestreo que se va a utilizar, en segundos minutos u horas.
- MAXSAMP: número máximo de muestreos que vamos a tomar.

Los parámetros que siguen son opcionales, al contrario que los anteriores, que deben aparecer de forma obligatoria.

- OFFSET: tiempo en segundos en el que queremos que empiece el muestreo a partir del comienzo de un minuto.
- EOFCHOICE: indica qué a de hacerse cuando se terminan los muestreo programados (finalizar, actualizar o continuar mientras haya espacio en disco).
- NOMSGS: flag para la impresión de mensajes en pantalla.

- WATCHDOG: avisa mediante la activación de un flag en la base de datos de que ha habido una parada en la recogida de datos.
- NUMSAMTAG: escribe en la base de datos el número de muestreos que se van realizando.
- STATUSTAG: en cada muestreo indica si se realizó correctamente o no mediante la escritura en una variable de la base de datos.

Finalmente en la segunda sección del archivo de configuración de los muestreos deben indicarse todos los tags, etiquetas o variables del proceso que queremos muestrear. Para ello tenemos que insertar una línea para cada variable con el siguiente formato:

“dispositivo”:unidad:”etiqueta de variable”:tipo de datos:formato

- “dispositivo” es el nombre del dispositivo (device name) CIM-IO que vamos a utilizar. Es un campo opcional.
- :unidad: es el número de la unidad (unit number) CIM-IO que utilizaremos. Campo opcional.
- “etiqueta” variable que queremos muestrear. Campo obligatorio.
- :tipo datos: se trata del formato de datos que utiliza la variable y es obligatorio.
- formato se trata de un código de dos caracteres que indica el formato de los datos que se tomen. ES un campo opcional.

Todos los campo anteriores tipificados como opcionales si no se especifican son tomados por defecto como se definieran en la primera sección del archivo de configuración en el parámetro CIMIOINFO.

#### OBSERVACIONES A TENER EN CUENTA:

A veces es posible que se produzcan fallos en la transmisión de datos a través de la red debido a retrasos en la misma o a una saturación en el envío de datos. Se ha detectado que, para la red en la que se utilizó DMCplus, para una frecuencia de muestreo de 1 segundo la pérdida de datos se hace considerable, impidiendo incluso la obtención de un modelo aceptable. Por ello se recomienda utilizar una frecuencia de 2 segundos como mínimo, frecuencia a la cual la pérdida de datos es prácticamente inexistente.

También hay que resaltar que el ensayo a realizar sobre la planta debe hacerse alrededor de un punto de reposo o de funcionamiento de la planta. Asimismo este ensayo a de hacerse en escalón, es decir, generando escalones en una variable independiente

mientras las demás se mantienen constantes y de igual forma con el resto. Se aconseja realizar trenes de escalones como los de la figura en los ensayos.

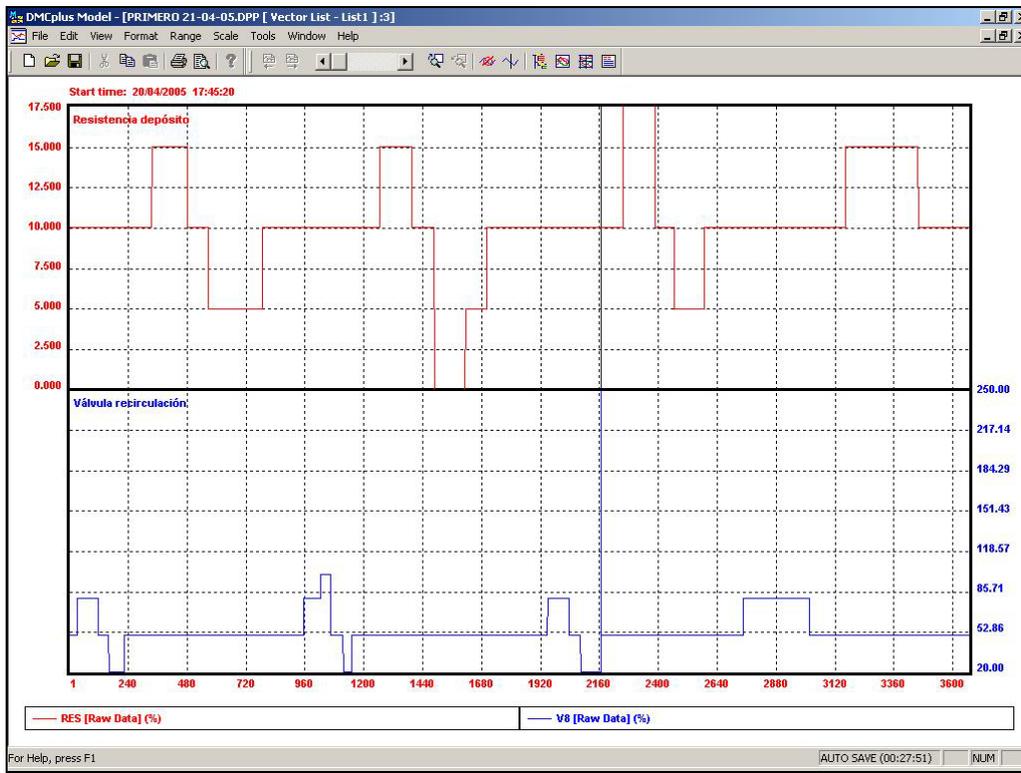


Figura 5.7.- Tren de escalones para el ensayo

### 5.2.2.- Extract

Una vez que tenemos los datos de la planta en formato binario ha de utilizarse la herramienta Extract para realizar un formateo de los datos de manera que puedan ser leídos por DMCplus Model.

Para hacer uso de Extract debemos abrir una sesión MS-DOS y situarnos en el directorio en el que se encuentre el archivo a extraer. Entonces ejecutamos la orden *dmcextract [nombre de archivo]*, siendo el nombre del archivo el utilizado en Collect. Al ejecutarse la orden anterior el programa nos guía de forma que podemos elegir los muestreos a extraer de entre los disponibles. Además nos crea dos nuevos archivos, uno de extensión .ext en el que se encuentran los tags de las variables muestreadas y otro de extensión .clc que contiene los datos formateados de la recolección.

```

plantapiloto
Recoleccion de datos de la planta piloto v.1
13
13
4-20-2005 17:45:20
2
3667
=====
TT1~~~~"::"PLANTA\TT1":REAL:~~~~
FT1~~~~"::"PLANTA\FT1":REAL:~~~~
V4~~~~"::"PLANTA\V4":REAL:~~~~
TT2~~~~"::"PLANTA\TT2":REAL:~~~~
FT2~~~~"::"PLANTA\FT2":REAL:~~~~
V5~~~~"::"PLANTA\V5":REAL:~~~~
TT4~~~~"::"PLANTA\TT4":REAL:~~~~
FT4~~~~"::"PLANTA\FT4":REAL:~~~~
V8~~~~"::"PLANTA\V8":REAL:~~~~
TT5~~~~"::"PLANTA\TT5":REAL:~~~~
LT1~~~~"::"PLANTA\LT1":REAL:~~~~
RES_D~~~~"::"PLANTA\RESIST_DIG":REAL:~~~~
RES~~~~"::"PLANTA\RESISTENCIA":REAL:~~~~
=====
4-20-2005 17:45:20,21.9038,G,-1.44072E-11,G,0.,G,17.2795,G,.306684,G,0.,G,20.2528,G,.705539,
G,50.,G,23.5203,G,95.2358,G,1.,G,10.,G
4-20-2005 17:45:22,21.8889,G,-6.90579E-12,G,0.,G,17.2611,G,.360478,G,0.,G,20.2573,G,.702078,
G,50.,G,23.5199,G,95.2322,G,1.,G,10.,G
4-20-2005 17:45:24,21.882,G,-1.58665E-12,G,0.,G,17.2426,G,.398622,G,0.,G,20.275,G,.704485,
G,50.,G,23.5203,G,95.2518,G,1.,G,10.,G
4-20-2005 17:45:26,-10000.,M,-10000.,M,-10000.,M,-10000.,M,-10000.,M,-10000.,M,-10000.,M,-
10000.,M,-10000.,M,-10000.,M,-10000.,M,-10000.,M,-10000.,M
4-20-2005 17:45:28,21.8805,G,-.174386,G,0.,G,17.2275,G,.248501,G,0.,G,20.2478,G,.701669,
G,50.,G,23.5206,G,95.2486,G,1.,G,10.,G
4-20-2005 17:45:30,21.8801,G,-.0687858,G,0.,G,17.2434,G,.346728,G,0.,G,20.2186,G,.707448,
G,50.,G,23.5218,G,95.2427,G,1.,G,10.,G
4-20-2005 17:45:32,21.8801,G,-.0329711,G,0.,G,17.2591,G,.379672,G,0.,G,20.2192,G,.708777,
G,50.,G,23.524,G,95.2517,G,1.,G,10.,G

```

**Figura 5.8.- Parte inicial del archivo .clc**

En el archivo .clc podemos observar varias secciones diferenciadas:

1. La sección primera contiene información sobre el archivo: nombre del archivo de configuración, descripción del mismo, número de variables muestreadas, cantidad por sección, momento inicial del muestreo, frecuencia y cantidad de muestreos realizados.
2. En la segunda sección nos encontramos con el mismo archivo que el .clc, en el cual está el listado de los tags extraídos.
3. La tercera y sucesivas secciones (aparecen automáticamente por limitaciones computacionales) contienen los datos tomados de la planta con el formato:

Tiempo,valor tag1,estado1,valor tag2,estado2,valor tag3,estado3,...

- Tiempo: MM-DD-AA hh:mm:ss
- Valor: dato tomado de la planta

- Estado: indica cómo se realizó el muestreo. Puede ser G (no hubo problemas), B (muestreo mal, valor -9999), M (pérdida de los datos, valor -10000) o S (se muestreó en un momento no debido)

### **5.3.- DMCplus Model**

Una vez que hemos obtenido los datos de la planta pasamos al modelado del sistema.

Mediante este programa podremos ver de una forma gráfica los datos extraídos de la planta para analizarlos y a partir de ellos realizar un modelo del proceso para utilizarlo más tarde en el controlador.

Para abrir el programa y comenzar a trabajar con él tenemos que tener activado en primer lugar el gestor de licencias y luego acudir a Inicio> Programas> AspenTech> Aspen Manufacturing Suite> DMCplus Desktop> Model.

#### **5.3.1.- Aspectos generales**

La interfaz del Model es la típica de los programas que trabajan en entornos Windows, por lo que su utilización es muy fácil e intuitiva, además de disponer de las herramientas típicas de este tipo de interfaces como copiar, pegar, nuevo, etcétera. También tendremos disponible el menú desplegable de Windows, el cual variará dependiendo de dónde lo situemos.

Esta interfaz está dividida en dos partes, la zona izquierda, llamada Report View, en la que tenemos un árbol del proyecto que estamos desarrollando y que nos permite un acceso rápido a todos los componentes de dicho proyecto; y la zona derecha, Vector/Model Plot Views, en la cual trabajaremos con los diferentes componentes.

A continuación se dan una serie de definiciones útiles para comprender el uso y funcionamiento de Model:

- Proyecto: podríamos identificarlo con un contenedor en el cual introducimos todas las cosas necesarias para la obtención del modelo que buscamos, como vectores, casos, predicciones. Un proyecto puede contener varios modelos de distintos procesos, aunque es recomendable que se abra un nuevo proyecto por cada proceso que deseemos modelar.
- Vectores: contiene los datos temporales de una variable.
- Casos: se tratan de especificaciones y pruebas para la identificación de un modelo. Contiene toda la información necesaria para ello.
- Modelos: se tratan de un modelo en el que se representan las curvas de respuesta de las variables dependientes ante escalones de las variables independientes.
- Predicciones: como indica su nombre son predicciones que realiza el programa de la respuesta que tendrá el proceso utilizando el modelo que hayamos indicado y como

señales de entrada las de los vectores que también hayamos elegido.

- Listas: son carpetas en las que podemos agrupar los distintos componentes según su tipo. También se permite la creación de listas generales en las que podemos agrupar componentes distintos.

### **5.3.2.- Pasos a seguir en el modelado**

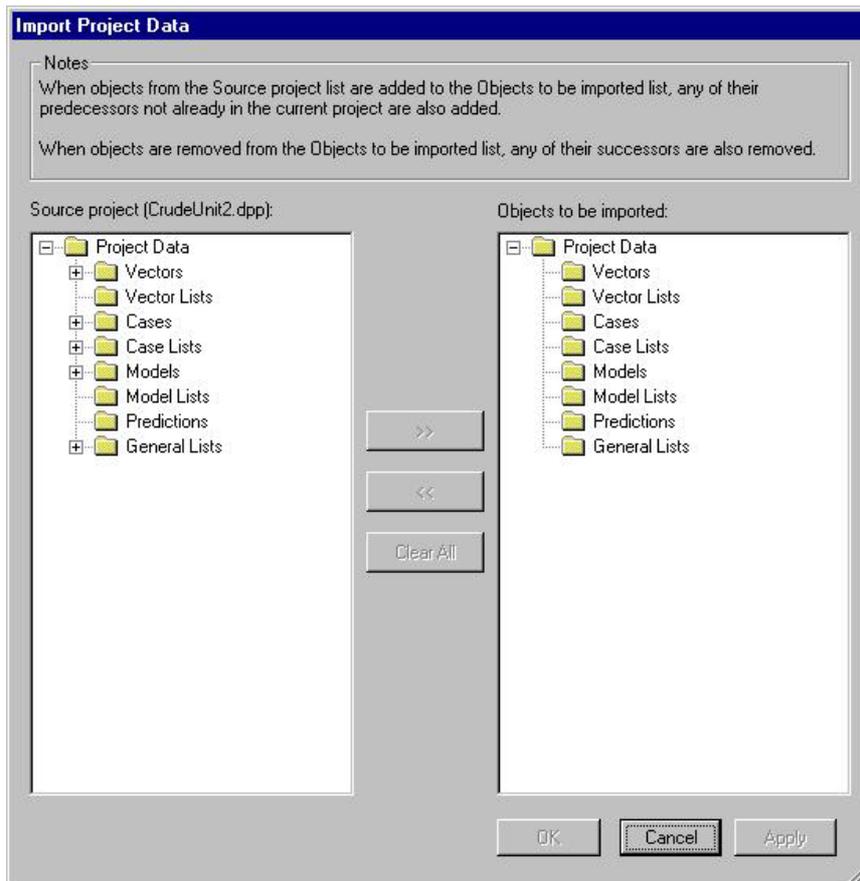
Se explicarán los pasos que han de darse para realizar el modelado de forma breve. Más adelante en los apartados correspondientes a cada componente del modelo se explicarán las características de cada uno de ellos.

- Al abrir un nuevo proyecto lo primero que hemos de hacer es obtener los datos tomados de la planta que se desea modelar importándolos del archivo .clc que se obtuvo a partir de Extract.
- Obtenidos los vectores de trabajo realizamos una inspección de los mismos para eliminar posibles errores y seleccionar los intervalos temporales que nos sean más convenientes. También se podrán cambios en los vectores.
- Tras el análisis de los vectores de datos pasamos a generar distintos Casos. Cada uno de éstos nos proporcionará varios modelos para las variables seleccionadas y según los parámetros que le marquemos, además de la posibilidad de realizar una predicción de uno de los modelos.
- A partir de los modelos obtenidos en los casos anteriores o bien de forma manual, pasaremos a generar el modelo que nos represente nuestro proceso. Para ello iremos añadiendo curvas que relacionen pares de variables independientes con dependientes hasta conseguir el modelo deseado. También en este caso tendremos la posibilidad de utilizar predicciones del modelo para comprobar que genera señales como las que se obtuvieron en el ensayo del proceso.
- Una vez tengamos el modelo más aproximado al proceso hemos de exportarlo en formato .mdl para que esté disponible para su utilización por DMCplus Build a la hora de configurar el controlador.

### **5.3.3.- Proyectos**

Para empezar tenemos que crear un proyecto nuevo. Para ello hacemos File> New Project o simplemente pulsamos en el botón de New Project, apareciéndonos un proyecto con un nombre por defecto que podremos cambiar. También es posible abrir

un proyecto existente (extensión del archivo .dpp) e incluso importar partes de otro. Para esto último nos es de gran ayuda la función de importación de proyectos, la cual nos permite copiar objetos de un proyecto en uno nuevo, eliminando la necesidad de copiar objetos de uno en uno. El acceso a esta función es mediante File> Import> Project. El aspecto de la interfaz de la función es el de la figura 5.9, y la forma de importar los objetos es seleccionando del proyecto fuente y trasladarlos al proyecto nuevo.



**Figura 5.9.- Función de importación de proyectos**

Asimismo es posible la exportación de objetos del proyecto actual simplemente situándonos encima de uno o varios objetos y desplegando el menú contextual, del cual elegimos la opción de exportar.

Por otro lado, la manera de guardar un proyecto es mediante el comando típico de Windows o bien programando para ello un autoguardado cada cierto tiempo, lo cual es muy recomendable para evitar la pérdida de modificaciones que se hayan realizado. El acceso a esta opción se realiza mediante Project> Options.

En las opciones del proyecto pueden realizarse además del autoguardado varias acciones más como:

- Utilizar gráficos de características extendidas: simplemente aumenta la capacidad de definición de los gráficos, aunque con una pérdida considerable de velocidad de procesado.
- Mensajes de confirmación en el autoguardado.
- Permitir la compresión de datos.
- Tolerancia de frecuencia para la unión de vectores.
- Tolerancia de comienzo para la unión de vectores: esta característica y la anterior son utilizadas a la hora de juntar vectores sin la misma frecuencia de muestreo ni el mismo momento de comienzo, por lo que se establece un porcentaje de tolerancia.

Por último, a continuación se muestran los distintos tipos de archivos que pueden coexistir en un proyecto:

- Proyecto  $\Rightarrow$ .dpp
- Vectores  $\Rightarrow$
- Vector ampliado  $\Rightarrow$ .dpv
- Listas de vectores  $\Rightarrow$ .vls
- Listas de importación de vectores  $\Rightarrow$ .lst
- Casos  $\Rightarrow$ .cas
- Listas de casos  $\Rightarrow$ .cls
- Modelos  $\Rightarrow$ .mdl
- Modelos ensamblados  $\Rightarrow$ .dpa
- Listas de modelos  $\Rightarrow$ .mls
- Listas generales  $\Rightarrow$ .gls
- Bandas de datos (ver en el apartado de vectores)  $\Rightarrow$ .dsl
- Archivos de fórmulas  $\Rightarrow$ .dpf
- Archivos de predicciones  $\Rightarrow$ .prd

### **5.3.4.- Vectores**

Como ya se ha indicado, los vectores son las unidades que contienen los datos con los que vamos a trabajar. Por ello es lo primero que necesitamos para comenzar con nuestro trabajo.

Dependiendo del tipo de datos que contengan podemos distinguir distintos tipos de

vectores como por ejemplo:

- Datos (Raw Data): nos referimos a estos vectores como aquellos que contienen datos tomados directamente del proceso sin modificaciones, salvo la posible interpolación de datos realizada cuando importamos los vectores y utilizada para evitar los datos clasificados como erróneos en la extracción desde la planta. Pueden modificarse y guardarse con el mismo nombre, aunque es recomendable utilizar otro nombre y extensión para poder distinguirlos más tarde.
- Vectores de cálculo: se tratan de vectores que provienen de vectores de datos a los cuales se les ha aplicado algún tipo de cálculo y por lo tanto han quedado modificados. Pueden utilizarse de la misma forma que los anteriores, ya que la única diferencia entre ellos es la fórmula aplicada a los últimos.
- Transformaciones: son linealizaciones estándar de DMCplus. Se pueden aplicar a los vectores anteriores. Tienen la particularidad de que no son en sí vectores independientes sino que se tratan como una cualidad del vector sobre el que se aplica y afecta en todos los lugares en los que se haga uso del vector al que se le ha aplicado la transformación.

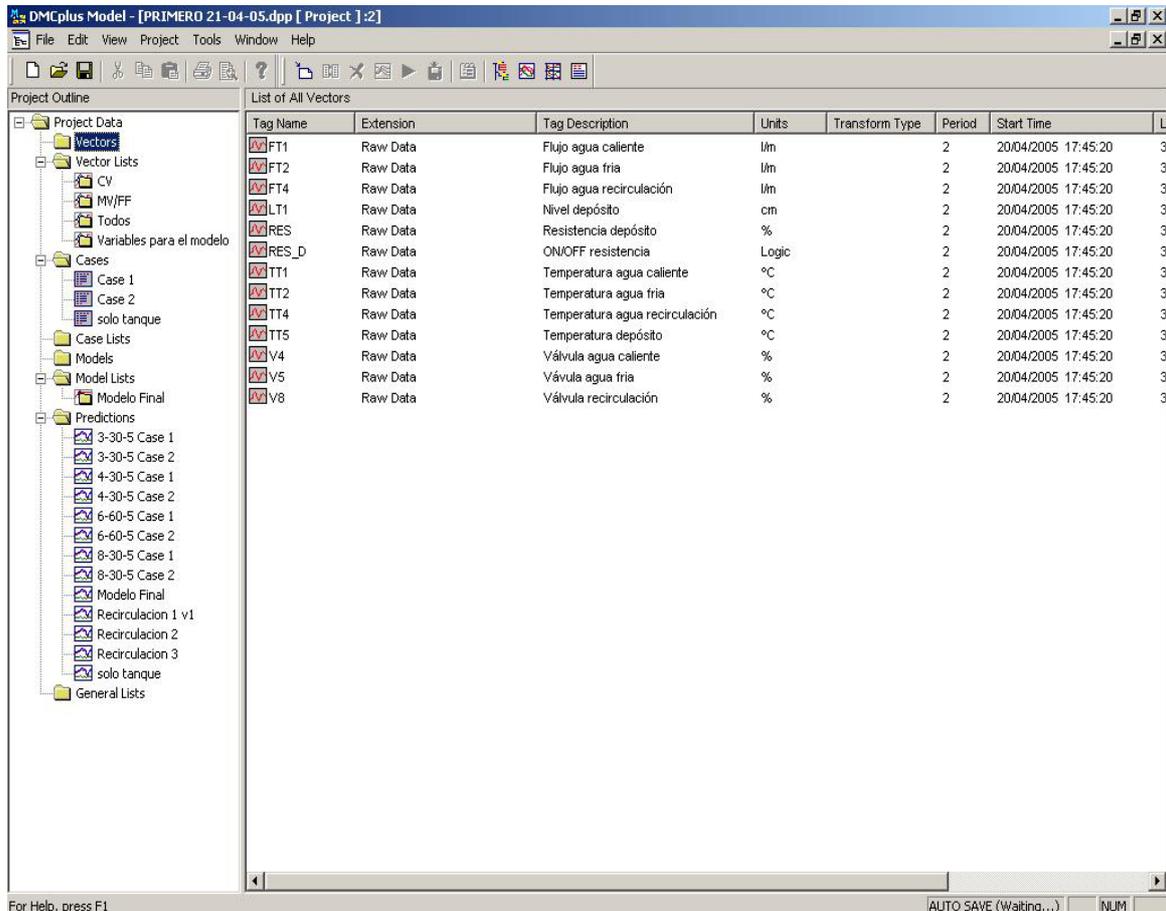


Figura 5.10.- Pantalla de vectores

### 5.3.4.1.- Importar vectores

La forma de importar vectores es acudir al menú File> Import> Vectors. Así se nos mostrará el diálogo de importación de vectores en el cual especificaremos el tipo de vectores que queremos importar así como su ubicación. Si, como es natural al principio de un proyecto totalmente nuevo, vamos a utilizar los datos tomados de la planta en un archivo .clc, buscaremos éste y lo abriremos.

