



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍAS INDUSTRIALES

Máster en Investigación en Ingeniería de Procesos y
Sistemas Industriales

CONTROL PREDICTIVO Y OPTIMIZACIÓN ECONÓMICA DE UNA PLANTA DE LABORATORIO CON EL SOFTWARE DMC3

Autor:

Riquelme Esteve, Pablo

Tutores:

De Prada Moraga, César

Zamarreño Cosme, Jesús María

Dpto. de Ingeniería de Sistemas y Automática

Valladolid, septiembre 2018

RESUMEN

El proyecto consiste principalmente en el diseño e implementación a escala de laboratorio de un controlador predictivo y un sistema de optimización en tiempo real asociado, utilizando un controlador industrial multivariable. Para empezar, se elige un tipo de control predictivo basado en modelo MPC, y para concretar el algoritmo del que se va a hacer uso, se comienza introduciendo el algoritmo conocido como control predictivo DMC, control por matriz dinámica o simplemente Dynamic Matrix Control. Para realizar este proyecto, en primer lugar, se aplican técnicas de control predictivo multivariable a una planta del laboratorio de Ingeniería de Sistemas y Automática, incluyendo la identificación de la planta y el diseño del controlador, utilizando para ello el software Aspen DMC3™, de la suite aspenONE® v10 de AspenTech®. En una segunda etapa, junto a las acciones de control, se define y se aplica una capa de optimización en línea, para que la planta opere continuamente en función de sus objetivos de producción.

Se redacta un manual de uso de cada una de las herramientas de software utilizadas en la realización del proyecto, además de una serie de consejos y aspectos importantes a la hora de enfrentar un proceso como el que se describe aquí.

PALABRAS CLAVE: control, predictivo, control avanzado, DMC, AspenTech, modelado, identificación, optimización, multivariable, robusto, restricciones.

ABSTRACT

The project consists mainly in the design and implementation at laboratory scale of a predictive controller and an associated real-time optimization system, using a multivariable industrial controller. To begin with, a type of predictive control based on the MPC model is chosen, and to specify the algorithm that will be used, it starts by introducing the algorithm known as DMC predictive control, control by dynamic matrix or simply Dynamic Matrix Control. To carry out this project, first of all, multivariable predictive control techniques are applied to a plant in the Systems and Automation Engineering laboratory, using the Aspen DMC3™ software of AspenTech® aspenONE® v10 suite. In a second stage, together with the control actions, an online optimization layer is defined and applied, so that the plant operates continuously according to its production objectives.

A manual of use of each of the software tools used in the realization of the project is written, as well as a series of tips and important aspects when facing a process like the one described here.

KEYWORDS: predictive, control, advanced control, DMC, AspenTech, modelling, identification, optimization, multivariable, robustness, constraints.

ÍNDICE GENERAL

Capítulo 1. INTRODUCCIÓN	1
1.1. ¿Qué ofrece este capítulo?	2
1.2. Motivación.....	2
1.3. Antecedentes.....	2
1.4. Objetivos	5
1.4.1. Objetivo principal	5
1.4.2. Procedimiento.....	5
1.4.3. Descripción de la planta	6
1.4.4. Software	7
1.5. Estructura de la memoria	8
Capítulo 2. TOMA DE DATOS	10
2.1. ¿Qué se ofrece en este capítulo?.....	11
2.2. ReadOPC.....	11
2.2.1. Introducción	11
2.2.2. Manejo	11
2.3. Softing OPC Toolbox Demo Client v4.1	14
2.3.1. Introducción	14
2.3.2. Manejo	15
2.4. ¿Por qué dos softwares distintos?	17
Capítulo 3. MODELADO	19
3.1. ¿Qué se ofrece en este capítulo?.....	20
3.2. Aspen DMC3 Model	20
3.2.1. Introducción	20
3.2.2. Descripción general	20
3.2.3. Manejo	20
3.2.4. Modelado	22
3.2.5. Proyectos	23
3.2.6. Vectores	25
3.2.7. Casos	32
3.2.8. Algoritmos de identificación	35
3.2.9. Modelos	37
3.2.10. Predicciones	45
3.2.11. Listas de objetos	46
Capítulo 4. SISTEMA DE CONTROL.....	48

4.1.	¿Qué se ofrece en este capítulo?.....	49
4.2.	Introducción al control predictivo	49
4.3.	Propuesta de control	50
4.3.1.	Aplicabilidad	50
4.3.2.	Entonces, ¿Por qué elegir DMC?.....	50
4.3.3.	Interés industrial.....	51
4.4.	Estudio del control.....	52
4.4.1.	Modelos de convolución.....	52
4.4.2.	<i>Move suppression</i>	55
4.4.3.	<i>Equal Concern</i>	55
Capítulo 5.	SIMULACIÓN Y OPTIMIZACIÓN.....	57
5.1.	¿Qué se ofrece en este capítulo?.....	58
5.2.	Aspen DMC3 Builder	58
5.2.1.	Introducción	58
5.2.2.	Manejo	59
5.3.	Pantalla de DMC3 Builder.....	61
5.3.1.	Descripción general	62
5.4.	Usando Aspen DMC3 Builder	63
5.5.	aspenONE® Exchange	64
5.5.1.	Menú de archivo.....	67
5.5.2.	Creando un Nuevo Proyecto.....	69
5.5.3.	Flujo de trabajo en el menú de cinta	75
5.6.	Datasets	76
5.6.1.	<i>Vista Datasets</i>	77
5.6.2.	<i>Distributed</i>	77
5.6.3.	<i>Overlay</i>	78
5.6.4.	<i>Common</i>	79
5.6.5.	<i>Manage Lists</i>	80
5.6.6.	<i>Slicing</i>	81
5.6.7.	<i>Mark Bad</i>	81
5.6.8.	<i>Mark Interpolated</i>	82
5.6.9.	<i>Edit (Slices)</i>	84
5.7.	<i>Controllers</i>	84
5.7.1.	<i>Create Model</i>	84
5.7.2.	Concepto de <i>Master Model</i>	87
5.7.3.	Vista <i>Master Model</i>	89
5.7.4.	Árbol <i>Controller</i>	90

5.7.5.	Vista de los casos	90
5.7.6.	Configuración.....	96
5.7.7.	Optimización.....	98
5.7.8.	Simulación	100
5.7.9.	<i>Calculations</i>	105
5.7.10.	<i>Deployment</i>	107
5.8.	<i>Composite</i>	108
5.9.	Online.....	109
Capítulo 6.	CONCLUSIONES Y TRABAJO FUTURO.....	111
6.1.	¿Qué se ofrece en este capítulo?.....	112
6.2.	Conclusiones.....	112
6.3.	Trabajo futuro.....	112
REFERENCIAS	114
ANEXOS	118

ÍNDICE DE FIGURAS

Figura 1 Configuración inicial de ReadOPC.....	12
Figura 2 Etiquetas en ReadOPC	13
Figura 3 Exportar datos desde ReadOPC	14
Figura 4 Pantalla Inicial de Softing OPC	15
Figura 5 Selección de entradas y salidas	16
Figura 6 Acceso a las variables de la planta.....	17
Figura 7 Funcionamiento de un controlador multivariable	58
Figura 8 Establecer punto como símbolo decimal.....	59
Figura 9 Secuencia de funcionamiento del controlador	62
Figura 10 Empezando a usar DMC3 Builder	63
Figura 11 Tipos de Nuevo proyecto en Aspen DMC3 Builder	64
Figura 12 Pestaña de recursos.....	64
Figura 13 aspenONE Exchange All Content	65
Figura 14 Resultados de búsqueda para DMC3 en aspenONE Exchange	66
Figura 15 Descripción del artículo en aspenONE Exchange	66
Figura 16 AspenTech Support Knowledge Base Solution	67
Figura 17 Menú de archivo en Aspen DMC3 Builder	68
Figura 18 Ventana de opciones en Aspen DMC3 Builder.....	69
Figura 19 Asignar nombre y ubicación al Proyecto	69
Figura 20 Vista por defecto para nuevo Proyecto.....	70
Figura 21 Vista Datasets de Aspen DMC3 Builder	71
Figura 22 Botones de navegación.....	72
Figura 23 Dataset Exp080518	72
Figura 24 Árboles de navegación para cada una de las secciones: Datasets, Controllers, Composite y Online.....	73
Figura 25 Pasos para construir el controlador en Aspen DMC3.....	74
Figura 26 Los limitados elementos de la cinta de opciones indican que el próximo paso en importar los datos	75
Figura 27 Tras importar los datos, aparece el resto de opciones	75
Figura 28 Funciones y acciones relacionadas, organizadas en grupos	76
Figura 29 Vista Datasets en Aspen DMC3 Builder	77
Figura 30 Los vectores en una gráfica con vista distribuida	78
Figura 31 Puede verse cada vector en una sola gráfica, con diferentes escalas, en la vista Overlay	79
Figura 32 Vectores mostrados en una sola gráfica, con la misma escala, usando la vista Common.....	79
Figura 33 Administrar listas de vectores	80
Figura 34 Botón de edición en la opción Slicing	81
Figura 35 Marcar puntos como Bad Global Slice	81

ACRÓNIMOS

- APC: *Advanced Process Control*. Es el control avanzado de procesos.
- CAD: *Computer-Aided Design*. Es el diseño asistido por ordenador.
- CCF: *Controller Configuration File*. Es el archivo de configuración del controlador.
- CLC: *Collect file*. Son los archivos de recolección de datos.
- CV: *Controlled Variable*. Son las variables controladas del proceso.
- DCS: *Distributed Control System*. Hace referencia a un sistema de control distribuido.
- DMC: *Dynamic Matrix Control*. Control por matriz dinámica.
- FF: *Feedforward Variable*. Son las variables de perturbación del proceso.
- FIR: *Finite Impulse Response*. Respuesta al impulso finita.
- GPC: *Generalized Predictive Control*. Control predictivo generalizado.
- KB: *Knowledge Base*. Base de datos de la documentación del software.
- MAC: *Model Algorithmic Control*. Control algorítmico basado en modelo.
- MBPC: *Model Based Predictive Control*. Control predictivo basado en modelos.
- MIMO: *Multiple-Input Multiple-Output*. Sistemas con varias entradas y varias salidas.
- MPC: *Model Predictive Control*. Control predictivo por modelo.
- MV: *Manipulated Variable*. Corresponden a las variables manipuladas del proceso.
- OP: *Output to Process*. Es la variable de salida de un controlador.
- OPC: *Open Platform Communications*. Es un estándar de comunicación, que especifica las comunicaciones entre los datos de planta en tiempo real y los dispositivos de control.
- PID: *Proportional Integral Derivative*. Hace referencia a un controlador proporcional, integral y derivativo.
- PV: *Process Variable*. Son las variables de entrada a un controlador.
- SP: *Set Point*. Es el punto de consigna de un controlador.
- SS: *Steady State*. Estado estacionario.

Capítulo 1. INTRODUCCIÓN

1.1. ¿Qué ofrece este capítulo?

En este capítulo se expone una breve introducción de la motivación tras la realización de este proyecto. Se presenta, además, una serie de objetivos a conseguir una vez terminado el trabajo, tanto para el autor, como para los lectores.

Por otra parte, se pone en antecedentes al lector en cuanto al control predictivo, con una pequeña formación sobre qué es lo importante a tener en cuenta a la hora de diseñar un controlador de estas características. En este mismo apartado, se introduce el software protagonista que se utilizará para realizar el diseño y se incluye una descripción de la planta es la que se desarrolla el proyecto.

1.2. Motivación

En busca de la optimización económica, la máxima explotación de la capacidad de producción y la mínima variabilidad en la calidad de los productos, el control predictivo y mantenimiento predictivo, actualmente, son aspectos clave para la industria. Sobre todo, teniendo en cuenta que las plantas son cada día más complejas, y las exigencias de producción y calidad, mayores.

1.3. Antecedentes

El control predictivo basado en modelos (*Model Predictive Control*, MPC) se ha desarrollado considerablemente en las últimas décadas, tanto en la industria como en la comunidad de investigación. Este éxito se debe a que el Control Predictivo basado en Modelo es quizás la forma más general de formular el problema de control en el dominio del tiempo. El control predictivo integra control óptimo, control de procesos con tiempos muertos, procesos multivariables y utiliza las referencias futuras cuando están disponibles. Al utilizar una estrategia con horizonte de control finito permite la consideración de restricciones y procesos no lineales. (Bordons & Camacho, Octubre 2004)

Entonces, ¿Qué es el Control Predictivo?

Se puede decir que es una técnica de control avanzado, tal y como se puede ver en la Ilustración 1, pero para ello primero se ha de definir qué es el control avanzado:

El término apareció en los años 60 y se aplicaba a toda técnica de control que no fuese un PID. Hoy en día, se define como un conjunto de técnicas de control que toma ideas de la Ingeniería de Control, la Instrumentación y Procesos. Es muy importante tener en cuenta que, para poder aplicar técnicas de control, se debe entender muy bien el proceso y la técnica que se quiere aplicar.

¿Cuáles son las ventajas del control avanzado?

Tabla 1 Control Clásico frente a Control Avanzado

CONTROL CLÁSICO	CONTROL AVANZADO
Sistemas sencillos	Sistemas multivariables
Control a bajo nivel	Dinámicas complejas
	Optimización económica
	Estabilidad y robustez
	Control de calidad
	Seguridad

El control avanzado puede verse como un control de supervisión. No pretende sustituir a las técnicas de control clásico en lazos simples, sino que se sitúa típicamente al nivel del control de una unidad de proceso con múltiples variables manipuladas, controladas y perturbaciones medibles.

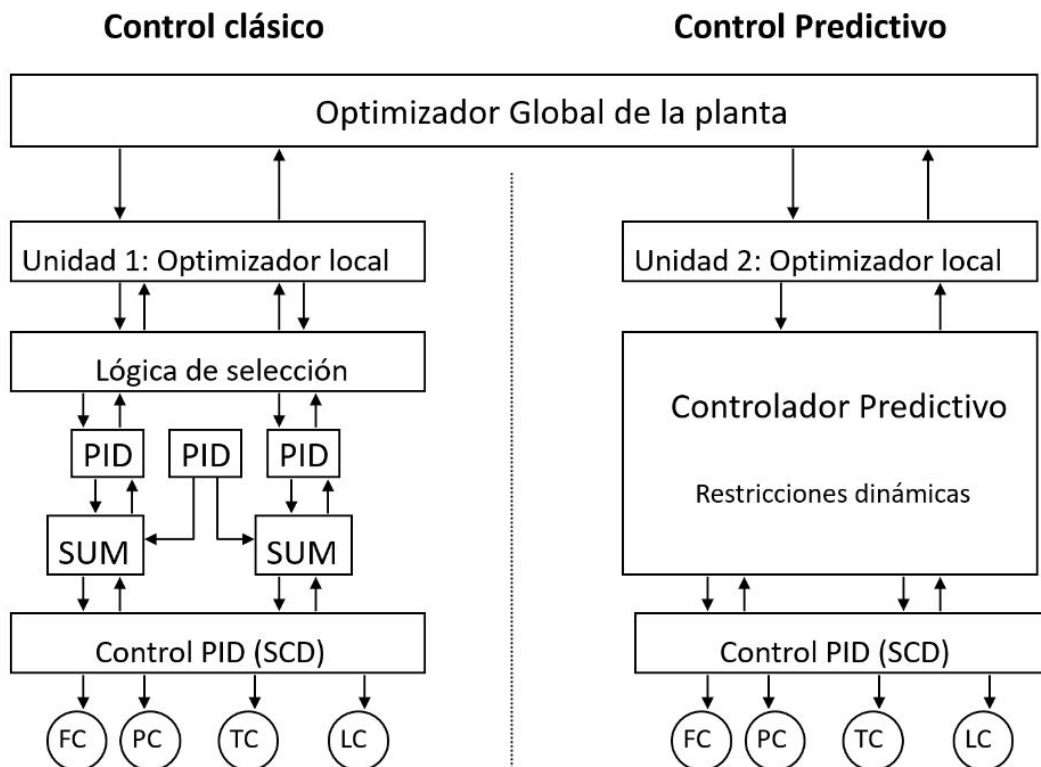


Ilustración 1 Control clásico frente a Control predictivo. (Badgwell & Qin, 2003)

Tabla 2 Jerarquía del control de una planta. (Richalet, et al., 1978)

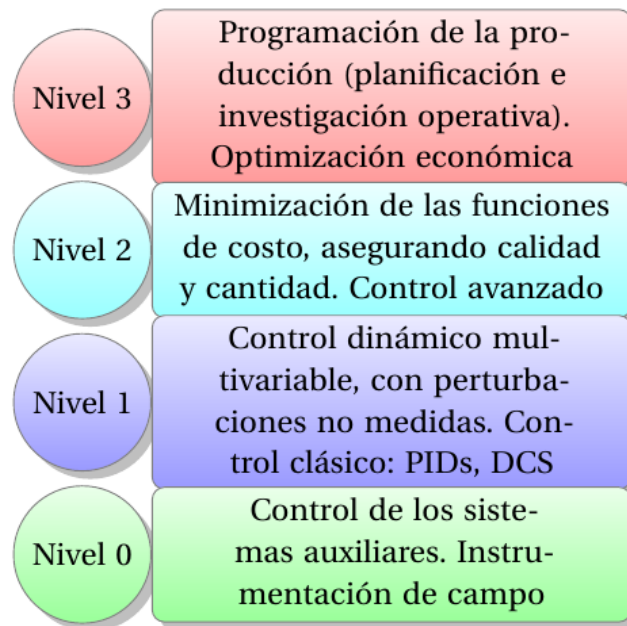


Tabla 3 Ventajas y desventajas del MPC. (Richalet, 1993)

VENTAJAS	DESVENTAJAS
Control mono o multivariable de procesos estables o inestables en lazo abierto que no se comportan bien: enfoque universal	Necesita un modelo dinámico de caja negra del proceso: se deben aplicar señales de prueba
Sin derivada explícita: sin efecto de ruido de medición	Exige más capacidad computacional: asignación de memoria y tiempo de cálculo. No es un problema para los procesos lentos, pero pueden ser críticos para una aplicación mecatrónica
Sin integrador explícito: <i>antiwind-up</i> no es necesario	Nuevo método
Gestionar las restricciones en las MV y CV (<i>override</i>)	Necesita personal técnico con entrenamiento en: modelado, identificación, control digital, control jerárquico y

	nuevos conceptos
Es sencillo hacer <i>feed-forwarding</i>	El enfoque en ingeniería es diferente
Sin error de retraso en una consigna polinomial	
Enfoque CAD: sin sintonización manual, se necesita menos personal	
Equilibrio entre robustez y la dinámica es negociable	
Mantenimiento fácil	
El modelado en línea abre el camino a la monitorización y el diagnóstico	

1.4. Objetivos

1.4.1. Objetivo principal

El objetivo de este proyecto es el diseño e implementación en un sistema real de un controlador predictivo capaz de controlar una planta usando software industrial. El software que se utiliza para ello es el Aspen DMC3™ (Aspen Technology, Inc., 2018). Posteriormente, se realiza una optimización económica en línea, para que la eficiencia sea máxima. En particular se pretende:

- Adquirir las habilidades y el conocimiento requeridos para liderar o participar en un equipo encargado del desarrollo y despliegue de una aplicación de control avanzado de procesos.
- Adquirir las habilidades y el conocimiento requeridos para usar y mantener controladores de Aspen DMCplus™ y DMC3.
- Aprender cómo usar DMCplus Model para identificar un modelo de controlador antes de implementarlo en línea.
- Incrementar la efectividad en la resolución de problemas de los controladores de Aspen DMCplus y DMC3.
- Incrementar el conocimiento sobre las características y capacidades de operación de los controladores de Aspen DMCplus y DMC3.

1.4.2. Procedimiento

- Conocer las características de los modelos y controladores del DMC3.
- Conocer cómo se desarrollan los modelos del DMC3 a través de las pruebas de respuesta a escalón.
- Aprender cómo concebir, diseñar y desplegar una aplicación de control de procesos avanzado.
- Conocer las características de modelos lineales, dinámicos y empíricos.

- Aprender cómo usar el DMC3 para identificar un modelo de proceso.
- Ser capaz de decidir cuándo usar FIR (*Finite Impulse Response*) o modelización por subespacios.
- Entender la importancia del análisis de colinealidad.
- Aprender a usar el DMCplus para desarrollar modelos de control, construir aplicaciones de control y realizar sintonización offline y simulación de aplicaciones de control.
- Conocer el rol de los rangos de las variables controladas y los factores de peso (factor γ (gamma) en la función de minimización $\min \sum(\gamma \cdot \Delta u)$) en el cálculo de los objetivos de consigna (set points en el futuro dados por el optimizador en línea).
- Aprender a usar los costes de las variables manipuladas y calculadas que hagan que la optimización en estado estacionario busque el punto de operación de mayor beneficio.
- Observar por qué se calculan múltiples movimientos para cada ciclo de control.
- Aprender a usar la penalización de los esfuerzos de control en las variables manipuladas y los factores de peso en las variables controladas para encontrar un equilibrio entre la minimización del movimiento frente a la minimización del error (Tradicionalmente, la sintonización de los controladores DMC multivariable se hace mediante ensayo y error. Se eligen valores iniciales arbitrarios y se comprueba el rendimiento en un controlador simulado, después, o se incrementan los valores para contener más el movimiento (penalización de esfuerzos de control), o se reducen para que las variables manipuladas se muevan más rápido (mayor precisión, menor error)).
- Aprender a implementar transformaciones de variables y cálculos del controlador personalizado.
- Familiarizarse con la infraestructura del Controlador Avanzado.
- Interactuar con el controlador en línea para operar una planta.
- Creación de una plantilla donde se reflejan todos los avances en el proyecto, donde se puede comparar el uso de un controlador predictivo como el DMC con el uso de un controlador convencional. Asimismo, se realiza una comparativa del software DMC3 con su antecesor, el DMCPlus.
- Realización de un experimento completo en una planta de laboratorio. Incluye un pretest, una toma de datos de experimentación y otra de datos de validación.
- Cálculo de un modelo del experimento.
- Identificación con el software de AspenTech®.
- Diseño del controlador predictivo.
- Programación de una optimización en línea.

1.4.3. Descripción de la planta

La planta sobre la que se realiza el experimento es una planta real del laboratorio del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Valladolid. La característica principal es que, además de su simplicidad, es una planta multivariable. El proyecto se desarrolla en ella, sobre todo por simplificar los manuales de uso del software, pero el diseño está preparado para asumir procesos con numerosas variables.

En este caso, se trata de una planta con dos depósitos conectados entre sí, con una bomba que impulsa el fluido a la parte alta del primer tanque, tal y como puede observarse en la Ilustración 2. En la tubería que conecta la bomba con el primer depósito, se encuentra un caudalímetro, y un transmisor de nivel mide el nivel del segundo depósito. La velocidad de la bomba puede manipularse por medio de un variador de frecuencia que admite una señal de control.

Esquema de la planta:

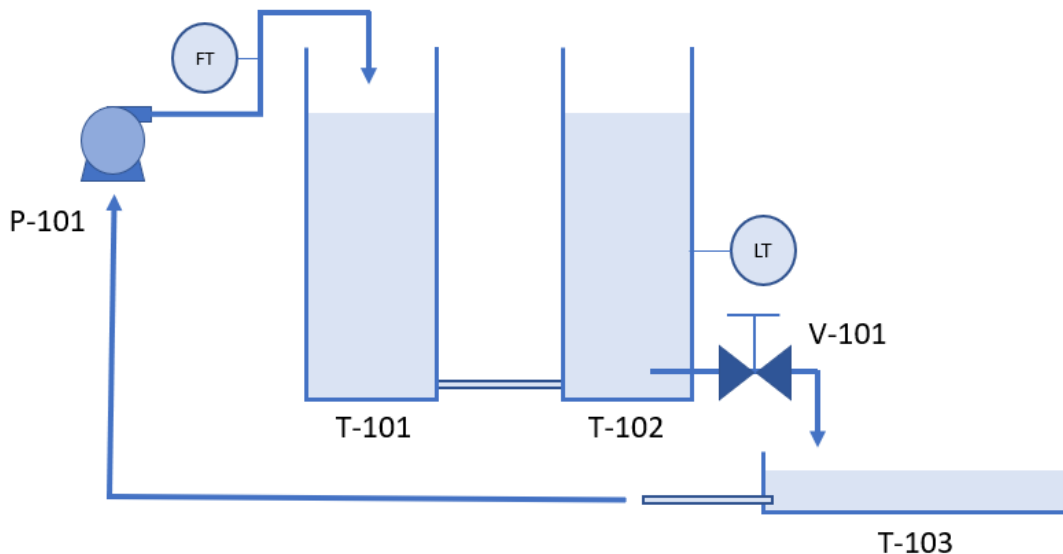


Ilustración 2 Esquema de la planta

1.4.4. Software

Toma de datos

Software: ReadOPC.

Mediante un servidor OPC, a través de la tarjeta de adquisición de datos, en este caso la USB-1408FS-Plus (Servidor OPC para USB-1408FS-Plus), se selecciona el periodo de muestreo y el tiempo durante el cual se desea estar capturando datos (se puede poner un máximo, pero se puede parar la toma de datos en cualquier momento si ya se tiene la cantidad que se necesita, así que conviene dar un valor alto). Una vez terminado este paso, se consigue un archivo de datos con un formato .clc (CLC: *Collect File*) que puede interpretar el DMC3, en este caso.

Identificación, diseño del controlador y optimización

Software: DMC3.

Para iniciar los vectores de datos en el DMC3, se utiliza el archivo CLC del ReadOPC.

Se realiza entonces un nuevo caso, donde:

- En los vectores correspondientes, se añaden las variables independientes por un lado y las dependientes por otro.
- Se determina el rango del muestreo de datos que se quiere utilizar.

- Como parámetros para el diseño del controlador predictivo, se especifica el tiempo que tarda en entrar en estado estacionario la planta.
- Se establece el número de coeficientes (debe ser mayor a $3 \cdot \text{tiempo estacionario}$) que debe ser un múltiplo de entre los siguientes: 30, 45, 60, 75, 90, 105 o 120.
- Se realiza la predicción.

1.5. Estructura de la memoria

Este proyecto se ha estructurado de la siguiente manera:

- En el primer capítulo, se expone una breve introducción de la motivación tras la realización de este proyecto. Se presenta, además, una serie de objetivos a conseguir una vez terminado el trabajo, tanto para el autor, como para los lectores. Por otra parte, se pone en antecedentes al lector en cuanto al control predictivo, con una pequeña formación sobre qué es lo importante a tener en cuenta a la hora de diseñar un controlador de estas características. En este mismo apartado, se introduce el software protagonista que se utilizará para realizar el diseño y se incluye una descripción de la planta es la que se desarrolla el proyecto.
- El segundo capítulo se centra en la toma de datos de funcionamiento de la planta. Se presenta el software que se utilizará a lo largo del proyecto con este fin, además, se reseñará lo más importante a la hora de realizar esta parte inicial del diseño del controlador, donde tanto influye un experimento adecuado, el conocimiento del proceso, y el control de la señal.
- En el tercer capítulo se describe la parte del modelado de la planta. Aunque el objetivo del proyecto es formar al autor y al lector sobre el diseño frente a cualquier proceso. En este apartado, se presenta el software mediante el cual se obtiene el modelo del proceso, así como una serie de consejos a la hora de realizar esta parte del proyecto y por qué son tan importantes los pasos previos a la construcción del controlador.
- En el cuarto capítulo se pretende abordar la parte de la construcción del controlador, para ello, se introduce el control predictivo por matriz dinámica, sus principales ventajas, aplicabilidad e interés industrial. Además, se realiza el estudio del sistema de control y se expone la propuesta del autor de este proyecto para el diseño final del controlador, el cual se rematará en el capítulo siguiente.
- En el quinto capítulo se introduce la parte final de este proyecto sobre el diseño del controlador. Además, se describe de forma general el software que se utiliza durante todo el proceso de diseño, así como una redacción de un manual de uso para que el lector pueda utilizar el programa. Por otro lado, se realiza en paralelo una comparación con versiones anteriores de la misma suite.

- En el sexto y último capítulo se desarrollarán las conclusiones que se han sacado de la realización de este proyecto, así como posibles mejoras que pueden realizarse en el proyecto.

Capítulo 2. TOMA DE DATOS

2.1. ¿Qué se ofrece en este capítulo?

Este capítulo se centra en la toma de datos de funcionamiento de la planta. Se presenta el software que se utilizará a lo largo del proyecto con este fin, además, se reseñará lo más importante a la hora de realizar esta parte inicial del diseño del controlador, donde tanto influye un experimento adecuado, el conocimiento del proceso, y el control de la señal.