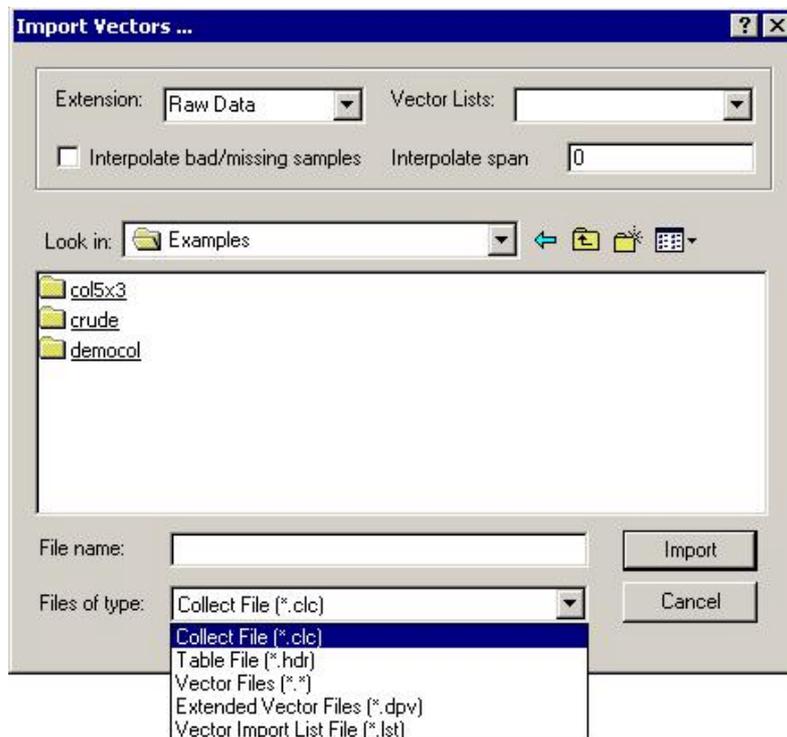


Figura 5.11.- Diálogo de importación de vectores

Podemos observar en la figura 5.11 que se nos ponen a nuestra disposición dos opciones para la importación de los vectores:

- Interpolación de datos erróneos o perdidos: se utiliza para eliminar de los vectores los datos que se han clasificado como tales en la extracción de los mismos. Para no dejar el “hueco” vacío se interpola con los datos contiguos.
- Extensión de la interpolación: se trata de los minutos de la muestra que se utilizarán para la interpolación. Por defecto son 5, pero hay que tener en cuenta que a mayor extensión peor será la identificación del modelo.

Una vez seleccionado el archivo .clc podremos visualizar los datos que estén disponibles en dicho archivo, como vemos en la figura 5.12. Ésto nos permitirá seleccionar los que deseemos, así como las muestras que queramos.

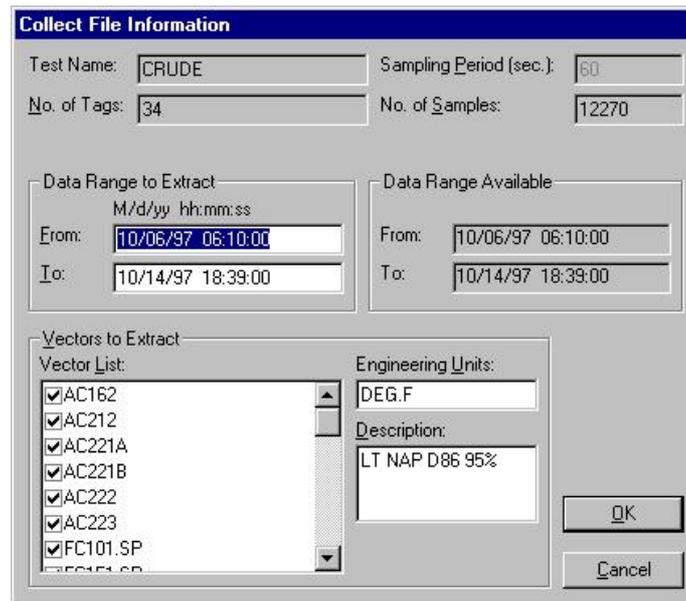


Figura 5.12.- Información de los datos disponibles

### **5.3.4.2- Trabajar con los vectores**

Ahora ya tenemos los vectores con los que vamos a trabajar. Para visualizar los datos que contienen (únicamente de forma gráfica) simplemente nos situamos sobre el vector o vectores que queramos ver, estando a nuestra disposición tres formas de visualizarlos:

- Doble clic con el ratón.
- Desplegar el menú contextual y elegir la opción Plot.
- Utilizar el botón de la barra de herramientas Plot.

Entonces se nos abrirá una ventana con los gráficos de los vectores que hayamos elegido.

Ésta nueva ventana puede configurarse para mostrar los datos como más nos guste. Estas configuraciones son accesibles a través de los menús existentes en la parte superior de la ventana de los gráficos. Algunas de estas opciones son las siguientes:

- Número de gráficos a mostrar por ventana (usar menú desplegable sobre los gráficos).
- Rango de datos a mostrar. Podemos ver el vector completo o bien por partes.
- Escala. Es posible visualizar todos los vectores en un mismo gráfico si establecemos los valores mínimos y máximos de cada vector.
- Formato, entendiendo por formato la escala temporal del vector, ésto es, podemos ver el gráfico bien por muestras o bien por su escala temporal.
- Otros. El programa permite establecer colores, los márgenes de impresión de los gráficos, títulos, leyendas y demás.

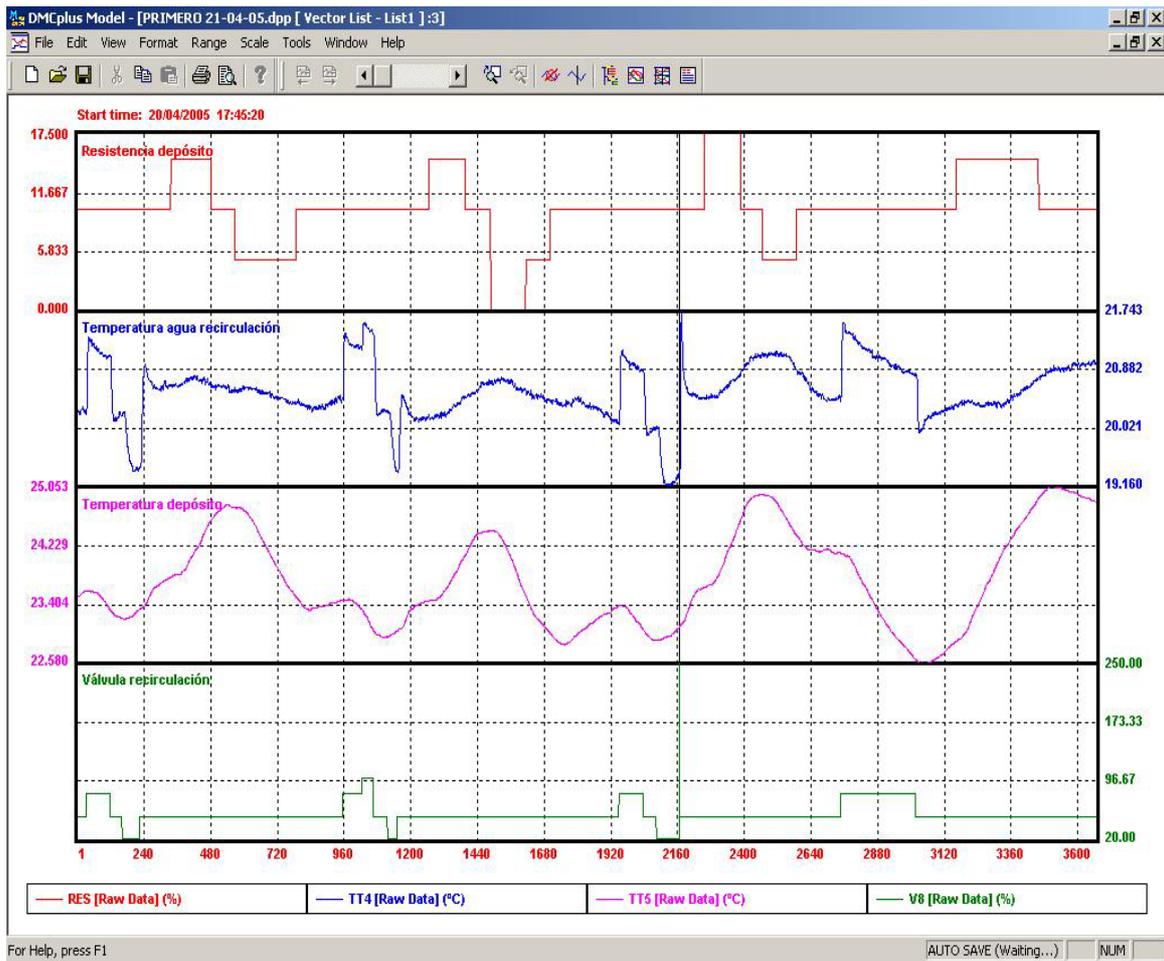


Figura 5.13.- Gráficos de vectores de datos

Otra herramienta muy útil para el análisis de los vectores es el zoom. Para agrandar zonas del gráfico basta con seleccionar la banda a ampliar, mediante un clic de ratón en un punto y arrastrando hasta el siguiente punto que deseemos, y seguidamente presionar el botón de la barra de herramientas Zoom In (Zoom Out para lo contrario), aunque también es accesible a través del menú contextual.

#### MARCAJE DE DATOS ERRÓNEOS:

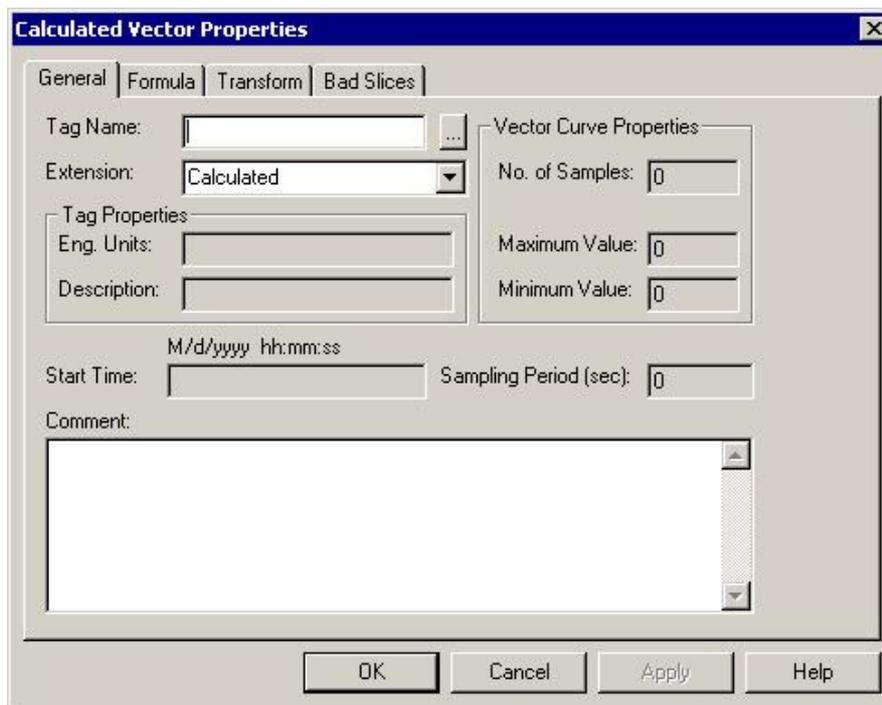
Los datos erróneos o Bad Slices, son bandas de datos que podemos marcar manualmente si detectamos que en el vector que estamos visualizando existen datos que pudiéramos clasificar como erróneos (movimientos extraños en las variables, datos no esperados u otros). Los datos que marquemos como Bad Slices no serán tenidos en cuenta a la hora de realizar la identificación del proceso aunque no serán borrados del vector.

La señalización de una banda de datos como un Bad Slice se hace seleccionando la zona deseada y utilizando luego el botón Mark Bad. Entonces la banda seleccionada tomará un color gris que indicará que ha sido marcada como Bad Slice.

Estos Bad Slice pueden ser locales, afectando únicamente al vector en el que se ha marcado, o bien pueden tomarse como globales, con lo cual todos los vectores del proyecto se verán afectados por dicho Bad Slice.

### VECTORES DE CÁLCULO:

Como ya se ha comentado, a los vectores de datos se le pueden aplicar una serie de funciones que nos permitirán transformar los datos que contengan según nuestras necesidades. Para crear un vector de cálculo nos posicionamos en la zona de vectores y seleccionamos el menú Project> New Vector, apareciéndonos entonces el panel de propiedades de los vectores calculados.



**Figura 5.14.- Panel de propiedades de los vectores de cálculo**

En este panel hemos de elegir un nombre para el nuevo vector y, lo más importante, la fórmula que vamos a aplicar.

Model pone a nuestra disposición una serie de fórmulas predeterminadas que podemos utilizar tal y como están, aunque existe la posibilidad de crear nuevas fórmulas a partir de las predeterminadas.

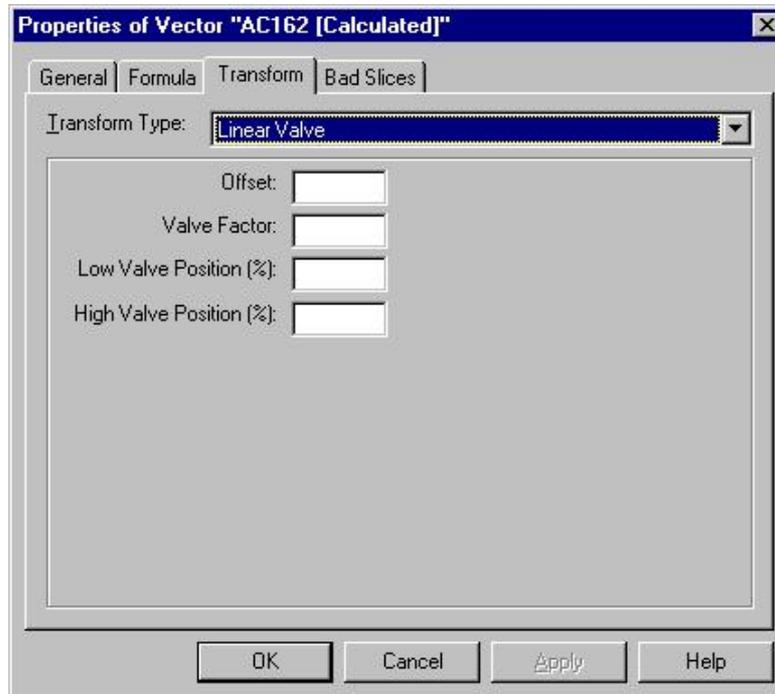
Algunas de las fórmulas predeterminadas son las siguientes (ver el resto en el Anexo I):

- @Log(value) Logaritmo natural de un valor real positivo
- @Abs(value) Valor absoluto de un valor
- @Diff(vector) Diferencia de un vector:  $out(1) = 0, out(i) = in(i) - in(i-1)$
- @Sqrt(value) Raíz cuadrada de un valor positivo real

### TRANSFORMACIÓN DE VECTORES:

El último cambio que puede aplicársele a un vector son las transformaciones lineales estándar que nos proporciona Model. Sin embargo, estas transformaciones no se traducen en la creación de un nuevo vector sino que son tratadas como una característica interna del propio vector y sólo es posible aplicar una transformación cada vez. Para aplicar otra transformación sin perder la que ya se tiene es necesario crear una copia del vector con el que estemos trabajando y aplicar sobre esta copia la nueva transformación. Así, cuando a un vector se le aplica una transformación, todos los objetos o componentes del proyecto en el que esté presente dicho vector se verán afectados por los cambios que produzca la transformación.

El acceso a las transformaciones de los vectores se realiza mediante la apertura de las propiedades del vector a través del menú contextual.



**Figura 5.15.- Transformación de un vector**

Las transformaciones que están disponibles son las siguientes:

- Válvula lineal: definición dada en 'Final Control Elements topic in Section

22 - Process Control of Perry and Chilton's Chemical Engineer's Handbook, Fifth Edition'

- Válvula parabólica: definición dada en 'Final Control Elements topic in Section 22 - Process Control of Perry and Chilton's Chemical Engineer's Handbook, Fifth Edition'
- Logaritmo natural
- Logaritmo en base 10
- Logaritmo natural modificado
- Logaritmo en base 10 modificado
- Shift Rate Power:  $y = b(x + a)^c$
- Piece-Wise Linear:  $y = \frac{x - x_i}{x_j - x_i} \cdot (y_j - y_i) + y_i$

#### PROPIEDADES DEL VECTOR:

Hablamos en este caso de las propiedades que nos definen las características del vector con el que trabajamos. Estas propiedades son el nombre del vector (Tag name), las unidades ingenieriles, el movimiento típico (generalmente el movimiento en el ensayo de la planta) y su descripción.

Accedemos a estas propiedades mediante el menú Edit> Tag Properties.



Figura 5.16.- Propiedades de los tag

#### **5.3.5.- Casos**

En general un caso incluye la siguiente información:

- Una lista de variables independientes.
- Otra con variables dependientes.
- El algoritmo de identificación que va a utilizarse, e información que necesite dicho algoritmo como el tiempo de establecimiento o Steady-State y número

de coeficientes del modelo.

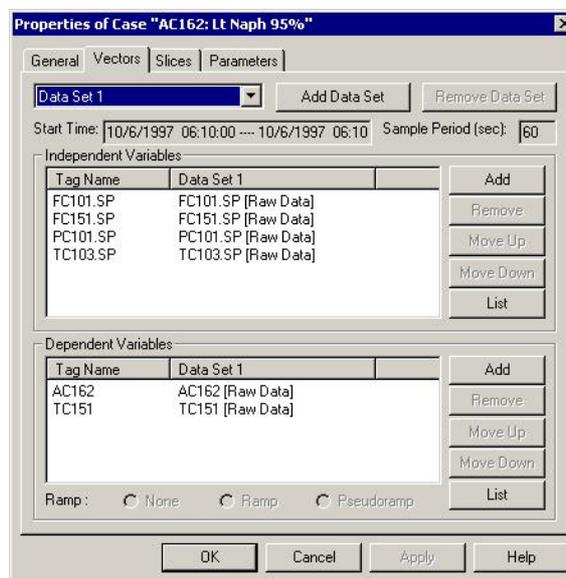
- Frecuencia del muestreo en segundos
- Bandas de datos de los vectores a usar en la identificación.

Crearemos un caso una vez que hayamos analizado los vectores, marcado los Bad Slice y realizado los cálculos y transformaciones que fuesen necesarias. El resultado que nos proporciona un caso es una serie de modelos de salida, cada uno de ellos con las características del algoritmo elegido y sus parámetros. Estos modelos contienen una serie de curvas de respuesta, correspondientes cada una de ellas a un par de variables independiente/dependiente.

Las curvas que obtengamos de los casos serán las que utilizemos para crear los modelos que se utilizarán finalmente, por lo que este paso es bastante importante en el modelado del proceso.

### **5.3.5.1.- Creación y ejecución de un caso**

Para crear un caso nos situamos en la carpeta Cases y abrimos el menú Project> New Case o bien nos situamos en el panel de Cases y utilizamos el menú contextual. Entonces se nos abrirá un diálogo en el que especificaremos las características de dicho caso.



**Figura 5.17.- Diálogo de propiedades de un caso**

En este diálogo nos encontramos con cuatro pestañas cuya edición es obligatoria. A continuación se detalla la edición de las pestañas:

- General:
  - Case Name: se elegirá un nombre para el caso en cuestión que no exista previamente.
  - Comments: no es obligatorio rellenar este apartado, aunque sí muy recomendable para identificar el caso de forma rápida.
- Vectors:
  - Data Set: los Data Set contienen datos de las variables en forma de vectores pero provenientes de diferentes ensayos. En principio no es necesario su uso puesto que con los vectores que ya tenemos es suficiente, por lo que podemos dejar el Data Set 1 que se tiene por defecto.
  - Independent Variables: hemos de añadir los nombres de las variables que serán las variables manipulables o de perturbación (llamadas más adelante Feedforward) del modelo.
  - Dependent Variables: además de añadir este tipo de variables para el modelo tenemos la posibilidad de especificar para cada una de ellas el tipo de variable del que se trate (rampa, pseudo rampa o normal), algo que luego tiene una gran repercusión en el modelo y el controlador a diseñar.
- Slices:
  - Data Set: como ya se indicó anteriormente podemos dejar el nombre por defecto.
  - Good Slices: son los tramos de datos correctos que utilizaremos para la identificación. Tenemos dos formas de especificar estos tramos: introducimos manualmente la muestra inicial y final cuando presionemos el botón Add, o bien utilizamos Rebuild, que nos suministra todos los tramos disponibles y que no se hubieran marcado como Bad Slice.  
Además tenemos la posibilidad de permitir a Model que siempre recalculé automáticamente estos Good Slices, así si añadimos algún Bad Slice, éste no será tenido en cuenta.
- Parameters:
  - Identification Parameters: hemos de añadir todas las identificaciones que deseemos realizar, presionando para ello el botón Add. Entonces tendremos que elegir el tipo de algoritmo a utilizar.

También es posible generar una predicción por caso, del comportamiento del modelo identificado usando los vectores reales de las variables independientes, permitiéndonos observar si el modelo se ajusta a los datos tomados de la planta.

Una vez que tengamos creado el caso lo ejecutaremos seleccionándolo y pulsando el botón de Run. Al ejecutar el caso nos aparecerá una ventana que nos irá indicando si la identificación se está llevando a cabo de forma favorable. Si todo transcurre bien obtendremos una serie de modelos (dos por cada identificación que especificáramos) en los cuales podremos ver la curva de respuesta para cada par de variables que hubiésemos especificado en el caso.

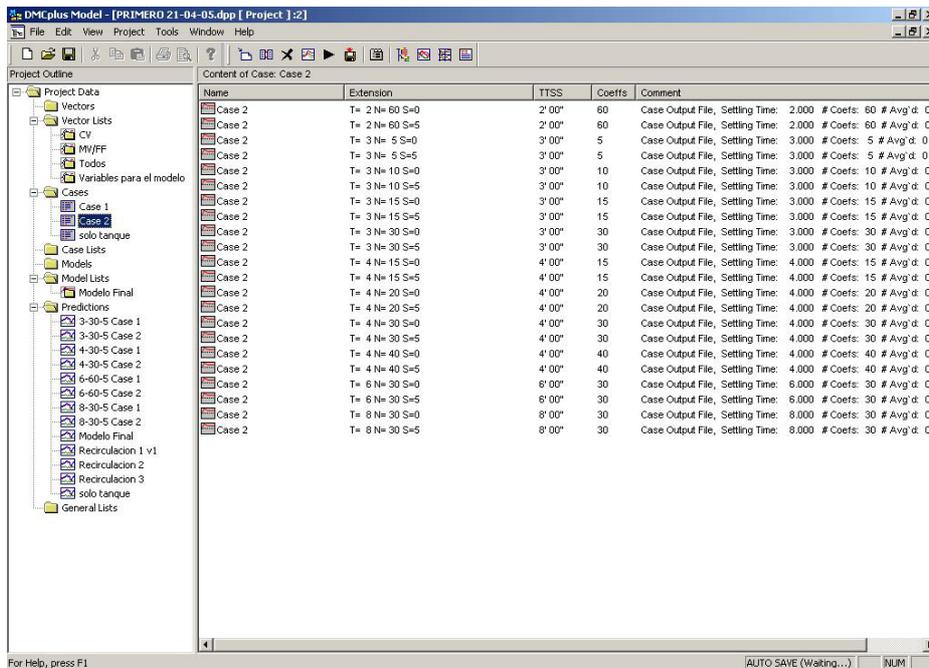


Figura 5.18.- Pantalla de casos ejecutados

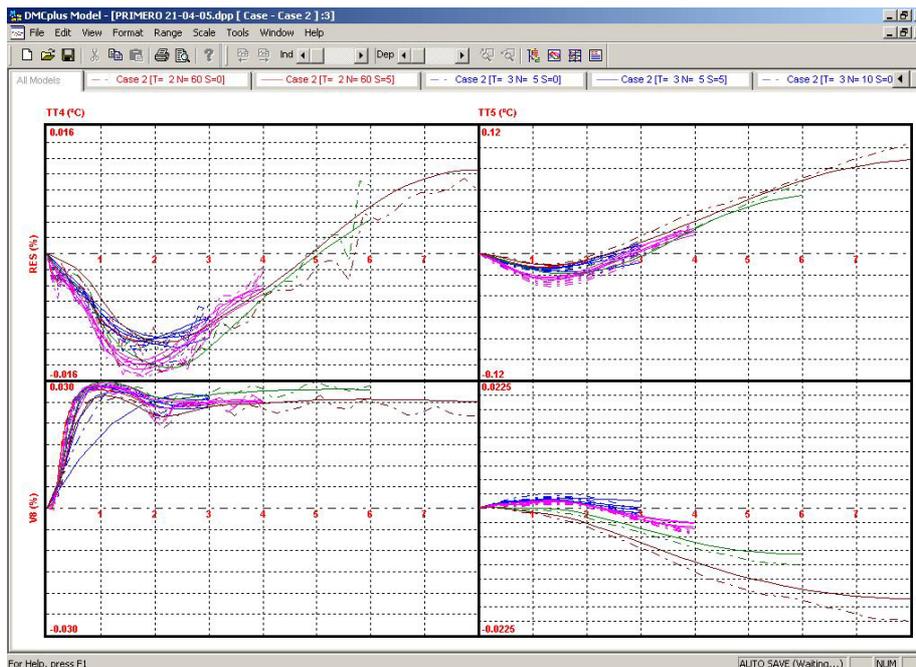


Figura 5.19.- Modelos obtenidos con varias identificaciones en un caso

Por último comentar la posibilidad de importar casos de otros proyectos existentes así como también es posible la exportación de los que nosotros creamos.