2.2. ReadOPC

2.2.1. Introducción

2.2.1.1. Descripción general

Es el software para la toma de datos del experimento necesarios para la identificación de un modelo, la principal desventaja de este software es que, a pesar de ofrecer la posibilidad de monitorizar el avance de las variables en gráficos, no está disponible esta opción a no ser que esté la ventana activa, en el momento en que se queda en segundo plano, no muestra el avance. Es por ello por lo que, si se quieren consultar las gráficas mientras dura el experimento, se aconseja el uso en paralelo del JavaRegula (JavaRegula, 2018) (software desarrollado en el Departamento de Ingeniería de Sistemas y Automática de la Universidad de Valladolid), que permite el control de la planta a la vez que se puede mostrar gráficamente la evolución de las variables, incluso en segundo plano.

2.2.2. Manejo

2.2.2.1. Funcionalidades

Abrir ReadOPC:

En la pestaña *Setup: OPC Server*, de la lista se selecciona el servidor donde opera la tarjeta de adquisición de datos, en este caso la USB-1408FS-Plus (Servidor OPC para USB-1408FS-Plus) → en *Parameters*, seleccionar el periodo de muestreo y el tiempo durante el cual se quiere estar capturando datos (se puede poner un máximo, pero se puede cortar en cualquier momento si ya se tienen los datos necesarios, así que conviene dar un valor alto) → en *Temporary File Name* seleccionar una carpeta donde el software guardará el archivo temporal que después se deberá pasar a un formato que pueda leer DMCPlus.

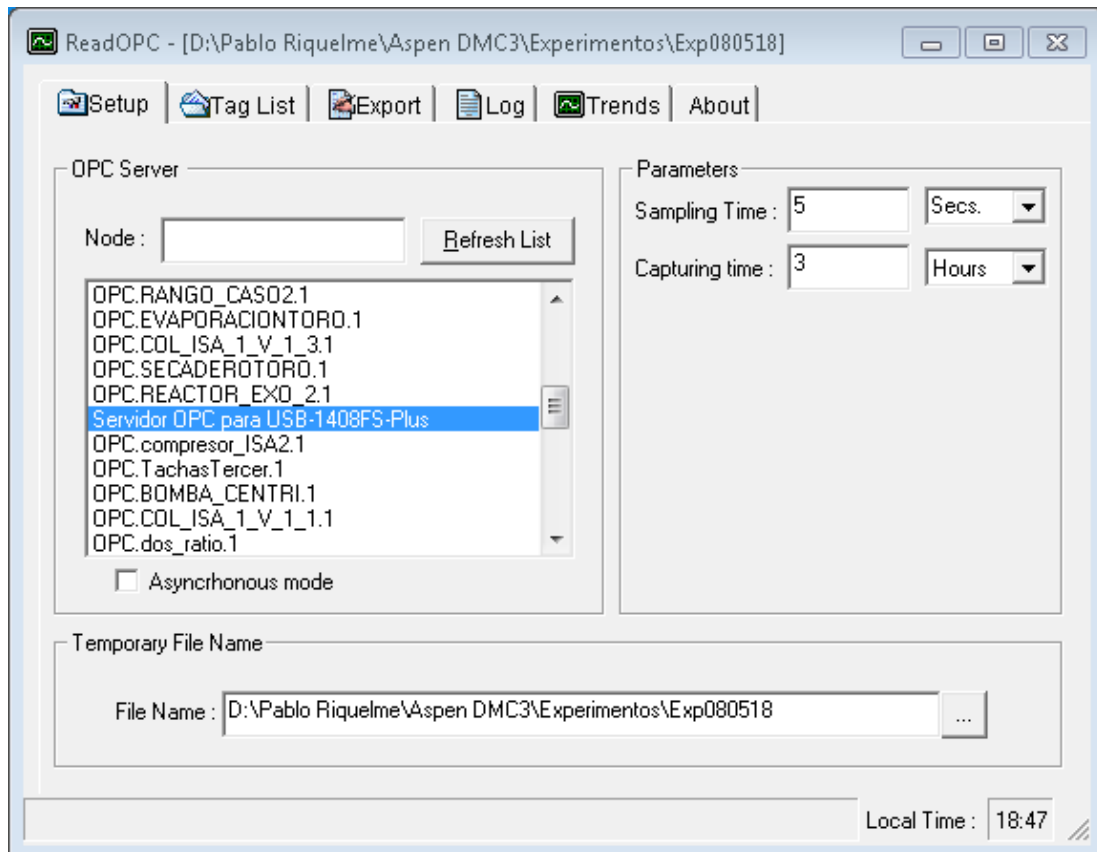


Figura 1 Configuración inicial de ReadOPC

En la pestaña *Tag List*: se puede proceder de dos maneras (es imprescindible que el nombre de la entrada o salida que se añade coincida completamente con el que se toma de Softing (véase apartado 2.3 Softing OPC Toolbox Demo Client v4.1, en la página 14)):

- Una vez que se lea desde Softing el nombre de cada una de las entradas y salidas, se puede crear una lista en un archivo de texto que se debe poner con extensión .lst, con cada una de las entradas y salidas separadas por un intro.
- Se pueden añadir una a una cada una de las etiquetas, poniendo el nombre íntegro e idéntico al del software de adquisición de datos. Una vez hecho esto, se puede guardar esa lista para poder utilizarla en sucesivas ocasiones.

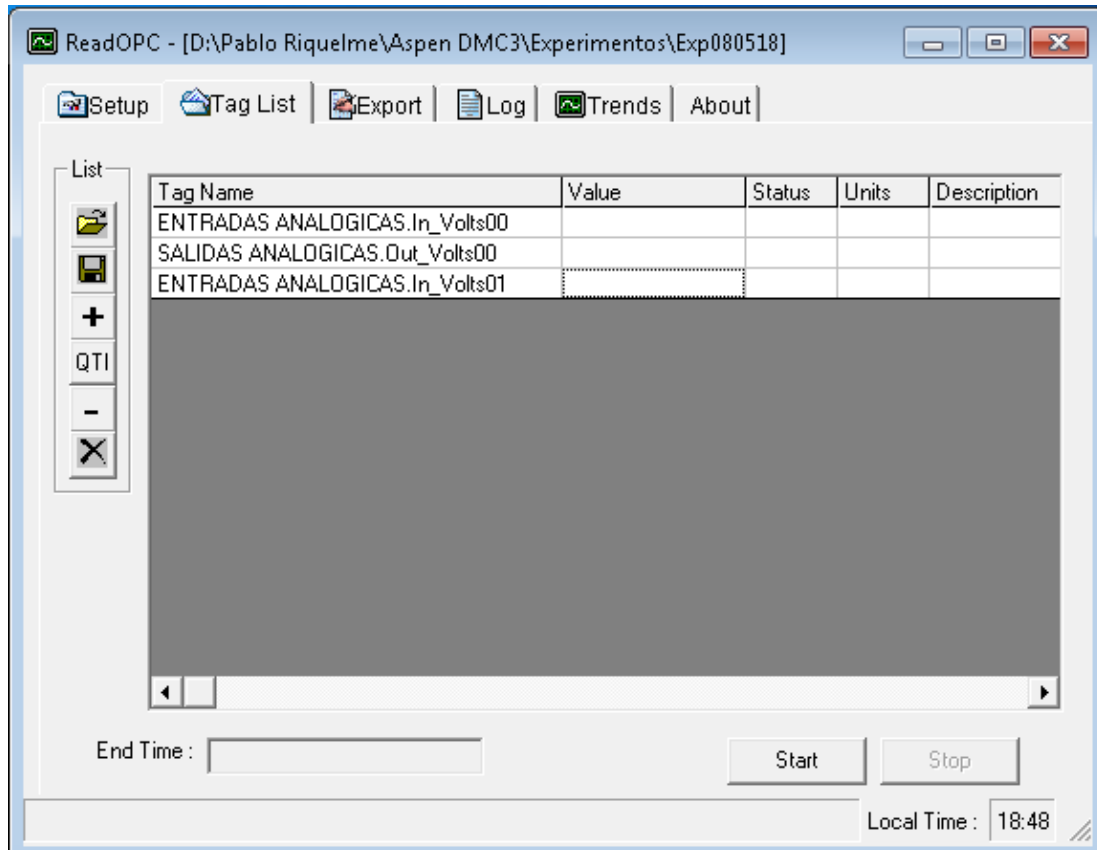


Figura 2 Etiquetas en ReadOPC

Se recomienda configurar como símbolo decimal el punto, en vez de la coma, en el Panel de control de Windows, sección Región, apartado Formato y Configuración adicional. De esta manera, el archivo final no será necesario ser modificado por el usuario.

Se hace clic en *Start* abajo a la derecha y se empieza a tomar datos. Los cambios de la señal de control se irán haciendo desde otro cliente OPC como el ofrecido gratuitamente por la empresa Softing. Cuando se haya terminado el experimento, hay que hacer clic en *Stop*.

En la pestaña *Export*: se seleccionan los archivos de lectura de los datos y la carpeta donde se quiera guardar el archivo .clc que después será utilizado en el DMCPlus. Se selecciona (CLC) *Collect File* → clic en *Extract*, entonces preguntará si se desea seleccionar una fecha inicial diferente, para poder unir dos experimentos con fechas distintas como si fuera un único (esta parte está bien explicada en el manual de uso de ReadOPC).

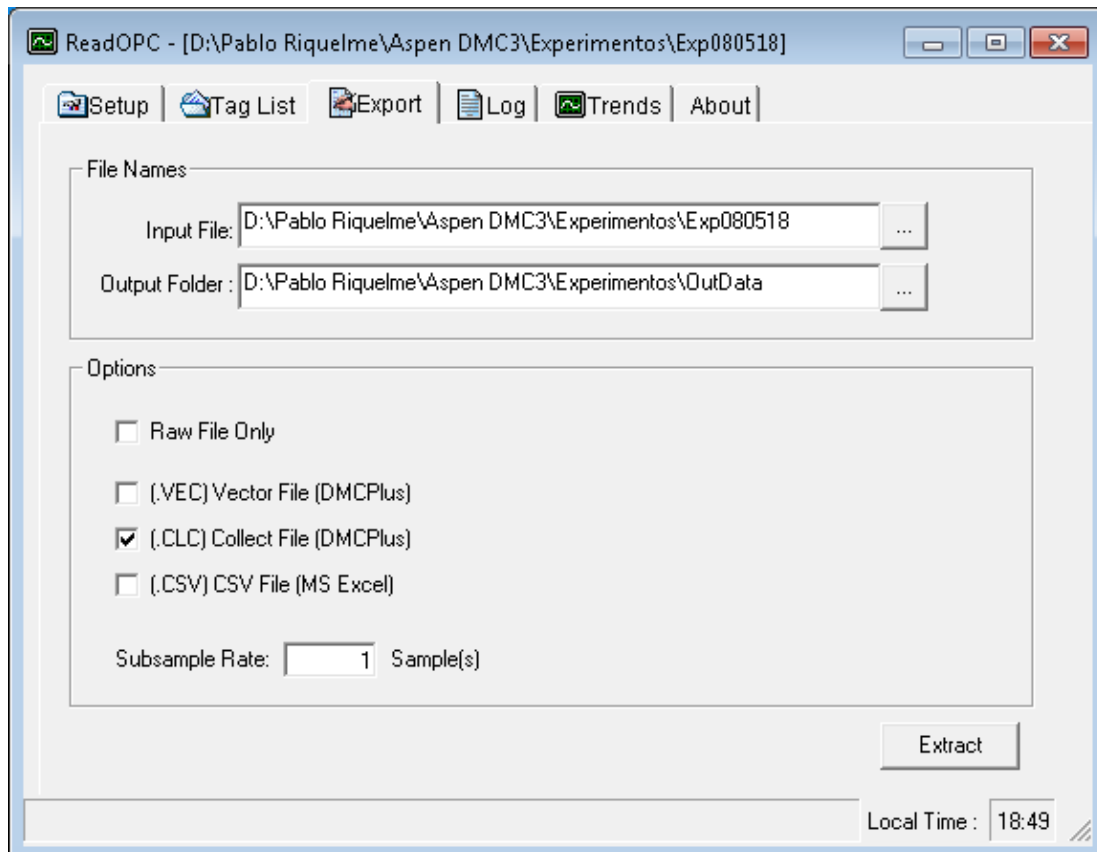


Figura 3 Exportar datos desde ReadOPC

2.2.2.2. Sinergias

Una vez se tenga el archivo CLC, se debe abreviar el nombre de las entradas y salidas antes de cargarlo en el DMCPlus, ya que éste tiene problemas de compatibilidad con nombres largos. Para ello, se diferencia cada una de las variables, definiendo las unidades y una pequeña descripción.

En este punto, si se ha definido en Windows el punto como símbolo decimal, no hará falta ninguna acción adicional. Si no fuera el caso, ha de modificarse el archivo de tal manera que las columnas correspondientes a cada variable estén separadas por comas, pero el decimal de cada valor, lo esté por puntos.

2.3. Softing OPC Toolbox Demo Client v4.1

2.3.1. Introducción

2.3.1.1. Descripción general

Es un software para la conexión vía OPC a las variables de la planta, donde se pueden monitorizar los valores de todas las variables, tanto las manipuladas como las calculadas.

Se utiliza el Softing OPC (Softing AG, 2018), de la misma forma que podría utilizarse JavaRegula, donde se tendrá control sobre la planta en la que se desee hacer el experimento. Pero a diferencia de este último, no se recogen los valores de los datos.

2.3.2. Manejo

2.3.2.1. Funcionalidades

Abrir Softing OPC:

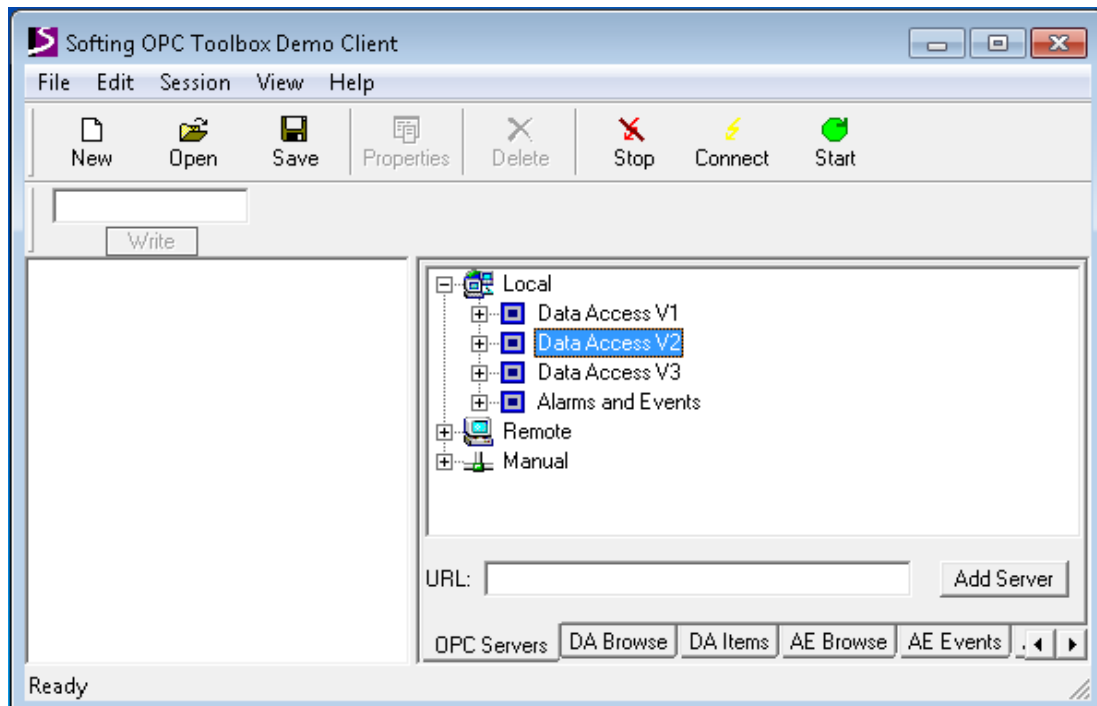


Figura 4 Pantalla Inicial de Softing OPC

En la pestaña inferior OPC Servers: *Local* → *Data Access V2* → Doble clic en Servidor OPC para USB-1408FS-Plus (DAQ device, es el nombre de la tarjeta de adquisición de datos de la planta).

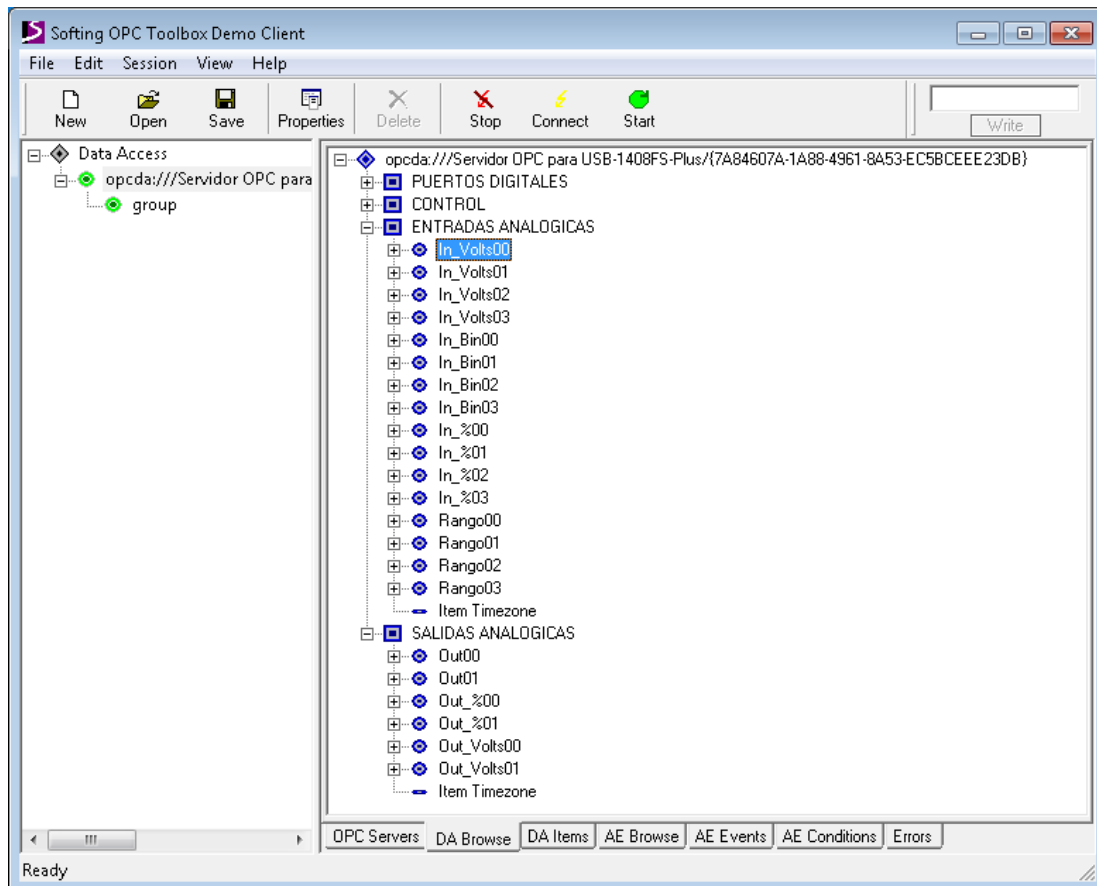


Figura 5 Selección de entradas y salidas

En la pestaña inferior DA *Browse*: En el apartado ENTRADAS ANALOGICAS, seleccionar In_Volts00 (hace referencia a la señal del transmisor de nivel LT situado en el depósito T-102, véase Ilustración 2 Esquema de la planta, pág. 7) y In_Volts01 (señal del transmisor de caudal FT situado en la conexión de la bomba P-101 al depósito T-101). En el apartado SALIDAS ANALOGICAS, seleccionar Out_Volts00 (señal de voltaje 0-5V transmitida a la bomba P-101, situada en la conexión del tanque T-103 al T-101). En la documentación de la planta, en este proyecto, se indican los canales de cada uno de los sensores y si pertenecen a Entradas o Salidas: en este caso Señal (Sensor de nivel) y Canal (CH0-IN, donde el cero indica el canal a usar en el Softing, IN a su vez indica que se está ante una señal de entrada).

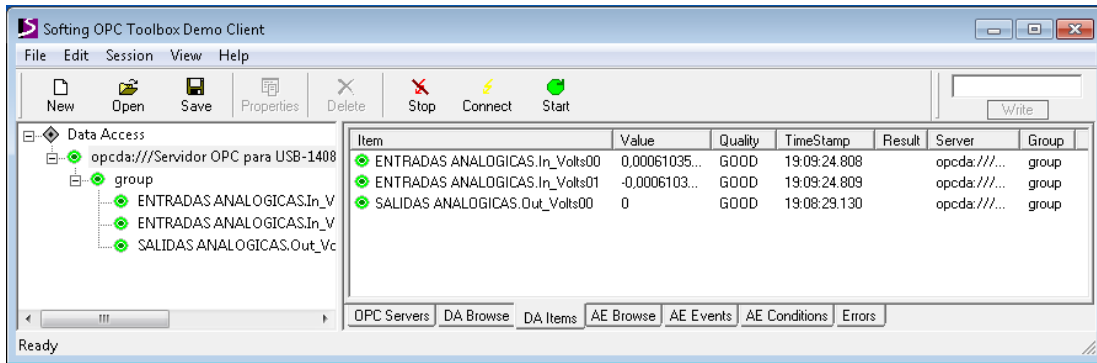


Figura 6 Acceso a las variables de la planta

2.3.2.2. Sinergias

Con esto se tiene acceso a las variables de la planta en cuestión, pero no se toman los datos, para ello se usa el ReadOPC, tal y como se puede observar en el diagrama de la ilustración 3.

2.4. ¿Por qué dos softwares distintos?

Para la toma de datos de la planta se utilizan dos softwares distintos, uno de ellos para tomar datos y otro para modificar las señales de la planta:

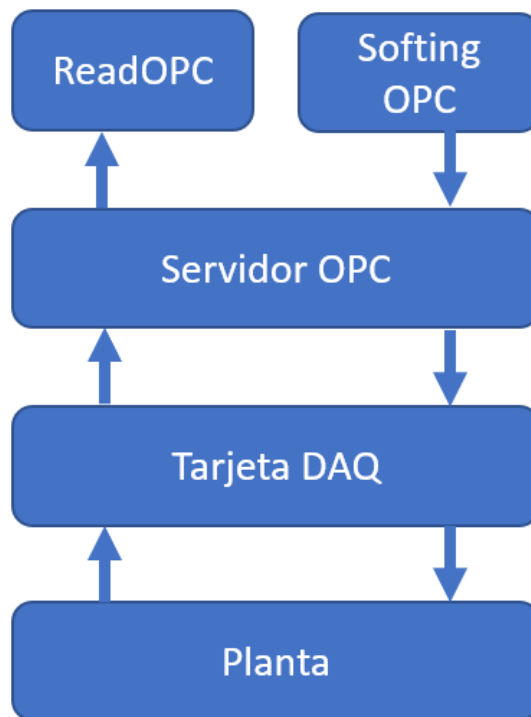


Ilustración 3 Diagrama Toma de datos y control de la planta

Capítulo 3. MODELADO

3.1. ¿Qué se ofrece en este capítulo?

En este capítulo se describe la parte del modelado de la planta. Aunque el objetivo del proyecto es formar al autor y al lector sobre el diseño frente a cualquier proceso.

En este apartado, se presenta el software mediante el cual se obtiene el modelo del proceso, así como una serie de consejos a la hora de realizar esta parte del proyecto y por qué son tan importantes los pasos previos a la construcción del controlador.

3.2. Aspen DMC3 Model

3.2.1. Introducción

Una vez se han obtenido los datos de la planta, el siguiente paso es modelar el sistema. El DMC3 Model permite una visualización gráfica de los datos de la planta, con los cuales se podrá realizar el modelo del proceso con el que se diseñará el controlador.

3.2.2. Descripción general

Software para obtener el modelo de un sistema dinámico.

3.2.3. Manejo

3.2.3.1. Funcionalidades

Para iniciar el software, se abre Inicio → Programas → AspenTech → Aspen Process Control → DMCplus Model (APC). En este momento, el programa permitirá elegir entre DMCplus y DMC3, para trabajar o bien con la versión antigua, o bien con la nueva.

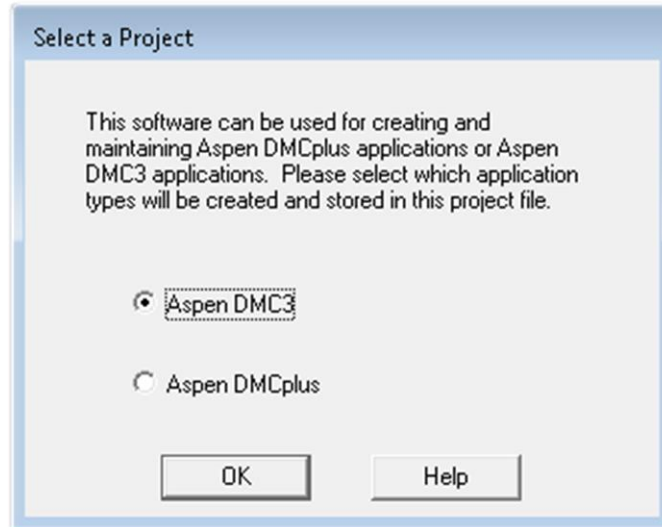


Figura 7 Menú de elección entre DMC3 y DMCPlus

La interfaz de este software es común a toda la suite aspenONE®, tanto en esta versión 10 como en la anterior, la versión 9. Por tanto, al trabajar con uno de los programas, la interfaz se vuelve mucho más amigable a la hora de trabajar con el resto. Además, es la común, en muchos aspectos, a la del software que se ejecuta en entorno de Windows, esto la hace intuitiva ya que dispone de las herramientas básicas de corte, copia y pegado, y demás menús desplegables.

Se puede hacer una división en las vistas que ofrece este software: por un lado, se tiene el árbol de proyecto, donde se dispone de todos los componentes en acceso directo. Por otro lado, se tiene la vista de gráficas, donde se visualizan todos los vectores y señales.

Para ofrecer una lista de los componentes principales, presentes en el árbol de proyecto, primero se identifica el contenedor principal, llamado Proyecto, donde se incluye el resto. Éste puede contener distintos modelos de diversos experimentos, pero en este caso se trabajará con un solo proceso a modelar.

Tal y como aparecen, en orden descendente, los componentes principales son:

- *Vectors*: es donde se almacenan los datos de las variables.
- *Vector Lists*: donde se agrupan todos los datos de vectores.
- *Cases*: donde se inicializan los métodos de identificación de los modelos.
- *Model*: es donde, una vez definidas las variables dependientes e independientes, se puede observar la respuesta escalón de las primeras frente a las segundas.
- *Predictions*: alberga las predicciones que se realizan, sobre la respuesta del proceso, cuando se den como señales de entrada los vectores y se use el modelo definido.

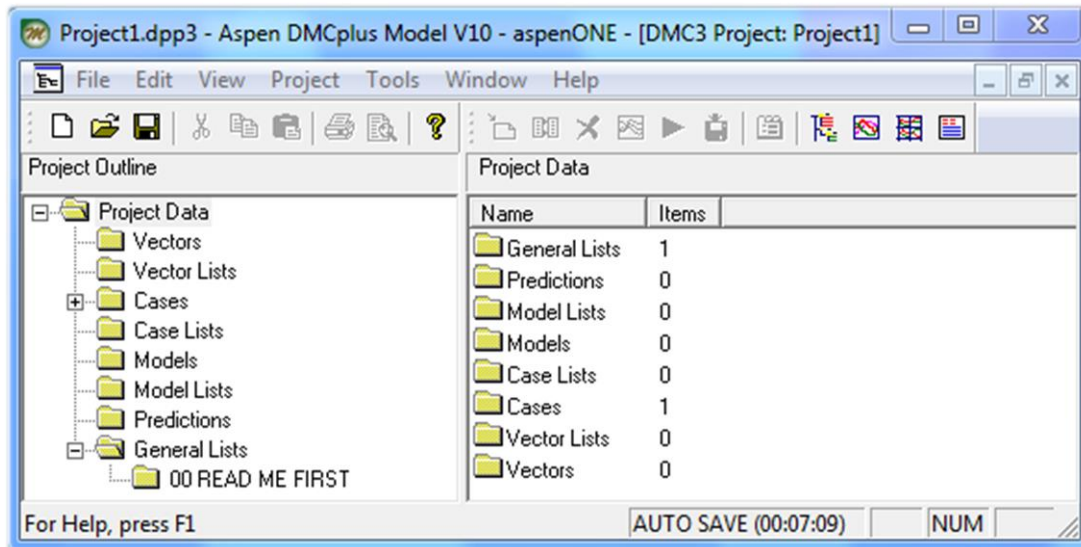


Figura 1.- Pantalla inicial. Árbol de proyecto y componentes principales

3.2.4. Modelado

Esta parte del proceso de diseño puede realizarse directamente en el Aspen DMC3 Builder si el usuario así lo desea, puesto que tiene incluidas prácticamente las mismas opciones que el software que aquí se describe, además de una interfaz más moderna. Se expone aun así un manual de uso de este programa con el objetivo de crear una serie de directrices hasta el diseño final del controlador, donde el procedimiento esté explicado paso a paso, y esto resulta más instructivo con esta explicación del módulo separado del Builder.