### **5.3.5.2.- Algoritmos de identificación**

#### **FINITE IMPULSE RESPONSE (FIR)**

Se trata del algoritmo de identificación típico del Control Predictivo basado en modelo (ver el capítulo 2). Se trata de un algoritmo muy simple y que permite una identificación rápida del proceso incluso si éste tiene retardos o no se tiene ningún conocimiento previo del proceso. Sin embargo no es aplicable a procesos inestables.

Cuando elegimos este algoritmo de identificación, Model nos preguntará por sus propiedades, siendo éstas:

- Time to Steady State: se trata del tiempo en minutos que queremos que tarde el modelo en alcanzar un estado de reposo.
- Number of Coefficients: debemos especificar el número de coeficientes que deseamos tenga el modelo. A mayor número de coeficientes se exigirá más tiempo de cálculo y de Steady State, aunque ganaremos en exactitud.
- Smooth factor: éste es un factor que permite que el modelo suavice sus respuestas. Con un número alto suavizaremos bastante el modelo pero perderemos exactitud. Al ejecutar un caso el programa nos proporciona un modelo sin suavizar y otro suavizado con el factor especificado.

#### **SUBSPACE IDENTIFICATON**

Este algoritmo ofrece varias ventajas y desventajas respecto al anterior como pueden ser:

- Utiliza un modelo en el espacio de estado para representar el proceso capaz de obtener tanto la baja como la alta frecuencia, siendo por lo tanto capaz de modelar procesos de dinámica compleja.
- Está indicado particularmente para procesos MIMO, que necesitan controladores muy ajustados.
- No está muy implantando demasiado, con los problemas de fiabilidad que ello conlleva.
- Necesita mayor cantidad de cálculo que el FIR.

Si elegimos este algoritmo nos aparecerá al igual que con el anterior una ventana de diálogo para especificar los parámetros, siendo éstos:

- Time to Steady State: igual que con el FIR.
- Maximum Order: se trata del máximo orden que estamos dispuestos a utilizar en nuestro modelo del proceso.

Si además activamos el botón Expert tendremos más opciones para configurar la identificación por este algoritmo, permitiendo exportar a las matrices en el espacio de estados a un archivo de texto y realizar un preprocesado a los vectores antes de utilizarlos en la identificación. Este preprocesado no genera un nuevo vector aunque sí se permite su visualización.

- Differencing: usa derivadas de las variables manipulables y de las variables controlables para la identificación. En modelos normales y con rampa.
- Detrending: usa datos a los que se les elimina parte de su tendencia, utilizable en ambos tipos de modelos.
- Zero Mean: elimina la media de los dos tipos de variables.
- Double Diff: utiliza Differencing en las variables controlables (CV) y Zero Mean con las manipulables (MV) para identificar modelos normales y luego los integra para obtener el modelo en rampa.

### **5.3.5.3.- Utilidades para los casos**

Model nos proporciona dos utilidades para poder realizar un modelado lo más aproximado a la realidad posible. En este apartado se indica para que sirven ambas.

Para acceder a ambas utilidades hemos de situarnos sobre un caso y seleccionar el menú Project.

### **MODEL UNCERTAINTY**

Se trata de una herramienta de validación para evaluar la calidad de un modelo identificado frente a los datos del ensayo de la planta. Así, mientras realizamos el análisis de los datos y del modelo, la información de la incertidumbre del modelo puede ayudarnos a determinar qué variables requerirán más ensayos y la frecuencia necesaria de los mismos para mejorar los modelos existentes.

Esta herramienta proporciona una serie de matrices en archivos de texto con la información. Para tratar esta información existen unas plantillas Excel en la web de

soporte técnico de Aspen. Sin embargo, en la nuevas versiones o actualizaciones de DMCplus esta utilidad viene completamente integrada en el programa y es de fácil uso.

### CROSS-CORRELATION

La herramienta Cross-Correlation se utiliza para ver las relaciones que se establecen entre las variables del modelo. La información proporcionada por esta utilidad puede ser utilizada para validar las relaciones entre variables identificar variables manipulables correladas.

Al igual que Model Uncertainty, Cross-Correlation genera un fichero de texto con toda la información, aunque es posible obtener una serie de plantillas Excel para tratar dichos datos.

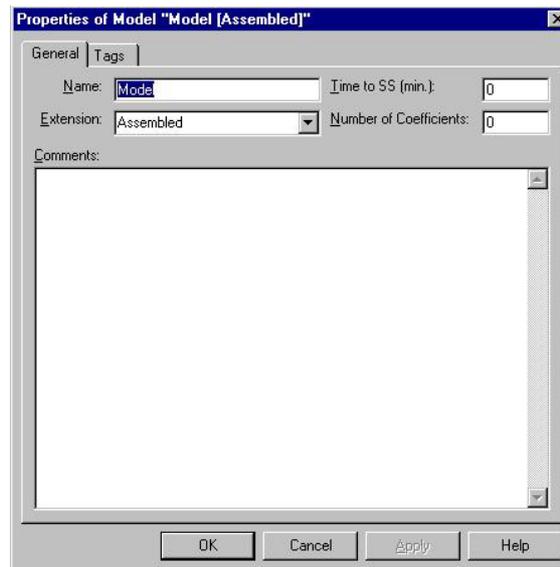
### **5.3.6.- Modelos**

Estos objetos modelan la dinámica del proceso. Se utiliza para generar y mantener una predicción de las variables a controlar.

Como ya hemos sabemos, los casos nos generan una serie de modelos de salida. Una vez que tenemos dichos modelos, los analizaremos y decidiremos qué curvas son significantes y cuales no. Si es necesario es posible realizar convoluciones de modelos para obtener relaciones que no es posible determinar de manera directa.

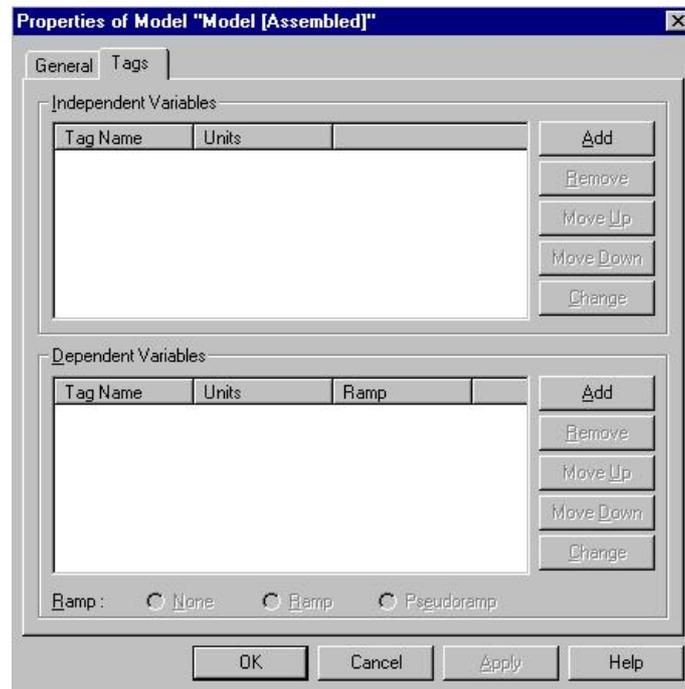
#### **5.3.6.1.- Creación de un modelo**

Cuando creamos un modelo, por defecto se toma que será un modelo Assembled, aunque podemos también importarlo. Para crearlo hemos de situarnos en la capeta Models y pulsar el botón de la barra de herramientas New o bien usar el menú contextual. Entonces nos aparecerá la ventana de propiedades del modelo, figura 5.20.



**Figura 5.20.- Propiedades del modelo**

En este diálogo deberemos introducir el nombre del modelo, una descripción del mismo y, de forma obligatoria, un tiempo de establecimiento o Steady State y el número de coeficientes que va a tener el modelo. Seguidamente pasamos a elegir las variables del modelo marcando la pestaña Tags.



**Figura 5.21.- Elección de variables para el modelo**

En esta ventana elegiremos el nombre de las variables que queremos de entre las que estén disponibles. Podemos observar que esta ventana es similar a la que nos aparecía al

crear un caso. Una vez que hemos elegido validamos y ya tendremos creado nuestro modelo.

Si abrimos el modelo recién creado veremos que no tiene ninguna curva de respuesta, por lo que aún debemos seguir trabajando con él. Tenemos dos formas de añadir curvas al modelo recién creado:

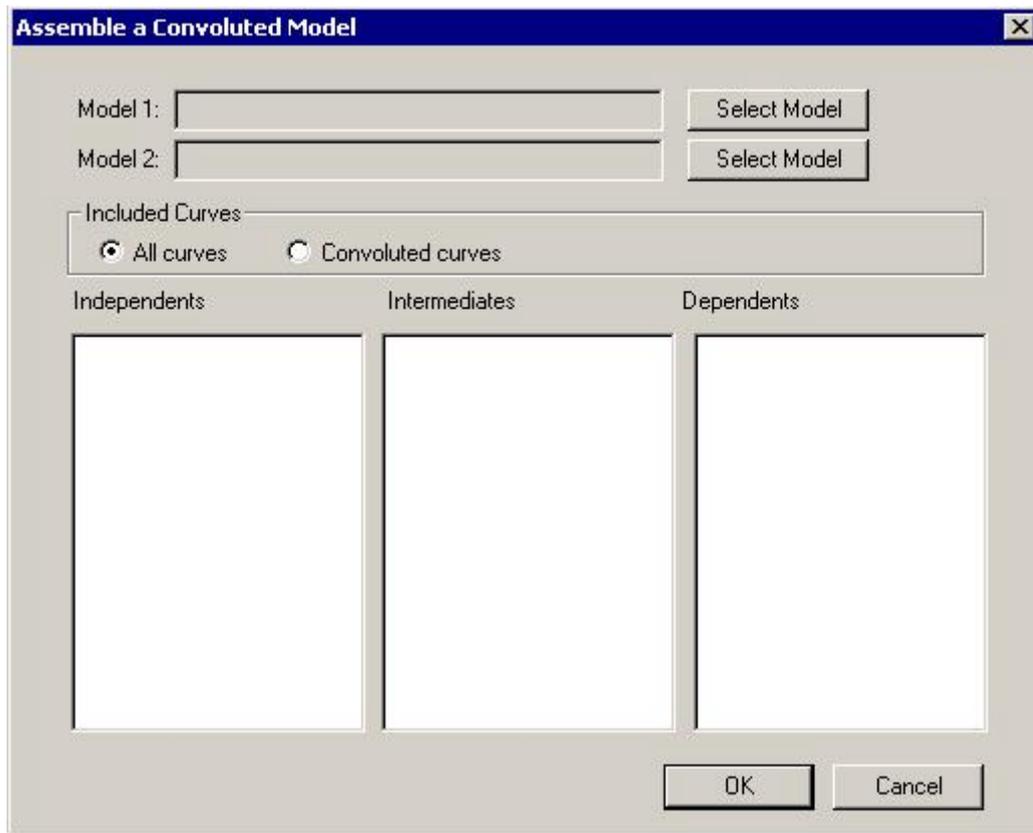
A. Arrastrar un modelo obtenido en la ejecución de un caso o un modelo existente hacia la ventana en la que tengamos nuestro modelo en blanco, con lo que las curvas del primer modelo serán añadidas en el nuestro. Esto puede hacerse curva a curva si seleccionamos únicamente una de las curvas del modelo (picamos dos veces con el ratón en la curva que queramos), lo que nos permite usar curvas de modelos distintos, o bien añadir el modelo al completo.

El añadido de curvas de forma independiente puede hacerse de forma más transparente si una vez seleccionada la curva usamos Edit> Get Curve, eligiendo entonces la curva fuente y la de destino.

B. Crear nosotros mismos las curvas de respuesta utilizando para ello las operaciones (Operation Curves) que pone a nuestra disposición Model. Basta con seleccionar el par de variables al que queremos añadir una curva y usar las Curve Operations. Más adelante se explicarán estas operaciones con más detalle.

### MODELOS DE CONVOLUCIÓN

Ya hemos comentado la posibilidad de realizar convoluciones de modelos para obtener relaciones entre variables que no son posibles de determinar de forma directa. Para realizar estas convoluciones debemos disponer de dos modelos creados previamente. Así, nos situamos encima de un modelo existente y desplegamos el menú contextual. Elegimos entonces la opción Assemble Convolute Model, lo cual nos lleva a una nueva ventana como la de la figura 5.22. En esta ventana tendremos que elegir el segundo modelo que se usará en la convolución, ya que el primer modelo será el que teníamos elegido al principio.



**Figura 5.22.- Creación de un modelo de convolución**

Una vez que se tienen seleccionados los modelos y las variables basta con validar, obteniendo el nuevo modelo de convolución.

### **5.3.6.2.- Trabajar con modelos**

#### **SUSTITUIR CURVAS**

Esta función podemos utilizarla para cambiar una curva existente por otra que le venga mejor a nuestro modelo final. Su funcionamiento es exactamente igual al de la creación de un modelo nuevo explicada en el apartado A.

Para cambiar una curva por otra simplemente hemos de seleccionar la curva a cambiar de nuestro modelo y acudir al menú Edit> Get Curve. Entonces nos aparecerá un diálogo en el que se nos pedirá información de la curva a insertar y de la curva a eliminar.

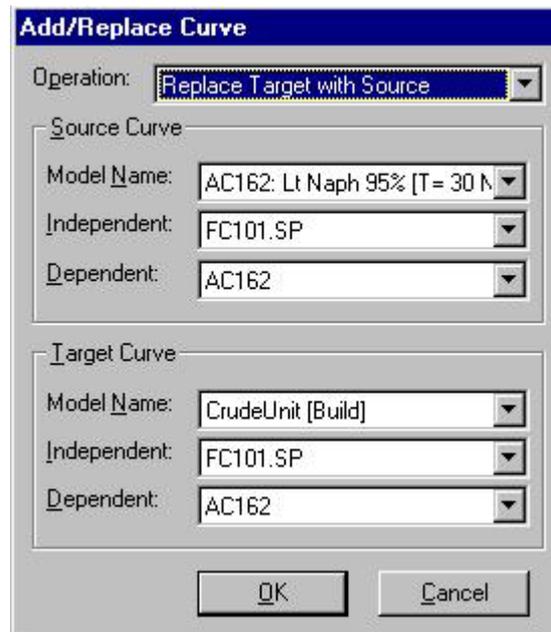


Figura 5.23.- Diálogo para la sustitución de curvas

Esta función también nos permite sumar la curva fuente a objetivo o bien sustraerla, todo ello según la opción que elijamos en el menú desplegable Operation del diálogo.

### CURVE OPERATIONS

Se trata de una serie de funciones que nos permiten modificar la forma de las curvas de respuesta que tengamos en nuestro modelo. Caben distinguir dos formas de operación:

- Cuando no existe aún una curva de respuesta para un par de variables: nos permite añadir curvas de respuesta estándar utilizando los parámetros que indiquemos. Estas funciones son reemplazar, curva cero, unidad, primer orden, segundo orden y convolución.
- Existe una curva de respuesta para el par de variables seleccionadas: entonces lo que se nos permite es modificar la curva existente añadiendo una serie de curvas estándar o modificaciones. Estas funciones son añadir, restar, ganancia, ganancia escalada, mover, multiplicar, tasa, tasa escalada, primer orden, segundo orden, 'leadlag' y 'rotate'.

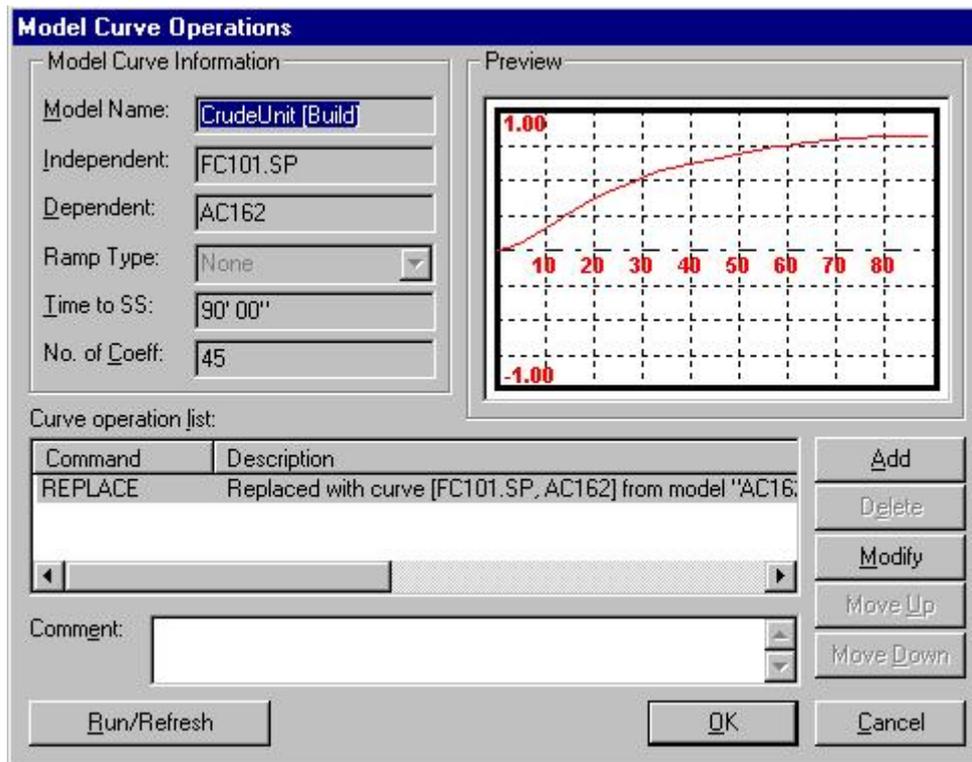


Figura 5.24.- Curve Operations

Para acceder a Curve Operations tenemos que seleccionar la curva de respuesta de un par de variables y dirigirnos al menú Edit o bien utilizar el menú contextual. Una vez que nos aparezca una ventana como la de la figura 5.24 simplemente hemos de añadir las curvas que nos parezcan convenientes.

Hay que tener especial cuidado con el orden de las curvas o funciones que se utilizan, ya que éste influye en el resultado final. Para ayudarnos a comprobar el resultado de los cambios que introduzcamos es de especial interés el uso del botón Run/Refresh, que nos permite ver los cambios realizados sin necesidad de validar la modificación.

Las funciones reseñadas anteriormente se corresponden con las siguientes referencias en el programa:

- REPLACE (reemplazar): se usa para insertar una curva externa al modelo actual.
- ZERO (curva cero): pone todos los coeficientes de la curva a cero.
- UNITY (unidad): ídem al anterior pero con valor unidad.
- CONVOLUTE (convolución): ya se ha comentado anteriormente.
- FIRSTORDER (primer orden): añade una curva de respuesta de primer orden según los parámetros que especifiquemos.
- SECONDORDER (segundo orden): ídem al anterior pero de segundo orden.

- ADD (añadir): lee una curva de un modelo existente y la añade a la curva seleccionada.
- SUBTRACT (restar): ídem al anterior pero sustrayendo.
- GAIN (ganancia): a partir del momento en el que especifiquemos y continuando hasta el final, hace que los coeficientes lleguen de forma lineal a un valor que indiquemos.
- GAINSCALE (ganancia escalada): especificamos un valor al que llegará la curva al final y los coeficientes de la curva serán multiplicados por (ganancia especificada / ganancia existente).
- SHIFT (mover): mueve una curva hacia la derecha o la izquierda en el tiempo (según el valor dado sea positivo o negativo).
- MULTIPLY (multiplicar): multiplica todos los coeficientes por el valor dado.
- RATE (tasa): modifica los coeficientes, desde el momento que digamos y hasta el final, de forma que los coeficientes generan u curva con pendiente igual a  $1/\text{coef}$ , siendo coef un valor dado por nosotros.
- RATESCALE (tasa escalada): igual que el anterior salvo que los coeficientes se multiplican por (pendiente dada / pendiente existente).
- LEADLAG: hace que los coeficientes formen una curva de respuesta lead-lag según los parámetros que introduzcamos.
- ROTATE: modifica la curva al completo para alcanzar el valor final que demos. Para ello calcula la diferencia entre el valor actual de la ganancia en el Steady State y el valor deseado y crea una curva que empieza en cero y acaba en el valor de la diferencia calculada. Luego añade la curva creada a la existente. Se trata de una modificación muy útil.

### GAIN MATRIX ANALISIS

Se trata de una herramienta para evaluar las propiedades numéricas de un modelo. En particular es un análisis para la evaluación de la colinealidad de la matriz de ganancias y evitar problemas estructurales que nos lleven a problemas numéricos en el controlador en tiempo real.

Así, esta herramienta puede usarse para chequear la presencia de submatrices colineales o casi, a la vez que se chequea la escala de la matriz.

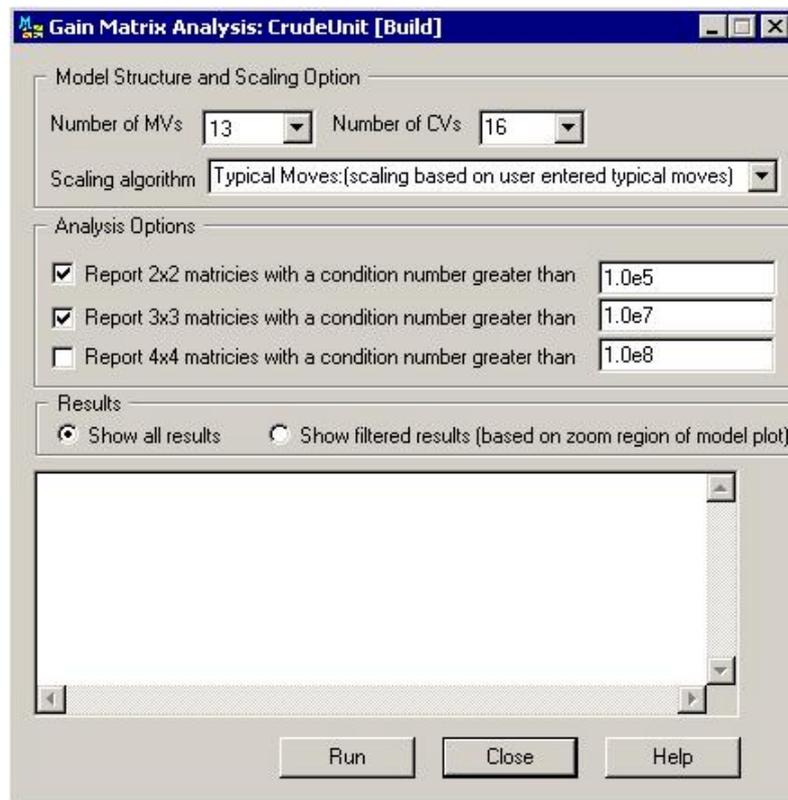


Figura 5.25.- Gain Matrix Analysis

### **5.3.6.3.- Exportar un modelo**

A lo largo de este capítulo ya se ha hablado de la exportación de componentes de un proyecto para tenerlos disponibles para otros proyectos. Sin embargo en este caso la exportación del modelo final del proceso que se desea controlar es necesario, ya que ha de incluirse dicho modelo en la construcción del controlador que vayamos a aplicar.

Por lo anterior debemos obtener un archivo .mdl (los .dpa son modelos exportados para ser usados con Model, no con Build) que contenga nuestro modelo final. Para realizar la exportación nos situamos sobre el modelo a exportar y acudimos al menú Project> Export. También, una situados sobre el modelo, podemos usar el botón Export o bien usar el menú contextual.

### **5.3.7.- Predicciones**

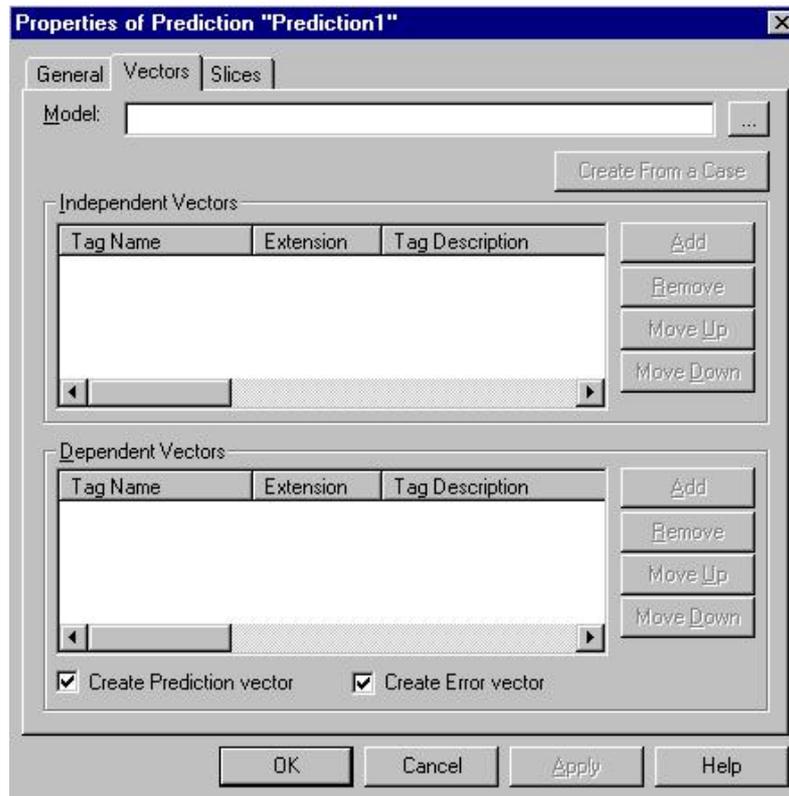
Las predicciones del comportamiento de las variables dependientes de un modelo son realizadas usando los vectores de las variables independientes que se hayan indicado en dicho modelo.

Podemos tener predicciones de dos tipos:

- Provenientes de los casos. Se tratan de los modelos que indicamos en la construcción de los casos que generaran predicciones.
- Creadas por el usuario para realizar predicciones de los modelos creado por él.

La creación de una predicción nos será de utilidad para comprobar que el modelo final se ajusta a los valores reales de la planta.

Para crear una predicción basta con situarnos en la carpeta de predicciones y entrar en Project> New Prediction.



**Figura 5.26.- Propiedades de una predicción**

Una vez que rellenemos los campos que se nos indican podremos generar la predicción del modelo pulsando el botón Run como se hacía en los casos. Entonces obtendremos una predicción que podremos visualizar para confirmar el funcionamiento del modelo utilizado en dicha predicción.

En la visualización de la predicción veremos los vectores que obtuvimos de la planta mediante Collect/Extract junto con los vectores de la predicción y unos vectores que nos indican el error encontrado entre los vectores con los datos de la planta y los vectores de la predicción.

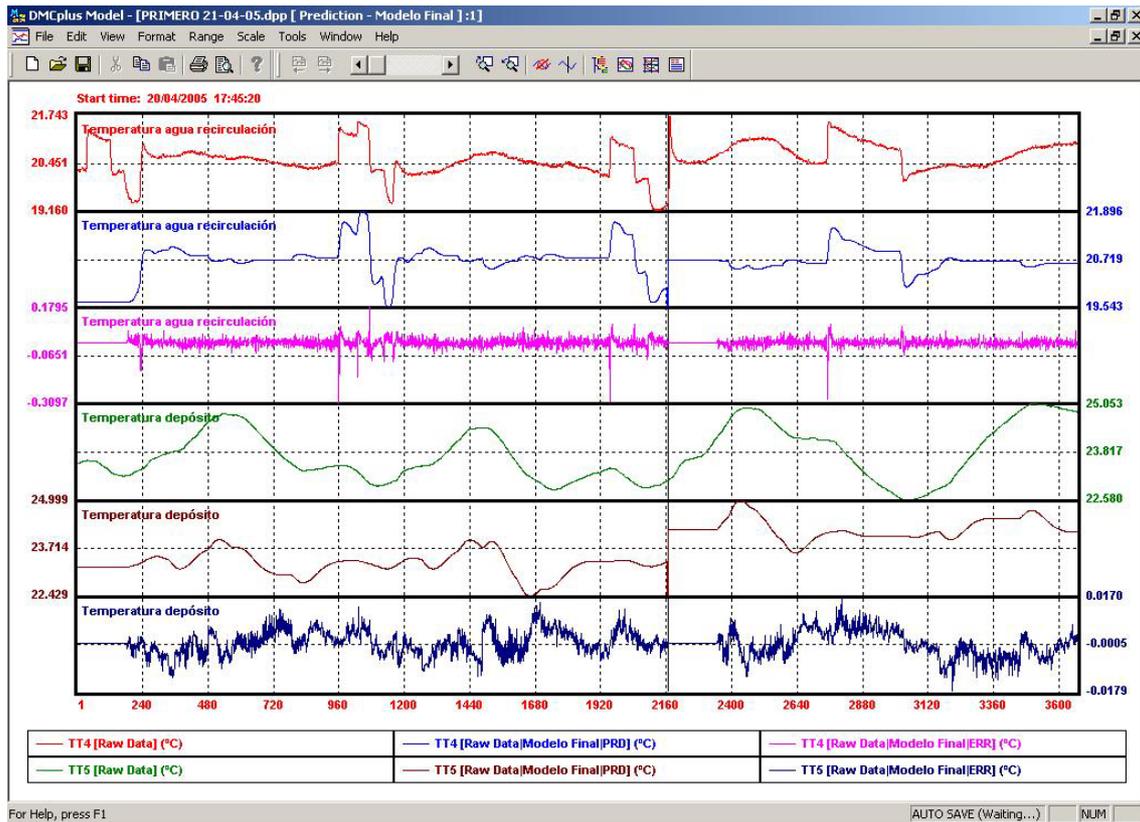


Figura 5.27.- Predicción generada a partir de un modelo de usuario

### 5.3.8.- Listas de objetos

Las listas son utilizadas para agrupar componentes del mismo tipo, salvo las listas generales que pueden agrupar distintos componentes, y poder tener los componentes del proyecto de una forma clasificados y fácilmente reconocibles. La creación de listas se hace de la misma forma que el resto de los objetos del proyecto.

Añadir componentes a una lista se puede hacer de forma fácil y rápida simplemente con arrastrar el componente que se desee a la lista elegida. Ésta acción no mueve el objeto de lugar sino que crea una copia del mismo en la lista.

## **5.4.- DMCplus Build**

Este módulo de DMCplus se utilizará para la creación y configuración del controlador que queremos aplicar al proceso. Es un paso previo al uso del Simulate y de ejercer el control sobre la planta.

La apertura de este programa se realiza desde Inicio> Programas> Aspen Manufacturing Suite> DMCplus Desktop> Build.

### **5.4.1.- Aspectos generales**

Para realizar el control sobre un proceso es necesario disponer de dos archivos con la información necesaria para ello: un archivo con el modelo del proceso (.mdl) y un archivo de configuración del controlador (.ccf)

El modelo del proceso a controlar determinará el tamaño del problema de control y las relaciones dinámicas entre las variables.

En cambio el archivo de configuración determina el lugar de los parámetros de salida o entrada del controlador, los tamaños de las variables, límites, cambios, ajustes, etcétera.

En este módulo se diferencian tres tipos de variables del proceso:

- MV (manipulated variables): variables manipulables. Son las variables independientes del proceso que podemos manipular para controlar el proceso.
- FF (feedforward variables): variables de perturbación. Son variables independientes pero que vienen establecidas, es decir, que no es posible manipular para ejercer el control.
- CV (controled variables): son las variables dependientes que se desean controlar.

Desde Build podremos generar dos tipos de archivos de configuración de controladores:

- .ccf  $\Rightarrow$  es el archivo de configuración del controlador (Controller Configuration File), el que se utilizará para ejercer el control.
- .tcc  $\Rightarrow$  es una plantilla base (Template Controller Configuration file) para la creación de .ccf. En este tipo de archivos sólo se permite un máximo de 1 variable de cada tipo.

Al igual que en el caso del Model la estructura del Build es similar, estando a nuestra disposición las mismas herramientas que en las aplicaciones diseñadas para Windows.

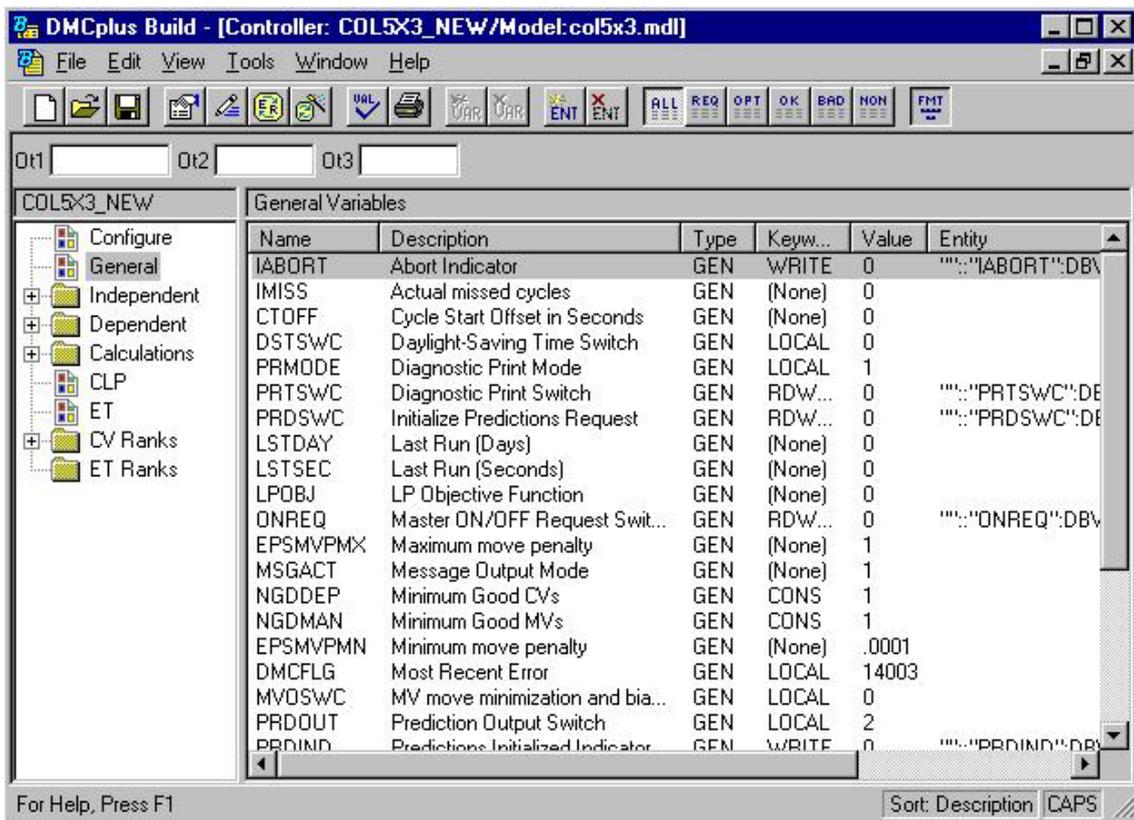


Figura 5.28.- Pantalla de DMCplus Build

Podemos observar en la figura anterior que la ventana de Build se divide en dos paneles, el de la izquierda o área de parámetros, desde donde podemos acceder de forma rápida a la configuración de los parámetros, y el panel derecho o panel de entradas, en el cual modificamos las características del controlador.

Los pasos a seguir para la configuración del controlador son los siguientes:

1. Crear una plantilla. Se trata de crear un archivo de configuración del controlador sólo para una variable manipulable y una controlable. Esto nos servirá para evitar la repetición de estructuras en la .pcf más adelante.
2. Especificar la estructura del control. Se realiza al crear una nueva .pcf y se especifican las variables que se tendrán en cuenta, así como el modelo que se va a utilizar.
3. Indicar opciones. Hemos de indicar una serie de opciones al iniciar la configuración de la .pcf como son: objetivos externos, generación de movimientos futuros y predicciones, conexión CIM-IO y dispositivo a utilizar, plantilla base, mensajería y utilización de subcontroladores.

4. Definir las etiquetas de las variables. Disponemos de tres formas de realizarlo: mediante el editor de entradas, usando la herramienta de sustitución o a través de la herramienta Tag Wizard.
5. Ajustar a nuestras necesidades el archivo .ccf. Debemos ajustar las entradas del controlador a nuestras necesidades revisando para ello las entradas del archivo.
6. Chequeo de errores. Se dispone de una herramienta de validación del archivo .ccf utilizando para ello el motor del programa Simulate. Así podremos estar seguros de que no existen errores de configuración en nuestro .ccf, aunque éste no garantiza que el control que se aplique funcione correctamente.
7. Utilizar el archivo .ccf con Simulate y Manage. Mediante Simulate tendremos la oportunidad de realizar ajustes, sintonizar el control y definir más concretamente los límites del control antes de utilizarlo sobre el proceso. Manage es la herramienta para ejecutar el control en línea sobre la planta.
8. Generar documentación. Podemos generar a partir del .ccf una documentación de los valores y de la configuración de nuestro controlador.

#### **5.4.2.- Controller Configuration File (CCF)**

La CCF está organizada en secciones, cada una de ellas conteniendo una o más entradas. Estas entradas son las unidades básicas de la CCF, ya que contienen la información necesaria para describir el controlador.

Cada parámetro del controlador puede describirse mediante una entrada, cuyo contenido hay que especificar y que es el siguiente:

- Name: es una etiqueta de texto única para cada entrada. Existen una serie de etiquetas reservadas para entradas estándar del controlador DMCplus, aunque es posible introducir nuevas entradas creadas por el usuario.
- Keyword: informa de si los datos de la entrada residen en el controlador de forma interna o son obtenidos del exterior, así como el tipo de acceso que se permite (lectura, escritura o ambos)
- Data Type: determina cómo ha de interpretarse el valor del parámetro.
- Valor: se trata del valor que deberá cargarse en el controlador la primera vez que se ejecute éste. Puede ser numérico, una cadena de caracteres o una ecuación dependiendo del Data Type.

- Entity: contiene la información necesaria para localizar el valor en una base de datos o en el propio DCS (sistema de control distribuido). A su vez está compuesta por varios ítems:
  - Source: palabra clave para describir el mecanismo de lectura o escritura.
  - Tag Name: cadena de caracteres que nos permite encontrar el punto a través de la conexión CIM-IO.
  - CIM-IO information: sólo si se utiliza la interfaz CIM-IO.

Por lo tanto la configuración del controlador se realiza especificando los valores para las entradas de la CCF. Las entradas, como ya se ha comentado, se estructuran en secciones, estando compuesta cada .ccf por varias de ellas, que se nos muestran en la zona izquierda de la pantalla, cada una de ellas con un cometido específico en la configuración del control. Estas secciones son:

- Configure: muestra las entradas de configuración general.
- General: entradas de descripción de la .ccf como número de MV y CV, tiempo de ejecución y demás.
- Independent: en esta sección tendremos un apartado para las entradas de cada variable MV (variable manipulable) o FF (variable independiente pero no manipulable) que se vayan a utilizar.
- Dependent: ídem al anterior pero con las variables CV (variables controlables).
- Calculations: dividido en dos, Declarations, donde declararemos nuevas entradas de la .ccf que creamos nosotros, y Ecuations, en donde podremos crear fórmulas para su uso en el archivo.
- Subcontrollers: sólo se muestra si permitimos que en nuestro controlador puedan participar subcontroladores.
- Composite: el uso de Composite se indica cuando vamos a utilizar varios controladores a la vez, y entre los cuales se establece una relación en la cual la salida de alguno de ellos va a ser la entrada para otros. Al igual que el anterior sólo aparece si incluimos el controlador en un conjunto gestionado por Composite para participar en un objetivo en común.
- ET (external targets): las External Targets son objetivos de control especificados desde el exterior de DMCplus. Esta sección no se muestra a no ser que especifiquemos dicha posibilidad.

Las entradas de todas las secciones pueden encontrarse en el anexo correspondiente.

A continuación pasaremos a ir explicando los pasos a seguir en la configuración de la .ccf desde el principio.

#### **5.4.2.1.- Entradas: Keyword**

La palabra clave sirve para agrupar las entradas por su mecanismo de almacenamiento y el mecanismo de entrada salida del valor. Además según la palabra clave que elijamos a la entrada podremos realizar una acción u otra y no todas las entradas disponen de todas las palabras clave.

Las keyword existentes son las siguientes:

- None: marcando una entrada con esta palabra clave dicha entrada no se reflejará en la CCF/TCC.
- AWRITE: escribe siempre (Always Write) el valor en una etiqueta del PCS (Process Control System). Es la clave para entradas definidas por el usuario.
- BUILD: reservado para entradas de Build solamente.
- CALGET: calculo de entrada (sólo disponible en la sección Calculation).
- CALPUT: igual al anterior pero de salida.
- CONFIG: parámetro de configuración que se actualiza sólo una vez al cargar la CCF.
- CONS: valor constante, actualizado únicamente una vez sin posibilidad de cambiarlo.
- INIT: valor que se inicializa tomando el dato del PCS en el momento de la inicialización.
- LOCAL: valor compartido en la memoria local de DMCplus. Se pueden cambiar mediante View, ecuaciones o transformaciones.
- LWRITE: escritura de baja prioridad hacia el PCS.
- PWRITE: escritura de alta prioridad al PCS.
- RDWRT: lectura y escritura de y hacia el PCS de prioridad media.
- READ: lectura de prioridad media del PCS.
- WRITE: escritura en el PCS de prioridad media. Cuando se trate de entradas definidas por el usuario debe usarse AWRITE.
- XFORM: transformaciones.

#### **5.4.2.2.- Creación de un nuevo CCF**

Para crear un nuevo archivo CCF hemos de presionar el botón New de la barra de herramientas. Entonces nos aparece una ventana para elegir la plantilla sobre la que nos basaremos para crear la .ccf. Además se nos indicará la posibilidad de elegir un modelo para el controlador.

Una vez hayamos especificado nuestras preferencias se nos preguntará por el número de variables MV (de 1 a 999), FF (entre 0 y 999 menos el número de MV) y CV (de 1 a 999) que incluiremos en nuestro controlador, así como por el nombre que le pondremos al controlador.

Hecho lo anterior se nos indicará si deseamos seleccionar una serie de opciones generales de la CCF. Aunque no es obligatorio sí se recomienda seleccionar las opciones en este momento, ya que nos ahorraremos tener que introducirlas luego en las entradas correspondientes. Las opciones que se pueden seleccionar son entre otras:

- Información para Composite.
- Uso de External Targets.
- Modelo a utilizar.
- Tipo de solución del controlador (lineal o cuadrática).
- Dispositivo de conexión CIM-IO.
- Mensajería a utilizar.
- Subcontroladores que se incluirán en nuestro controlador.

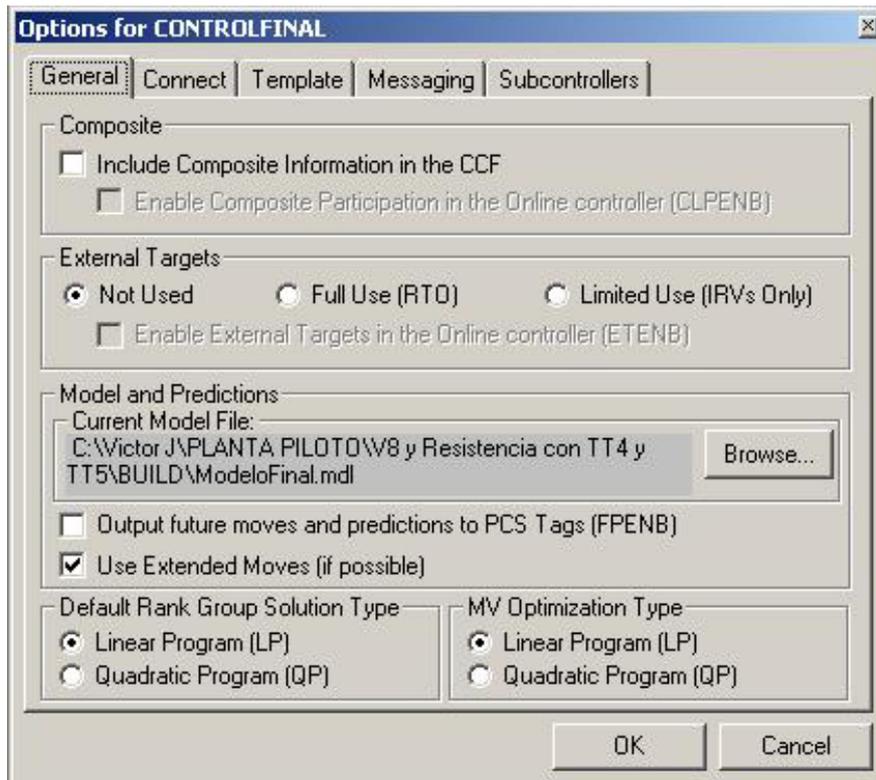


Figura 5.29.- Opciones de la CCF

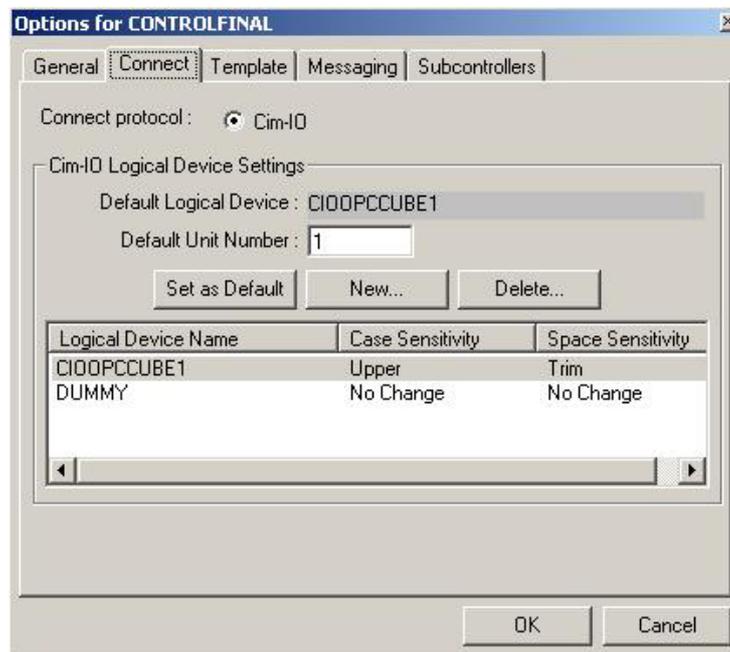


Figura 5.30.- Opciones de la CCF

Indicadas las opciones anteriores se nos preguntará si queremos ejecutar la herramienta Tag Wizard. Esta herramienta se utiliza para rellenar una serie de parámetros de forma automática sin necesidad de hacerlo uno a uno. Los parámetros que se introduzcan serán

añadidos de forma automática en los lugares de la CCF en los que en la TCC se introdujeron los valores \*, ~ y ^.

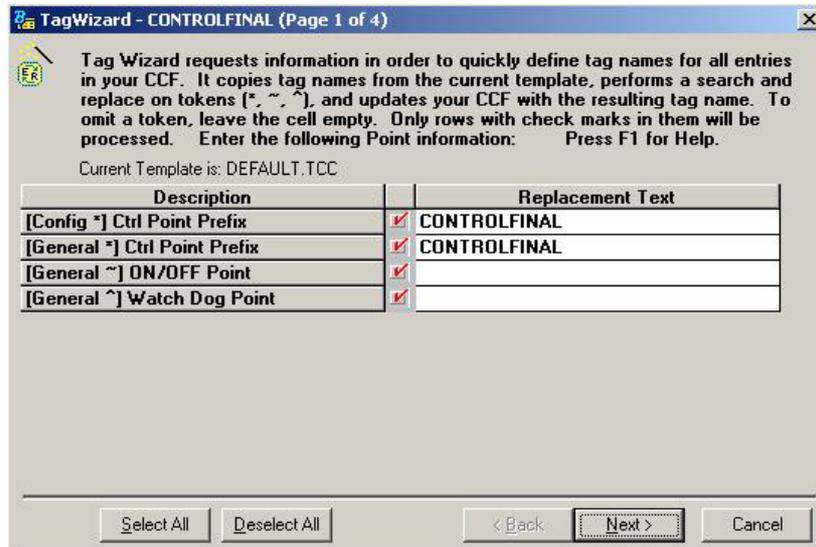


Figura 5.31.- Uso de Tag Wizard

En este caso podemos observar una ventaja de la creación de una TCC en la que basarnos para crear más adelante nuestra CCF, ya que si elegimos bien dónde colocar los valores “comodín” anteriores ahorraremos tiempo y necesidad de escribir.