Para realizar el modelado, lo primero es crear un nuevo proyecto e importar los datos de los experimentos realizados en la planta, utilizando para ello el archivo tipo *collect* que genera el ReadOPC. Clic derecho en *Vectors* → *Import* → *Vectors* y seleccionar el archivo CLC donde se han guardado los datos con el ReadOPC.

Una vez obtenidos los vectores, se revisan para evitar errores, haciendo interpolaciones donde sea necesario, o dividiendo en intervalos. Tras el tratamiento de las señales, el siguiente paso es crear un caso, que generará los modelos para las variables seleccionadas siguiendo las indicaciones que proporcionan los distintos parámetros introducidos por el usuario.

Clic derecho en *Cases* → *New case*:

- En la pestaña *Vectors*, añadir las variables independientes por un lado y las dependientes por otro.
- En la pestaña *Slices*, determinar el rango del muestreo de datos que se quiere utilizar.
- En la pestaña *Parameters*, determinar el tiempo que tarda en entrar en estado estacionario la planta, el número de coeficientes (debe ser mayor a $3 \times$ tiempo estacionario) que debe ser un múltiplo de entre los siguientes: 30, 45, 60, 75, 90, 105 o 120.

A partir de los modelos obtenidos en los casos anteriores, se pasa a generar el modelo que represente el proceso. Para ello se añaden curvas que relacionen pares de variables independientes con dependientes hasta conseguir el modelo deseado. También, en este caso, se pueden utilizar predicciones del modelo para comprobar que genera señales como las que se obtuvieron en el ensayo del proceso.

Clic derecho en *Models* → *Assemble a New model*:

- En la pestaña *General*, Introducir el nombre, el tiempo a estado estacionario y el número de coeficientes.
- En la pestaña *Tags*, introducir las variables.

Arrastrar el caso al gráfico vacío del modelo. Una vez hecho esto, seleccionar cada una de las variables (una por una, no todas juntas) y hacer clic en *Edit* → *Get Curve*.

Cuando ya se tienen las curvas del modelo más aproximado al proceso, ya se puede exportar el archivo *.mdl3*, para poder importarlo en el DMC3 Builder.

Clic derecho en *Predictions* → *New prediction*.

3.2.5. Proyectos

Para empezar, se debe que crear un proyecto nuevo. Para ello hacer *File* → *New Project* o simplemente se pulsa en el botón de *New Project*, apareciendo un proyecto con un nombre por defecto que se puede cambiar. También es posible abrir un proyecto existente (extensión del archivo *.dpp3*) e incluso importar partes de otro.

Para esto último, es de gran ayuda la función de importación de proyectos, la cual 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 que se puede observar en la figura 1, y la forma de importar los objetos se consigue seleccionando del proyecto fuente y trasladarlos al proyecto nuevo.

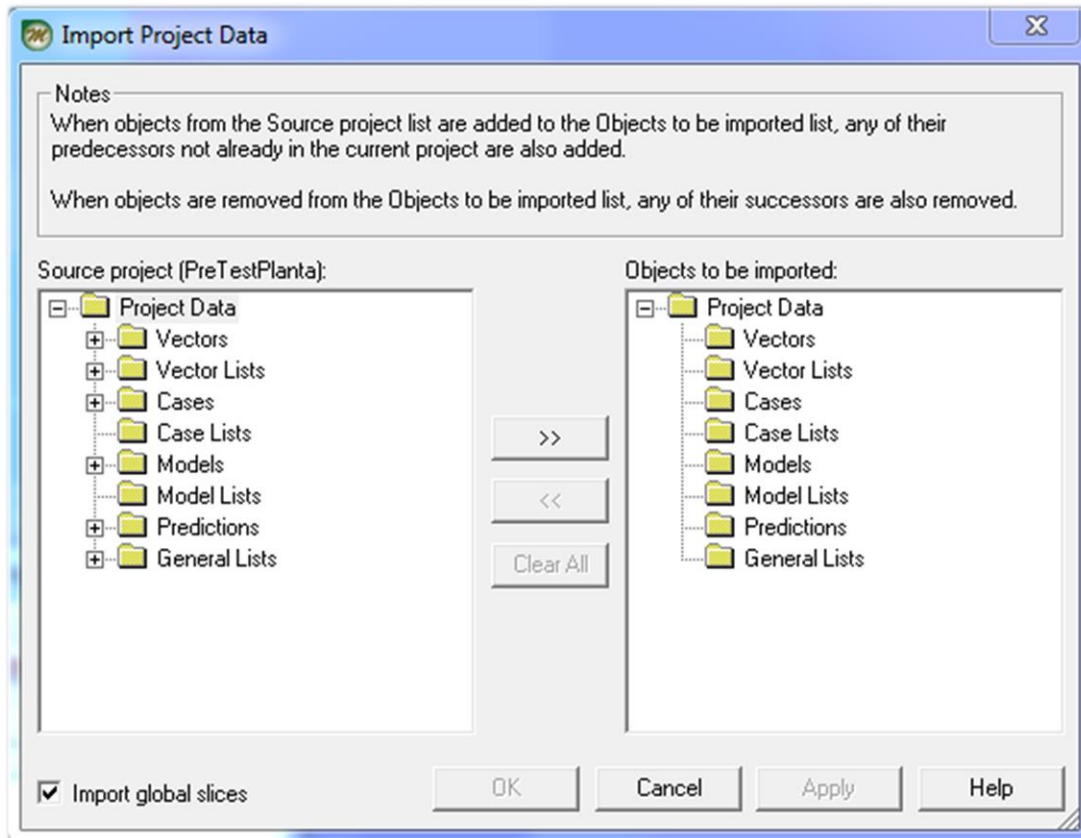


Figura 2.- Función de importación de proyectos

Asimismo, es posible la exportación de objetos del proyecto actual simplemente con situarse encima de uno o varios objetos y desplegando el menú contextual, del cual se elige la opción de exportar. Por otro lado, la manera de guardar un proyecto es mediante el comando de Windows o bien programando para ello un autoguardado cada cierto tiempo (activada por defecto en el software), 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 instante de comienzo, por lo que se establece un porcentaje de tolerancia.

3.2.6. Vectores

Como ya se ha indicado, los vectores son las unidades que contienen los datos con los que se va a trabajar. Por ello es lo primero que se necesita para comenzar con el trabajo. Dependiendo del tipo de datos que contengan se pueden distinguir distintos tipos de vectores como, por ejemplo:

- Datos (*Raw Data*): estos vectores son aquellos que contienen datos tomados directamente del proceso sin modificaciones, salvo la posible interpolación de datos realizada cuando se importan 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 DMC3. 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.

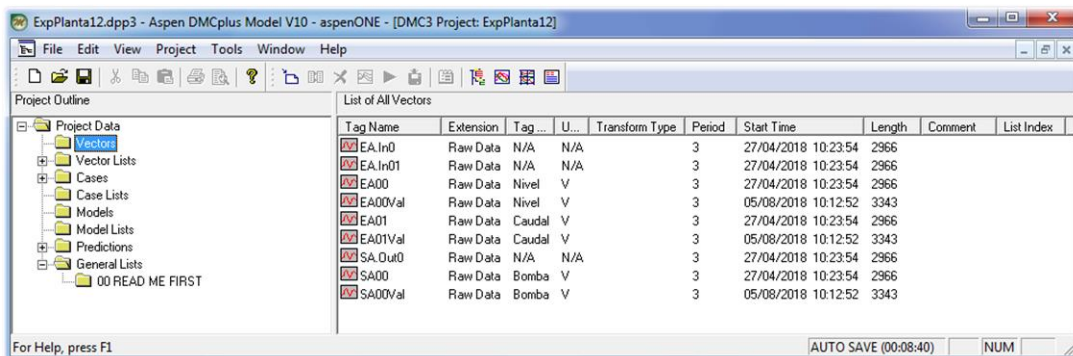


Figura 3.- Pantalla de vectores

3.2.6.1. Importar vectores

La forma de importar vectores es acudir al menú *File* → *Import* → *Vectors*. Así se mostrará el diálogo de importación de vectores en el cual se especificará el tipo de vectores que se quiere importar, así como su ubicación. Si, como es natural al principio de un proyecto totalmente nuevo, se utilizan los datos tomados de la planta en un archivo .clc, se busca éste y se abre.

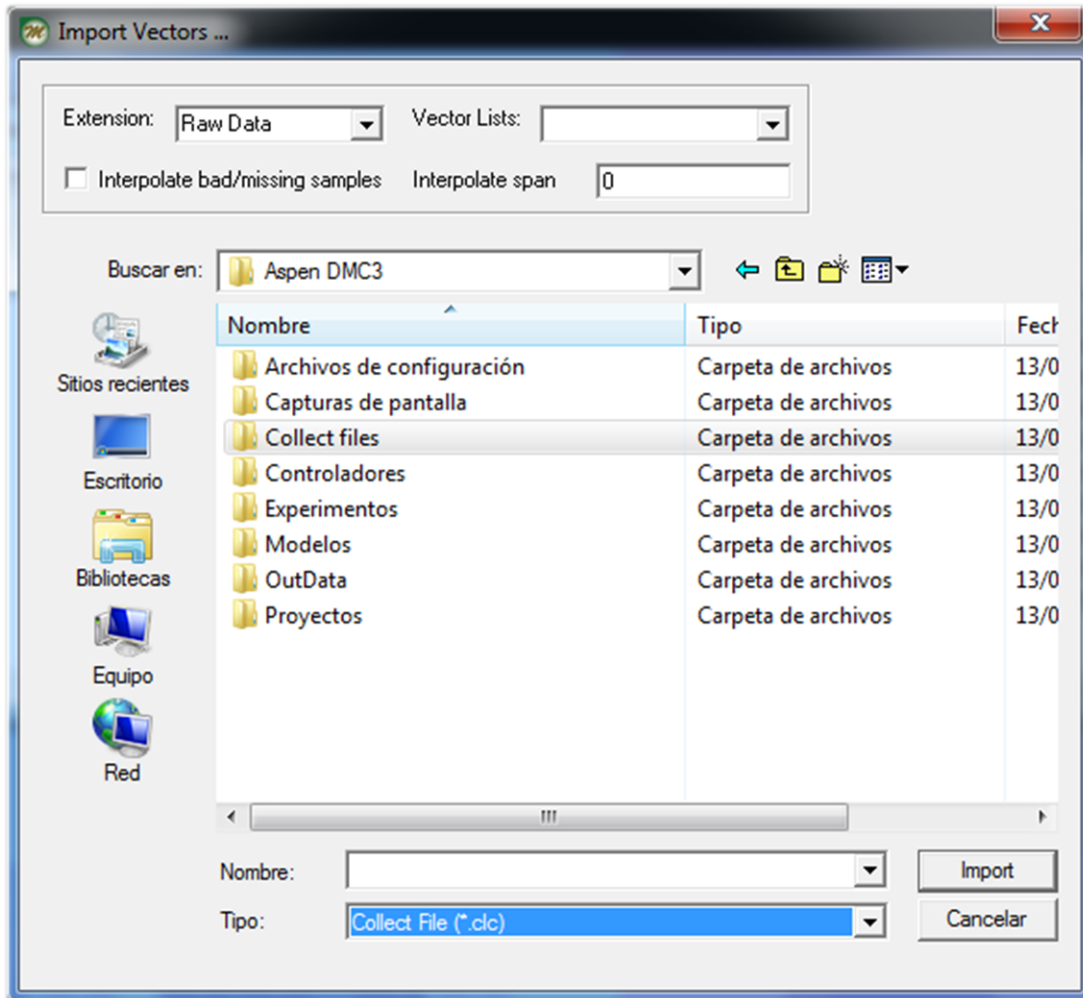


Figura 4.- Diálogo de importación de vectores

Tal y como se puede observar en la figura 4, se ponen a disposición del usuario dos opciones para la importación de los vectores:

- Interpolación de datos erróneos o perdidos (*Interpolate bad/missing samples*): se utiliza para eliminar de los vectores los datos que se han clasificado como tales en la extracción de éstos. Para no dejar el "hueco" vacío se interpola con los datos contiguos.
- Extensión de la interpolación (*Interpolate span*): se trata de los minutos de la muestra que se utilizarán para la interpolación. Por defecto son cinco, pero hay que tener en cuenta que a mayor extensión peor será la identificación del modelo.

Una vez seleccionado el archivo .clc se podrán visualizar los datos que estén disponibles en dicho archivo, como se ve en la figura 5. Esto permitirá seleccionar los que se deseen, así como las muestras que se quiera.

Figura 5.- Información de los datos disponibles

3.2.6.2. Trabajar con los vectores

Ahora ya se tienen los vectores con los que se va a trabajar. Para visualizar los datos que contienen (únicamente de forma gráfica), simplemente hay que situarse sobre el vector o vectores que se quieran ver, estando a disposición del usuario 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 abrirá una ventana con los gráficos de los vectores que se hayan elegido. Esta nueva ventana puede configurarse para mostrar los datos como más se ajuste. 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. Se puede ver el vector completo o bien por partes.
- Escala. Es posible visualizar todos los vectores en un mismo gráfico si se establecen los valores mínimos y máximos de cada vector.

- Formato. Entendiendo por formato la escala temporal del vector, esto es, se puede 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.

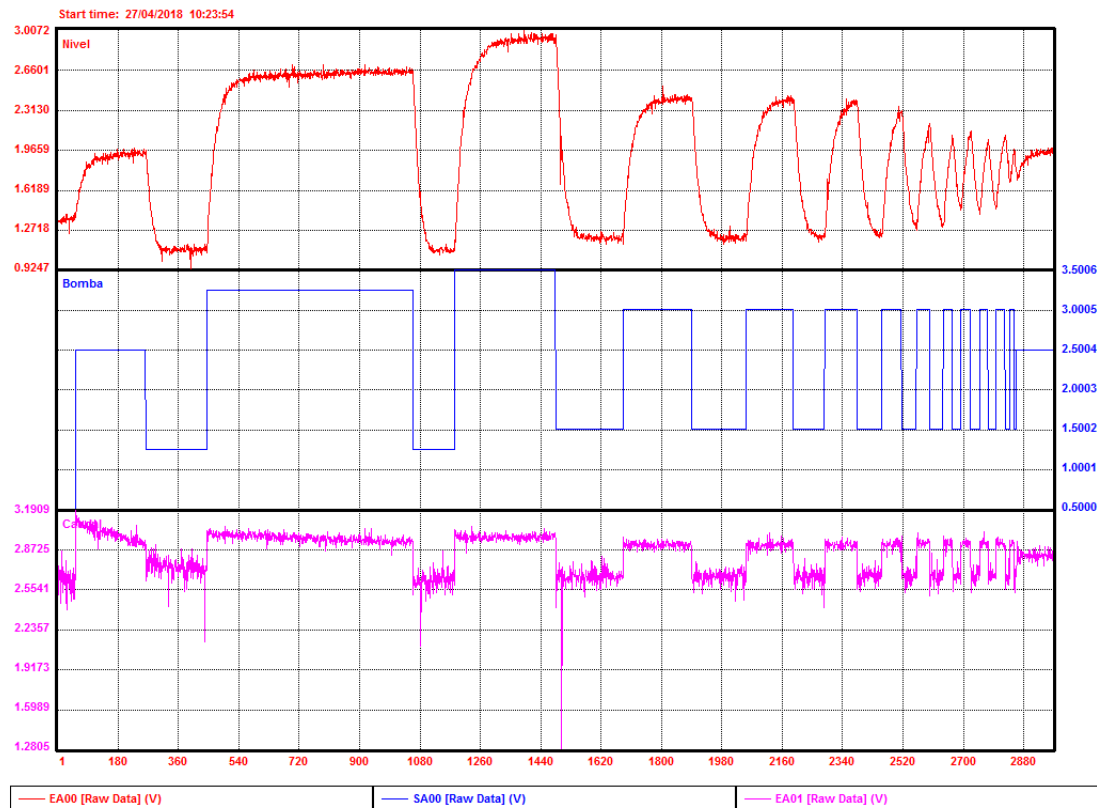


Figura 6.- Gráficos de vectores de datos

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 se desee, 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 se pueden marcar manualmente si se detecta que en el vector que se está visualizando existen datos que se pudieran clasificar como erróneos (movimientos extraños en las variables, datos no esperados u otros). Los datos que se marquen 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 les puede aplicar una serie de funciones que permitirán transformar los datos que contengan según las necesidades propias. Para crear un vector de cálculo, hay que posicionarse en la zona de vectores y seleccionar el menú *Project* → *New Vector*, apareciendo entonces el panel de propiedades de los vectores calculados.

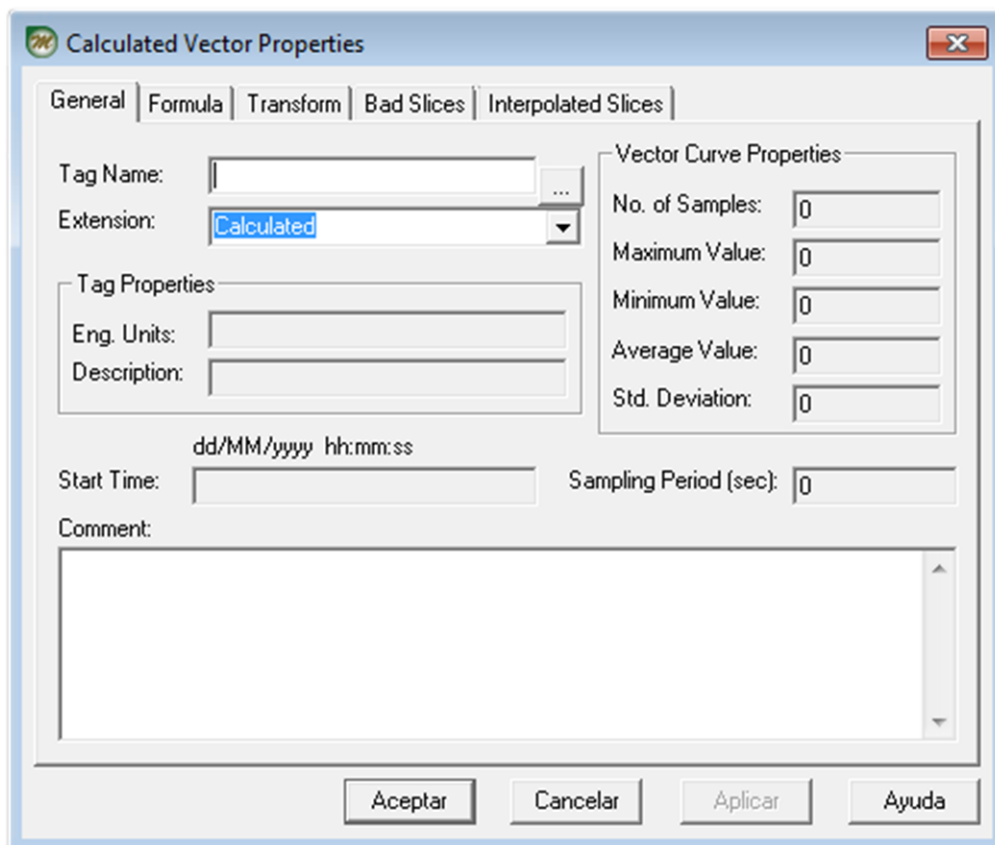


Figura 7.- Panel de propiedades de los vectores de cálculo

En este panel se ha de elegir un nombre para el nuevo vector y, lo más importante, la fórmula que se va a aplicar. Model pone a disposición del usuario una serie de fórmulas predeterminadas que se puede utilizar tal y como está, 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):

- @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 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 se esté 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.

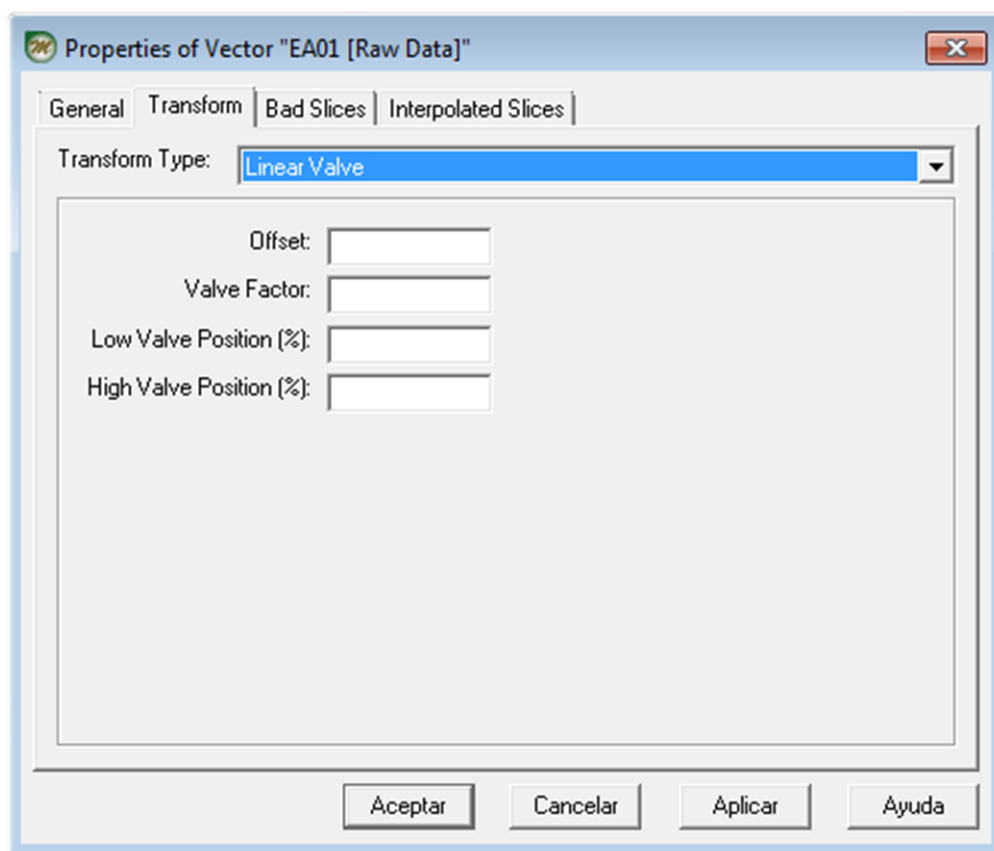


Figura 8.- Transformación de un vector

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:

Se trata en este caso de las propiedades que definen las características del vector con el que se trabaja. 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. Se accede a estas propiedades mediante el menú *Edit* → *Tag Properties*.

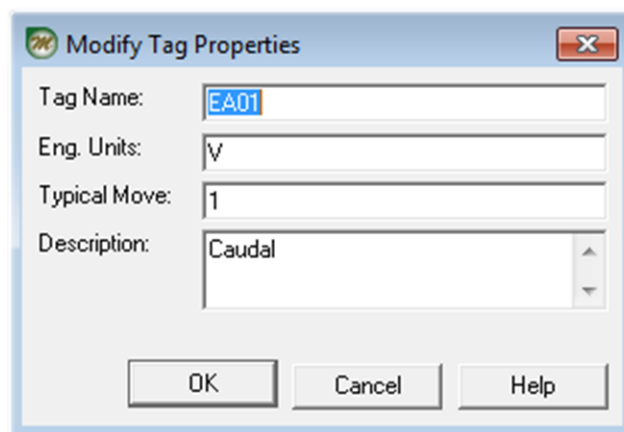


Figura 9.- Propiedades de tag

Lo recomendable es diferenciar y nombrar las variables directamente en la cabecera del archivo CLC que se importa en el DMC3, tal y como se ha explicado en el apartado correspondiente.

3.2.7. 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, por ejemplo, el tiempo de establecimiento o *Steady State* y el número de coeficientes del modelo.
- Frecuencia del muestreo en segundos
- Bandas de datos de los vectores a usar en la identificación.

Se crea un caso una vez que se hayan analizado los vectores, marcados los *Bad Slice* y realizados los cálculos y transformaciones que fuesen necesarios. El resultado que 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 se obtengan de los casos serán las que se utilicen para crear los modelos que se utilizarán finalmente, por lo que este paso es bastante importante en el modelado del proceso.

3.2.7.1. Creación y ejecución de un caso

Para crear un caso hay que situarse en la carpeta *Cases* y abrir el menú *Project* → *New Case* o bien situarse en el panel de *Cases* y utilizar el menú contextual. Entonces, se abrirá un diálogo en el que se especificarán las características de dicho caso.

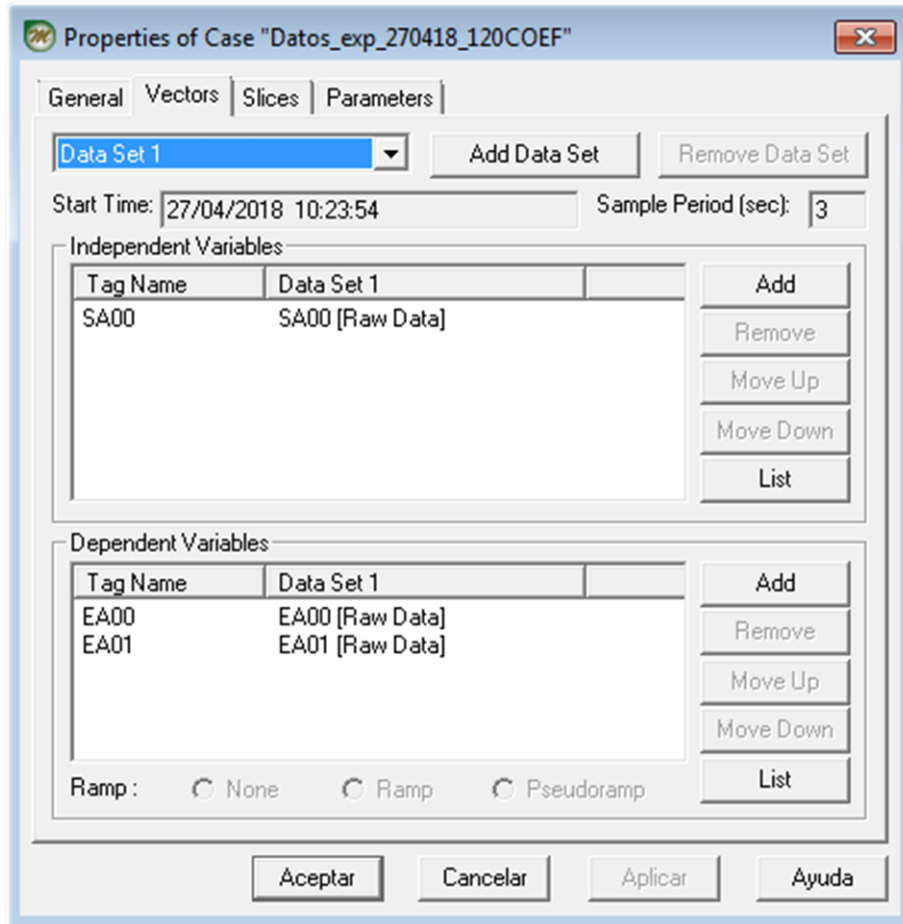


Figura 10.- Diálogo de propiedades de un caso

En este diálogo se encuentran 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 se tienen es suficiente, por lo que se puede dejar el *Data Set 1* que se tiene por defecto.
- *Independent Variables:* se han de añadir los nombres de las variables que serán las variables manipuladas o de perturbación (llamadas más adelante *Feedforward*) del modelo.

- *Dependent Variables*: además de añadir este tipo de variables para el modelo se tiene 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, se puede dejar el nombre por defecto.

- *Good Slices*: son los tramos de datos correctos que se utilizarán para la identificación. Hay dos formas de especificar estos tramos: se introduce manualmente la muestra inicial y final al presionar el botón *Add*, o bien se utiliza *Rebuild*, que suministra todos los tramos disponibles y que no se hubieran marcado como *Bad Slice*. Además, se tiene la posibilidad de permitir a Model que siempre recalculé automáticamente estos *Good Slices*, así si se añade algún *Bad Slice*, éste no será tenido en cuenta.