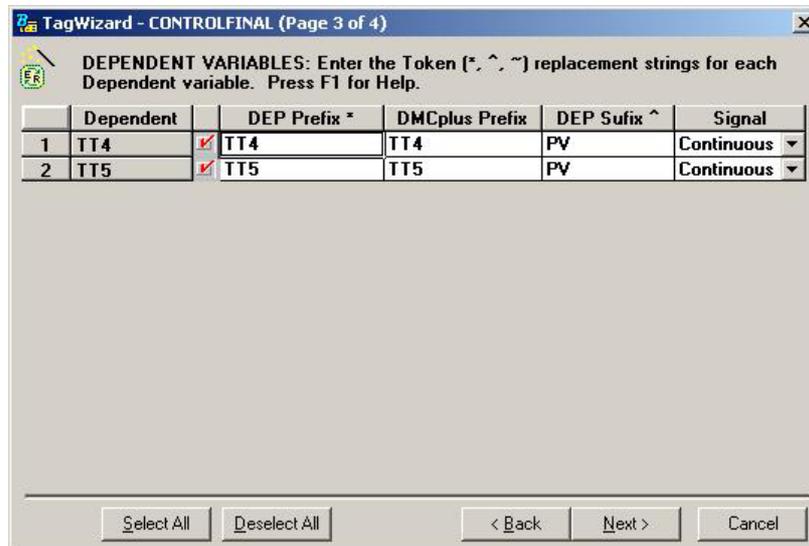


Figura 5.32.- Uso de Tag Wizard

Hay que indicar que ni las opciones anteriores ni el Tag Wizard son de ejecución obligatoria, aunque sí se recomienda su uso.

Una vez hayamos cerrado las opciones y Tag Wizard habremos terminado de crear la nueva CCF y estaremos en disposición de variar los parámetros de las entradas de la CCF.

### **5.4.2.3.- Configure**

La sección de configuración muestra las entradas de configuración de la CCF (o la TCC) como son el mostrar las predicciones y los movimientos que se van a realizar, el uso de External Targets, la participación de Composite y demás, aunque como se indicó en el apartado anterior, muchas de estas entradas se configuran utilizando el panel de opciones, al que se puede acceder en cualquier momento a través del menú Tools> Options.

Es de reseñar también que muchas de estas entradas son configuradas automáticamente por Build a partir del modelo que se utilice.

Algunas de las entradas más interesantes de esta sección son las siguientes:

- CTLINT: entrada de sólo escritura y de configuración automática según el modelo. Es una entrada utilizada para verificar que el controlador se está ejecutando en el periodo de control correcto.
- DEFSOLT: indica el ranking de restricciones que se utilizará por defecto. Ver más adelante en el apartado de las variables dependientes. También es configurada por Build.
- INTSUM: es una constante, por defecto a 0 (off). Se usa para indicar al controlador que mantenga una copia de los setpoints con una resolución mayor que la del sistema de control del proceso. Así se previene la pérdida de pequeños movimientos que el proceso no es capaz de reflejar por su poca resolución.
- SSMINS: introducido de forma automática por Build, son los minutos establecidos para el Steady State.

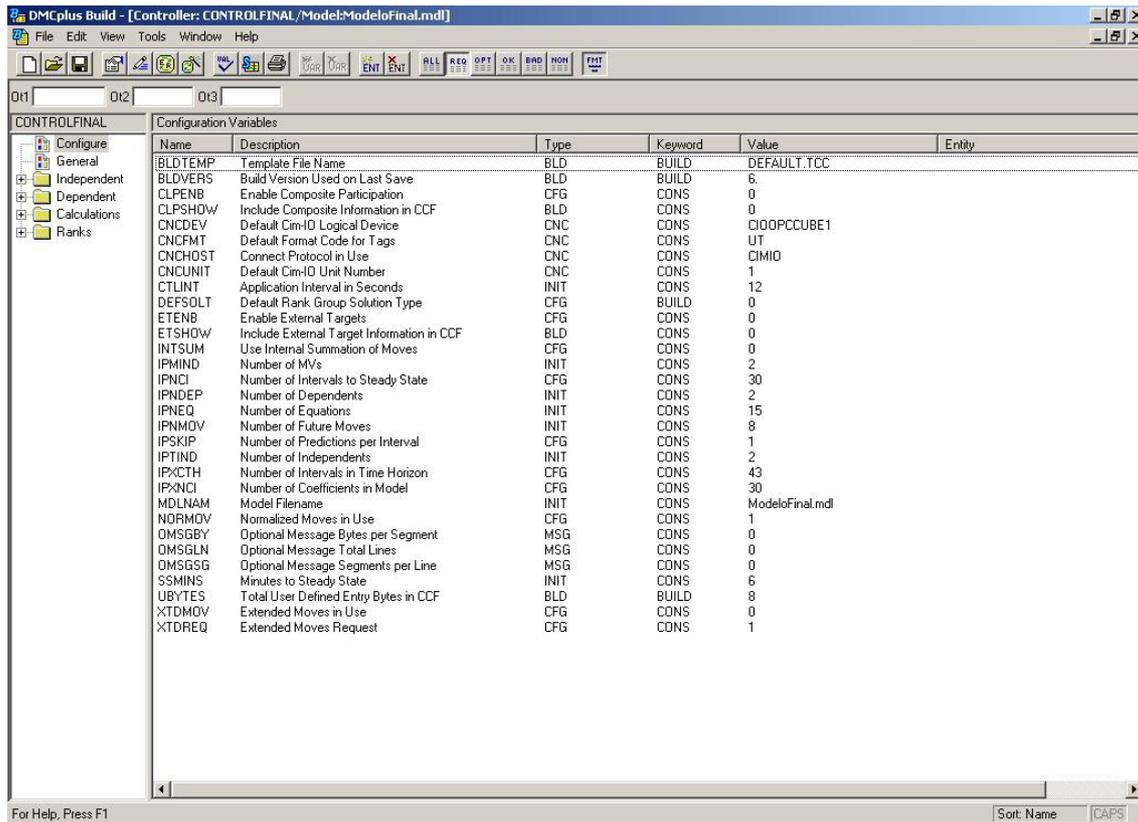


Figura 5.33.- Entradas obligatorias de Configure

### 5.4.2.4.- General

Esta sección muestra una lista con entradas generales como el número mínimo de variables CV o MV que deben cumplir las restricciones, tiempo de ejecución, indicador de parada y otros.

Presenta la misma estructura que la sección Configure y que el resto de secciones, aunque en esta sección es en la que aparece un menor número de entradas, siendo sólo cuatro las entradas obligatorias.

Cabe destacar de entre todas las entradas:

- CNTDWN: se trata del típico perro guardián que se utiliza en toda programación con el fin de verificar que el controlador está ejecutándose. Es una entrada de escritura o local.
- MVOSWC: tipo de solución a adoptar para los movimientos de las variables manipulables (0 es lineal, 1 cuadrático). Es LOCAL, READ o CONS.
- ONREQ: gestor de inicio del controlador. Cuando está a 1 (ON) y se indica que debe comenzar el control se permite que se inicie el control si se dan las condiciones

para ello. De lo contrario se pone a cero y no permite la ejecución del controlador. Puede ser RDWRT, LOCAL, READ o CONS.

- LPOBJ: entrada que almacena el valor de la función objetivo durante la ejecución del controlador. Es None, WRITE o LOCAL.
- NGDDEP, NGDMAN: son el número mínimo de variables CV y MV respectivamente que deben mantenerse en buen estado pues de lo contrario el control deja de ejecutarse. None, LOCAL, CONS o READ.
- ONSTS: muestra el resultado de ONREQ obtenido al intentar ejecutar el controlador. None, WRITE o LOCAL.
- PRDSWC: indica si han de inicializarse las predicciones. None, LOCAL o RDWRT.

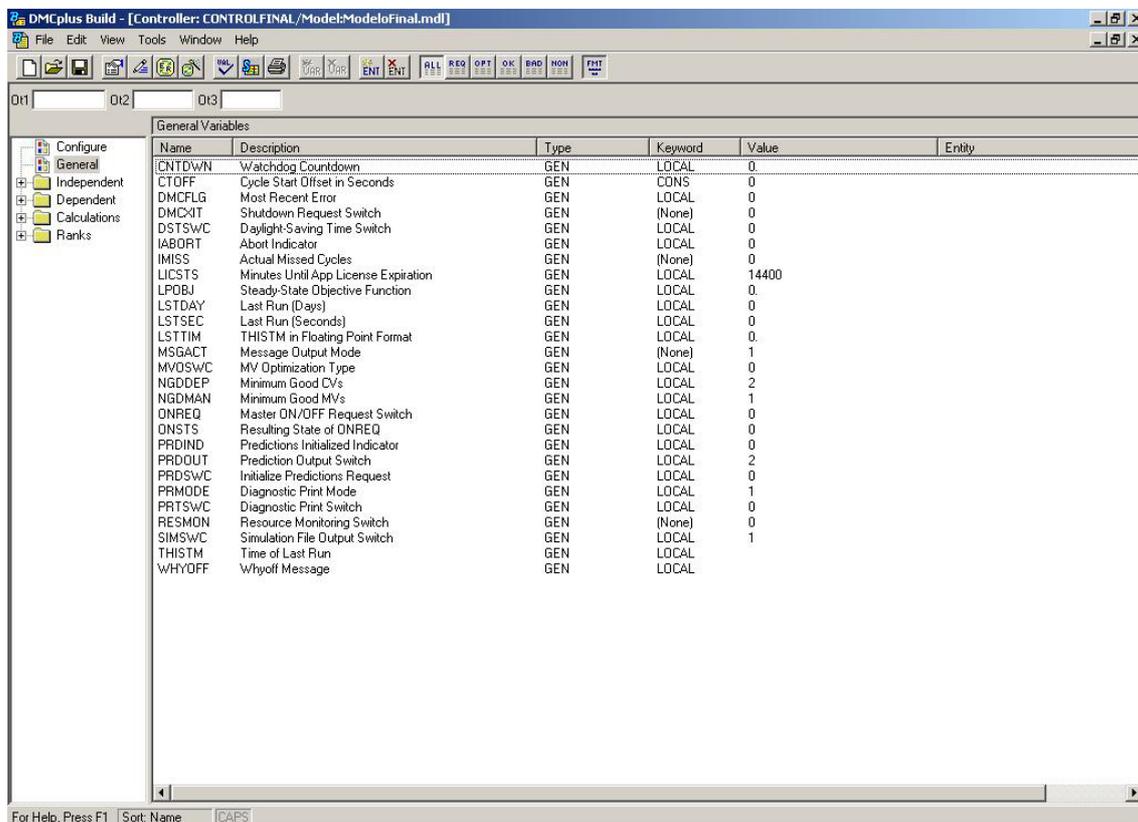


Figura 5.34.- Panel de entradas de la sección General

### 5.4.2.5.- Independents

En la sección de variables independientes tendremos que definir las mismas entradas para todas las variables que se encuentren en esta sección. Las variables independientes pueden ser de dos tipos como ya se ha indicado con anterioridad:

- Manipulated Variables (MV): son las variables independientes que podemos manejar para controlar el proceso.
- Feedforward Variables (FF): se tratan de variables independientes del proceso pero que nos es imposible su manipulación, como por ejemplo una entrada de agua corriente en la cual no podemos actuar sobre su temperatura. Estas variables pueden tratarse como una perturbación al sistema.

Para seleccionar una variable independiente como feedforward simplemente hemos de seleccionar la variable y validar la ‘check-box’ situada debajo de la barra de herramientas. En ese momento, las variables independientes que estén situadas por debajo de la seleccionada, y por supuesto la seleccionada, se convertirán en variables feedforward.

Para deseleccionar hemos de hacer lo mismo, es decir, desvalidar la misma ‘check-box’. Si queremos deseleccionar todas las variables FF basta con situarnos en la ultima de estas variables y deseleccionar el ‘check-box’.

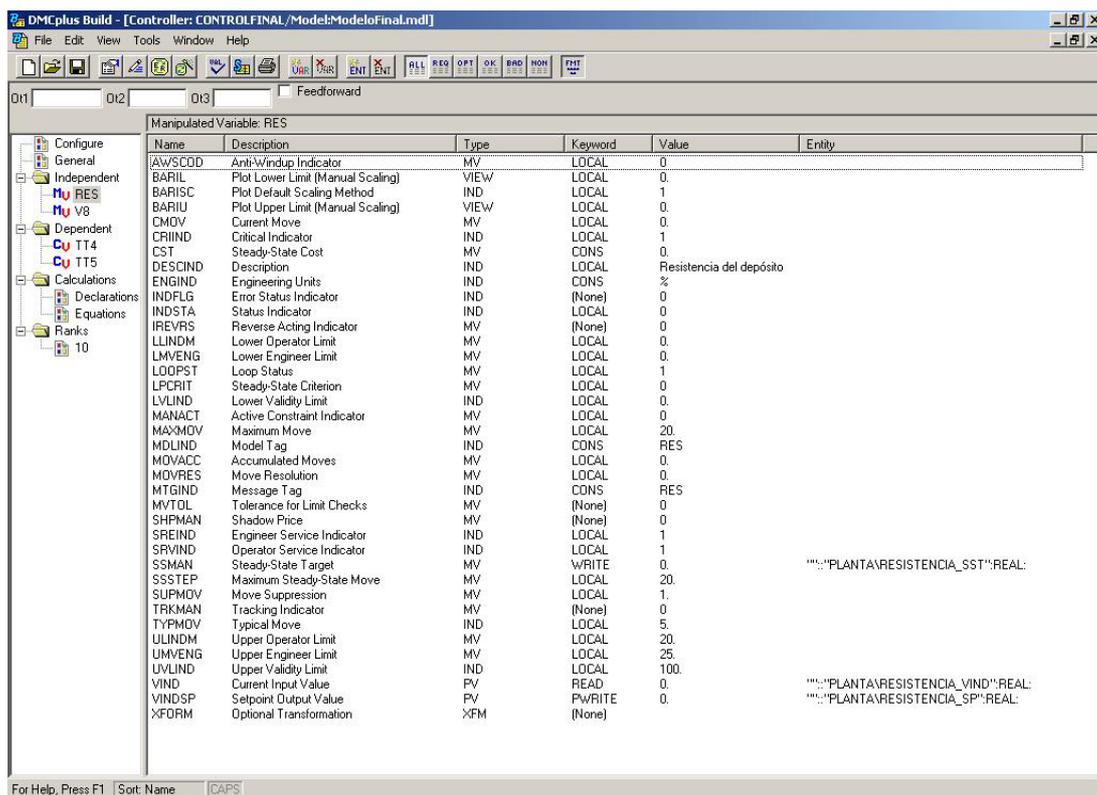


Figura 5.35.- Panel de entradas de Independents

En esta sección las entradas más interesantes son las siguientes:

- CRIIND: indica si la variable es crítica para el control del proceso. Si se indica que lo es, el control dejará de funcionar en el momento que dicha variable se salga de los límites que se le impongan. None, LOCAL, CONS o READ.

- CST: es la penalización en la que incurrimos por incrementar en una unidad el valor de la variable. LOCAL, READ o CONS.
- LLINDM, ULINDM: son los límites inferior y superior respectivamente que son válidos para la variable. El programa siempre tratará de buscar una solución que mantenga a la variable dentro de estos límite, que podríamos llamar límites de operación. Se tratan básicamente de un par de restricciones sobre las variables independientes. READ, LOCAL o RDWRT.
- MAXMOV: se trata del movimiento máximo en unidades ingenieriles que puede realizar el set point de la variable en un ciclo. Debe ser mayor o igual a SSSTEP. LOCAL, READ o CONS.
- SSMAN: valor de Steady State para la variable. El control moverá la variable hasta llevarla a este valor al final del horizonte temporal del controlador. LOCAL o WRITE.
- SSSTEP: es el movimiento máximo en el valor del Steady State permitido para una variable independiente. Esto permite que el controlador sea capaz de mover la variable al nuevo valor de Steady State durante el horizonte temporal. LOCAL, READ o CONS.
- VIND: valor actual de la variable medida en el proceso. READ o LOCAL.
- VINDSP: se trata del set point calculado por el controlador para la variable. Este valor es enviado al proceso al final de cada ciclo de ejecución del controlador. PWRITE o LOCAL.

### NOTAS IMPORTANTES

- A. CMOV es la entrada para las variables independientes que le indica al proceso el movimiento que debe realizar dicha variable (si el control está OFF se trata del movimiento propuesto). Sin embargo en la planta sobre la que se probó el programa este movimiento no puede implementarse ya que deben ser valores absolutos no incrementales como es CMOV. Para solucionar este problema simplemente se definió una entrada de usuario, ABSRES y ABSV8, en la cual sumábamos al valor que tuviera en ese momento la variable el valor de CMOV mediante ecuaciones, y el total era el valor que le proporcionábamos a la planta.
- B. También es importante resaltar que debe tenerse especial cuidado con los valores de las variables que se leen y se escriben en la planta, puesto que las prioridades de estas acciones pueden hacer que estemos leyendo valores que no sean los que se

deseaban. Esto ocurre por ejemplo en la planta en la que se realizaron las pruebas con el valor que lee la entrada VIND, que es el valor que en un momento tiene la variable, y el valor que escribe cualquiera de las entradas que definimos anteriormente. Así, si se escribe antes de leer el valor de la variable entonces en VIND no tendremos el valor de la variable en el momento actual sino el del ciclo siguiente que es el que escribe el controlador al ejecutarse. Para solucionar esto podemos crear una nueva variable en la base de datos de la planta, V8\_VIND y RESISTENCIA\_VIND, que sea una copia del valor que queremos leer y escribir, evitando de esta manera que la escritura borre los datos que queremos leer.

#### **5.4.2.6.- Dependent**

Esta sección es parecida a la anterior ya que en ésta se agrupan todas las variables del proceso que se desean controlar. Las variables de esta sección tienen unas características propias que son:

- Intermittent: son variables que no se actualizan de forma continua sino cada cierto tiempo según configuremos la entrada adecuada para ello.
- Ramp CV: se trata de variables dependientes de tipo rampa. Las variables de este tipo llevan esta característica activada según se marcaran en el modelo del proceso.

Al contrario de la característica Ramp, la característica Intermittent sí puede activarse a voluntad para cada variable con solo validar la ‘check-box’ que está debajo de la barra de herramientas. La validación de esta característica sólo afecta a la variable para la que se activa o desactiva, al contrario de lo que ocurría con el marcaje de FF de las variables independientes en la sección anterior.

Como en las secciones precedentes, a continuación listamos algunas de las entradas más importantes con las que nos encontramos en la sección Dependents:

- DEP: se trata del valor actual de la variable leído directamente del proceso. Puede ser READ o LOCAL.

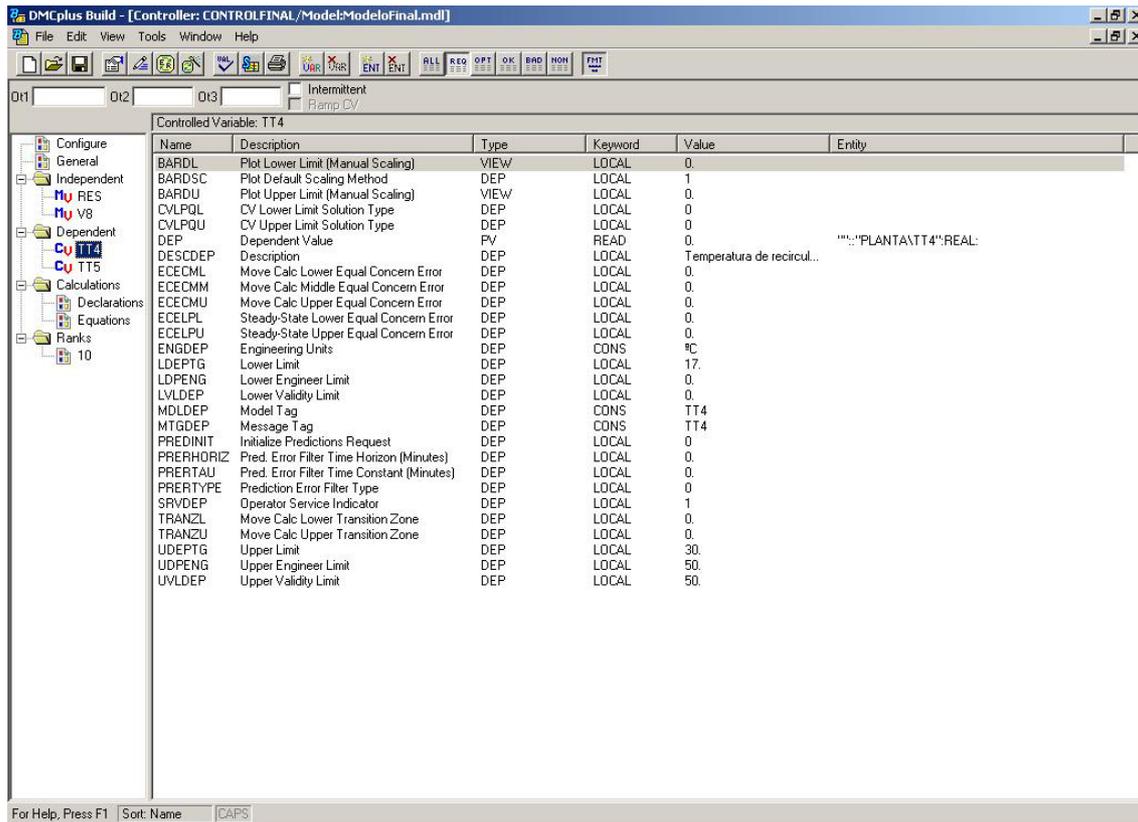


Figura 5.36.- Entradas obligatorias de una variable en la sección Dependent

- LDEPTG, UDEPTG: son las entradas correspondientes a los límites inferior y superior para el objetivo que debe alcanzarse en el régimen permanente o Steady State. Debe estar dentro de los márgenes establecidos por los límites de validación (LVLDEP, UVLDEP) e ingenieriles (LDPENG, UDPENG). Con estos límites lo que delimitamos es una zona válida en la que el programa debe encontrar una solución en régimen permanente para esta variable. Si queremos forzar la variable para que se dirija a un valor concreto deberemos tomar el mismo límite inferior y superior, con lo cual restringimos la banda válida a un solo valor, aunque esto tiene el inconveniente de que puede ser que no exista una solución conjunta para todas las variables, con lo cual no se podrá ejercer el control sobre el proceso, a no ser que la variable no sea cr. RDWRT, LOACL o READ.
- SRVDEP: es la entrada que nos indica si la variable debe entrar en los cálculos del controlador para buscar la solución óptima en el régimen permanente. READ, LOCAL o CONS.
- ACPRER: se trata de la entrada que nos guarda el error acumulado en la predicción. Esta entrada nos sirve para detectar posibles errores o desajustes en el modelo

utilizado para la predicción del proceso. El controlador actualiza este valor en cada ciclo de ejecución. None, WRITE o LOCAL.

- CVRANKL, CVRANKU: mediante estas entradas podemos definir los rankings de las soluciones que busque el programa simplemente asignando un número de ranking para el límite inferior y el superior. None, LOCAL, READ o CONS.
- CVSTEP: máximo cambio permitido en el valor de Steady State de la variable en un ciclo de control. Es interesante utilizarlo para variables que no sean rampas para limitar los cambios del objetivo en régimen permanente. None, READ, LOCAL o CONS.
- SSDEP: nos indica el objetivo Steady State o de régimen permanente para la variable con la que estemos trabajando. None, WRITE o LOCAL.
- SSERR: esta entrada muestra el error cometido en el Steady State. Es la diferencia entre SSDEP y los límites (si se ha violado alguno, de lo contrario se resetea a cero) UDEPTG o LDEPTG. None, WRITE o LOCAL.

#### **5.4.2.7.- Calculations**

Esta sección se divide en dos subsecciones a su vez, la correspondiente a la definición de entradas del usuario y la correspondiente a las ecuaciones que se quieran utilizar.

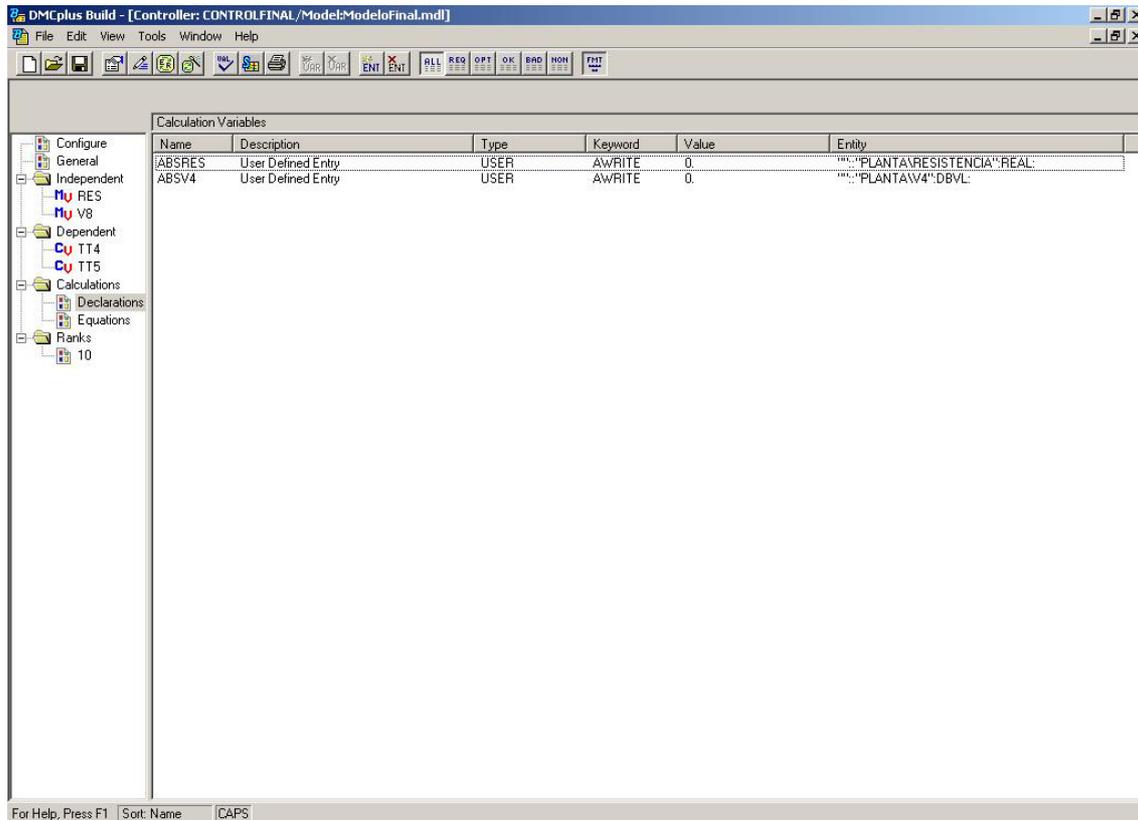
#### **DECLARATIONS**

En esta subsección definiremos las entradas que deseemos crear para completar la configuración de la CCF y que Build no proporciona de forma predeterminada.

Para crear una nueva entrada basta con pulsar el botón de la barra de herramientas Define a New Entry. Entonces el programa nos pedirá un nombre para la nueva entrada, que debe ser distinto de los nombres que están reservados para las entradas que Build nos proporciona de forma predeterminada, y además deberemos elegir el tipo de datos que utilizará la entrada.

Una vez validado lo anterior aparecerá en la sección la nueva entrada y podremos editarla. Al editar podremos observar que se tienen a disposición del usuario todas las palabras clave o Keyword de que dispone Build, esto es: None, CONS, LOCAL, INIT, READ, WRITE, LWRITE, PWRITE, AWRITE y RDWRT. Así pues podremos elegir cualquiera de las Keyword disponibles, con todas las posibilidades que esto nos ofrece.

Una vez editada la nueva entrada ésta estará disponible para usar como si fuese una entrada más de Build, sin diferencias con las que vienen predeterminadas en Build.



**Figura 5.37.- Sección Declarations**

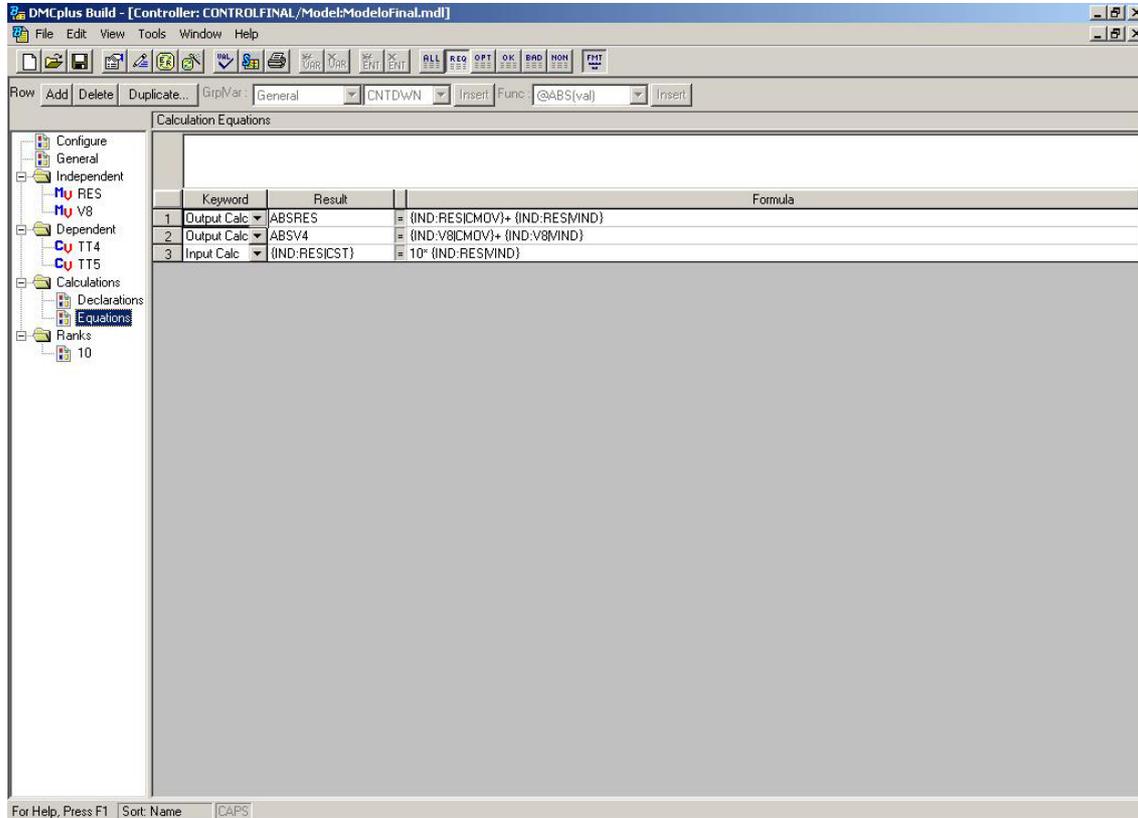
## EQUATIONS

Build nos ofrece la posibilidad de utilizar una serie de fórmulas sencillas, lo que nos permite hacer que el valor de alguna entrada de la CCF esté en función de otra serie de entradas. Esto resulta de gran utilidad si se desea tener valores de entradas condicionadas a valores del proceso que vamos a controlar por ejemplo.

Para crear una ecuación hemos de seguir una serie de pasos:

1. Pulsar el botón de herramientas Add.
2. Abrir la lista desplegable Keyword que nos aparece y seleccionar el tipo que queramos: None, Input Calculation o Output Calculation.
3. Seleccionar el campo Result y seleccionar del desplegable Grp|Var la variable o el grupo de variables a la que se aplicará el cálculo así como la entrada en la que irá el resultado del cálculo. Una vez seleccionado lo que queramos pulsamos el botón Insert.

4. Posicionarse luego en el campo Formula. En este campo podremos insertar también entradas de variables o grupo de variables, así como algunas de las funciones que Build trae predeterminadas (ver el anexo correspondiente).
5. Repetir los pasos para seguir insertando nuevas ecuaciones.



**Figura 5.38.- Sección Equations de Build**

Las ecuaciones también pueden ser duplicadas de forma rápida y sencilla.

#### **5.4.2.8.- Subcontrollers, Composite, ET**

El trabajo en estas secciones se realiza de igual forma que en las secciones anteriores, aunque sólo aparecerán si previamente en las opciones de la CCF se habilitaron estas posibilidades.

#### **5.4.3.- Set Point frente a Steady State Target**

Hemos de distinguir en la configuración del controlador los conceptos de Set Point Y Steady State.

Como Set Point (SP) se establece el valor que ha de tener una variable al final de cada ciclo de ejecución del controlador. En procesos en los que a nivel de campo las señales de control estén reguladas mediante PID por ejemplo, este valor es el que se le

suministraría a dicho PID para que moviese la variable hacia el valor de SP proporcionado en el ciclo del controlador.

Por otro lado hay que distinguir del SP el valor de Steady State (SS), ya que éste es el valor objetivo al que se debe llegar al final del horizonte de predicción del controlador, esto es, el tiempo de establecimiento que se indicó en su momento.

Por lo tanto los SP que se suministren a la planta en cada ciclo de ejecución del controlador irán variando durante el tiempo de establecimiento hasta alcanzar el valor objetivo en el SS. Esto no quita sin embargo que el valor de SS pueda variar también a lo largo del proceso de control hasta que se encuentre un valor óptimo para el mismo.

En nuestro caso, al no disponer de PID en la planta sobre la que se utilizó el programa, los valores de SP no se fueron teniendo en cuenta, aunque podrían haber sido utilizados como valores a suministrar a las variables de campo.

### 5.4.4.- Rankings

Hemos hablado anteriormente de rankings de soluciones. La sección Rank nos muestra todos los grupos de ranking o categoría que hayamos creado mediante los valores dados a las entradas correspondientes (CVRANKL, CVRANKU).

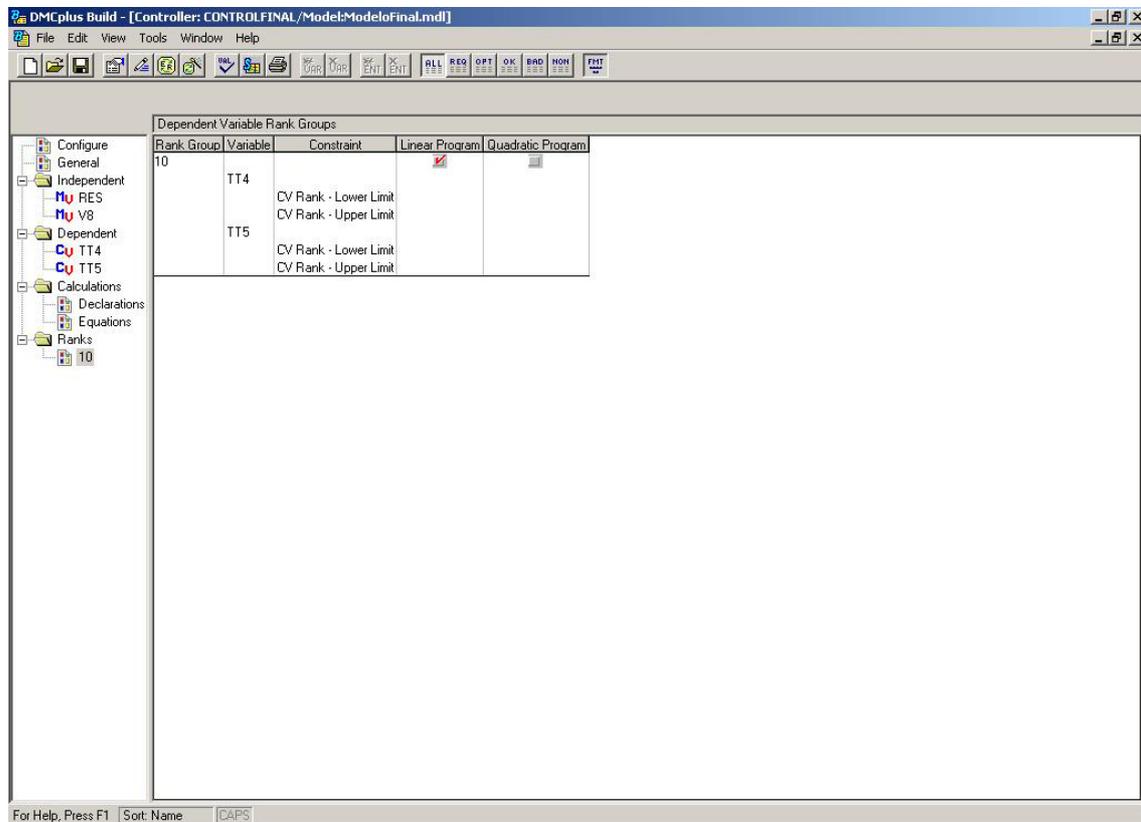


Figura 5.39.- Sección Ranks

Los rankings creados sirven para relajar las restricciones del problema de control en un orden determinado para asegurar la consecución de una solución en régimen permanente. Así pues, el controlador intenta buscar una solución para el problema de control utilizando las restricciones del rango de menor número. Si para ese rango obtiene una solución para el problema, dichas restricciones se convertirán en restricciones fuertes, es decir, que se tendrán en cuenta siempre y se continúa con el siguiente rango de restricciones. Este proceso termina cuando todos los rangos son considerados como restricciones fuertes, y se ejecuta entonces la optimización económica.

Por lo tanto aquellas limitaciones que no se hayan incorporado a un rango pueden no ser cumplidas y serán tenidas como restricciones relajadas.

#### **5.4.5.- Validación de la CCF**

Una vez que tengamos la CCF configurada, Build nos permite realizar un chequeo de la misma para comparar la estructura del modelo con la CCF, verificar que todas las entradas obligatorias han sido introducidas y llevar a cabo una validación completa del controlador utilizando para ello el motor de Simulate. Esta validación también se hará antes de guardar el archivo.

Para realizar la validación basta con utilizar el botón Validate situado en la barra de herramientas.

Hay que tener en cuenta que las etiquetas de las variables normalmente no podrán validarse debido a que la conexión con el proceso no esté disponible. De todas formas cuando se ejecute el controlador usando el programa Manage, éste realizará dicha validación de las etiquetas justo en el momento anterior a la ejecución del controlador, y si existe algún error se nos comunicará.

#### **5.4.6.- Creación de una plantilla**

Una plantilla de controlador es un archivo TCC (Template Controller Configuration), como ya se explicó al principio de este capítulo.

La creación de un archivo de este tipo se realiza de la misma manera que una CCF. Lo único que hay que tener en cuenta es que no debemos asignarle ningún modelo al archivo. Una vez hayamos creado la TCC, debe hacerse notar que sólo es posible la inclusión de una variable de cada tipo, es decir, una MV, FF y CV.

Además al guardar la plantilla deberemos cambiar el tipo de archivo a .tcc y tener cuidado de guardarlo en la carpeta \dmcpplus\templates en donde tengamos instalado el programa para que más tarde nos aparezca la plantilla cuando vayamos a crear la CCF.

#### **5.4.7.- Generación de documentación**

Por último podemos generar una documentación del estado de configuración del controlador, ya que en momentos posteriores puede que la configuración inicial del controlador sea cambiada por el usuario. La documentación nos permite pues no perder la configuración inicial y tenerla siempre disponible para cuando sea necesaria.

Para obtener el informe de la configuración sólo hemos de ir al menú File> Print. Entonces podremos elegir el formato del informe y realizar cambios en el mismo.

## **5.5.- DMCplus Simulate**

Tras haber configurado el controlador que se aplicara sobre la planta utilizaremos este módulo de DMCplus para comprobar el funcionamiento del control sin ejecutarlo en línea, es decir, sin necesidad de estar físicamente conectado al proceso para evitar que posibles errores de configuración del controlador pudieran inducir un funcionamiento incorrecto del proceso.

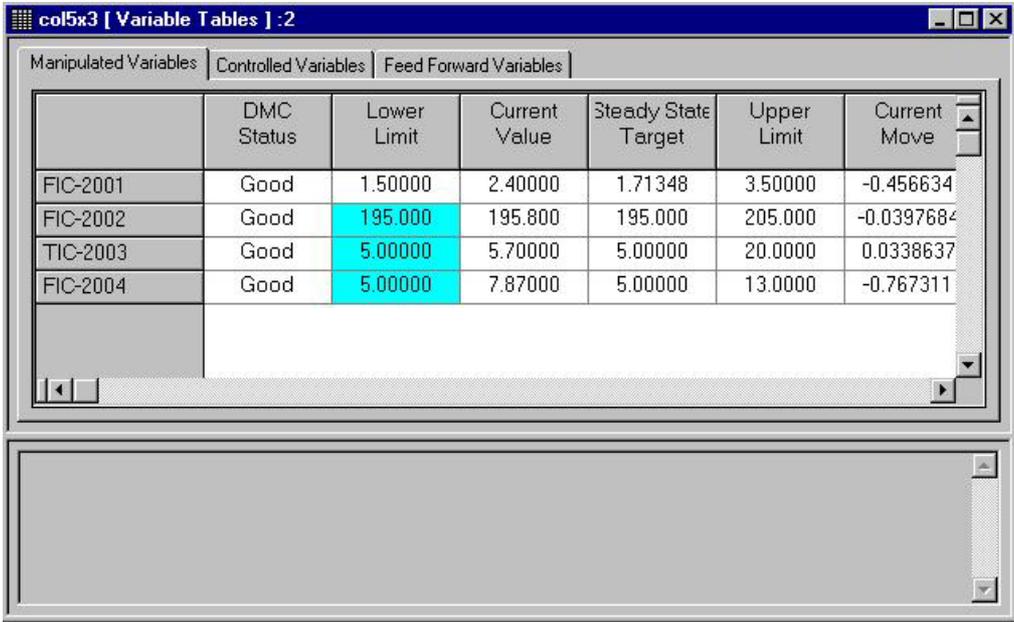
Para abrir este módulo iremos a Inicio> Programas> Aspen Manufacturing Suite> DMCplus Desktop> Simulate una vez que el servidor de licencias esté ejecutándose.

### **5.5.1.- Generalidades**

Para poder realizar una simulación del funcionamiento del proceso es necesario disponer en la misma carpeta de los archivos .mdl del modelo y .ccf de la configuración del controlador.

Simulate se compone básicamente de dos clases de ventanas: la de tablas de variables y las de simulación gráfica.

En la ventana de tabla de variables se nos mostrarán tres tablas: una con las variables MV, otra con las CV y una tercera con las variables FF.



	DMC Status	Lower Limit	Current Value	Steady State Target	Upper Limit	Current Move
FIC-2001	Good	1.50000	2.40000	1.71348	3.50000	-0.456634
FIC-2002	Good	195.000	195.800	195.000	205.000	-0.0397684
TIC-2003	Good	5.00000	5.70000	5.00000	20.0000	0.0338637
FIC-2004	Good	5.00000	7.87000	5.00000	13.0000	-0.767311

**Figura 5.40.- Vista de las tablas de Simulate**

Por otra parte, las ventanas de simulación gráfica pueden ser varias, según la necesidad del usuario, pudiéndose abrir tantas ventanas de simulación como variables existan en las tablas anteriores.

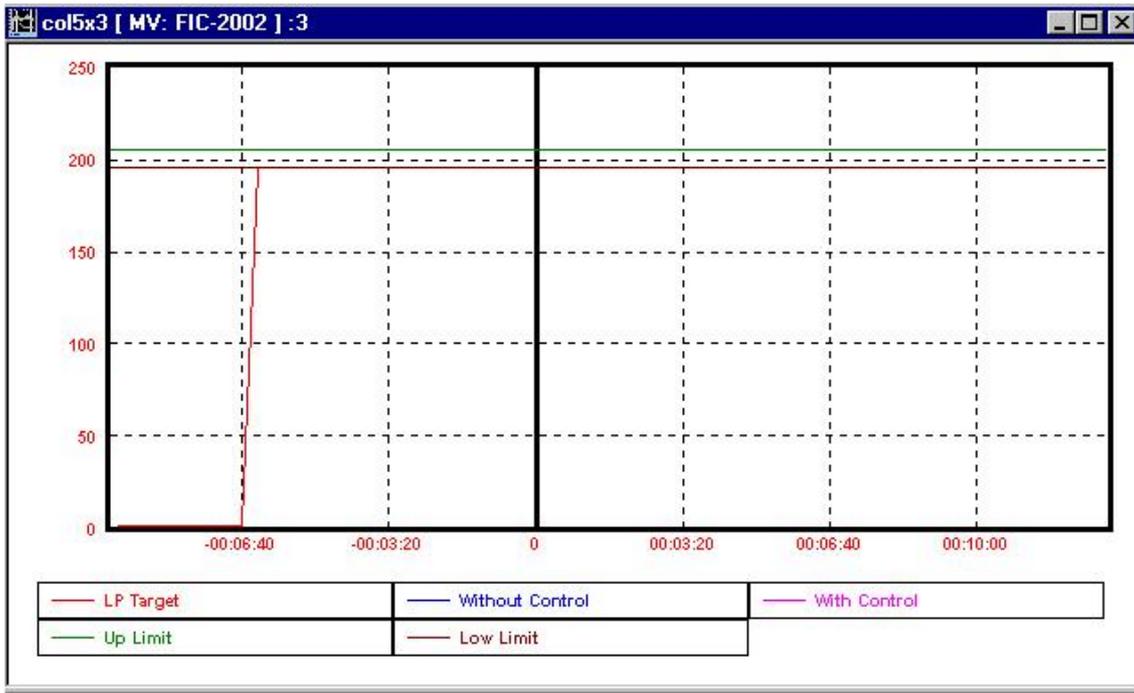


Figura 5.41.- Ventana de simulación para una variable

En general, los pasos a seguir para realizar una simulación son los siguientes:

1. Abrir el archivo de configuración CCF y crear un PSM.
2. Realizar un ajuste de los parámetros.
3. Ejecutar la simulación.
4. Analizar los resultados obtenidos.
5. Guardar los estados iniciales y los ajustes.
6. Actualizar la CCF con los datos nuevos.
7. Generar la documentación asociada a la simulación. Controller> Create Report File, generando un archivo PRT.

### **5.5.2.- Archivos utilizados en Simulate**

En Simulate se pueden utilizar una serie de archivos de distinto tipo cada uno de con un cometido distinto. Estos archivos se enumeran a continuación:

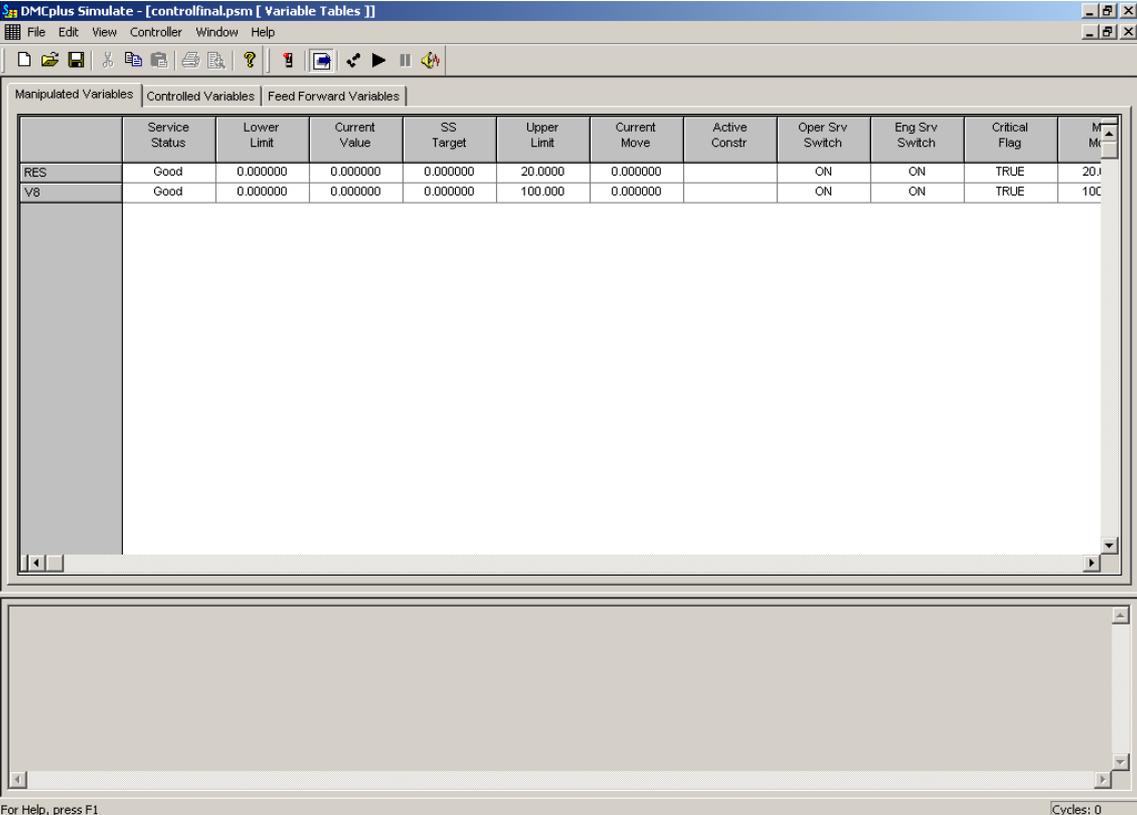
- .ccf  $\Rightarrow$  es el archivo de configuración del controlador que vamos a utilizar. Simulate lo carga para realizar simulaciones con él y comprobar el funcionamiento del mismo. Desde Simulate es posible ajustar algunos parámetros del controlador y actualizarlo.
- .psm  $\Rightarrow$  este tipo de archivo se utiliza para salvar el estado actual de una simulación. Pueden crearse distintos archivos de este tipo usando el mismo CCF

para guardar diferentes ajustes de los parámetros. Este archivo sólo se utiliza con Simulate.