Parameters:

- *Identification Parameters*: se han de añadir todas las identificaciones que se deseen realizar, presionando para ello el botón *Add*. Entonces habrá 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, permitiendo así observar si el modelo se ajusta a los datos tomados de la planta.

Una vez que se tenga creado el caso se ejecutará seleccionándolo y pulsando el botón de *Run*. Al ejecutar el caso aparecerá una ventana que irá indicando si la identificación se está llevando a cabo de forma favorable. Si todo transcurre bien, se obtendrá una serie de modelos (dos por cada identificación que se especificara) en los cuales se podrá ver la curva de respuesta para cada par de variables que se hubiese especificado en el caso.

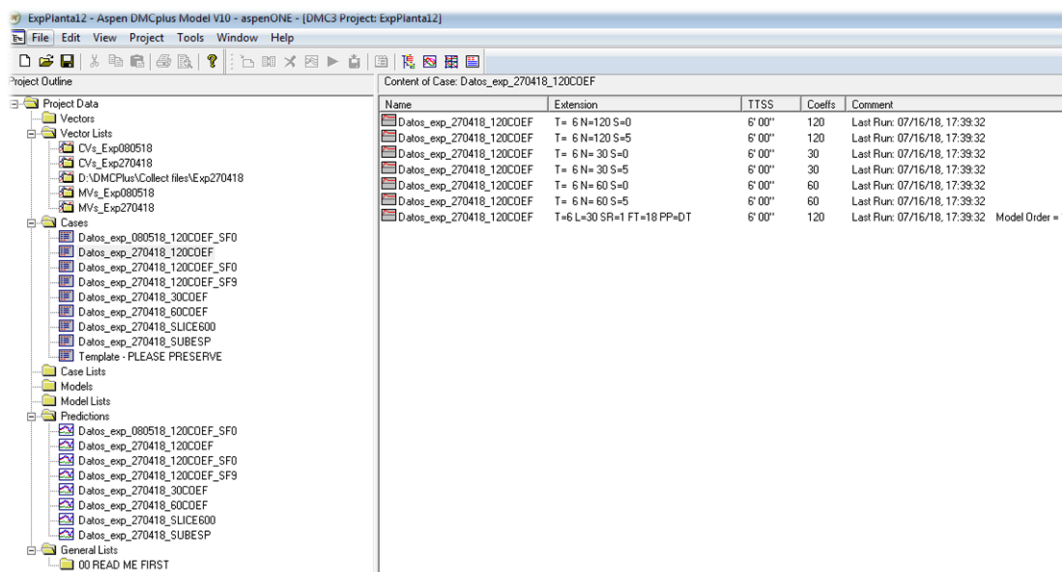


Figura 11.- Pantalla de casos ejecutados

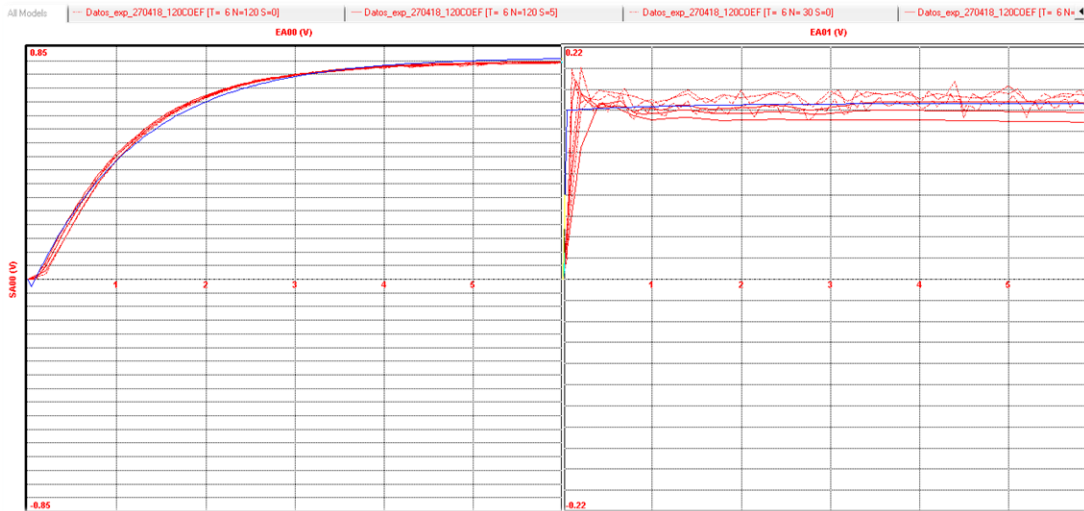


Figura 12.- Modelos obtenidos con varias identificaciones en un caso

Por último, es importante comentar la posibilidad de importar casos de otros proyectos existentes, así como también es posible la exportación de los que se crean por el usuario.

3.2.8. Algoritmos de identificación

FINITE IMPULSE RESPONSE (FIR)

Se trata del algoritmo de identificación típico del Control Predictivo basado en modelo (ver el capítulo 1). 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 se elige este algoritmo de identificación, Model preguntará por sus propiedades, siendo éstas:

- *Time to Steady State*: se trata del tiempo en minutos que se quiere que tarde el modelo en alcanzar un estado de reposo.
- *Number of Coefficients*: se debe especificar el número de coeficientes que se desea tenga el modelo. A mayor número de coeficientes, mayor tiempo de cálculo y de *Steady State*, aunque se gana en exactitud.
- *Smooth factor*: éste es un factor que permite que el modelo suavice sus respuestas. Con un número alto se suavizará bastante el modelo, pero se perderá exactitud. Al ejecutar un caso el programa 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 estados 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 implantado, con los problemas de fiabilidad que ello conlleva.
- Necesita mayor cantidad de cálculo que el FIR.

Si se elige este algoritmo, 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 a utilizar en el modelo del proceso.

Si además se activa el botón *Expert*, habrá más opciones para configurar la identificación por este algoritmo, permitiendo exportar 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 manipuladas 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 manipuladas (MV) para identificar modelos normales y luego los integra para obtener el modelo en rampa.

3.2.8.1. Utilidades para los casos

Model 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 hay que situarse 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 se realiza el análisis de los datos y del modelo, la información de la incertidumbre del modelo puede ayudar a determinar qué variables requerirán más ensayos y la frecuencia necesaria de los mismos para mejorar los modelos existentes.

Para tratar esta información, en esta versión de DMC3, el usuario no tiene que realizar ningún paso, puesto que está integrada la herramienta y es de fácil uso. En versiones anteriores, había que recurrir a unas plantillas en Excel alojadas en la web de soporte técnico de Aspen.

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 manipuladas correlacionadas.

3.2.9. Modelos

Estos objetos modelan la dinámica del proceso. Se utilizan para generar y mantener una predicción de las variables a controlar. Como ya se ha indicado, los casos generan una serie de modelos de salida. Una vez que se tienen dichos modelos, se analizan y se decide qué curvas son significantes y cuáles no. Si es necesario es posible realizar convoluciones de modelos para obtener relaciones que no es posible determinar de manera directa.

3.2.9.1. Creación de un modelo

Cuando se crea un modelo, por defecto se toma que será un modelo *Assembled*, aunque se puede también importar. Para crearlo hay que situarse en la carpeta *Models* y pulsar el botón de la barra de herramientas *New* o bien usar el menú contextual. Entonces aparecerá la ventana de propiedades del modelo, figura 13.

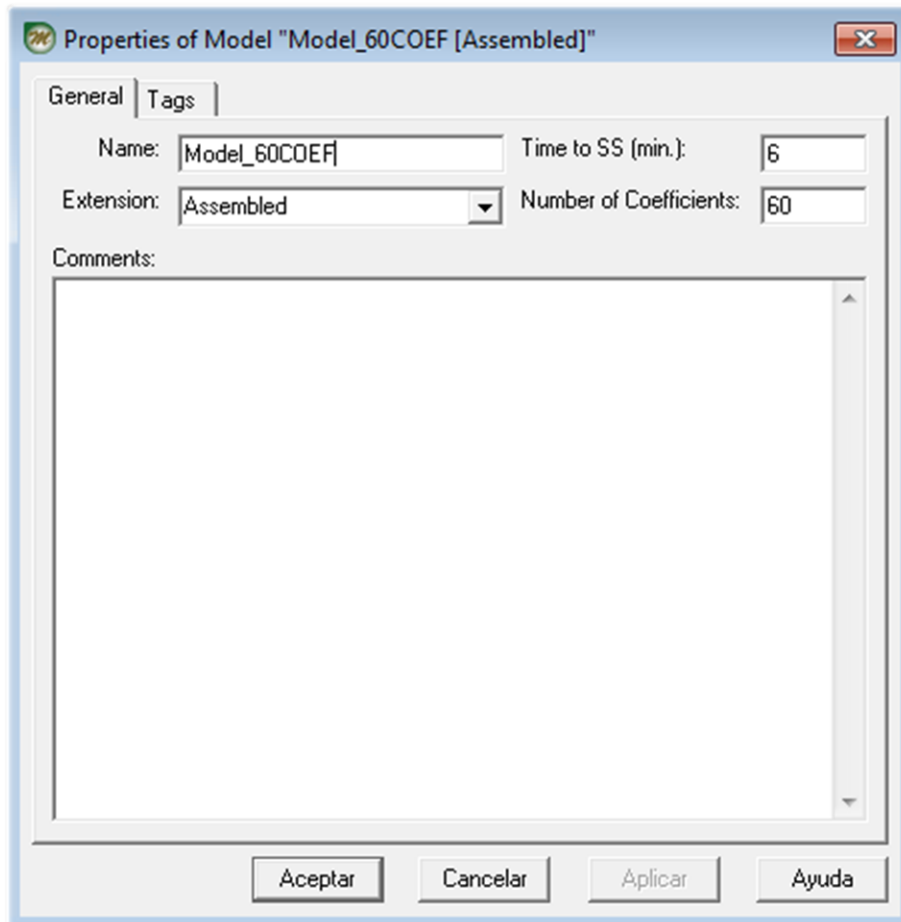


Figura 13.- Propiedades del modelo

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

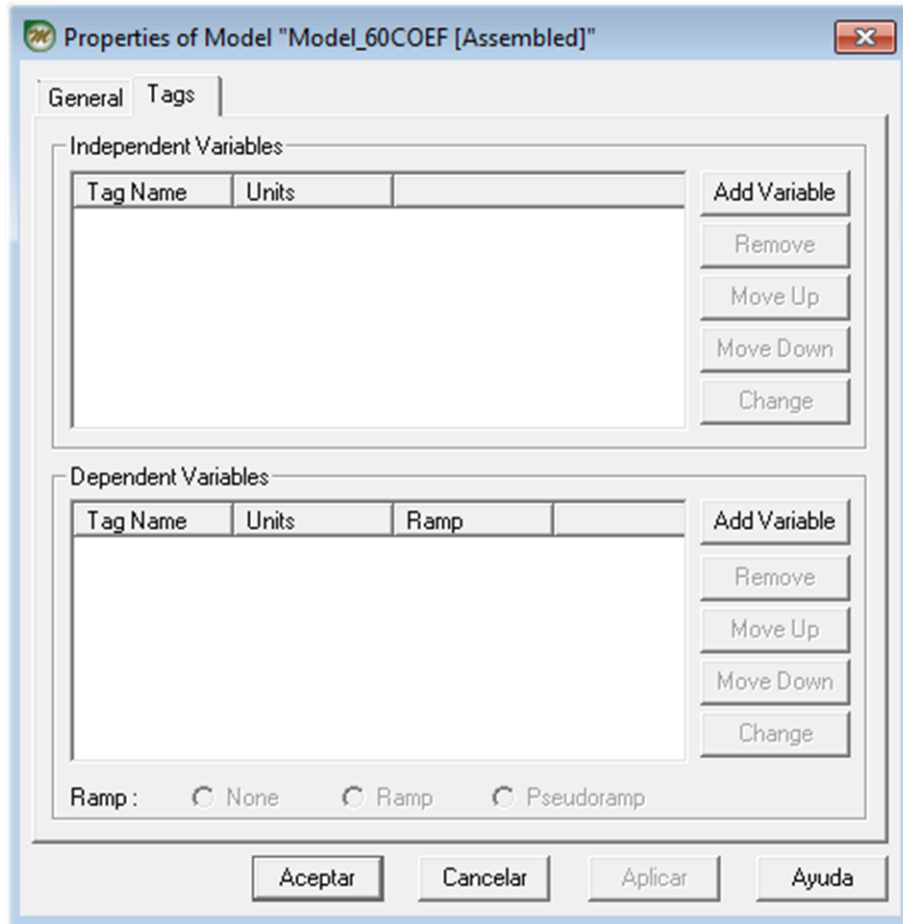


Figura 14.- Elección de variables para el modelo

En esta ventana se elegirá el nombre de las variables que se desee de entre las que estén disponibles. Se puede observar que esta ventana es similar a la que aparecía al crear un caso. Una vez que se ha elegido, se valida y ya se tiene creado el modelo.

Si se abre el modelo recién creado, se verá que no tiene ninguna curva de respuesta, por lo que aún se debe seguir trabajando con él. Hay 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 esté el modelo en blanco, con lo que las curvas del primer modelo serán añadidas en el actual. Esto puede hacerse curva a curva si se selecciona únicamente una de las curvas del modelo (se hace doble clic con el ratón en la curva que se desee), lo que 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 se usa *Edit* → *Get Curve*, eligiendo entonces la curva fuente y la de destino.

B. El mismo usuario puede crear las curvas de respuesta utilizando para ello las operaciones (*Operation Curves*) que hay a disposición del usuario en Model. Basta con seleccionar el par de variables al que se quiere añadir una curva y usar las *Curve Operations*. Más adelante se explicarán estas operaciones con más detalle.

MODELOS DE CONVOLUCIÓN

Ya se ha 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 se debe disponer de dos modelos creados previamente. Así, hay que situarse encima de un modelo existente y desplegar el menú contextual.

Se elige entonces la opción *Assemble Convolved Model*, lo cual lleva a una nueva ventana como la de la figura 15. En esta ventana hay que elegir el segundo modelo que se usará en la convolución, ya que el primer modelo será el que se había elegido al principio.

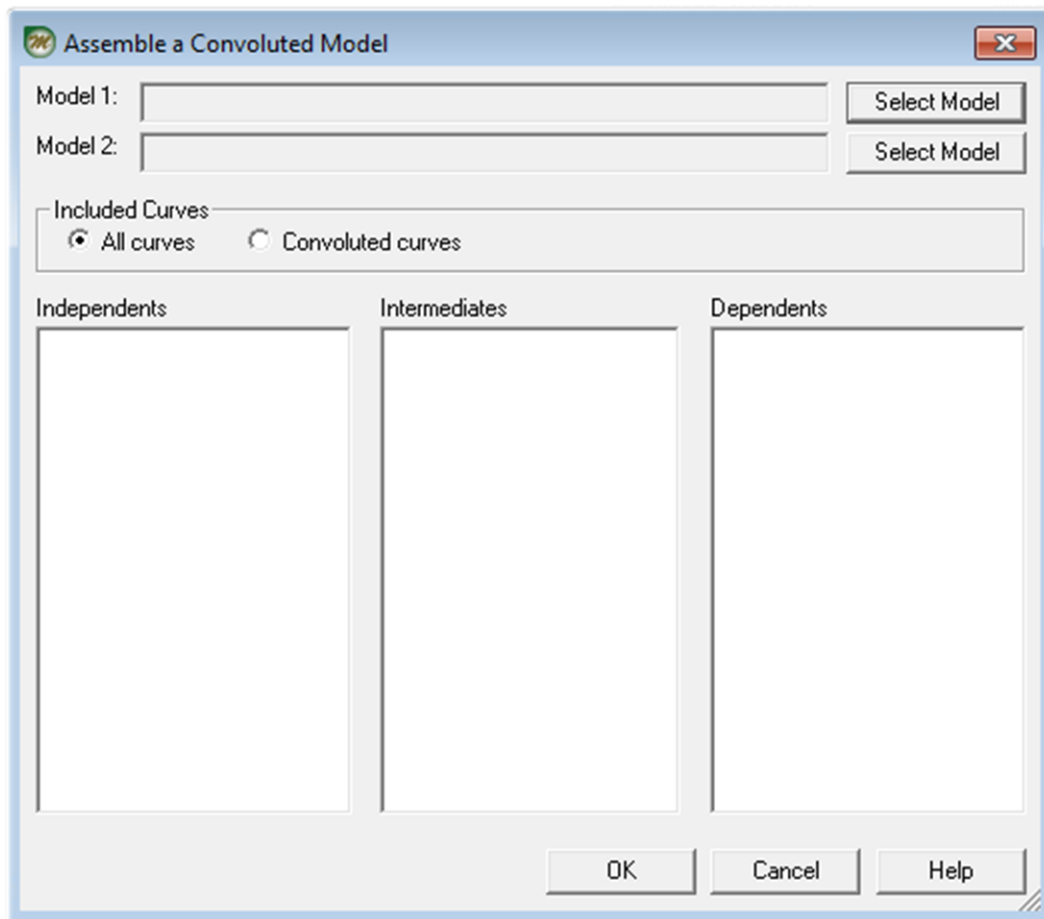


Figura 15.- Creación de un modelo de convolución

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

3.2.9.2. Trabajar con modelos

SUSTITUIR CURVAS

Esta función se puede utilizar para cambiar una curva existente por otra que le sea más favorable al modelo final. Su funcionamiento es exactamente igual al de la creación de un modelo nuevo explicada en el apartado A, página 39.

Para cambiar una curva por otra, simplemente se ha de seleccionar la curva a cambiar del modelo y acudir al menú *Edit* → *Get Curve*. Entonces aparecerá un diálogo en el que se pedirá información de la curva a insertar y de la curva a eliminar.

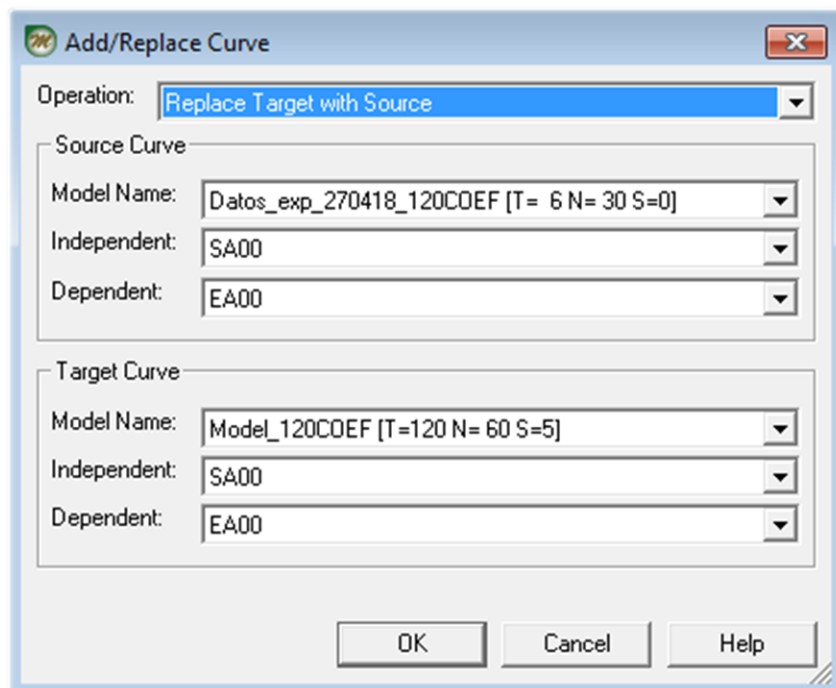


Figura 16.- Diálogo para la sustitución de curvas

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

CURVE OPERATIONS

Se trata de una serie de funciones que permiten modificar la forma de las curvas de respuesta que se tenga en el modelo propio. Cabe distinguir dos formas de operación:

- Cuando no existe aún una curva de respuesta para un par de variables: permite añadir curvas de respuesta estándar utilizando los parámetros que se indiquen. Estas funciones son: remplazar, curva cero, unidad, primer orden, segundo orden y convolución.

- Existe una curva de respuesta para el par de variables seleccionadas: entonces lo que se 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'.

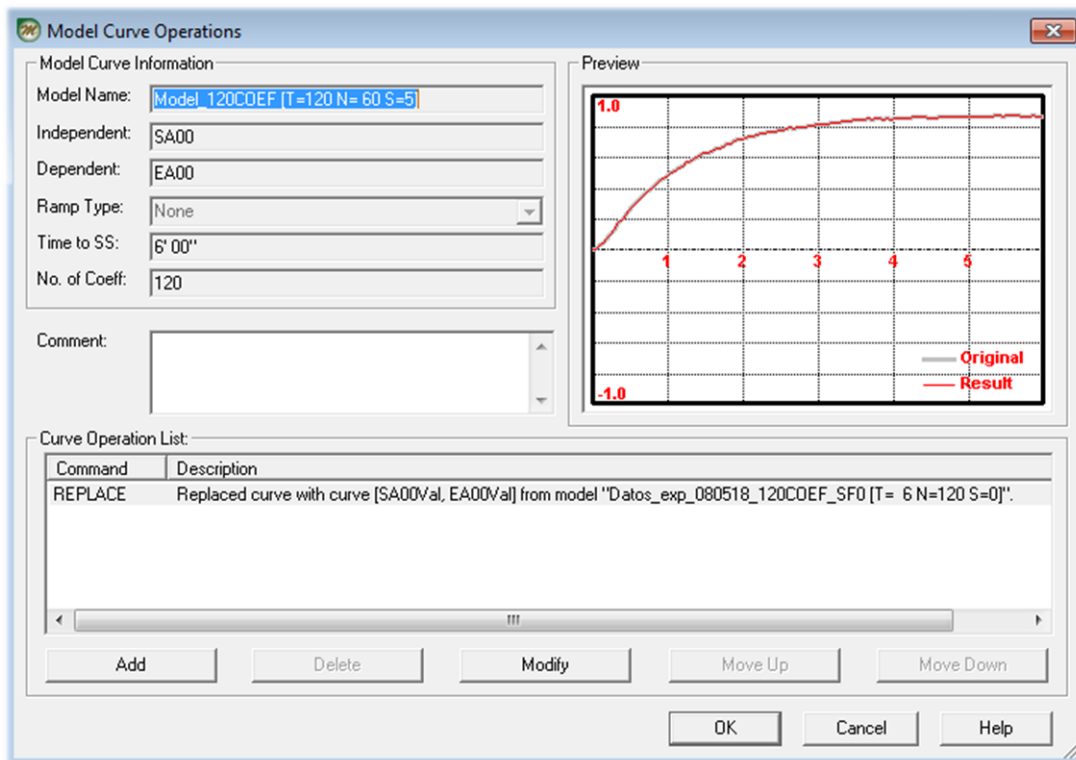


Figura 17.- Curve Operations

Para acceder a *Curve Operations* se ha de seleccionar la curva de respuesta de un par de variables y dirigirse al menú *Edit*, o bien utilizar el menú contextual. Una vez que aparezca una ventana como la de la figura 17, simplemente se han de añadir las curvas que sean 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 ayudar al usuario a comprobar el resultado de los cambios que se introduzcan, es de especial interés el uso del botón *Run/Refresh*, que 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 se especifiquen.
- *SECONDDORDER* (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 que se especifique y continuando hasta el final, hace que los coeficientes lleguen de forma lineal a un valor que se indique.
- *GAINSCALE* (ganancia escalada): se especifica 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 en que se diga y hasta el final, de forma que los coeficientes generen u curva con pendiente igual a $1/\text{coef}$, siendo coef un valor dado por el usuario.
- *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 se introduzcan.
- *ROTATE*: modifica la curva al completo para alcanzar el valor final que se pida. 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 de esta forma problemas estructurales que 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.

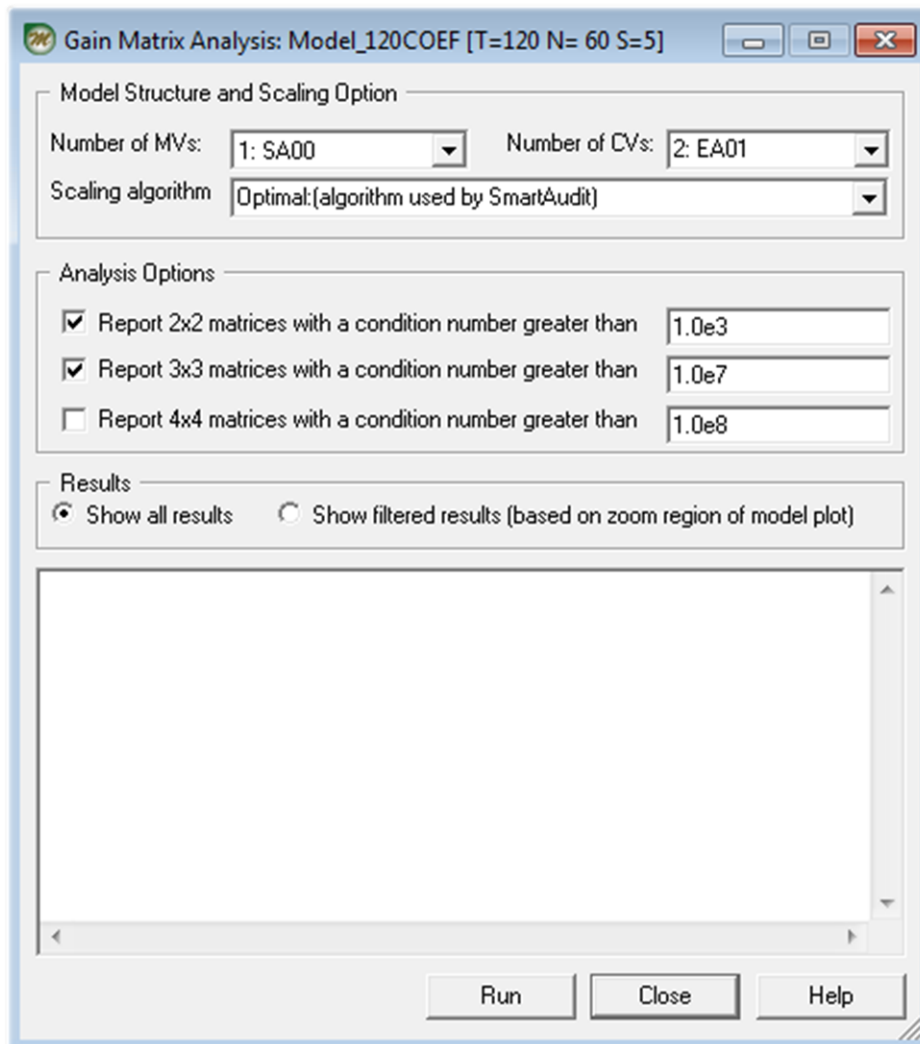


Figura 18.- Gain Matrix Analysis

3.2.9.3. 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 se vaya a aplicar.

Por lo anterior se debe obtener un archivo .mdl3 (los .dpa son modelos exportados para ser usados con Model, no con Builder) que contenga el modelo final. Para

realizar la exportación hay que situarse sobre el modelo a exportar y acudir al menú *Project* → *Export*. También, una vez se está situado sobre el modelo, se puede usar el botón *Export* o bien usar el menú contextual.

3.2.10. 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.

Se pueden tener predicciones de dos tipos:

- Provenientes de los casos. Se tratan de los modelos que se indican en la construcción de los casos que generarán predicciones.
- Creadas por el usuario para realizar predicciones de los modelos creados por el mismo.

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

Para crear una predicción basta con situarse en la carpeta de predicciones y entrar en *Project* → *New Prediction*.