- .prt ⇒ es el archivo en el cual se guarda la documentación generada por Simulate y que contiene información acerca de la simulación, los parámetros del controlador y ciertos valores de las variables. Su formato y su cantidad de información pueden variarse.
- .dbg ⇒ son archivos generados por el programa que gestiona la ejecución en línea del controlador. Contiene el estado y los parámetros del controlador y las predicciones en el momento de generar dicho archivo. La configuración de este archivo se realiza a través de la entrada de la CCF SIMSWC. Este archivo puede cargarse en Simulate como si se tratase de un .ccf.

### **5.5.3.- Ventana de variables**

Al cargar un archivo de configuración de un controlador (CCF), menú File> Open lo primero que nos muestra Simulate es la ventana que contiene las tablas con las variables que intervienen en el controlador.



	Service Status	Lower Limit	Current Value	SS Target	Upper Limit	Current Move	Active Constr	Oper Srv Switch	Eng Srv Switch	Critical Flag	M
RES	Good	0.000000	0.000000	0.000000	20.0000	0.000000		ON	ON	TRUE	20.0
V8	Good	0.000000	0.000000	0.000000	100.000	0.000000		ON	ON	TRUE	100

**Figura 5.42.- Tablas de variables**

Puede observarse que en cada una de las tablas nos aparece una fila para cada variable que hubiéramos indicado en el momento de realizar el controlador. Estas filas están compuestas por una serie de parámetros del controlador que podremos modificar desde estas tablas para realizar un mejor ajuste de los mismos.

A continuación se detallan los parámetros que podemos variar desde Simulate para las variables MV y FF:

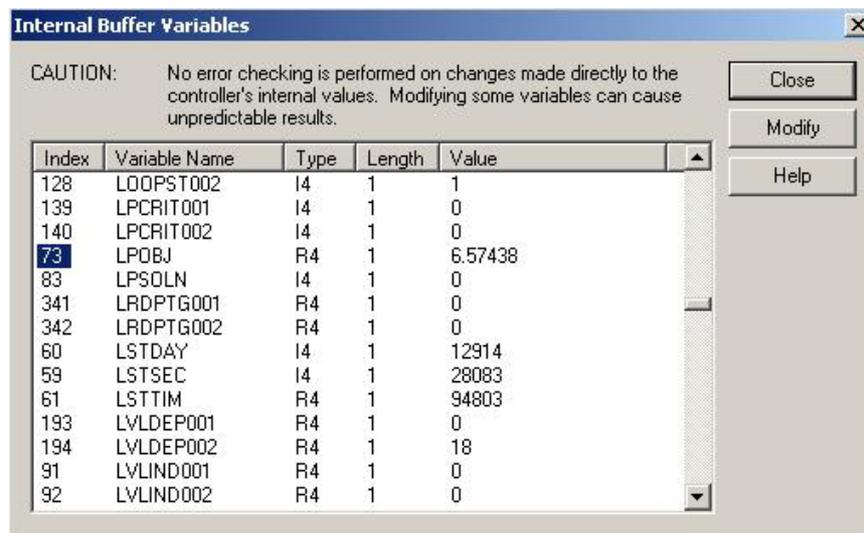
- Service status
- Lower limit
- Current value
- Steady State target
- Upper limit
- Current move
- Active constraint
- Oper. Srv. Switch
- Engr. Srv. Switch
- Critical flag
- Maximum move
- Move suppression
- Maximum SS step
- SS criterion
- SS cost
- Shadow price
- Lower Eng limit
- Upper Eng limit
- Lower Val limit
- Upper Val limit
- Loop status
- Windup status
- Reverse acting
- Limit tolerance
- Tracking lag
- Future moves
- Subctrl name
- ET criterion
- ET cost
- ET Lower rank
- ET Upper rank
- ET Srv switch
- ET status
- ET mode switch
- ET Upd time (day)
- ET Upd time (sec)
- External Target
- ET track
- ET range
- New ET flag
- ET age
- ET Lower Sol type
- ET Upper Sol type
- ET Lower ECE
- ET Upper ECE
- Typical move

Y en las siguientes columnas aparecen los parámetros que podemos variar para las variables CV:

- Service status
- Lower limit
- Current value
- Steady State target
- Upper limit
- Ramp Setpoint
- Active constraint
- Steady State error
- Oper. Srv. Switch
- Engr. Srv. Switch
- Critical flag
- Lower Eng limit
- Upper Eng limit
- Lower Val limit
- Upper Val limit
- Use prediction
- Max pred cycles
- Limit tracking
- Shadow price
- Prediction error
- Avg. Pred. Error
- Prediction Filter type
- Prediction Filter time constant
- Prediction Filter horizon

- Prediction initialization switch
- Maximum CV step
- Lower limit rank
- Upper limit rank
- SS lower Eq. Concern
- SS upper Eq. Concern
- Dyn Lower Eq. Concern
- Dyn Middle Eq. Concern
- Dyn Upper Eq. Concern
- Lower Trans. Zone
- Upper Trans. Zone
- Control Weight
- Noise Standard desviation
- Ramp Rot factor
- Ramp rate
- Ramp horizon
- Ramp max imbalances
- Prediction
- Prediction with control
- Subctrl names
- ET Lower rank
- ET Upper rank
- ET Srv switch
- ET status
- ET mode switch
- ET Upd time (day)
- ET Upd time (sec)
- External Target
- Lower rank sol type
- Upper rank sol type
- ET track
- ET range
- ET Lower Sol type
- ET Upper Sol type
- ET Lower ECE
- ET Upper ECE
- New ET flag
- ET age

Existe también la posibilidad de ver otra serie de parámetros del controlador, como por ejemplo la función objetivo del controlador, los cuales también podemos variar pero no se aconseja, ya que influyen directamente en el funcionamiento interno del mismo. Para poder acceder a estos parámetros hemos de ir al menú Controller> Internal Variables. Entonces se nos abre una nueva ventana en la que aparecen los parámetros internos del controlador. En la figura 5.43 se observa esta ventana, siendo el parámetro que se encuentra resaltado el valor de la función objetivo.



Index	Variable Name	Type	Length	Value
128	LOOPST002	I4	1	1
139	LPCRIT001	I4	1	0
140	LPCRIT002	I4	1	0
73	LPOBJ	R4	1	6.57438
83	LPSOLN	I4	1	0
341	LRDPTG001	R4	1	0
342	LRDPTG002	R4	1	0
60	LSTDAY	I4	1	12914
59	LSTSEC	I4	1	28083
61	LSTTIM	R4	1	94803
193	LVLDEP001	R4	1	0
194	LVLDEP002	R4	1	18
91	LVLIND001	R4	1	0
92	LVLIND002	R4	1	0

Figura 5.43.- Parámetros internos del controlador

### ACTUALIZACIÓN DEL CONTROLADOR

Una vez que se han realizado los ajustes necesarios en los parámetros del controlador, para hacerlos efectivos para su ejecución en línea con el proceso hemos de hacer que se reflejen en el archivo de configuración del controlador. Para realizar esta operación Simulate nos proporciona una función para hacerlo de forma automática. Así, accediendo al menú Controller> Update Configuration, el controlador queda actualizado automáticamente con los valores que se tengan en Simulate, sin necesidad de volver a Build y hacerlo de forma manual.

### ARCHIVOS PSM

Hemos comentado anteriormente que este tipo de archivos se utiliza para salvar las distintas configuraciones que se hagan de un mismo controlador. Así podremos hacer los cambios que veamos necesarios y poder guardar todos ellos de forma ordenada y sin

perder información. Además este tipo de archivos sólo se utilizan en Simulate, con lo que no existe la posibilidad de que se modifique mediante otro programa.

Para guardar el estado de un controlador en el formato PSM sólo hay que usar la función de guardar File> Save As, eligiendo antes de aceptar el formato de archivos .psm.

#### **5.5.4.- Ventanas de simulación**

Las ventanas gráficas o de simulación nos dan la posibilidad de seguir la evolución de las variables según la simulación que nos proporciona Simulate. Para abrir estas ventanas, una por cada variable del controlador, simplemente picamos con el ratón dos veces en la ventana de las tablas sobre la variable en cuestión, generándose al momento una nueva ventana con un gráfico para dicha variable.

Podemos observar en la figura 5.44 que en estos gráficos se muestran varios trazos, los cuales corresponden a los siguientes datos:

- LP Target: objetivo a alcanzar según solución lineal.
- Without Control: evolución de la variable sin utilizar el control.
- With Control: evolución de la variable de forma controlada.
- Up Limit: valor máximo para situar el objetivo a alcanzar de la variable.
- Low Limit: valor mínimo para situar el objetivo a alcanzar de la variable.

Es posible además cambiar el aspecto de estas ventanas mediante el cambio de colores, de líneas, escalas y demás. Para ello la forma más fácil de acceder a estas funciones es utilizar el menú contextual picando con el botón derecho del ratón cuando estemos situados en la ventana que nos interese.

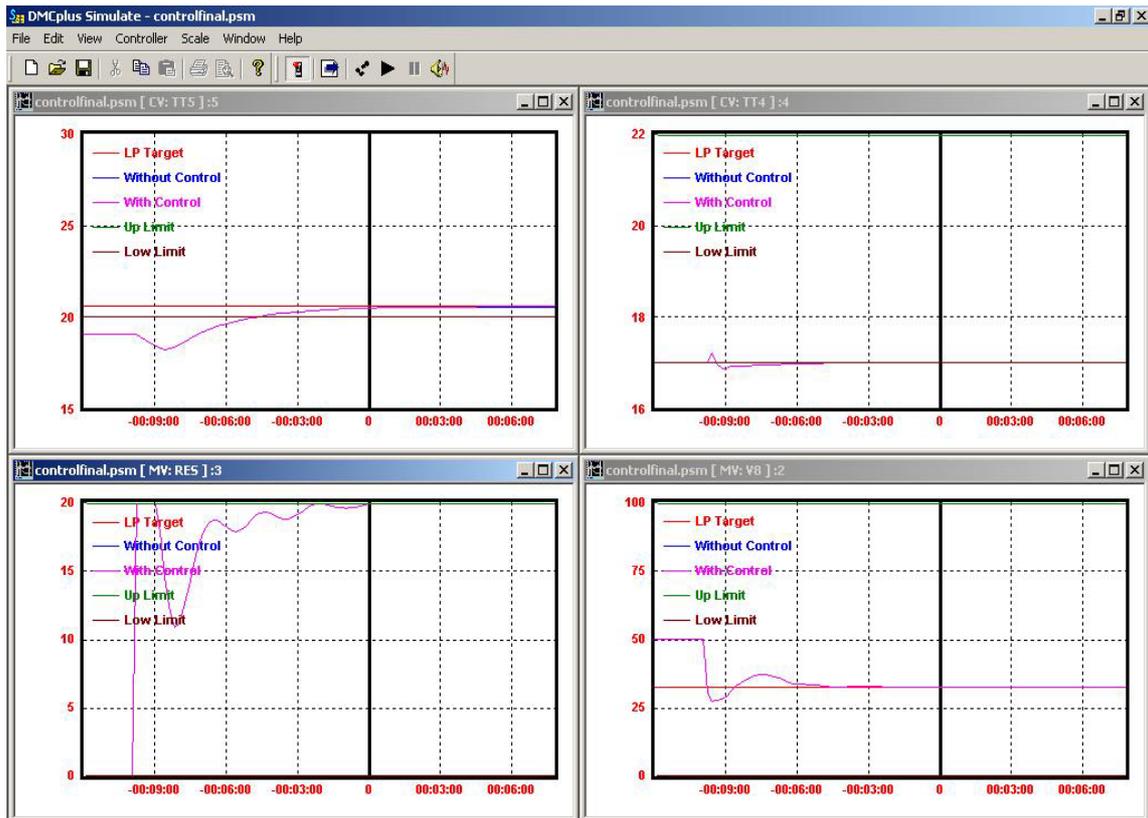


Figura 5.44.- Gráficos de simulación

### 5.5.5.- Realizar una simulación

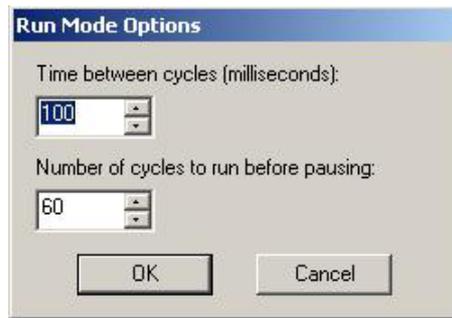
Para realizar una simulación del proceso una vez que tenemos cargado el controlador o un archivo .psm solamente hay que pulsar el botón Run Simulation de la barra de herramientas. Entonces comenzará la simulación del proceso según las opciones que hayamos marcado antes de ejecutar la simulación.

#### 5.5.5.1.- Opciones de simulación

##### RUN MODE OPTIONS

Accesibles a través de Controller> Run Mode Options, podemos especificar los siguientes parámetros de la simulación:

- Time between cycles (ms): se trata del tiempo que transcurrirá entre cada ciclo de ejecución del controlador. Esto nos sirve para poder seguir la evolución de la simulación pausadamente.

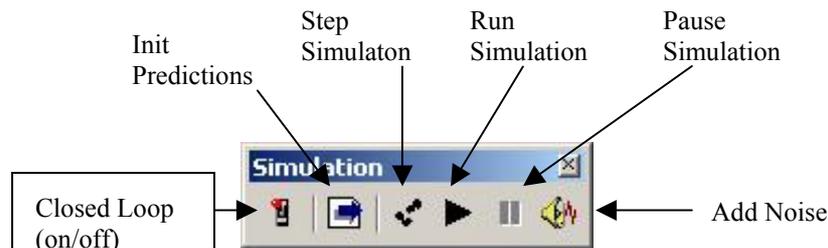


**Figura 5.45.- Run Mode Options**

- Number of cycles to run before pausing: como indica su nombre, son el número de ciclos del controlador que se simularán antes de parar la simulación.

### HERRAMIENTAS DE SIMULACIÓN

Entendemos por éstas a las funciones disponibles en la barra de herramientas de simulación. Estas funciones nos ayudan a trabajar con la simulación de una forma bastante cómoda:



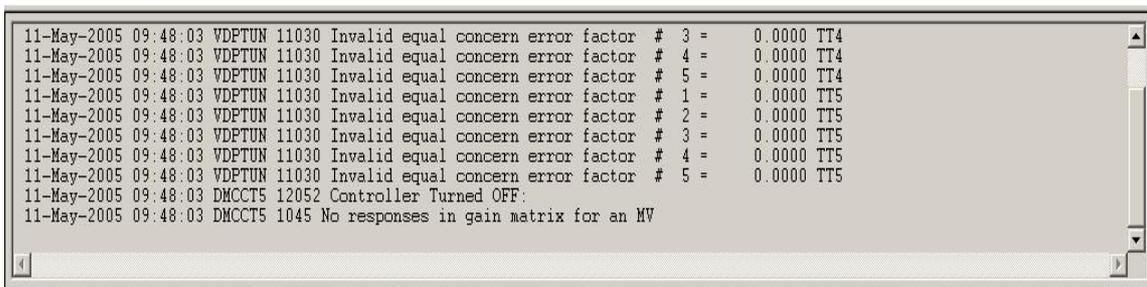
**Figura 5.46.- Barra de herramientas de simulación**

- Run Simulation: inicia la simulación del proceso en el estado actual del controlador.
- Pause Simulation: para la simulación en cualquier instante.
- Step Simulation: inicia la simulación pero de forma que se para cada vez que se ejecuta un ciclo del controlador.
- Add Noise : se utiliza para añadir ruido a las variables CV. Para que tenga efecto deben haberse configurado correctamente el parámetro correspondiente de cada variable en el controlador (SIMCVSTDDEV). El ruido que se añade es un ruido blanco, con una distribución Normal de media cero y desviación estándar dada por el parámetro anterior.
- Init Predictions: podemos utilizar este botón para inicializar las predicciones del controlador en cualquier momento.

- Closed Loop (on/off): se utiliza para poner en marcha la acción del controlador. Pulsar el botón para que empiece a ejercer su acción no indica que se ejecute el controlador, puesto que puede ocurrir que no se den las condiciones para que el controlador pueda arrancar (variables críticas fuera de rango, algún parámetro no configurado...).

### **5.5.5.2.- Área de mensajes**

Volviendo la vista atrás hasta la figura 5.42 podemos observar que debajo de las tablas de las variables aparece un área gris. Esta área es la denominada zona de mensajes. En esta área Simulate nos insertará una serie de mensajes informativos cuando se realice la simulación.



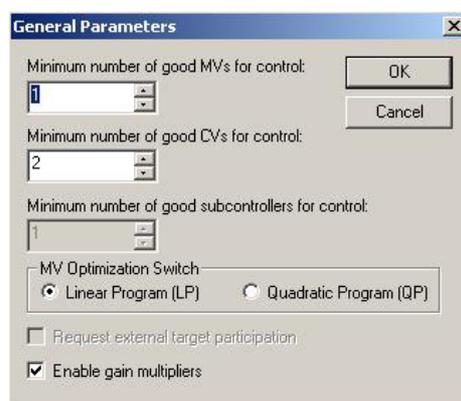
**Figura 5.47.- Mensajería generada por Simulate al realizar una simulación**

Los mensajes generados son de gran utilidad para detectar errores en la configuración del controlador, como se ve en la figura anterior.

### **5.5.5.3.- Edición del controlador**

#### **PARÁMETROS GENERALES**

Se tratan de varios parámetros que tienen gran importancia en el funcionamiento del controlador y que se pueden variar si accedemos a ellos mediante Controller> General.



**Figura 5.48.- Parámetros generales que podemos cambiar**

### EDICIÓN DEL MODELO DE LA PLANTA

Podemos editar el modelo de la planta de tal forma que elegiremos entre el modelo que lleva incorporado el controlador o un modelo importado, aunque es recomendable utilizar el modelo del controlador ya que si utilizamos otro modelo puede que se den incompatibilidades entre el controlador y el nuevo modelo.

Para ello acceder al editor del modelo, Controller> Edit Plant Model.

### EDICIÓN DEL MODELO DEL CONTROLADOR

El modelo del controlador puede ser modificado de forma independiente al modelo de la planta cambiando los multiplicadores de ganancia del controlador. Si se habilitan en Simulate los multiplicadores de ganancia, el controlador aplicará estos multiplicadores de ganancia a los cálculos llevados a cabo por el controlador. El modelo de la planta nunca aplicará los multiplicadores de ganancia a las predicciones y por lo tanto podrá analizarse en el entorno de simulación los posibles desajustes del modelo.

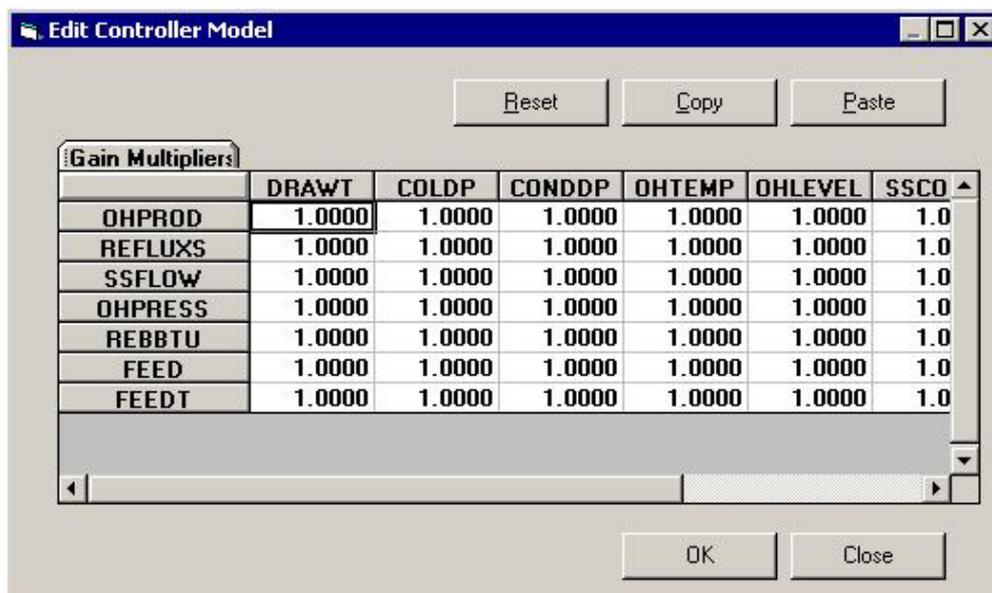


Figura 5.49.-Controller's Gain Multipliers

Para poder editar los multiplicadores de ganancia del controlador es necesario habilitar esta característica en los parámetros generales (ver figura 5.49). Una vez habilitados, basta con dirigirnos a Controller> Edit Controller Model.

Hay que tener en cuenta que los valores que podemos introducir deben estar dentro del intervalo que se especifique en Build mediante las entradas de la sección Configure GMULTL y GMULTU, por defecto 0.5 y 2.0, activando además el flag GMULTE.

## **5.6.- DMCplus Manage**

Tras haber pasado por los módulos de identificación del proceso, configuración del controlador y simulación del control del proceso llegamos finalmente al módulo mediante el cual gestionamos los controladores existentes en el sistema y ejecutamos los controladores que deseemos.

Para lanzar este módulo de DMCplus existen dos formas, o bien abrimos una sesión MS-DOS y ejecutamos el comando *dmcpmanage*, o bien a través de Inicio> Programas> AspenTech> Aspen Manufacturing Suite> DMCplus Online> Manage.

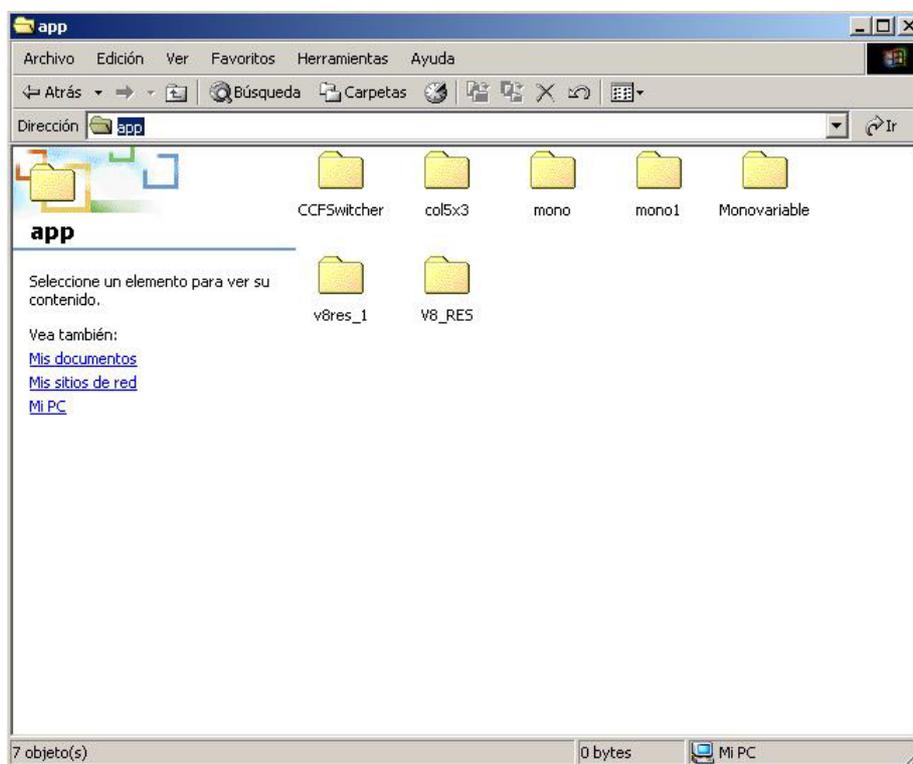
### **5.6.1.- Antes de empezar**

Para poder ejecutar un controlador mediante este programa es necesario disponer del archivo del modelo del proceso a controlar, MDL, juntamente con el archivo de configuración del controlador, CCF, ambos guardados en una carpeta determinada.