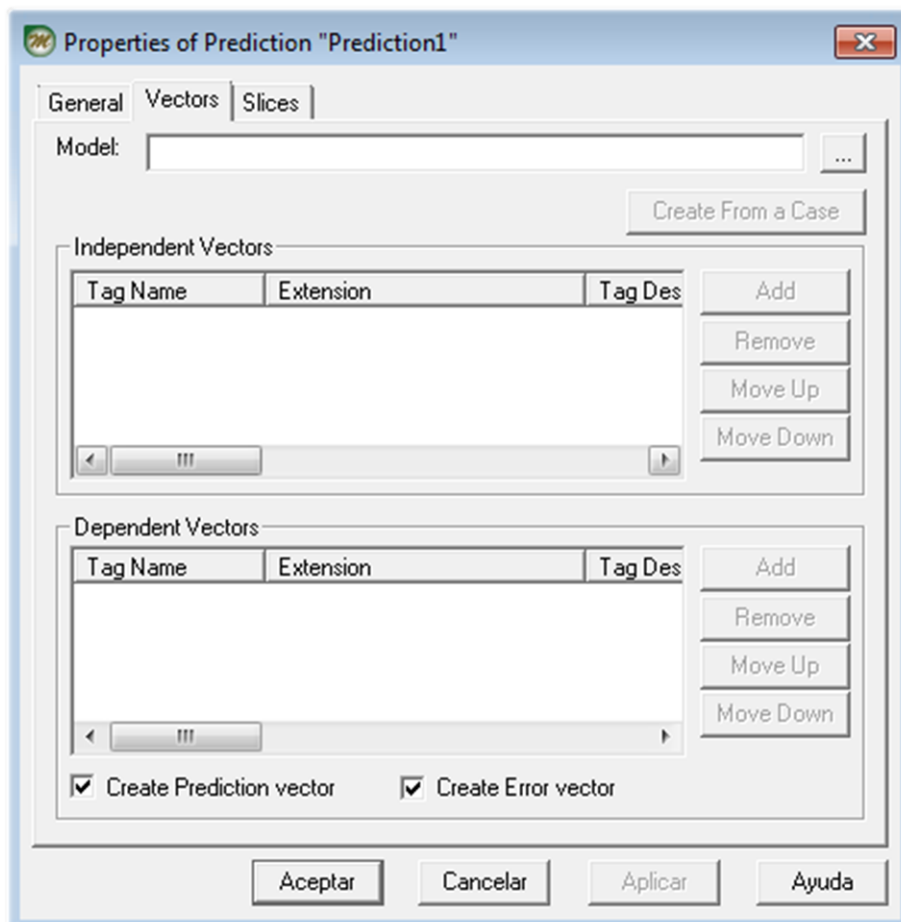


Figura 19.- Propiedades de una predicción

Una vez que se rellenen los campos que se indican, se podrá generar la predicción del modelo pulsando el botón *Run* como se hacía en los casos. Entonces se obtendrá una predicción que se podrá visualizar para confirmar el funcionamiento del modelo utilizado en dicha predicción.

En la visualización de la predicción se verán los vectores que se obtuvieron de la planta mediante *Collect/Extract* junto con los vectores de la predicción y unos vectores que indican el error encontrado entre los vectores con los datos de la planta y los vectores de la predicción.

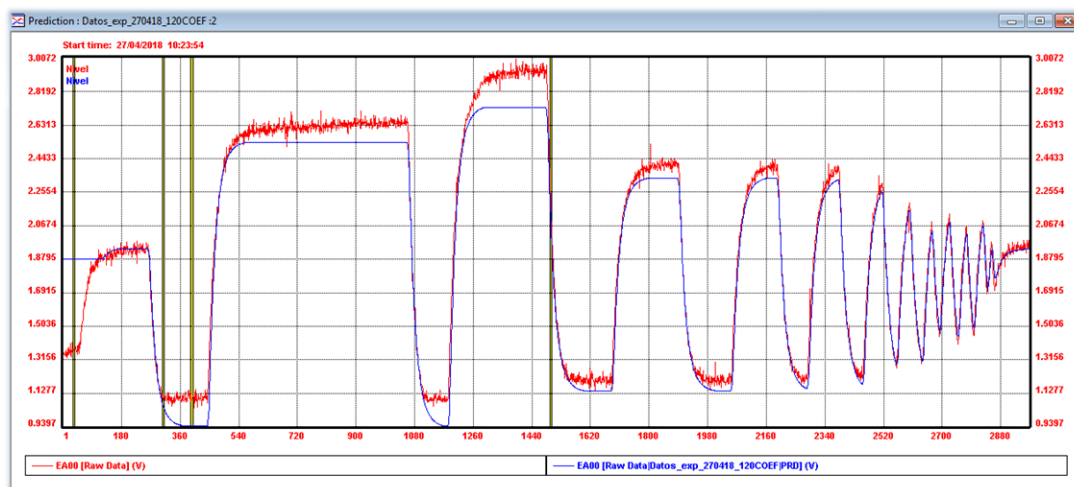


Figura 20.- Predicción generada a partir de un modelo de usuario

3.2.11. 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. Esta acción no mueve el objeto de lugar, sino que crea una copia de éste en la lista.

Capítulo 4. SISTEMA DE CONTROL

4.1. ¿Qué se ofrece en este capítulo?

En este capítulo se pretende abordar la parte de la construcción del controlador, para ello, se introduce el control predictivo por matriz dinámica, sus principales ventajas, aplicabilidad e interés industrial.

Además, se realiza el estudio del sistema de control y se expone la propuesta del autor de este proyecto para el diseño final del controlador, el cual se rematará en el capítulo siguiente.

4.2. Introducción al control predictivo

Se puede nombrar este Control predictivo de dos maneras:

- Control Predictivo (MPC)
- Control Predictivo basado en modelos (MBPC)

Como características principales:

- Las técnicas de control predictivo se aplican en diferentes entornos industriales: papeleras, azucareras, automoción, petroquímica, etc.
- El control predictivo da una forma de diseño, no un algoritmo específico.
- Cada usuario puede desarrollar su propio controlador en base a sus necesidades.

MPC se refiere a un conjunto de algoritmos que utilizan el modelo de un sistema para predecir la respuesta futura de la planta. Esa predicción se utiliza para calcular una secuencia de control óptima.

¿Cuándo se debe utilizar Control Predictivo?

Se utiliza para procesos difíciles de controlar con algoritmos clásicos: constante de tiempo grande, retardos, fase no mínima, etc. También cuando son sistemas MIMO y cuando hay restricciones en las variables manipuladas y/o controladas.

¿Qué ventajas tiene?

- Formulación 'sencilla'.
- Uso de restricciones.
- Uso de un modelo.
- Parámetros de sintonía claros:
 - Horizonte de predicción.
 - Problema de optimización.
- Desarrollo más rápido que otras técnicas de control avanzado.
- Mantenimiento sencillo.

Algunas de las aplicaciones más conocidas del MPC son:

- Refino, petroquímica y polímeros.
- Planificación de la producción de semiconductores.
- Control del tráfico aéreo.
- Anestesia clínica.
- Extensión del tiempo de vida de los sistemas de turbina y caldera.
- Robots.
- Industria del cemento y desecadoras.

4.3. Propuesta de control

4.3.1. Aplicabilidad

Para este proyecto, se ha decidido utilizar un tipo de control predictivo de control de matriz dinámica llamado DMC (*Dynamic Matrix Control*), utilizado sobre todo en la industria petroquímica actualmente.

Ventajas:

- Modelo sencillo de implementar.
- Fácil de comprender.
- Atractivo para el uso industrial.
- No es necesario hacer supuestos sobre el orden del modelo.

Desventajas:

No sirve en plantas inestables en lazo abierto.

4.3.2. Entonces, ¿Por qué elegir DMC?

Aufderheide B. et al (Aufderheide & Wayne, 2003), en su investigación, observan el comportamiento de un DMC bajo distintos regímenes de operación utilizando modelos múltiples de control predictivo, comentan que estas estrategias tienen gran potencial para manejar sistemas con alta variabilidad donde las posibilidades de obtener modelos basados en principios básicos son bajas o cuando múltiples perturbaciones están presentes. Bentes F. et al (Bentes, et al., 2006) presentaron un análisis comparativo entre el control PI clásico y estrategias de MPC (*Model Predictive Control*) en un proceso de secado, encontraron que para este tipo de procesos el control mediante DMC presenta un comportamiento satisfactorio incluso para el caso de las mayores perturbaciones.

Fischer M. et al (Fischer, et al., 1998) por su parte analizaron una estrategia de control predictivo basada en lógica difusa y compararon sus resultados con los obtenidos en un control mediante DMC, comentan que las dos estrategias mostraron gran efectividad en mantener la estabilidad cuando se usaron en un intercambiador de calor a escala industrial.

Demircan M. et al (Demircan, et al., 1999) usaron modelos de lógica difusa para implementar un controlador DMC, en sus resultados comentan que el comportamiento del sistema es bueno en la mayor parte de la zona de operación. Vasconcelos L. et al. (Vasconcelos & Maciel, 1998) desarrollaron una estrategia de control para el secado utilizando matriz dinámica como medio para obtener mejores resultados en la producción, en sus resultados mostraron que la estrategia tiene un comportamiento satisfactorio frente a variaciones en el punto de control y a perturbaciones.

Cutler C. (Cutler & Johnston, 1985) en su investigación propone establecer criterios de evaluación del desempeño que permitan observar el comportamiento de las estrategias de DMC y PID en ambientes similares. Ruihua W. and Lihong X. (Ruihua & Lihong, 2009) por su parte implementaron una estrategia en la cual un controlador DMC y un PID clásico trabajaban en paralelo en un proceso, en sus

resultados establecen que este tipo de estrategia conjugada también presenta buenos resultados en términos de variaciones en los parámetros y frente a perturbaciones. Serra M. et al (Serra, et al., 2001) analizan distintas posibilidades de control para una columna de separación, comentan sobre la capacidad de controladores DMC para manejar perturbaciones y cambios en el punto de control frente a el control clásico o convencional.

Sanjuán M. et al (Sanjuán, et al., 2006) en su investigación proponen una ecuación de sintonización para controladores de matriz dinámica en lazo simple. En sus resultados comentan que la ecuación lograda permite un desempeño adecuado de la estrategia. Kember G. et al (Kember, et al., 2005) en su investigación muestran una formulación para la sintonización de controladores DMC en lazo simple, además realizan su comparación con otras técnicas de sintonización de parámetros, mostrando resultados adecuados. Kember G. et al (Dubay, et al., 2005) continuaron con sus investigaciones y desarrollaron un set de ecuaciones para la sintonización de controladores de matriz dinámica, pero para el caso multivariable, en sus resultados aparece la comparación de su formulación con el método *m-shifted*.

4.3.3. Interés industrial

Tal y como se ha mencionado anteriormente, este tipo de sistemas de control se utiliza en la industria azucarera, en la de aluminio, en la papelera y de fibras, pero, sobre todo, y es la motivación de este proyecto, se utiliza en la industria petroquímica.

Sirve para controlar columnas de destilación, unidades de cracking, hornos, etc. Y, trabajando a la par con una de las empresas más representativas a nivel nacional en el campo de la petroquímica, se utiliza para el modelado, desarrollo y posterior diseño del controlador, el software DMC3 (heredero del DMCPlus) desarrollado por AspenTech. Cuyo uso están ejerciendo hoy en día muchas de las grandes empresas petroquímicas.

Se ha elegido DMC3 en lugar de DMCPlus, porque a diferencia del segundo, el primero incorpora un algoritmo de identificación por subespacios, además de una interfaz gráfica más moderna, depurada y atractiva para el usuario.

¿Por qué usar DMC3 de AspenTech?

En primer lugar, abre el acceso a la zona privada a la web de Aspen DMC, donde se realiza intercambio de conocimientos, tan importante a la hora de resolver dudas. Además, DMC3 tiene la opción de implementar *gain scheduling*, a diferencia de versiones anteriores, como el DMCplus, que no la tiene.

Este software permite hacer la identificación del modelo, e implementar las funciones peso, muy complicadas, pero realmente necesarias.

El propio optimizador del software es el que se utiliza. Además, el optimizador designa la consigna en el futuro, para que el controlador tenga una referencia que seguir.

Es importante recordar que la planta en la que se trabaja es no lineal, luego es importante que los cambios sobre el punto de operación sean pequeños. De otro modo, el controlador podría no resultar válido. Se recuerda que este tipo de controladores se utiliza para sistemas estables en lazo abierto.

4.4. Estudio del control

4.4.1. Modelos de convolución

Los modelos de convolución engloban los modelos de respuesta impulsional y respuesta ante escalón, de gran éxito en la industria por ser muy intuitivos y permitir un procedimiento de identificación relativamente sencillo. Estos tipos de modelo han dado lugar a dos de los controladores más extendidos en la práctica: *Dynamic Matrix Control*, (DMC) y *Model Algorithmic Control* (MAC). Se muestra aquí la formulación del primero de ellos, siendo la del segundo muy similar.

El controlador DMC se desarrolló a finales de los años setenta del siglo pasado por Cutler y Ramaker (Cutler & Ramaker, n.d.) de la Shell Oil Co. y ha sido ampliamente aceptado en el entorno industrial, principalmente por la industria petroquímica (Qin y Badgwell, 1997 (Badgwell & Qin, 1997)). Su gran éxito proviene de su capacidad para tratar procesos multivariables, aunque en esta sección sólo se aborda el caso monovariable ya que permite comprender mejor los fundamentos sin perderse en notación matemática. (Bordons & Camacho, 2010)

Como se ha indicado previamente, se usa un modelo de respuesta ante escalón para capturar la dinámica del proceso, mientras que la perturbación se considera constante a lo largo del horizonte. El procedimiento para obtener las predicciones se muestra a continuación.

Al usar el siguiente modelo de respuesta ante escalón:

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t - i)$$

los valores predichos de la salida a lo largo del horizonte serán:

$$\begin{aligned} \hat{y}(t + k | t) &= \sum_{i=1}^{\infty} g_i \Delta u(t + k - i) + \hat{n}(t + k | t) \\ &= \sum_{i=1}^k g_i \Delta u(t + k - i) + \sum_{i=k+1}^{\infty} g_i \Delta u(t + k - i) \\ &\quad + \hat{n}(t + k | t) \end{aligned}$$

Como se considera que las perturbaciones son constantes en el futuro, $\hat{n}(t + k | t) = \hat{n}(t | t) = y_m(t) - \hat{y}(t | t)$ se puede escribir:

$$\hat{y}(t+k | t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + y_m(t) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) - \sum_{i=1}^{\infty} g_i \Delta u(t-i)$$

Es decir, la predicción viene dada por:

$$\hat{y}(t+k | t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + f(t+k)$$

donde $f(t+k)$ es la respuesta libre del sistema, que como se sabe no depende de las acciones de control futuras, y viene dada por:

$$f(t+k) = y_m(t) + \sum_{i=1}^{\infty} (g_{k+i} - g_i) \Delta u(t-i)$$

Si el proceso es asintóticamente estable los coeficientes g_i de la respuesta ante escalón tienden a valores constantes tras un cierto número N de periodos de muestreo, por lo que se puede considerar que

$$g_{k+i} - g_i \approx 0, \quad i > N$$

y por lo tanto se puede calcular la respuesta libre como

$$f(t+k) = y_m(t) + \sum_{i=1}^N (g_{k+i} - g_i) \Delta u(t-i)$$

Si el sistema no es asintóticamente estable, no existe tal valor N y $f(t+k)$ no se puede calcular (aunque existe una generalización en el caso de que la inestabilidad sea debida a integradores puros). Ahora se pueden calcular las predicciones a lo largo del horizonte de predicción ($k = 1, \dots, p$), considerando m acciones de control:

$$\begin{aligned} \hat{y}(t+1 | t) &= g_1 \Delta u(t) + f(t+1) \\ \hat{y}(t+2 | t) &= g_2 \Delta u(t) + g_1 \Delta u(t+1) + f(t+2) \\ &\vdots \\ \hat{y}(t+p | t) &= \sum_{i=p-m+1}^p g_i \Delta u(t+p-i) + f(t+p) \end{aligned}$$

Si se define ahora la matriz dinámica del sistema \mathbf{G} (que da nombre al algoritmo) como

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_m & g_{m-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_p & g_{p-1} & \cdots & g_{p-m+1} \end{bmatrix}$$

se puede escribir la predicción como

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}$$

que tiene la misma forma que las predicciones calculadas por el GPC. \mathbf{G} está formada por m (horizonte de control) columnas de la respuesta ante escalón del sistema deslizadas hacia abajo de forma apropiada. $\hat{\mathbf{y}}$ es un vector de dimensión p que contiene las predicciones, \mathbf{u} tiene dimensión m y contiene los incrementos en las acciones de control y \mathbf{f} es el vector de la respuesta libre. \mathbf{f} depende del vector de estados $\mathbf{x}(t)$, que en este caso viene dado por $\mathbf{x}(t)^T = [y_m(t)u(t-1)u(t-2) \dots u(t-N-1)]$ y se puede expresar como $\mathbf{f} = \mathbf{F}\mathbf{x}(t)$, con lo que se puede escribir la predicción como:

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{F}\mathbf{x}(t)$$

Las variables manipuladas se calculan, como en cualquier controlador predictivo, de forma que minimicen la siguiente función objetivo cuadrática, que incluye el esfuerzo de control:

$$J = \sum_{j=1}^p [\hat{y}(t+j | t) - w(t+j)]^2 + \sum_{j=1}^m \lambda [\Delta u(t+j-1)]^2$$

Si no hay restricciones, la solución a la minimización de la función de coste $J = \mathbf{e}\mathbf{e}^T + \lambda\mathbf{u}\mathbf{u}^T$, donde \mathbf{e} es el vector de errores futuros a lo largo del horizonte, se puede obtener de forma analítica igualando a cero la derivada de J (igual que en el GPC), proporcionando:

$$\mathbf{u} = (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f})$$

Conviene recordar que, como ocurre en todas las estrategias de control predictivo, sólo el primer elemento del vector \mathbf{u} ($\Delta u(t)$) se envía realmente a la planta. No es

aconsejable implementar la secuencia de control calculada completa (m valores), ya que es imposible estimar perfectamente las perturbaciones en el horizonte y por tanto es imposible anticiparse de forma precisa a las inevitables perturbaciones que hacen que la salida real difiera de las predicciones que se han usado en el cálculo. Además, la referencia podría cambiar en dicho intervalo.

4.4.2. *Move suppression*

La formulación básica del DMC calcula un plan de movimientos futuros para la variable manipulada de forma que se minimice el error a lo largo del horizonte de predicción. Esta solución, a pesar de ser la más óptima para minimizar el error, puede ser demasiado agresiva para un proceso real, pudiendo provocar cambios muy fuertes de las variables manipuladas. Cuando se controla un proceso real, normalmente se desea una acción del controlador que sea más suave, a pesar de perder capacidad de respuesta del controlador y a costa de sacrificar el objetivo de minimización del error.

A un controlador multivariable predictivo se le puede añadir fácilmente un segundo objetivo consistente en minimizar la magnitud de los movimientos del vector que minimiza el error a lo largo del horizonte de predicción. Este objetivo entra en conflicto con el de minimizar el error, pero es precisamente el algoritmo de mínimos cuadrados el que se encarga de encontrar una solución de compromiso entre los dos objetivos.

Para añadir el segundo objetivo se usa un factor de peso configurable, k , por ejemplo, que intenta cumplir dicho objetivo. Añadiendo este objetivo al de minimizar el error se obtiene una ecuación en la cual, cuanto mayor sea k , más importancia tiene en la resolución por mínimos cuadrados el que el vector que minimiza el error a lo largo del horizonte de predicción sea cero. Y, por tanto, más se alejará del objetivo de minimizar el error. Es decir, para mayores valores de k , se obtiene una acción de control más suave. En procesos normales k varía entre 0.1 y 50 (sin unidades). (González & Cifuentes, 1999)

En el controlador DMC, la constante k , se denomina *Move Suppression* en DMC3.

4.4.3. *Equal Concern*

En un entorno multivariable, el controlador está generando la ley de control para minimizar el error en todas las variables controladas al mismo tiempo. En un contexto de mínimos cuadrados, todos los errores, E_N , se intentarán minimizar con el mismo rigor, sin embargo, en el proceso no todas las variables controladas tienen la misma importancia. Por lo tanto, es necesario dotar al controlador de algún mecanismo que le permita decidir qué error E_N es más importante.

Multiplicando artificialmente las ecuaciones de minimización de error de una variable controlada por un coeficiente de peso, W_{CVn} , positivo, provocará que los residuos asociados con el CV sean mayores y, por lo tanto, el algoritmo de mínimos cuadrados intentará minimizar esos errores con más énfasis.

Se puede definir el concepto de *Equal Concern Error* como el nivel de error admisible para una variable controlada. Se define en las mismas unidades de

ingeniería que la CV. El *Equal Concern Error* no significa que el controlador permita dicha magnitud de error, si no que comparativamente pondere más o menos los errores en una CV con respecto a otra. En la medida que el *Equal Concern Error* es mayor, menos importancia relativa se da a dicha CV.

El factor de peso W_{CVn} definido anteriormente es el inverso del *Equal Concern Error*:

$$W_{CVN} = \frac{1}{\text{Equal Concern Error } CV_b}$$

En resumen, el *Equal Concern Error* normaliza los errores en las CVs para poner todas las magnitudes de error en una misma escala matemática. Y prioriza las variables controladas por orden de importancia relativa de unas con respecto a otras.

Capítulo 5. SIMULACIÓN Y OPTIMIZACIÓN

5.1. ¿Qué se ofrece en este capítulo?

En este capítulo se introduce la parte final de este proyecto sobre el diseño del controlador. Además, se describe de forma general el software que se utiliza durante todo el proceso de diseño, así como una redacción de un manual de uso para que el lector pueda utilizar el programa.

Por otro lado, se realiza en paralelo una comparación con versiones anteriores de la misma suite.

5.2. Aspen DMC3 Builder

5.2.1. Introducción

Este módulo de DMC3 se utilizará para la creación y configuración del controlador que se quiere aplicar al proceso. Es un paso previo al uso del simulador y de ejercer el control sobre la planta.

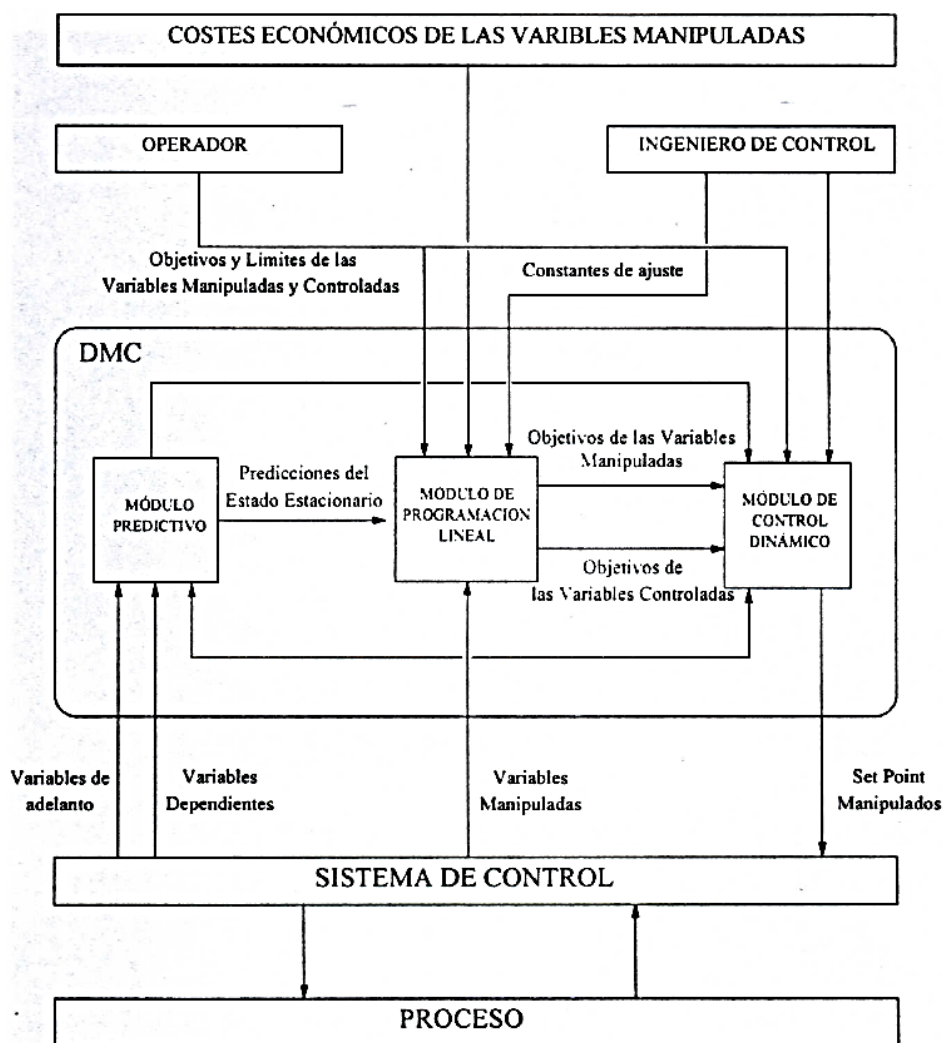


Figura 8 Funcionamiento de un controlador multivariable

5.2.2. Manejo

5.2.2.1. Funcionalidades

La apertura de este programa se realiza desde Inicio → Programas → AspenTech → Aspen Process Control → DMC3 Builder (APC).

Antes de comenzar, es importante recalcar que en esta nueva versión DMC3 no se exige, pero en la anterior DMCPlus, no procedía a la apertura del software sin tener configurado el punto como símbolo decimal. Con lo cual, aparecía un mensaje como el que se muestra a continuación:

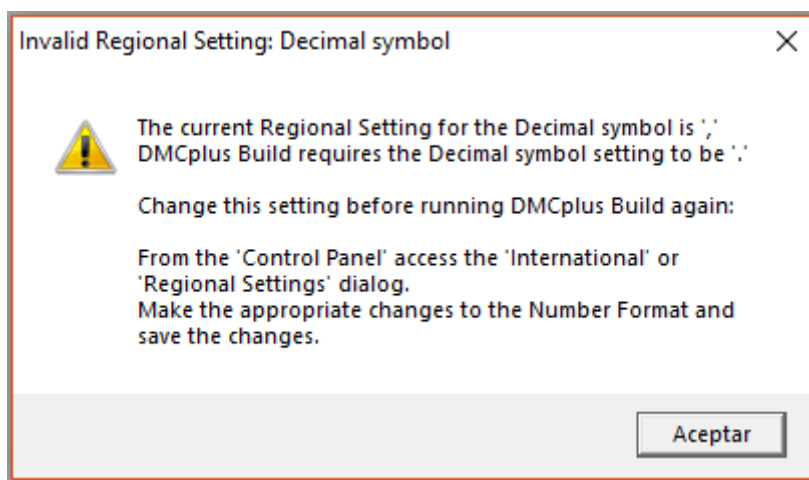


Figura 9 Establecer punto como símbolo decimal

Es por ello que se recomienda realizar este cambio en la configuración si no estuviera por defecto, tal y como se ha indicado en la parte final del apartado 2.2.2.1 Funcionalidades, en la página 11.

En la pestaña *Datasets*, se importan los vectores de datos con los que se trabaje.

En la pestaña *Controllers*, se importa el modelo que se ha generado en el DMC3 Model.

Para realizar el control sobre un proceso es necesario distinguir entre dos archivos con la información necesaria: un archivo con el modelo del proceso (.mdl3) 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.

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.

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

- MV (*Manipulated Variables*): variables manipuladas.
- FF (*Feedforward variables*): variables de perturbación.
- CV (*Controlled Variables*): son las variables que se desean controlar.

Que se pueden clasificar en dos grandes grupos en el programa:

Independents

En la sección de variables independientes se tendrán 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 se pueden manejar para controlar el proceso.
- *Feedforward Variables* (FF): se trata de variables independientes del proceso pero que es imposible su manipulación, como por ejemplo una entrada de agua corriente en la cual no se puede actuar sobre su temperatura. Estas variables pueden tratarse como una perturbación al sistema. Para seleccionar una variable independiente como *feedforward* simplemente se ha 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 se ha de hacer lo mismo, es decir, invalidar la misma '*check-box*'. Si se quiere deseleccionar todas las variables FF basta con situarse en la última de estas variables y deseleccionar la '*check-box*'.

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 se configure 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.

- .ccf \Rightarrow es el archivo de configuración del controlador (*Controller Configuration File*), el que se utilizará para ejercer el control.

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

5.3. Pantalla de DMC3 Builder

La ventana de Builder se divide en dos paneles principalmente, el de la izquierda o árbol de proyecto, desde donde se puede acceder de forma rápida a la configuración de los parámetros, y el panel derecho o panel de entradas, en el cual se modifican las características del controlador.

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

1. Se recomienda crear listas de vectores para las variables manipulada y para las controladas. Esto servirá para evitar confusiones en la estructura del controlador más adelante.
2. Especificar la estructura del control. Se especifican las variables que se tendrán en cuenta, así como el modelo que se va a utilizar.
3. Indicar opciones. Se ha de indicar una serie de opciones al iniciar la configuración del controlador, 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. Se dispone 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. Se deben ajustar las entradas del controlador a las necesidades del usuario 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 se está seguro de que no existen errores de configuración en el .ccf, aunque esto no garantiza que el control que se aplique funcione correctamente.
7. Utilizar el archivo .ccf con *Simulate* y *Manage*. Mediante *Simulate* se tiene 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.

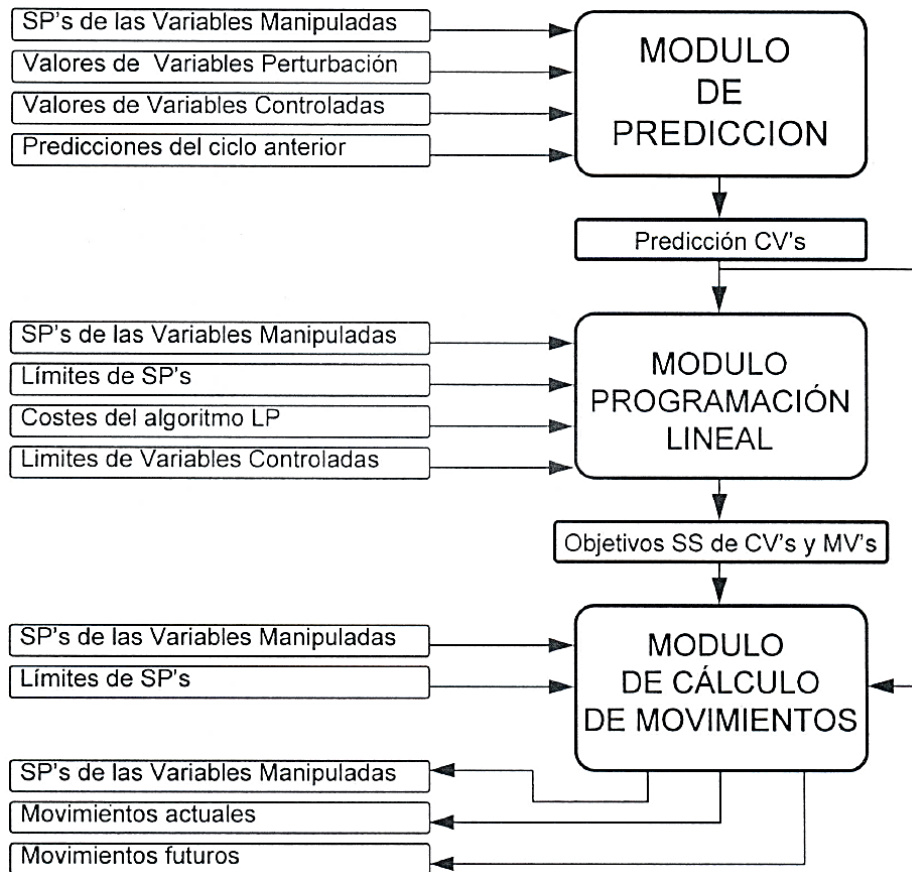


Figura 10 Secuencia de funcionamiento del controlador

Generación de documentación

Por último, es posible 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 permite pues no perder la configuración inicial y tenerla siempre disponible para cuando sea necesaria.

5.3.1. Descripción general

Aspen DMC3 Builder es una única plataforma integrada que se puede utilizar para construir y desplegar todas las aplicaciones de control avanzado de AspenTech (Aspen DMC3, Aspen DMCplus, Aspen Nonlinear Controller y Aspen State-Space Controller).

Aspen DMC3 Builder presenta una interfaz de usuario moderna que aprovecha las pantallas con pestañas y un despliegue de opciones contextual de estilo cinta, lo que facilita su navegación, mejorando la experiencia del usuario. El diseño orientado al flujo de trabajo de APC incluye todo desde la recopilación de datos hasta la implementación y administración del controlador, todo en un solo lugar. Funciones potentes como el análisis de colinealidad, identificaciones restringidas

y transformaciones visuales integradas en Aspen DMC3 Builder, lo que elimina la necesidad de realizar una transición entre varias aplicaciones, que a su vez preserva el conocimiento del proceso y aumenta la eficiencia general.

5.4. Usando Aspen DMC3 Builder

Haciendo clic en el icono Aspen DMC3 Builder (APC) en la página de inicio, se abre Aspen DMC3 Builder. Cuando se abre Aspen DMC3 Builder, se debería ver la página Empezar a usar DMC3 Builder. Aquí se verá la lista de proyectos y botones accedidos recientemente para crear un nuevo proyecto, se abre uno existente o se accede a los controladores en línea, lo que no requiere la apertura de ningún proyecto.

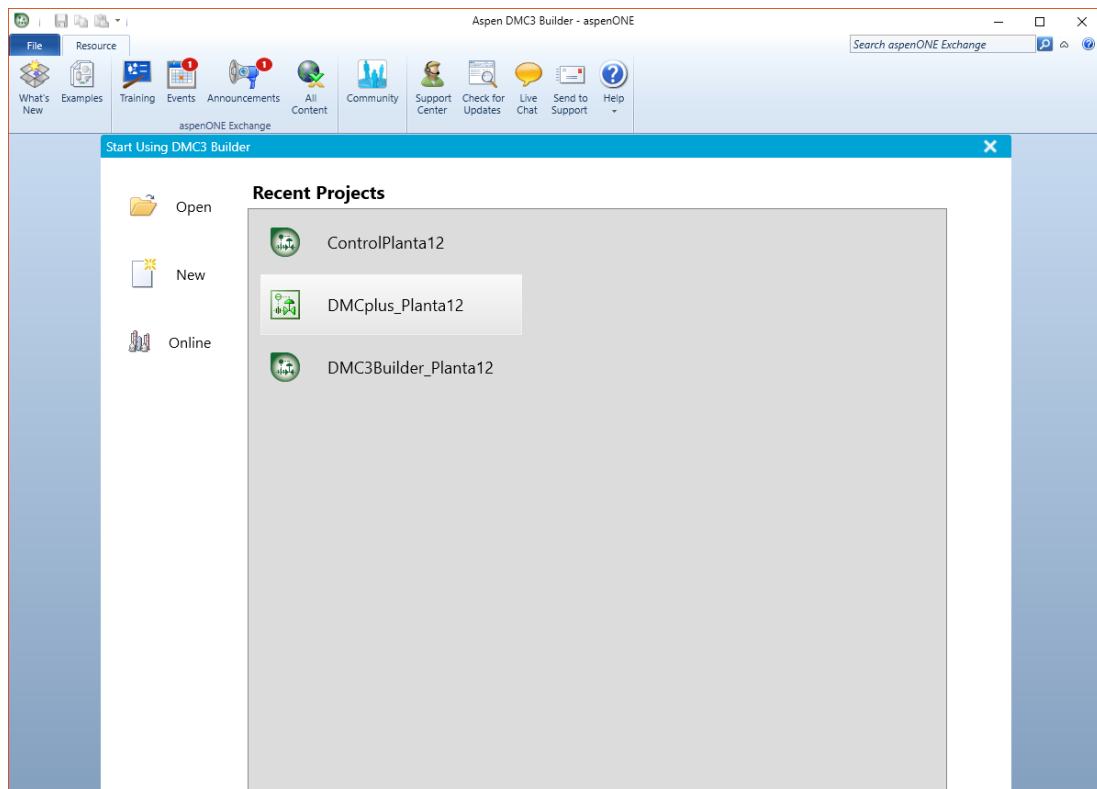


Figura 11 Empezando a usar DMC3 Builder

Se observa en la figura anterior cómo se muestran dos tipos de proyectos DMC3 Builder (dos iconos diferentes). El DMC3 Builder es un proyecto DMC3, mientras que uno llamado DMCplus es un tipo de proyecto APC. Cuando se hace clic en el botón Nuevo, se pueden ver las diferencias entre ellos, como se muestra en la figura a continuación.

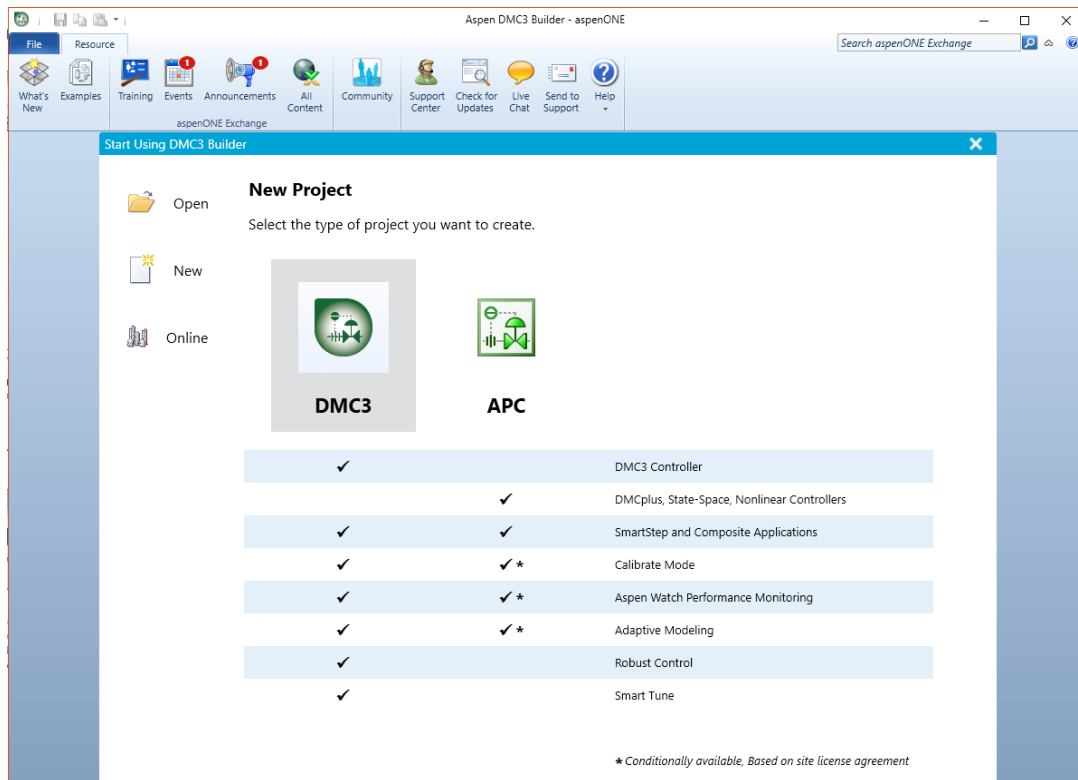


Figura 12 Tipos de Nuevo proyecto en Aspen DMC3 Builder

Se selecciona el tipo de proyecto DMC3 si se planea usar todas las funciones disponibles, como *Smart Tune* y *Robust Control*. Se selecciona el Proyecto APC si se planea construir controladores Aspen DMCplus, Aspen State-Space o Aspen Nonlinear Control (aunque en la versión 10 de la suite, esta última opción ya no se incluye).

5.5. aspenONE® Exchange

La primera vez que se abra Aspen DMC3 Builder, hay una pestaña de Recursos:

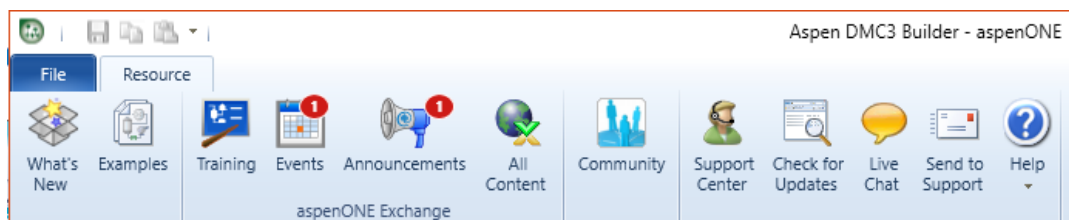


Figura 13 Pestaña de recursos

aspenONE Exchange permite acceder a información sobre productos de APC y compartir conocimientos y sugerencias de la industria de procesos con otros

usuarios. Al hacer clic en *All Content*, se accederá a una página en la que se pueden buscar artículos de *Knowledge Base (KB)*, información de capacitación, vídeos y tutoriales. La opción *E-Learning* incluye información de capacitación en línea relacionada. Por ejemplo, se hace clic en el botón de *All Content*, que debería abrir lo que se muestra en la Figura 4 a continuación:

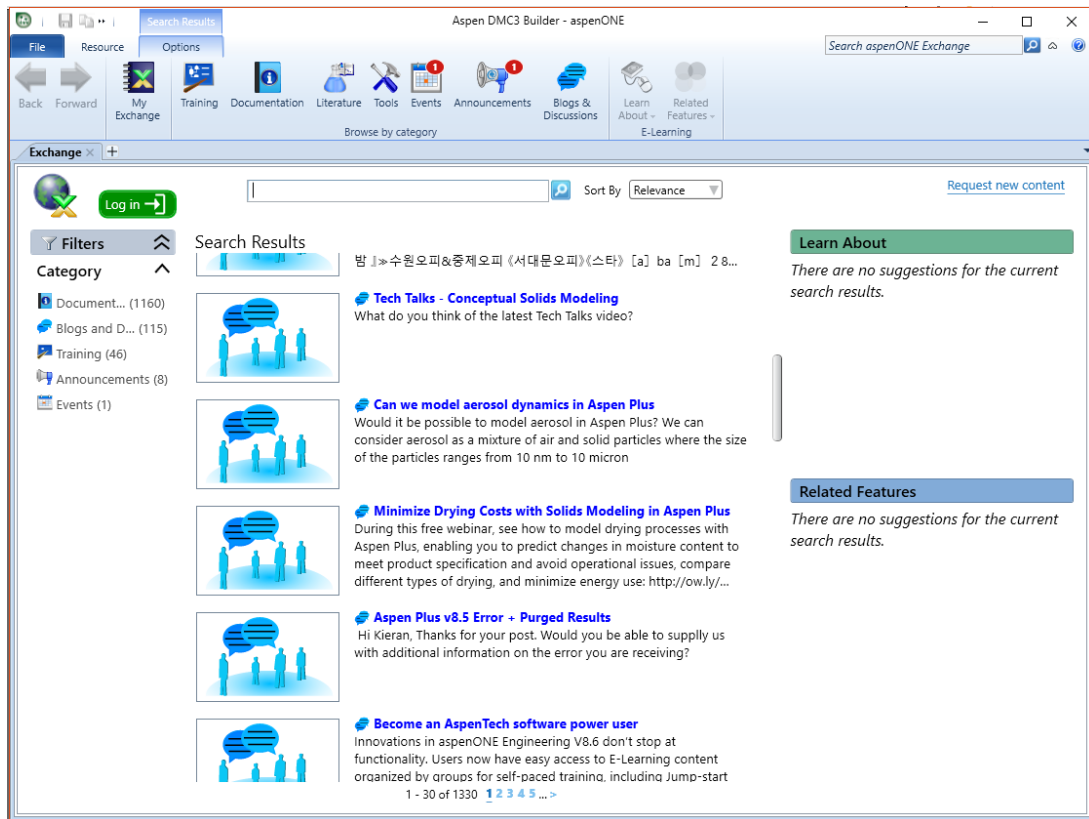


Figura 14 aspenONE Exchange All Content

Ahora, si se busca una solución específica haciendo clic en Documentación y buscando DMC3, los resultados se muestran en la Figura 5:

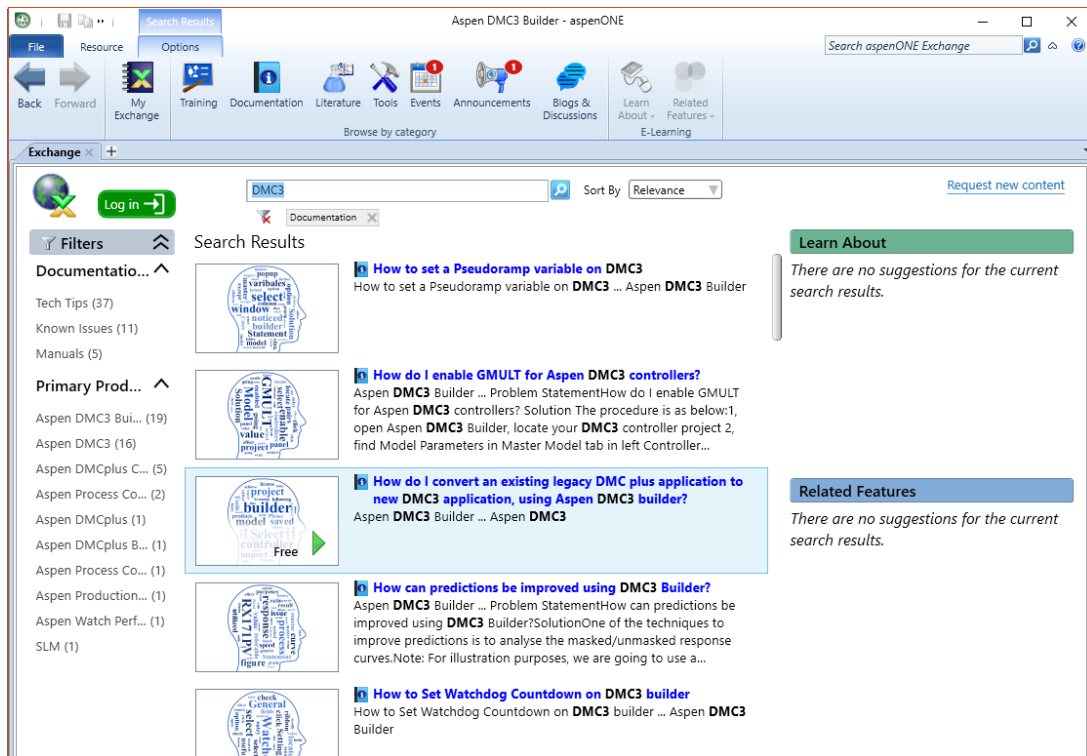


Figura 15 Resultados de búsqueda para DMC3 en aspenONE Exchange

Haciendo clic en uno de los ítems, aparece el diálogo mostrado en la figura 6:



Figura 16 Descripción del artículo en aspenONE Exchange

Al hacer clic en la flecha verde en la esquina superior derecha del cuadro de diálogo, o en el enlace *View Article*, se accederá al Soporte de AspenTech. Base de conocimiento (KB), que se muestra en la Figura 7:

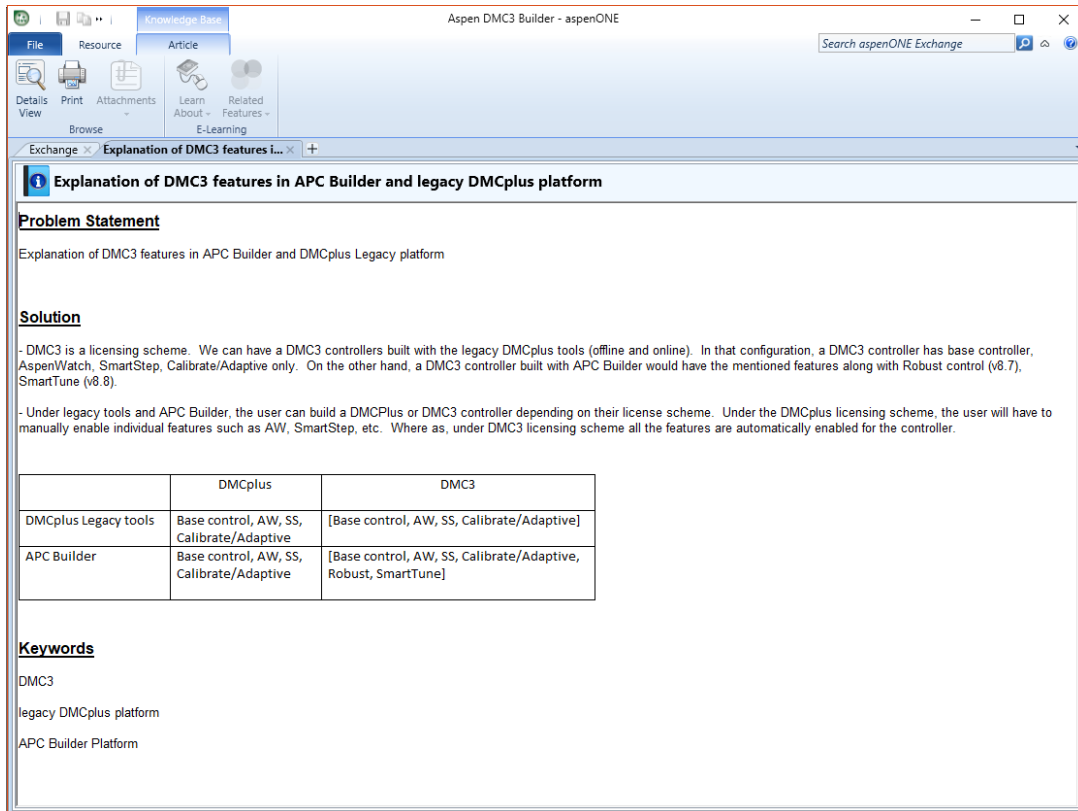


Figura 17 AspenTech Support Knowledge Base Solution

Se necesita una cuenta de soporte para que el intercambio aspenONE esté completamente habilitado. Esto también requerirá acceso a Internet para buscar y visualizar las soluciones. Se hace clic en el icono *My Exchange* o en *Log in* en la esquina superior izquierda de la vista aspenONE Exchange para ingresar la información de la cuenta y comenzar a usar aspenONE Exchange.

5.5.1. Menú de archivo

En Aspen DMC3 Builder, un menú de archivo está disponible en la esquina superior izquierda. Contiene varios elementos de menú, incluido el nuevo aspenONE Exchange. El menú *File* permite abrir, guardar y crear un proyecto. También se pueden importar aplicaciones, modelos y conjuntos de datos. La creación de un nuevo proyecto utilizando el menú *File* es la misma que en la Página de inicio, como se muestra a continuación:

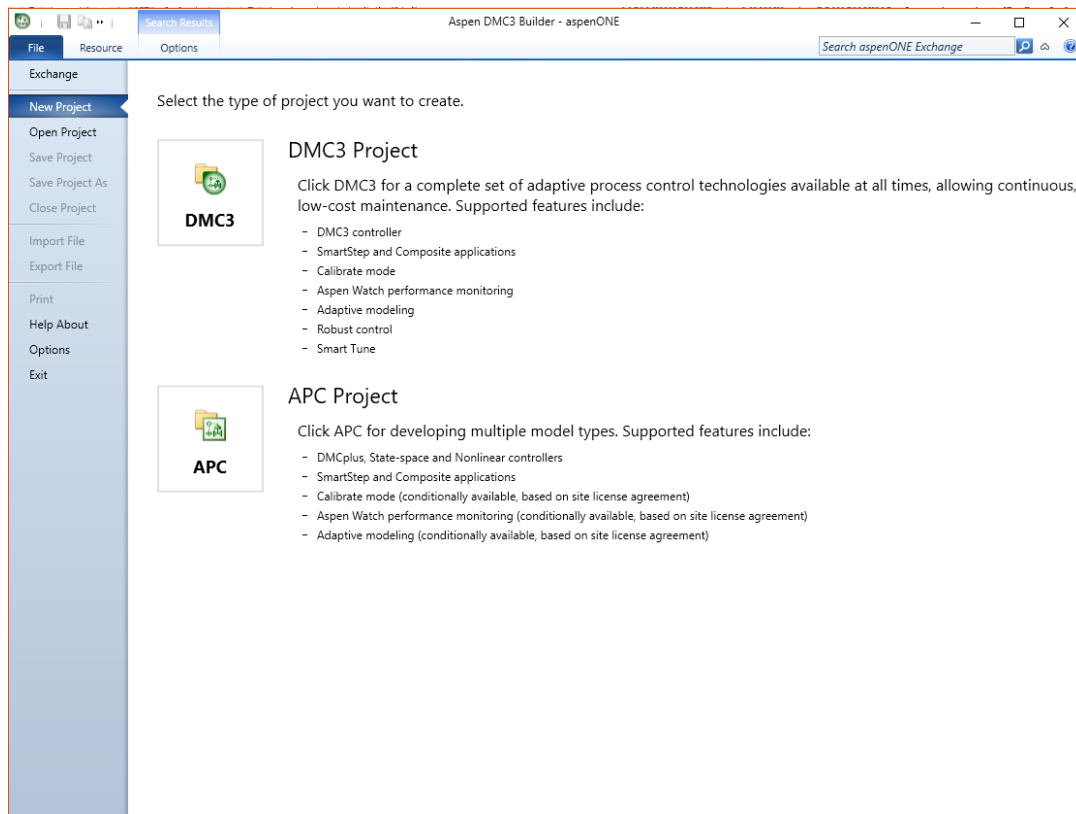


Figura 18 Menú de archivo en Aspen DMC3 Builder

Otro elemento importante en el menú Archivo es *Options*. Aquí se pueden revisar y ajustar las opciones para toda la aplicación. Haciendo clic en las opciones se mostrará el siguiente cuadro de diálogo, que se muestra en la Figura 9.

La Figura 9 muestra la vista de conjunto de datos seleccionada. Si se cambia el ancho de línea utilizado en los cuadros de datos de 2 a 1 píxel. Hay que hacer clic en Aplicar y luego OK para guardar los cambios.

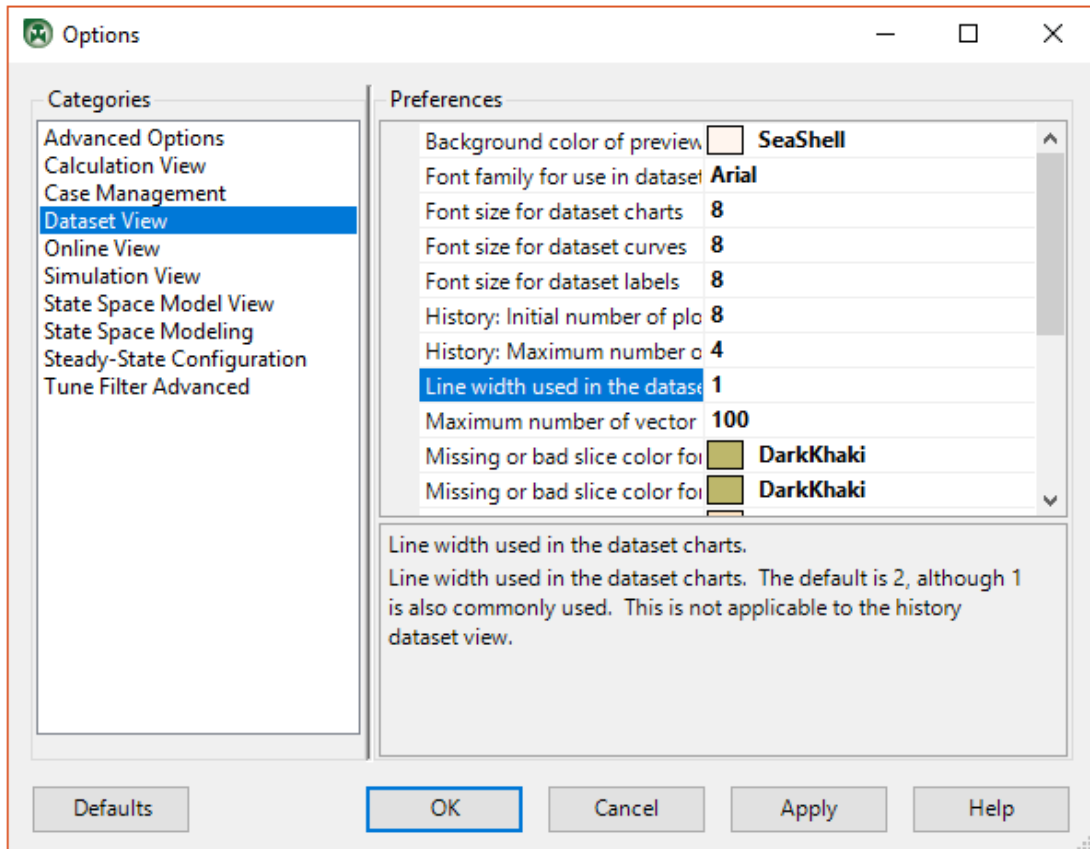


Figura 19 Ventana de opciones en Aspen DMC3 Builder

5.5.2. Creando un Nuevo Proyecto

Para crear un nuevo Proyecto hay que hacer clic en New desde la página de inicio y seleccionar el tipo de proyecto DMC3. Se abre el cuadro de diálogo Nuevo proyecto DMC3. Es preciso darle al proyecto un nombre y una ubicación.