Se creará una carpeta o directorio nuevo para cada controlador que vayamos a utilizar. Estas carpetas irán en colgadas del directorio raíz app/ de Aspen Tech. En nuestro caso este directorio es el siguiente:

C:\Archivos de programa\AspenTech\AC Online\app\

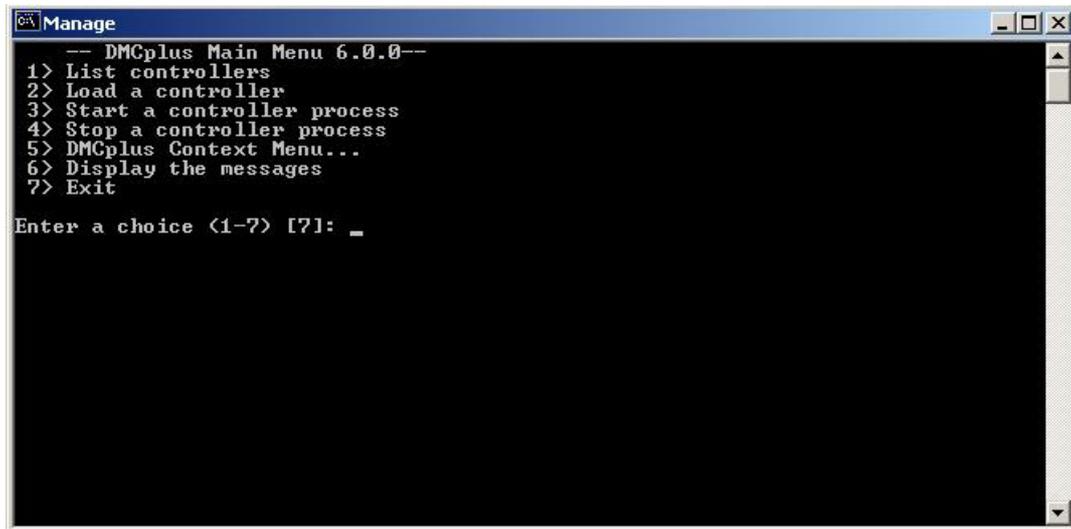
Así se tendrán una serie de carpetas en dicho directorio, como en la figura 5.50.



**Figura 5.50.- Carpetas de los controladores en el directorio app\**

### 5.6.2.- Menú Principal

Una vez ejecutado el comando *dmcmanage* nos aparecerá la pantalla principal tal y como sale en la figura 5.51.



```

-- DMCplus Main Menu 6.0.0--
1> List controllers
2> Load a controller
3> Start a controller process
4> Stop a controller process
5> DMCplus Context Menu...
6> Display the messages
7> Exit

Enter a choice <1-7> [?]: _

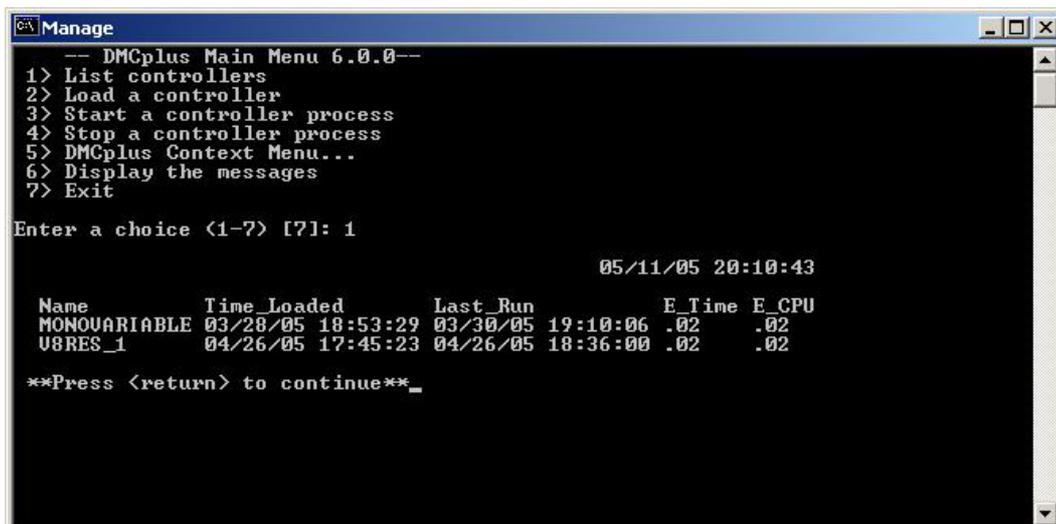
```

Figura 5.51.- Menú principal de la aplicación

A continuación detallamos la funcionalidad de cada una de las opciones que nos aparecen:

1 > List controllers: muestra todos los controladores cargados en forma de lista, de los cuales se da la siguiente información: nombre del controlador, momento en que se cargó, hora de la última ejecución y tiempo transcurrido para la última ejecución.

Además al principio de cada controlador pueden aparecer otros dos símbolos: \*, indicando que el controlador marcado está en ejecución, y A, que indica que el controlador está marcado para funcionar en autoarranque.



```

-- DMCplus Main Menu 6.0.0--
1> List controllers
2> Load a controller
3> Start a controller process
4> Stop a controller process
5> DMCplus Context Menu...
6> Display the messages
7> Exit

Enter a choice <1-7> [?]: 1

                                05/11/05 20:10:43

Name          Time_Loaded      Last_Run          E_Time  E_CPU
MONOVARIABLE  03/28/05 18:53:29 03/30/05 19:10:06  .02   .02
U8RES_1       04/26/05 17:45:23 04/26/05 18:36:00  .02   .02

**Press <return> to continue**_

```

Figura 5.52.- Lista de controladores cargados en Manage

- 2 > Load a controller: esta opción se utiliza para cargar un controlador para poder usarlo normalmente. Al elegir esta opción el programa nos guía para realizar la carga del controlador.
- 3 > Start a controller process: si el controlador que queremos ejecutar está cargado, esta opción se usa para comenzar su ejecución.  
Hay que tener claro que indicar a Manage que comience la ejecución del controlador no implica que éste lo haga, sino que dependerá del parámetro ONREQ del archivo CCF.
- 4 > Stop a controller process: como indica el nombre de esta opción, detendremos el controlador que se desee.
- 5 > DMCplus Context Menu: esta opción nos muestra un nuevo menú con opciones para la configuración de los controladores. Ver en un capítulo más adelante.
- 6 > Display the messages: para mostrar los mensajes generados por uno o por todos los controladores.

Estos mensajes pueden ser configurados y controlados. Para ello es necesario modificar el archivo que controla dicha configuración:

C:\Archivos de programa\AspenTech\AC Online\cfg\message.dat

```

C:\ Manage
You may specify 'all' to get the messages for all controllers
Enter controller name, or 'quit' to stop [ALL]:
04/26/05 18:36:11 U8RES_1 Control process exiting
04/26/05 17:45:49 U8RES_1 12050 U8RES_1 PRED5 Prediction Initialized
by Engine.
04/26/05 17:45:49 U8RES_1 14007 U8RES_1 MCYCLE Too many missed
cycles: 92877524
04/26/05 17:45:41 U8RES_1 Control process startup complete
04/26/05 17:45:41 U8RES_1 Controller re-mapped
04/26/05 17:45:40 U8RES_1 Control process startup beginning
04/26/05 17:45:23 U8RES_1 Controller definition loaded from
CCF control
04/26/05 17:39:33 U8RES_1 Control process exiting
04/26/05 17:23:49 U8RES_1 12050 U8RES_1 PRED5 Prediction Initialized
by Engine.
04/26/05 17:23:49 U8RES_1 14007 U8RES_1 MCYCLE Too many missed
cycles: 92877418
04/26/05 17:23:42 U8RES_1 Control process startup complete
04/26/05 17:23:42 U8RES_1 Controller re-mapped
04/26/05 17:23:42 U8RES_1 Control process startup beginning
04/26/05 17:22:54 U8RES_1 Controller definition loaded from
CCF control
04/26/05 17:22:25 U8RES_1 1045 U8RES_1 DMCCT5 No responses in gain
matrix for an MU
Continue? (Y/N) [Y]:
    
```

Figura 5.53.- Mensajes de los controladores

- 7 > Exit: salimos de Manage. Presionar Intro sin haber elegido ninguna opción tiene el mismo efecto.

### 5.6.3.- Menú secundario

Este segundo menú o Context Menu se utiliza para controlar la configuración de los controladores que vamos a cargar.

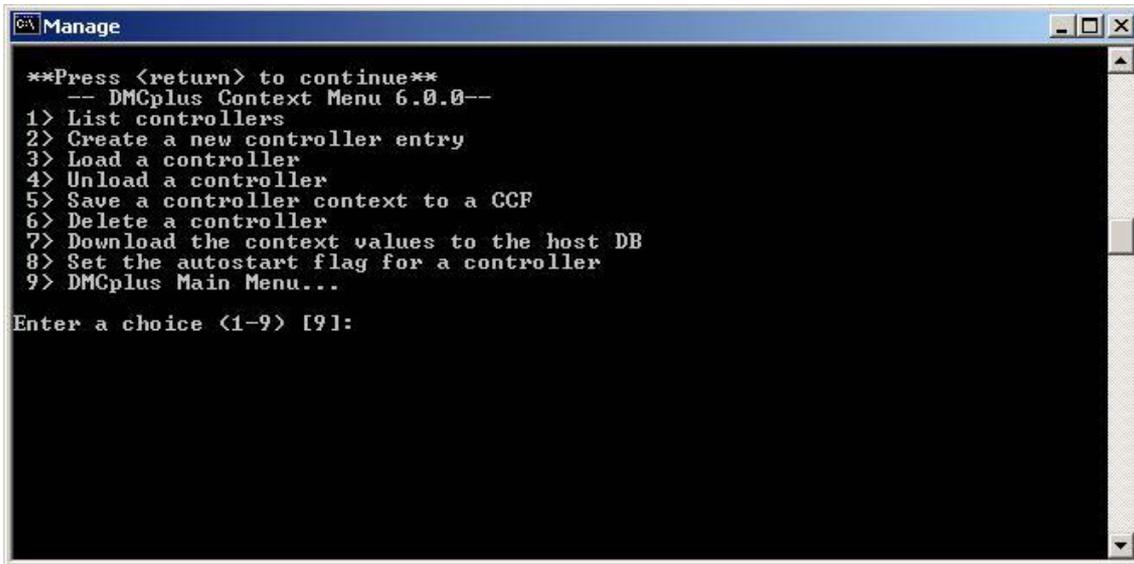


Figura 5.54.- Context Menu

Algunas de las opciones de este menú son ya conocidas por lo que se detallarán solo aquellas que sean nuevas:

- 2 > Create a new controller entry: se utiliza para incluir un nuevo controlador en la lista de controladores. Aunque se trata de una opción no necesaria puesto que cuando se carga un controlador este paso se realiza de forma automática, es bueno hacer este paso para especificar el orden de los controladores.
- 4 > Unload a controller: al utilizar esta opción el controlador se mantendrá en la lista aunque no podrá ejecutarse hasta que lo volvamos a cargar. Para eliminar completamente el controlador de la lista debe usarse la opción 6.
- 5 > Save a controller context to CCF: mediante esta opción se guarda una fotografía de los valores actuales del controlador en un archivo CCF para poder verlo desde Build o Simulate.
- 6 > Delete a controller: elimina el controlador de la lista. Antes de ser eliminado el controlador debe detenerse si está ejecutándose, además es recomendable guardarlo mediante la opción 5 antes de borrarlo.
- 7 > Download the context values to the host DB: esta opción hace que se envíen los valores del controlador a la base de datos del DCS (Distributed Control System). Es recomendable realizar esta operación antes de resintonizar los parámetros con

Simulate, pero no debe realizarse cuando el controlador esté en ejecución.

Los parámetros que se envían son los que estén configurados en:

C:\Archivos de programa\AspenTech\AC Online\cfg\download.dat

8 > Set the autostart flag for a controller: esta opción permite marcar un controlador para que se inicie de forma automática. Sin embargo, igual que en el caso de la ejecución manual, el proceso de control empieza en el momento que lo permita ONREQ.

## **5.7.- CIM-IO**

Aunque la configuración de la parte Cim-IO se detalló en el capítulo dedicado a la instalación de las herramientas necesarias para trabajar, tenemos a nuestra disposición una serie de herramientas para trabajar con nuestra interface Cim-IO para archivos OPC cuya utilización se describe en este apartado.

### **5.7.1.- Inicio de la interface**

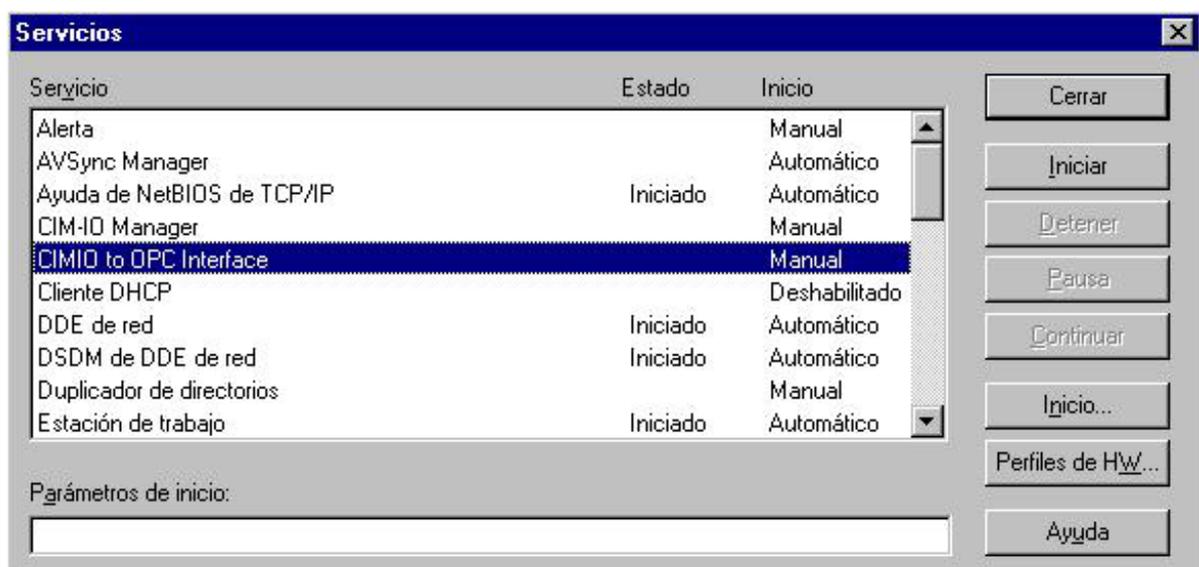
Para comunicar DMCplus es necesario tener activada la interfaz Cim-IO, de lo contrario los módulos online de DMCplus que utilicemos (en nuestro caso Collect y Manage) no podrán acceder a los datos del proceso.

El arranque de la interfaz es un proceso muy simple que puede realizarse de dos formas:

- A. Utilizando la utilidad StartAndStop, situada en Inicio> Programas> Aspen Tech> Aspen Manufacturing Suite> CIM-IO> Servers> CIM-IO for OPC> CIM-IO for OPC StartStop. En la ventana que nos aparezca deberemos especificar el servidor OPC al que queremos tener acceso.

Este método tiene el inconveniente de que ha de estar ejecutándose previamente el Manager del Cim-IO, aunque no es nada importante.

- B. Accediendo a los servicios del sistema operativo, Inicio> Configuración> Panel de Control> Servicios. Nos encontraremos entonces con una serie de servicios en ejecución o no entre los que se encontrará “Cim-IO for OPC interface”. Lo seleccionamos y lo arrancamos de forma manual.



**Figura 5.55.- Inicio de la interfaz mediante Servicios de Windows NT**

## OBSERVACIONES IMPORTANTES

1. La interfaz deberá arrancarse antes de utilizar alguno de los productos online que necesiten datos OPC.
2. Previamente al arranque de la interfaz el servidor de licencias ALM ha de estar en ejecución.
3. El servidor OPC, en nuestro caso Cube cuyo nombre del servidor es OPC.Cube.1, deberá estar listo antes también de que se lance la interfaz, pues de lo contrario no reconocerá al servidor y habrá que reiniciar el arranque de la interfaz.

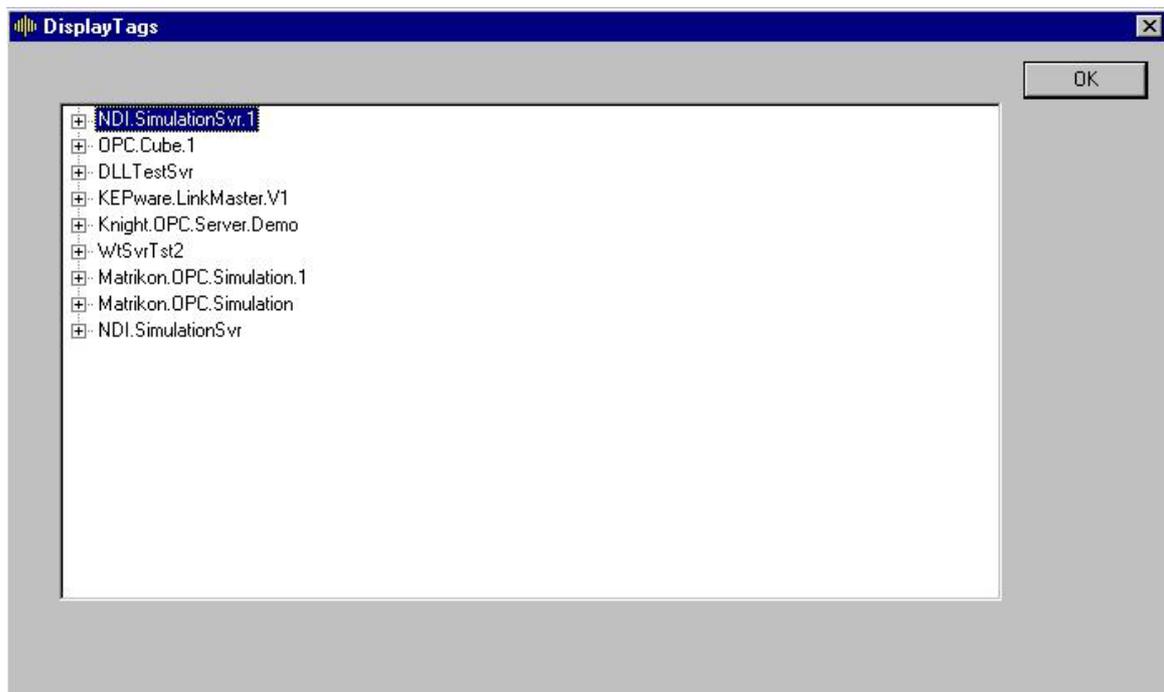
Para poner en marcha el servidor OPC de Cube basta con iniciar una sesión de Cube con la planta que vayamos a utilizar.

## **5.7.2.- Utilidades de la interfaz CIM-IO**

### **5.7.2.1.- Display Tags Utility**

Esta utilidad es nos permite comprobar todos los servidores OPC existentes en el equipo y las etiquetas de las variables de cada uno de ellos.

Accedemos a través de Inicio> Programas> Aspen Tech> Aspen Manufacturing Suite> CIM-IO> Servers> CIM-IO for OPC> CIM-IO for OPC DisplayTags.



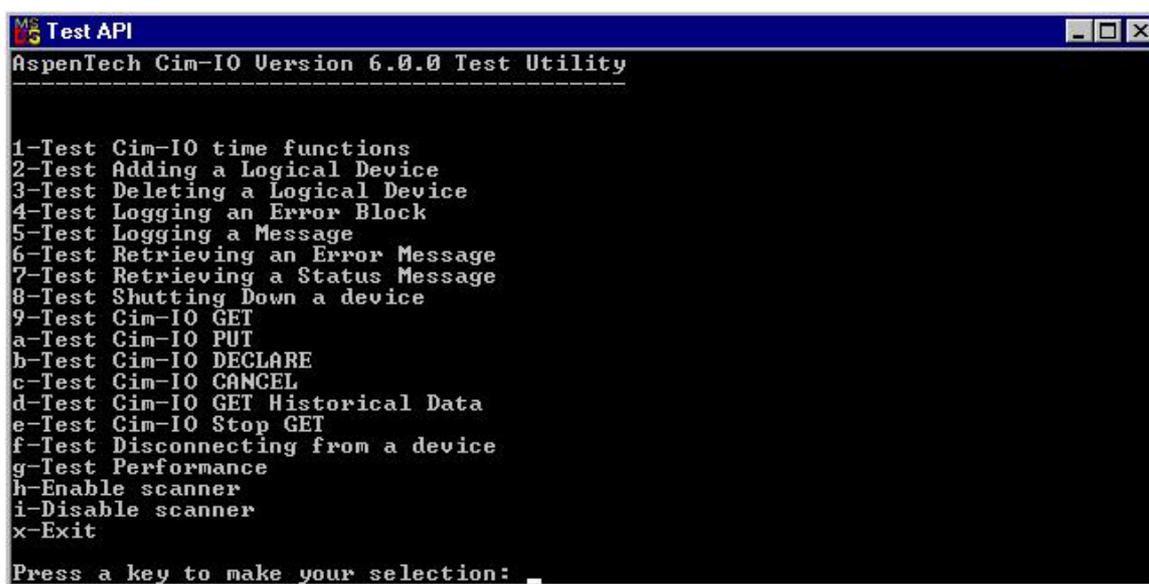
**Figura 5.56.- Servidores OPC disponibles en el equipo**

### 5.7.2.2.- Test API

Se trata de una utilidad que nos permite comunicarnos de forma interactiva con el servidor OPC, muy útil para comprobar el funcionamiento de la comunicación entre el servidor OPC, en nuestro caso Cube, y el cliente, que es la interfaz Cim-IO.

Para ejecutar esta herramienta hemos de ir a Inicio> Programas> Aspen Tech> Aspen Manufacturing Suite> CIM-IO> Servers> Test API.

Se nos mostrará entonces una sesión MS-DOS en la cual aparecen una serie de opciones para comprobar la conexión cliente-servidor OPC.



```
MS-DOS Test API
AspenTech Cim-IO Version 6.0.0 Test Utility
-----
1-Test Cim-IO time functions
2-Test Adding a Logical Device
3-Test Deleting a Logical Device
4-Test Logging an Error Block
5-Test Logging a Message
6-Test Retrieving an Error Message
7-Test Retrieving a Status Message
8-Test Shutting Down a device
9-Test Cim-IO GET
a-Test Cim-IO PUT
b-Test Cim-IO DECLARE
c-Test Cim-IO CANCEL
d-Test Cim-IO GET Historical Data
e-Test Cim-IO Stop GET
f-Test Disconnecting from a device
g-Test Performance
h-Enable scanner
i-Disable scanner
x-Exit
Press a key to make your selection: _
```

Figura 5.57.- Sesión MS-DOS de Test API con sus opciones

Las opciones del test de conexión nos permiten llevar a cabo funciones simples como GET y PUT para verificar el funcionamiento del Cim-IO. Al seleccionar alguna de ellas deberá proporcionarse cierta información como:

- List ID: grupo de una o más etietas o tag names. Han de ser números positivos a no ser que sea la primera vez que se usa el grupo o las etiquetas han cambiado, por lo que será un número negativo.
- Access Type: síncrono o asíncrono, aunque el acceso asíncrono puede que no sea soportado por todos los servidores.
- Tagname Entry Options: se permite la introducción de información sobre las etiquetas a través de archivos, aunque no se suele utilizar.
- Output Type: debemos decidir el tipo de dato de salida. Suelen ser datos reales enteros, aunque ha de consultarse la documentación del producto que vayamos a usar.

### OPCIONES DEL TEST

- 1- Test CIM-IO time functions: comprueba las funciones de conversión de tiempo.
- 2- Test Adding a Logical Device: añade un dispositivo lógico en el archivo cimio\_logical\_devices.def como se hacía en la configuración general.
- 3- Test Deleting a Logical Device: realiza la acción contraria al anterior.
- 4- Test Logging an Error Block: busca en el archivo cimio\_msg.log información acerca de los errores en las peticiones I/O.
- 5- Test Logging a Message: comprueba la mensajería en el archivo cimio\_msg.log.
- 6- Test Retrieving an Error Message: permite mostrar un mensaje de error en pantalla.
- 7- Test Retrieving a Status Message: nos muestra información acerca de un error.
- 8- Test Shutting Down a Device: utilizado para cerrar un dispositivo.
- 9- Test Cim-IO GET: para leer u obtener un dato del servidor.
  - a- Test Cim-IO PUT: contrario al anterior.
  - b- Test Cim-IO DECLARE: especifica etiquetas cuyos valores serán obtenidos de forma asíncrona.
  - c- Test Cim-IO CANCEL: notifica al servidor la finalización del procesado de una lista de etiquetas declarado anteriormente.
  - d- Test Cim-IO GET Historical Data
  - e- Test Cim-IO Stop GET
  - f- Test Disconnecting from a Device
  - g- Test Performance: permite realizar varios GET o PUT y mostrar los valores por pantalla, proporcionando además al final del test una pequeña estadística de los datos tratado.
  - h- Enable Scanner: se usa con el Cim-IO Store & Forward.
  - i- Disable Scanner: ídem al anterior.
- x- Exit