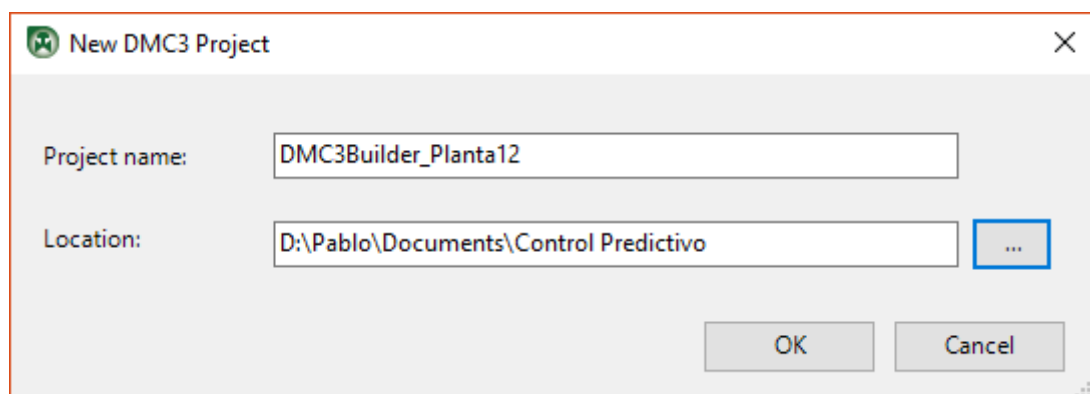


Figura 20 Asignar nombre y ubicación al Proyecto

Hay que hacer clic en Aceptar para crear el proyecto. Debería verse la vista *Datasets*, como se muestra en la Figura 11, después de hacer clic en Aceptar. Esta es la vista por defecto cuando se ha creado un nuevo Proyecto:

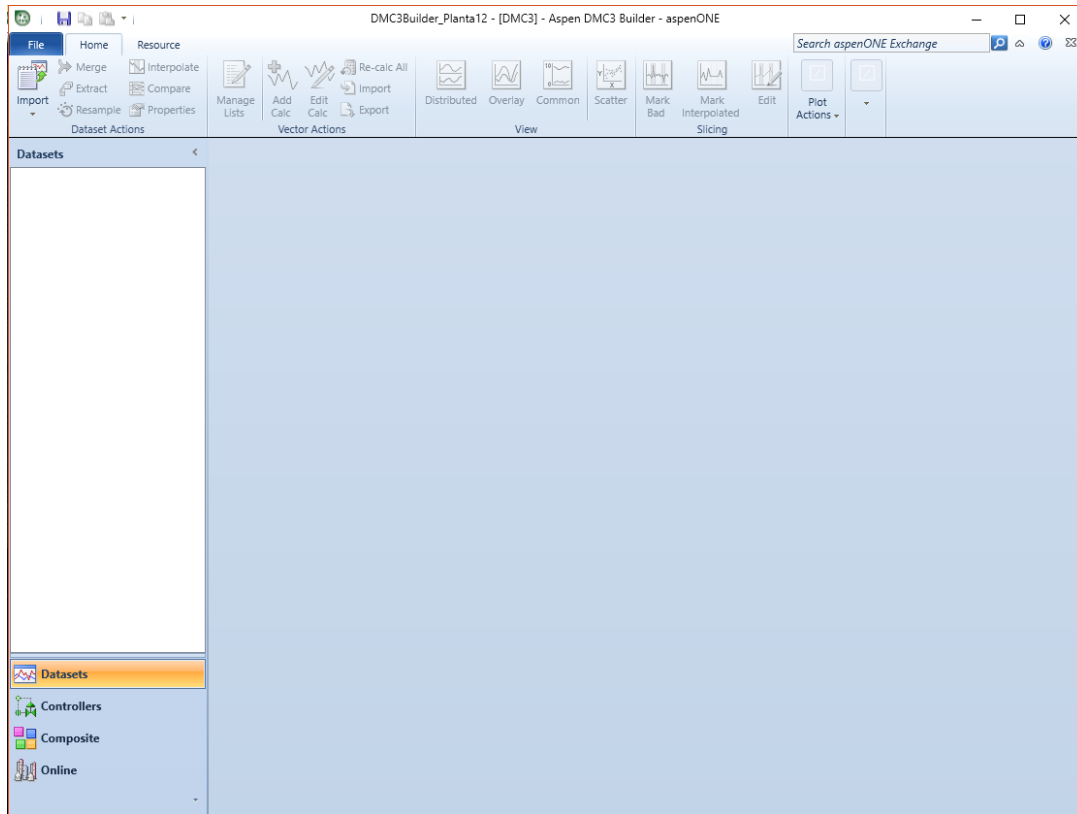


Figura 21 Vista por defecto para nuevo Proyecto

Se observa que el proyecto está vacío y que sólo el ícono *Import* está habilitado en la vista *Datasets*. Hay que hacer clic en *Import* | Elemento del menú *Datasets*, y a continuación, seleccionar el conjunto de datos que se desea utilizar. En este ejemplo, se está usando Exp080518. Esta acción importará el conjunto de datos seleccionado a la vista *Datasets* de DMC3 Builder, como se muestra a continuación:

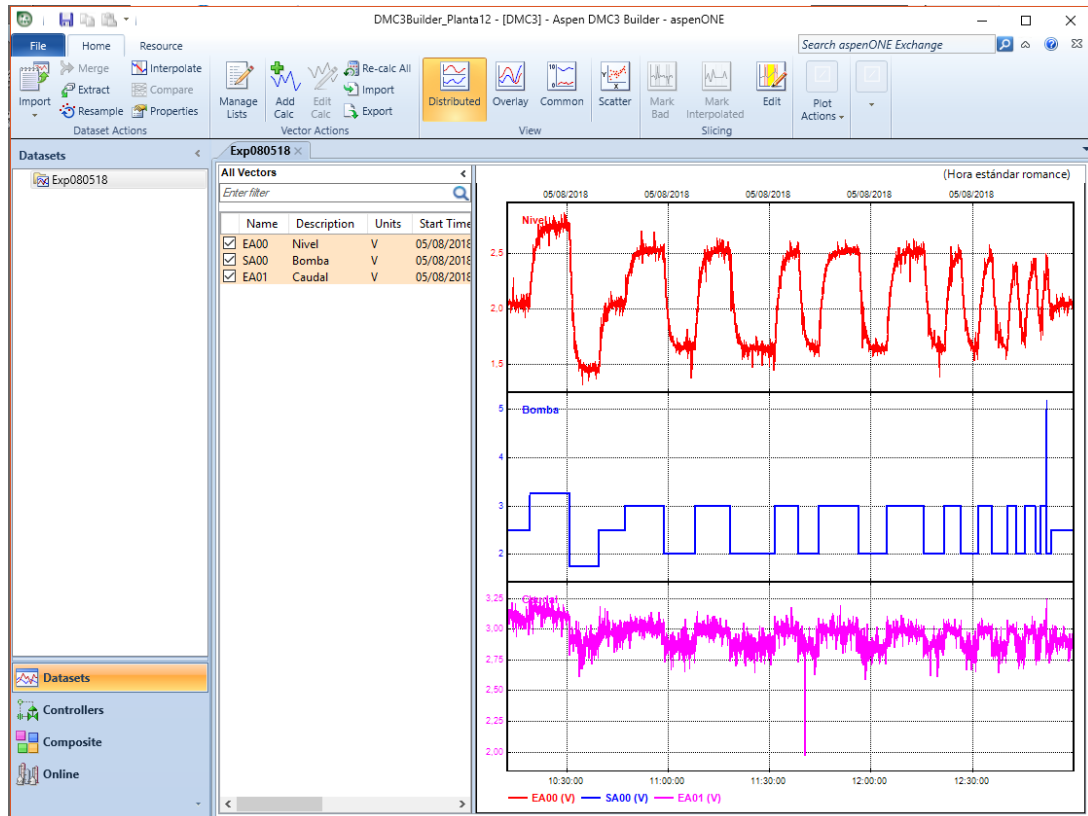


Figura 22 Vista Datasets de Aspen DMC3 Builder

Hay cuatro secciones principales en esta vista:

- Árbol de Proyecto o árbol de navegación: aquí se pueden tener conjuntos de datos, listas de vectores, controladores, casos, etc.
- Botones de navegación: estos botones se usan para navegar entre las diferentes partes del software.
- Vectores: localizados en la vista *Datasets*, solo para mostrar la lista de vectores en un conjunto de datos.
- *Plot Area*: es un área reservada para mostrar gráficas y tendencias.

Antes de seguir adelante, hay que repasar el árbol de navegación, los elementos de cinta y los flujos de trabajo en Aspen DMC3 Builder; esto será importante a medida que se avance. De forma predeterminada, debe haber cuatro botones disponibles en el área de botones de navegación: *Datasets*, *Controllers*, *Composite* y *Online*, como se muestra a continuación:

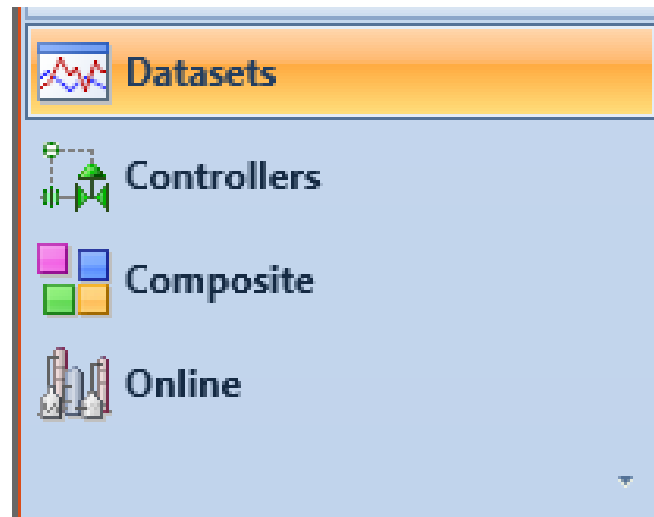


Figura 23 Botones de navegación

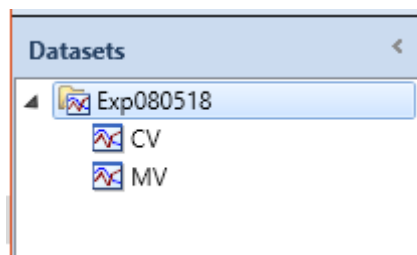


Figura 24 Dataset Exp080518

Desde que se selecciona *Datasets*, se puede ver la vista de árbol de navegación de conjuntos de datos. A la derecha, está el conjunto de datos Exp080518. Este conjunto de datos contiene dos listas de vectores, llamadas CV y MV.

Si se selecciona *Controllers*, se puede ver la Vista de árbol del panel de navegación Controladores. Al hacer clic derecho en diferentes áreas y objetos en la navegación de los controladores se abre un menú contextual específico para la vista de árbol de los controladores.

En las próximas dos secciones, se explicarán los flujos de trabajo de navegación y cinta.

Flujo de trabajo del árbol de navegación

La figura a continuación muestra el árbol de navegación para cada sección: *Datasets*, *Controllers*, *Composite* y *Online*. En la sección Conjuntos de datos, se verán en el nodo principal los conjuntos de datos y posiblemente las listas de vectores debajo de él. En la sección Controladores, el nodo principal es el controlador aplicación, y debajo de ella está el flujo de trabajo que comienza con el Modelo Maestro y finaliza con la Implementación. La sección Compuesto

muestra el flujo de trabajo relacionado con una aplicación compuesta (Tuning y Simulación). La sección en línea tiene dos elementos en el árbol de navegación:

Servidores (enumerando todos los servidores disponibles) y Aplicaciones (enumerando todas las aplicaciones en ejecución para los servidores conectados).

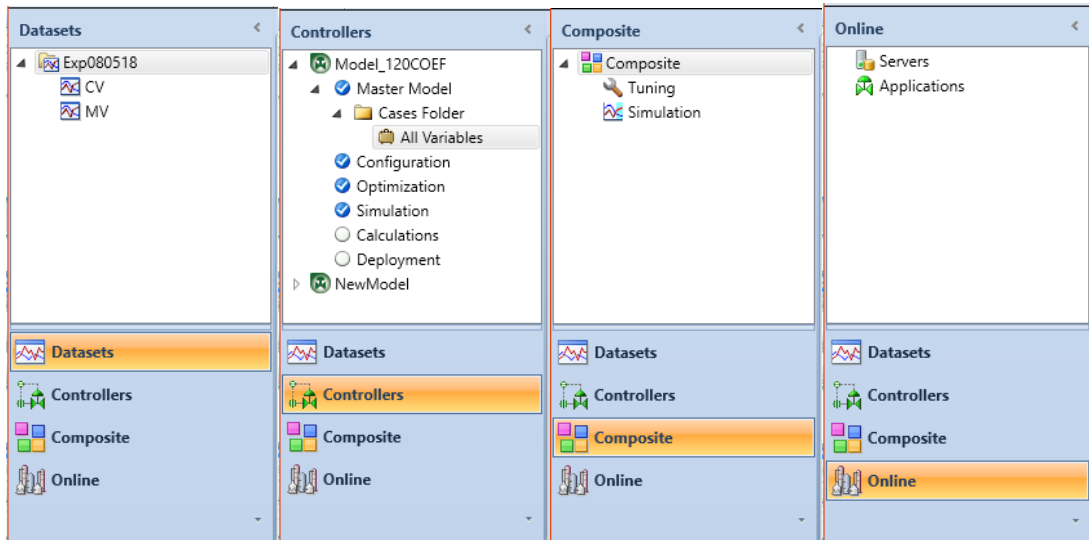


Figura 25 Árboles de navegación para cada una de las secciones: Datasets, Controllers, Composite y Online

En el panel de navegación Controladores, se pueden ver los pasos para crear los diversos componentes y etapas de las aplicaciones del controlador. Se puede expandir una aplicación y ver los seis pasos (también llamados nodos de flujo de trabajo) involucrados en la construcción de un controlador Aspen DMC3, como se muestra en la Figura 16.

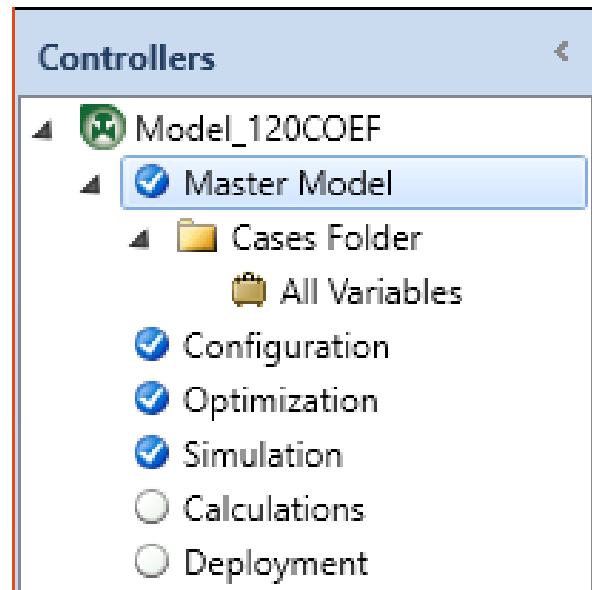


Figura 26 Pasos para construir el controlador en Aspen DMC3

Los nodos de flujo de trabajo bajo un controlador incluyen *Master Model* (para operaciones de modelo, casos e identificación del modelo), *Configuration* (para configurar filtros de retroalimentación y subcontroladores), *Optimization* (para configurar el optimizador, *Smart Tune*, objetivos externos, etc.), *Simulation* (para simular la aplicación), *Calculations* (para agregar cálculos de entrada y salida para el controlador) y *Deployment* (utilizado para configurar etiquetas IO e implementar la aplicación a un entorno en línea).

Los siguientes íconos aparecen en los nodos de nivel superior del árbol de navegación de Controladores. Indican el tipo de modelo de la aplicación.



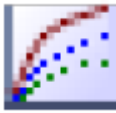
- Aplicaciones de tipo de modelo Aspen DMC3 (respuesta de impulso finito), que es el único tipo de aplicación disponible en Proyectos DMC3.





- Tipo de modelo Aspen DMCplus (respuesta de impulso finito), construido sólo dentro de los proyectos de APC.





- Tipo de modelo Aspen State-Space (entrada múltiple, salida múltiple), construido sólo dentro de los proyectos de APC.



- Tipo de modelo del controlador no lineal de Aspen (entrada múltiple, salida única), construido sólo dentro de los proyectos de APC (como ya se ha mencionado anteriormente, en la versión 10 de la suite, ya no está disponible esta opción).

Para cada etapa del desarrollo de la aplicación, ciertos íconos indican si las configuraciones y el ajuste no se iniciaron , se completaron , o si se

requieren configuraciones y ajustes adicionales . Si algún error está

asociado con un nodo, se muestra el siguiente ícono: 

5.5.3. Flujo de trabajo en el menú de cinta

Los elementos de la cinta están organizados para proporcionar un flujo de trabajo de izquierda a derecha. Por ejemplo, el primer paso es importar un nuevo conjunto de datos en un proyecto vacío, por lo que hasta que se complete ese paso, Importar es el único elemento disponible en la cinta de opciones.

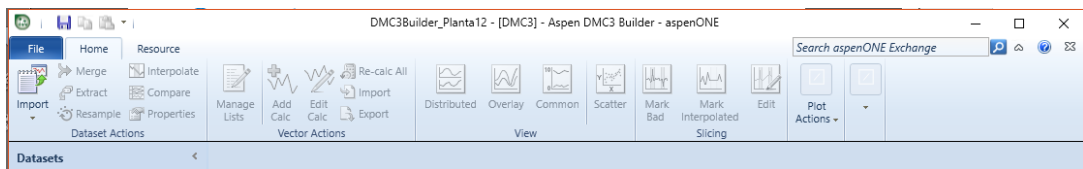


Figura 27 Los limitados elementos de la cinta de opciones indican que el próximo paso es importar los datos

Después de hacer clic en Importar y seleccionar un conjunto de datos, los elementos de cinta adicionales estarán disponibles.

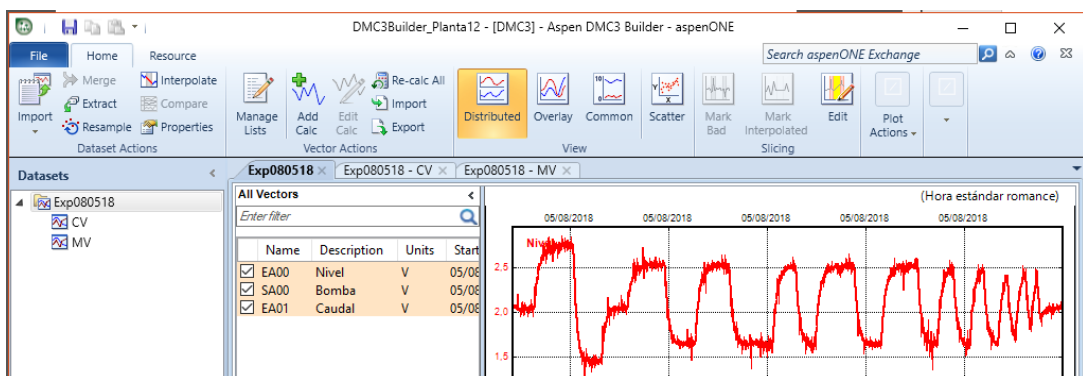


Figura 28 Tras importar los datos, aparece el resto de opciones

Otro aspecto importante son los grupos de cintas. Por ejemplo, en el lado izquierdo se encuentra el grupo de cinta llamado *Dataset Actions*, donde toda la funcionalidad relacionada con *datasets* está organizada.

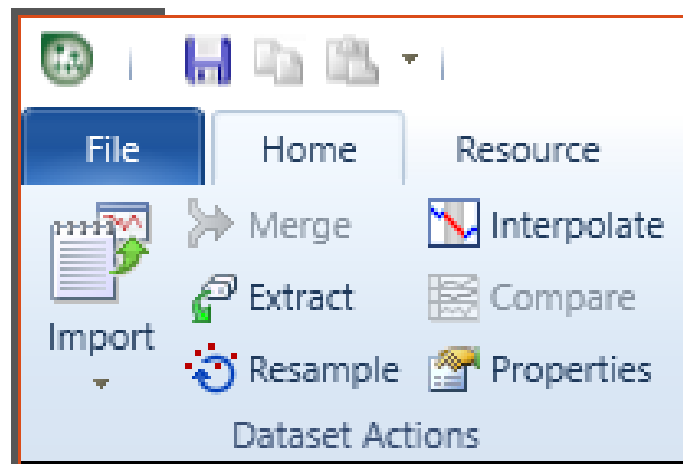


Figura 29 Funciones y acciones relacionadas, organizadas en grupos

Se entrará en más detalles sobre el flujo de trabajo de conjuntos de datos y cómo realizar algunos básicos pasos antes de usar el comando Crear modelo. Esto creará una estructura de modelo de esqueleto basado en el conjunto de datos y lo dirigirá a la sección de controladores para el desarrollo del modelo y configuración de la aplicación.

5.6. Datasets

Después de crear un nuevo proyecto, la aplicación se dirige a la vista *Datasets*. Esta vista permite ver gráficamente las tendencias trazadas de datos para vectores (o variables) que se incluyen en un solo conjunto de datos. Otras características en la vista de conjuntos de datos permiten lograr lo siguiente:

- Agregar vectores calculados al conjunto de datos actual.
- Agregar y administrar sectores de datos en el conjunto de datos.
- Modificar el conjunto de datos y las propiedades del vector, como eliminar o renombrar vectores.
- Acceder al cuadro de diálogo Editar datos de vectores para modificar los valores de datos reales para un vector seleccionado.

La Figura 20 a continuación muestra la vista de *Datasets* en Aspen DMC3 Builder con tres vectores que se trazan en el modo Distribuido. Este es el comportamiento predeterminado después de importar un nuevo conjunto de datos.

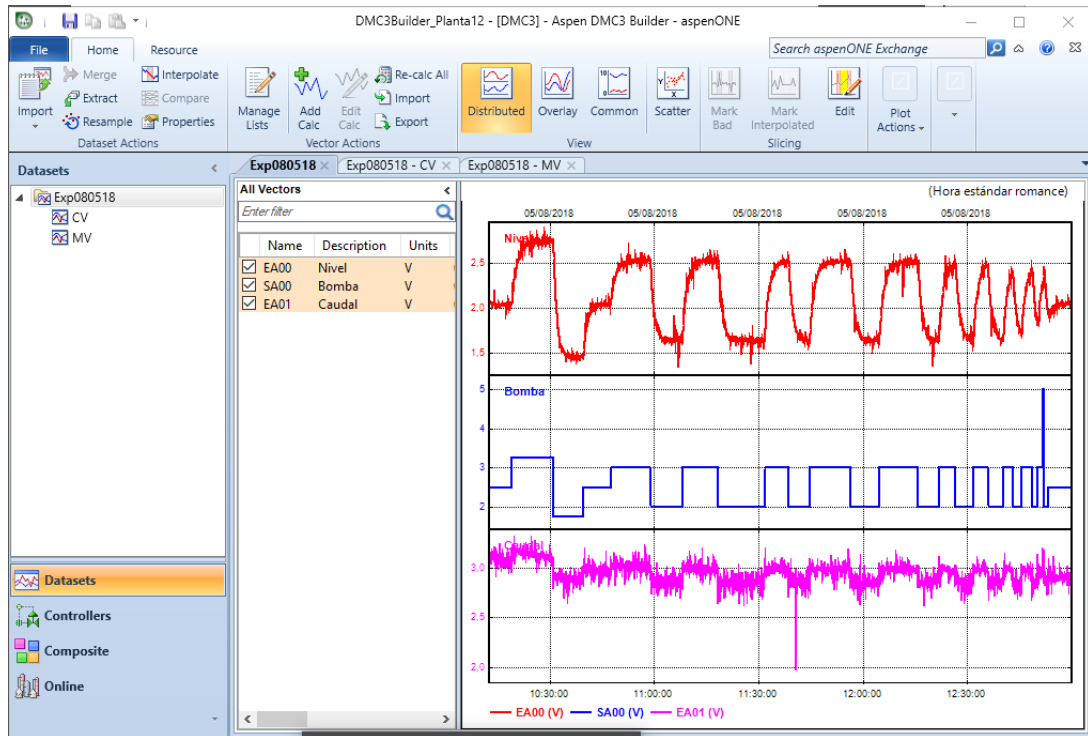


Figura 30 Vista Datasets en Aspen DMC3 Builder

Si se observan las funcionalidades disponibles en la vista Conjuntos de datos, después se podrá construir un modelo de esqueleto y un controlador.

5.6.1. Vista *Datasets*

El grupo de cinta *View* incluye diferentes opciones para trazar y mostrar vectores, incluidos *Distributed*, *Overlay* y *Common*.

5.6.2. *Distributed*

El botón *Distributed* se seleccionará automáticamente en la sección *View* de la cinta con los primeros tres vectores seleccionados (por defecto) para la visualización de diagramas de tendencia para visualizar un conjunto de datos por primera vez. Usando la vista distribuida, se puede ver cada vector mostrado en su propia trama. En la figura siguiente, se muestran EA00 y EA01, los vectores de las variables controladas.

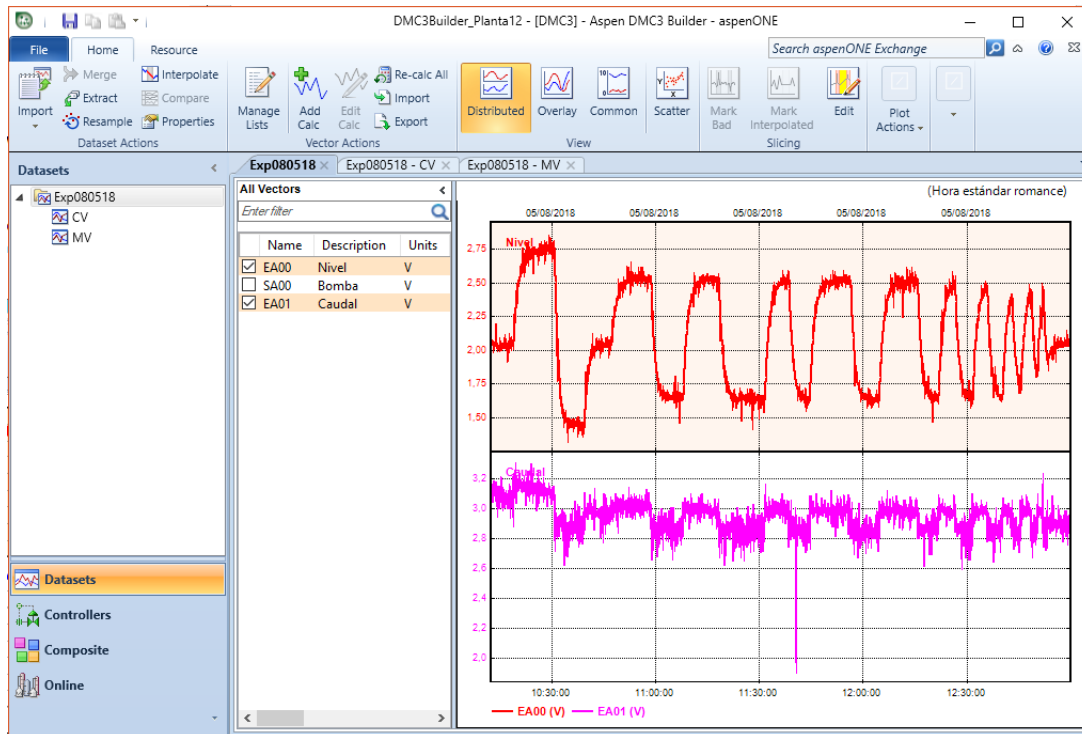


Figura 31 Los vectores en una gráfica con vista distribuida

Como se puede observar, el EA00 (Nivel) tiene un color de fondo diferente. Esto se usa para indicar qué vector está seleccionado. Se puede cambiar el ancho de línea y el color de fondo de la trama seleccionada accediendo a Opciones en el menú File.

5.6.3. Overlay

La siguiente vista es la superposición, donde se puede ver cada vector en una sola gráfica, con diferentes escalas para cada vector.

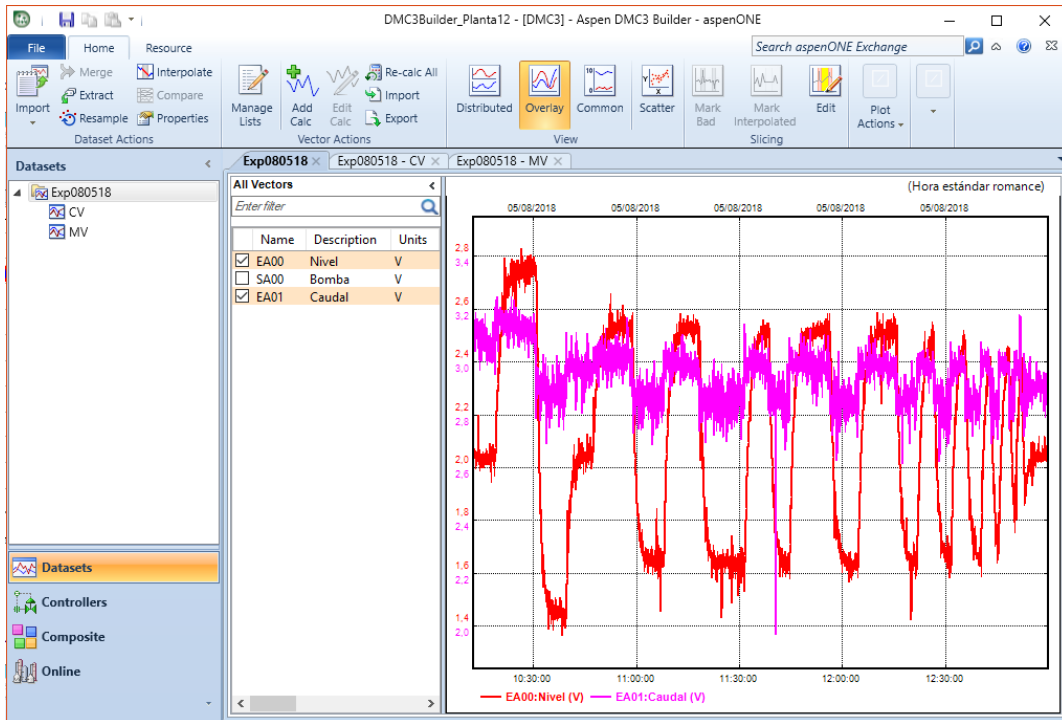


Figura 32 Cada vector en una sola gráfica, con diferentes escalas, en la vista Overlay

5.6.4. Common

Aquí se puede ver cada vector mostrado en un solo gráfico con la misma escala:

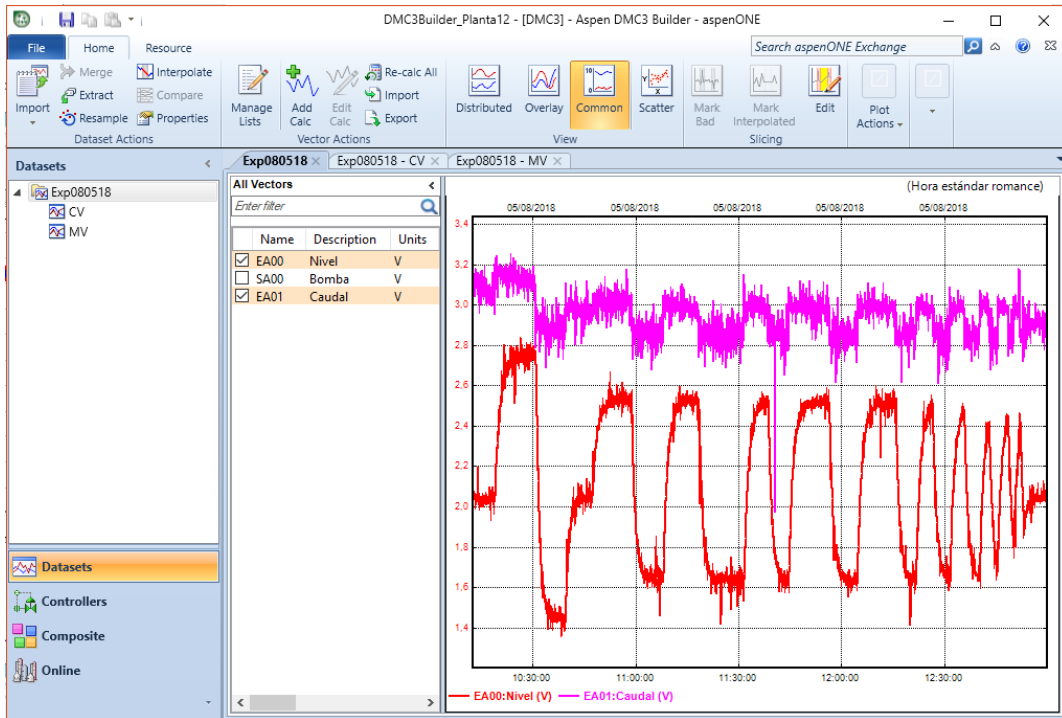


Figura 33 Vectores en una sola gráfica, con la misma escala, usando la vista Common

5.6.5. Manage Lists

Usar el cuadro de diálogo *Manage Lists* para crear conjuntos de vectores personalizados que se pueden visualizar en la vista *Datasets*. Después de crear una lista de vectores, se almacena dentro del conjunto de datos y queda disponible para su selección en el Árbol de navegación de conjuntos de datos. En el cuadro de diálogo a continuación, se observa que el título es *Manage Vector Lists - Exp080518*, por lo tanto, se refiere a los vectores en el conjunto de datos Exp080518.

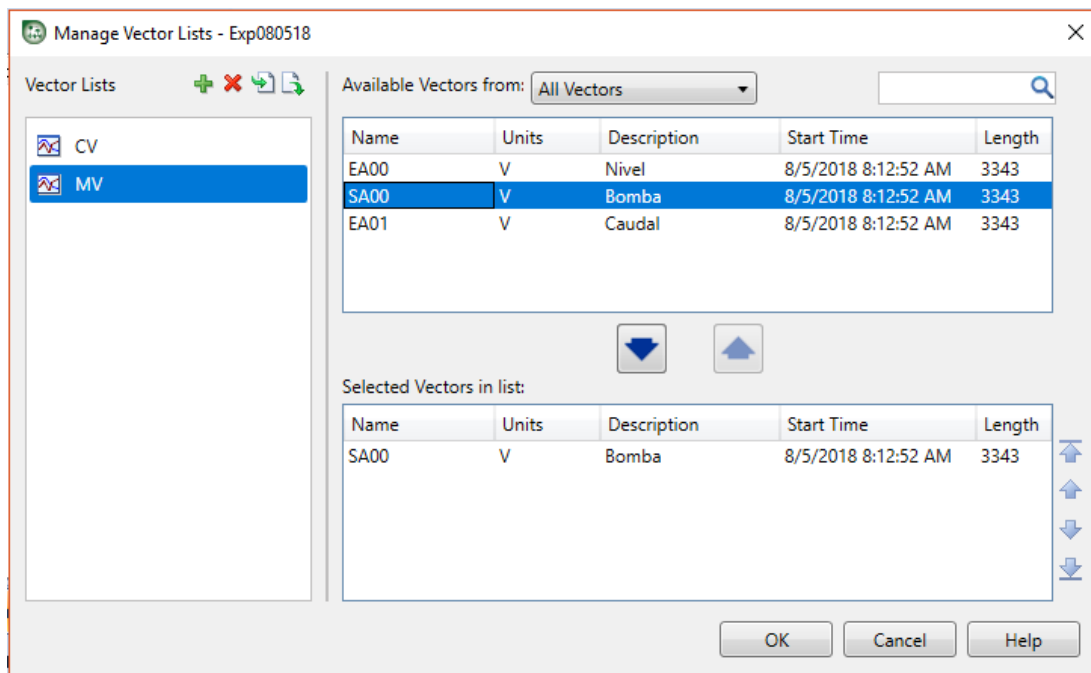


Figura 34 Administrar listas de vectores

En el lado izquierdo de este cuadro de diálogo, se pueden ver las listas de vectores en el conjunto de datos seleccionado y se pueden agregar, eliminar, importar y exportar listas de vectores. Al crear una nueva lista de vectores, se pueden seleccionar y agregar vectores de *All Vectors* o diferentes listas de vectores asociadas con el conjunto de datos seleccionado. Por ejemplo, si se hace clic en el icono + para agregar una nueva lista de vectores y se cambia el nombre a OP, como se muestra en la figura anterior.

Después de eso, se escribe OP en la barra de búsqueda en la parte superior derecha para ver todas las etiquetas que contienen el carácter OP (el filtro incluye nombre, unidades y descripción). Hay que seleccionar algunas etiquetas con la extensión .OP y hacer clic en la flecha hacia abajo para agregarlas a la lista de OP.

5.6.6. Slicing

La sección de corte de la cinta incluye operaciones de corte de datos. Aquí se puede marcar los malos datos que deberían excluirse de la identificación del modelo. En esta sección también se puede marcar un conjunto de puntos de datos que deben ser interpolados. El botón Editar proporciona acceso al cuadro de diálogo *Slices*. Todas estas tres funciones están cubiertas en las siguientes secciones.

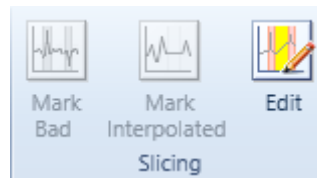


Figura 35 Botón de edición en la opción Slicing

5.6.7. Mark Bad

Los vectores en el conjunto de datos tienen datos que están fuera del rango que no se quieren usar en la identificación del modelo. Se seleccionan las etiquetas EA00, EA01 y SA00 y la tendencia en una escala común. Hay algunos puntos indicados en la figura a continuación que deben ser marcados como un mal sector global.

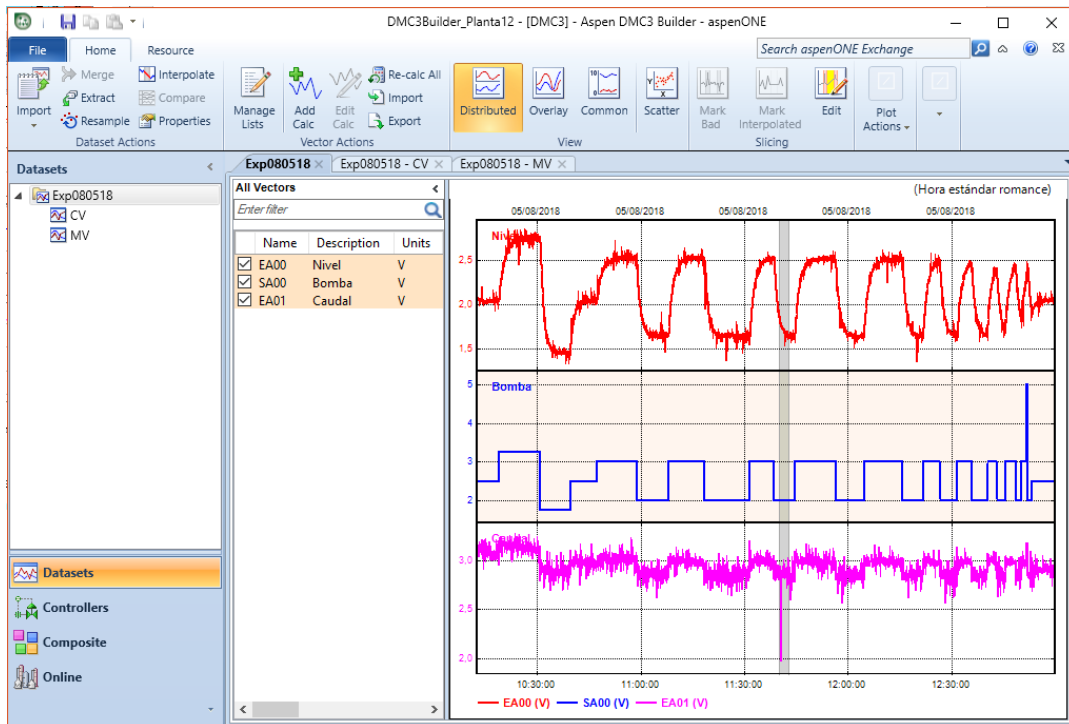


Figura 36 Marcar puntos como Bad Global Slice

Hay que seleccionar un sector con datos incorrectos, ampliar y hacer clic en el botón *Mark as bad* para marcar el intervalo de tiempo seleccionado como inutilizable para la identificación. Además, se ha de seleccionar la primera opción: *Global Slice*, como se muestra en la Figura 27. Esta opción ampliará esta porción incorrecta a todo el conjunto de datos.

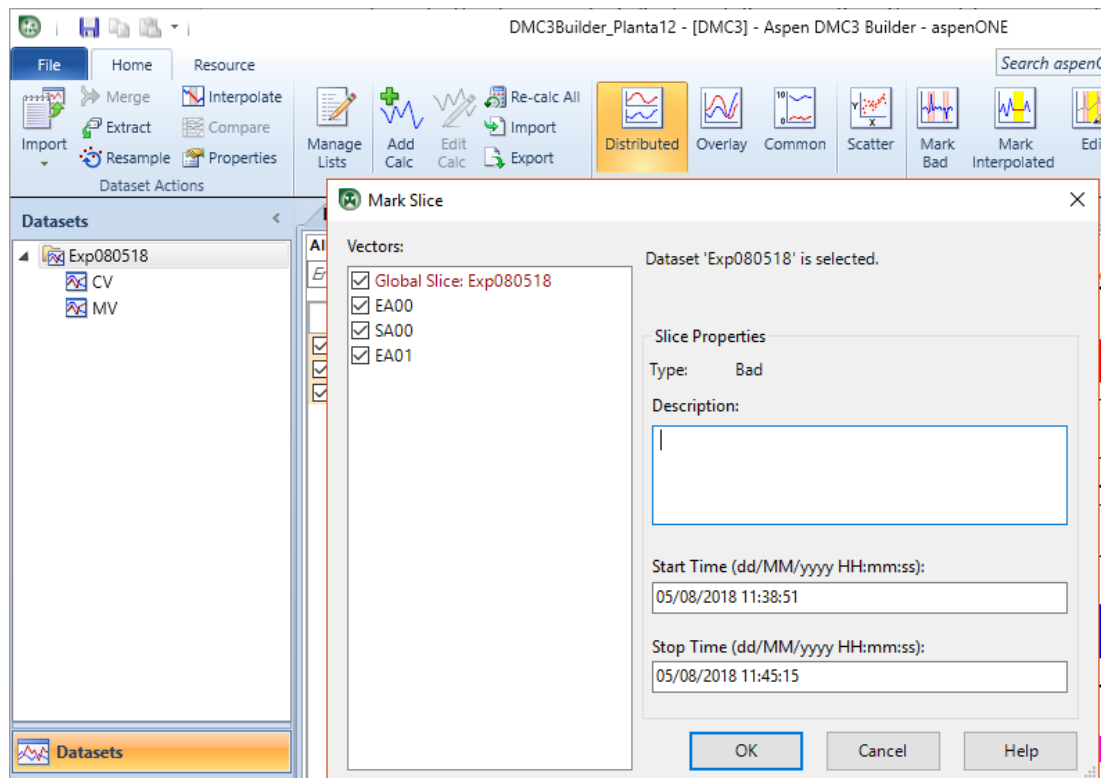


Figura 27 Seleccionar Global Slice para todo el conjunto de datos

5.6.8. Mark Interpolated

Si se selecciona la etiqueta SA00 y luego se selecciona el intervalo de tiempo de interés (Figura 28) y se hace clic en *Mark Interpolated*, se crea una interpolación desde el inicio del rango de tiempo seleccionado hasta el final.

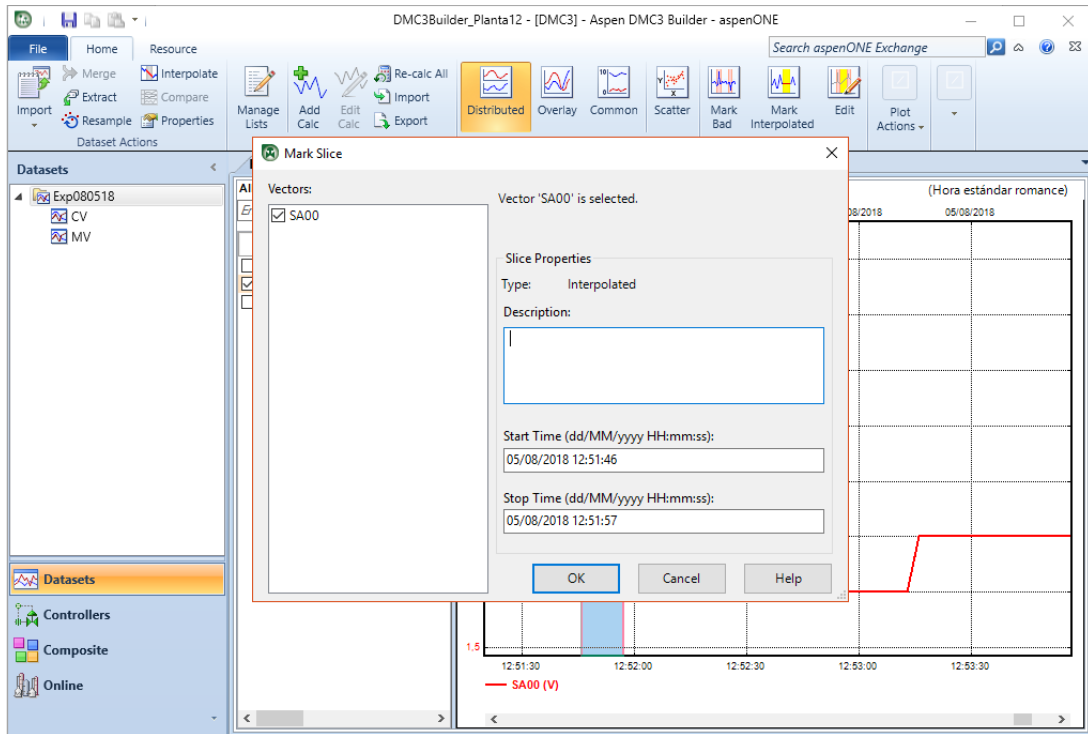


Figura 28 Crear una interpolación lineal

Haga clic en OK y aparecerá un área amarilla donde tuvo lugar la interpolación, como se muestra a continuación:

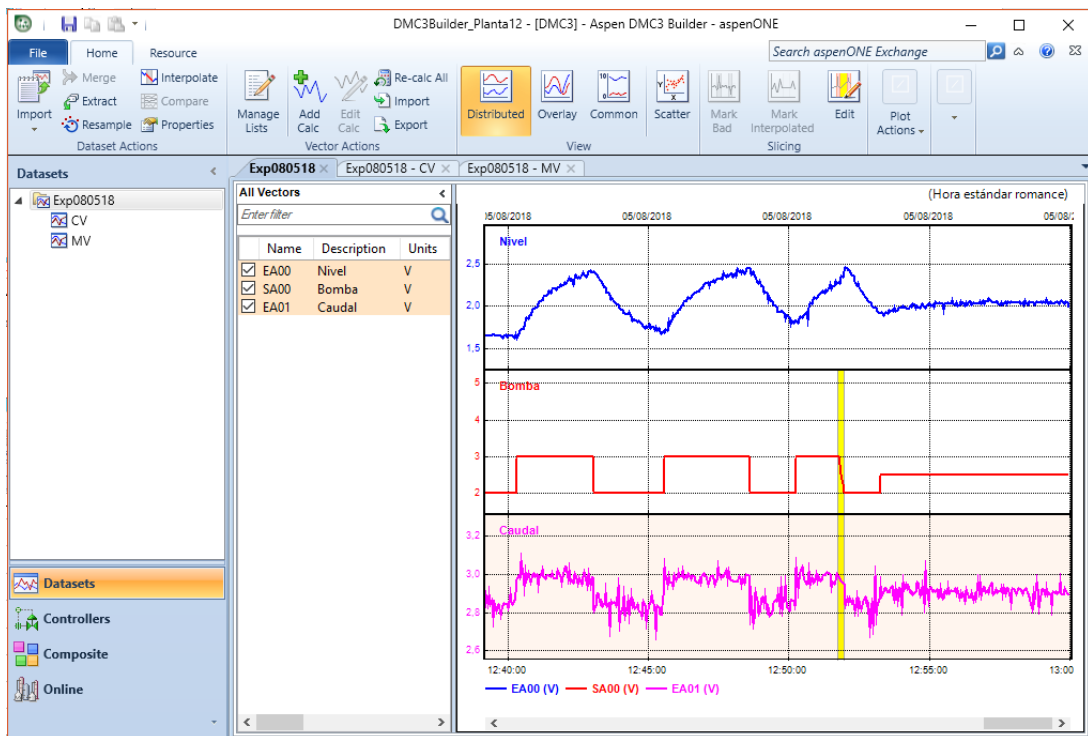


Figura 29 Ver dónde tuvo lugar la interpolación

5.6.9. *Edit (Slices)*

Se puede usar el cuadro de diálogo Editar sectores para ver los tipos de segmentos que existen, globalmente o para vectores específicos, con la opción de eliminar un segmento existente del conjunto de datos actual. También es posible agregar un nuevo segmento de datos incorrectos o un segmento interpolado al ingresar la fecha y hora de los límites inicial y final de la porción o selección gráfica del rango de tiempo de la nueva porción en el área de la gráfica.

5.7. *Controllers*

5.7.1. *Create Model*

Esta parte del proceso de diseño se expone en este apartado en virtud del usuario que no había realizado este paso con anterioridad, pero puede obviarse si se ha realizado el modelado en el Aspen DMC3 Model, explicado en el capítulo 3 de este documento. Con lo que simplemente se ha de hacer clic en *Import* y seleccionar el modelo que se desea importar.

Se recomienda crear el modelo usando el Aspen DMC3 Model, con una interfaz más intuitiva a la hora de realizar todos los pasos necesarios por orden, además de una cinta de opciones más sencilla a la par que instructiva con el fin de que el usuario tenga un mayor conocimiento sobre el proceso y su modelado. Una vez hecho esto, es recomendable refinar el modelo en el Aspen DMC3 Builder.

Si se quiere crear un modelo, después de revisar los datos y marcar segmentos defectuosos, crear listas, etc., el paso final del flujo de trabajo de la Vista de conjuntos de datos es crear un Nuevo modelo haciendo clic en Crear modelo. Esto abrirá el cuadro de diálogo Identificar modelo - Especificar estructura, donde deberá especificar al menos una entrada y una salida, período de muestra y tiempo para estado estacionario.

Al hacer clic en Aceptar llevará a la Vista de Controladores, que muestra la estructura del modelo, configurada siguiendo los pasos a continuación. Hay que ponerle nombre al controlador 2TNQSDMC3 y establecer el tiempo en *Steady State* en 6 minutos. Después de eso, se selecciona la lista llamada MV, y luego hay que verificar todos los vectores en esa lista como entradas, como se muestra en la figura a continuación:

Identify Model - Specify Structure

Properties

Name:

Sample Period: Time to Steady State:

Empty model Select variables from a dataset

Variables

Dataset:

List: Filter:

	Name	Tag Name	Input	Output	Units	Description
▶	SA00	SA00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	V	Bomba

Buttons: Add, Delete, Move Up, Move Down, Check, Uncheck

Buttons: OK, Cancel, Help

Figura 30 Identificar el modelo, marcar los components de la lista MV como entradas

Se ha de cambiar la selección de lista de MV a CV, y verificar todos los vectores en esa lista como salidas, como se muestra a continuación:

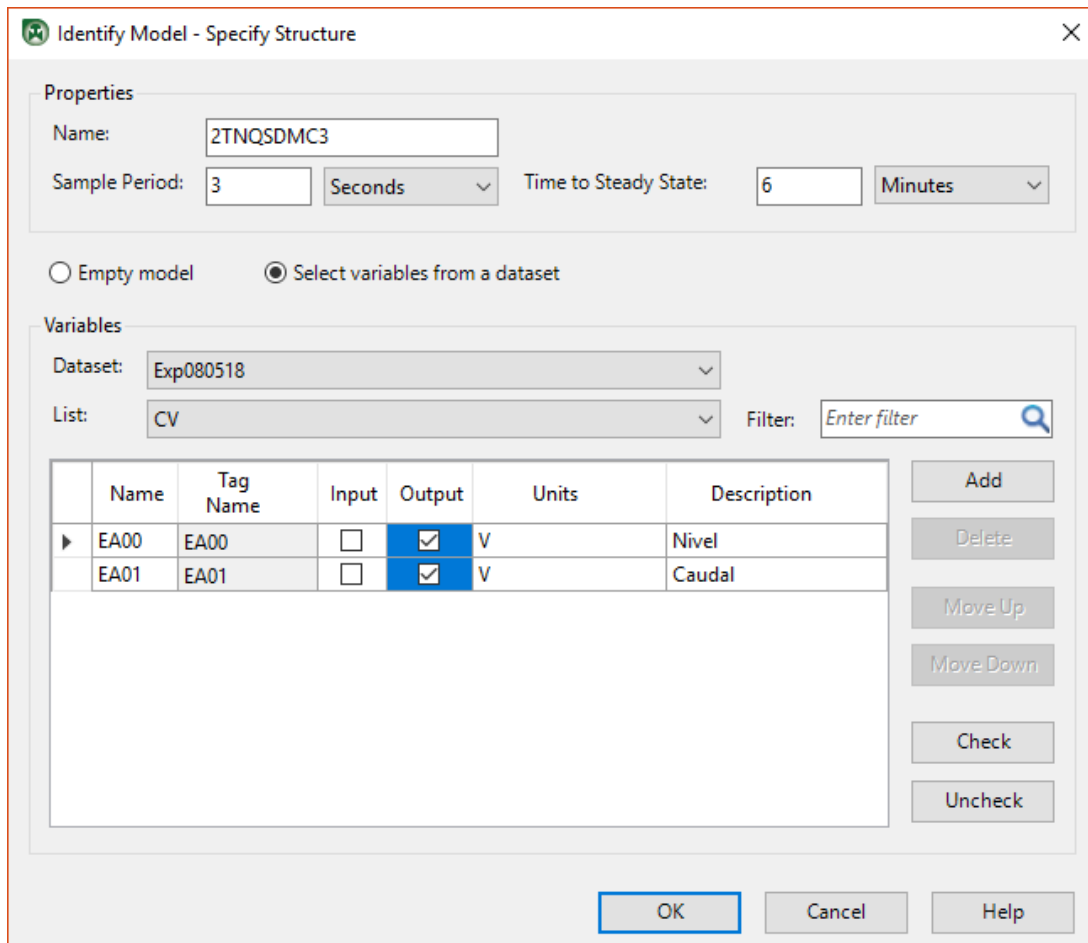


Figura 31 Identificar el modelo, marcar los componentes de la lista CV como salidas

Hay que hacer clic en Aceptar para crear el modelo de esqueleto para este controlador. Se debería ver un modelo en blanco. La vista Controladores ahora está seleccionada en el panel de navegación e incluye los pasos del flujo de trabajo para comenzar a construir este controlador.

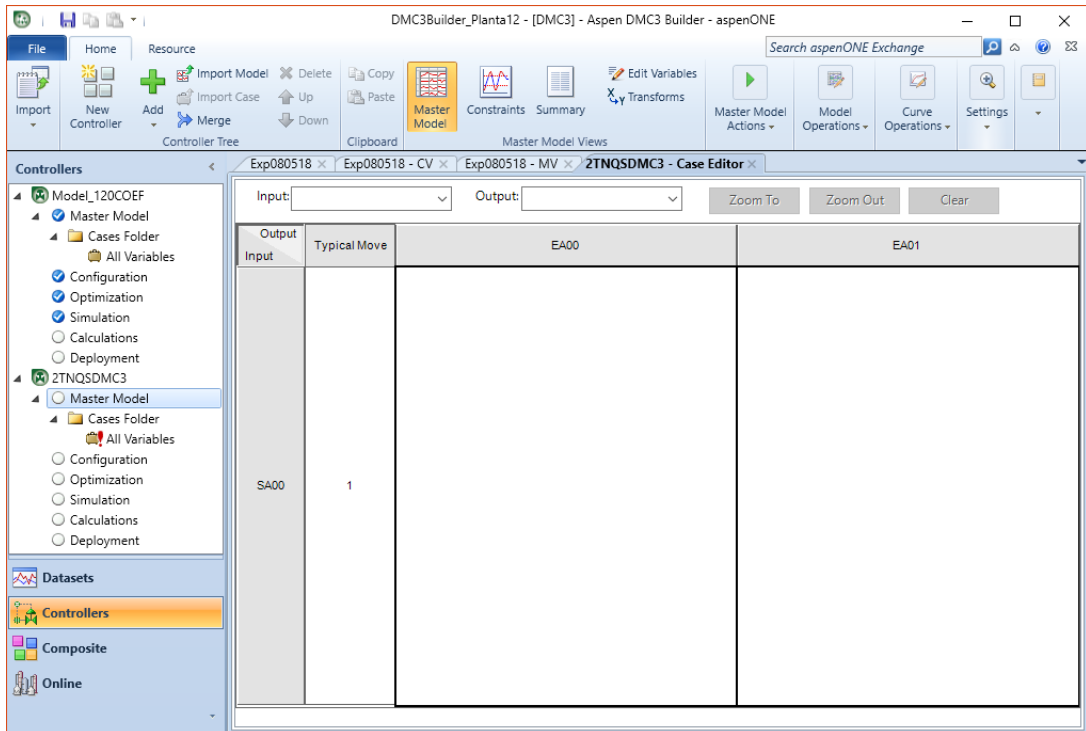


Figura 32 Modelo en blanco

5.7.2. Concepto de *Master Model*

Antes de comenzar a construir el controlador Aspen DMC3, se ha de repasar el concepto del modelo maestro. A diferencia del Aspen DMCplus Model, Aspen DMC3 Builder combina el modelo de controlador con la aplicación. El modelado ocurre bajo el nodo Modelo Maestro. La figura a continuación explica este concepto en el que las curvas del modelo de los casos de identificación se transfieren al modelo maestro.

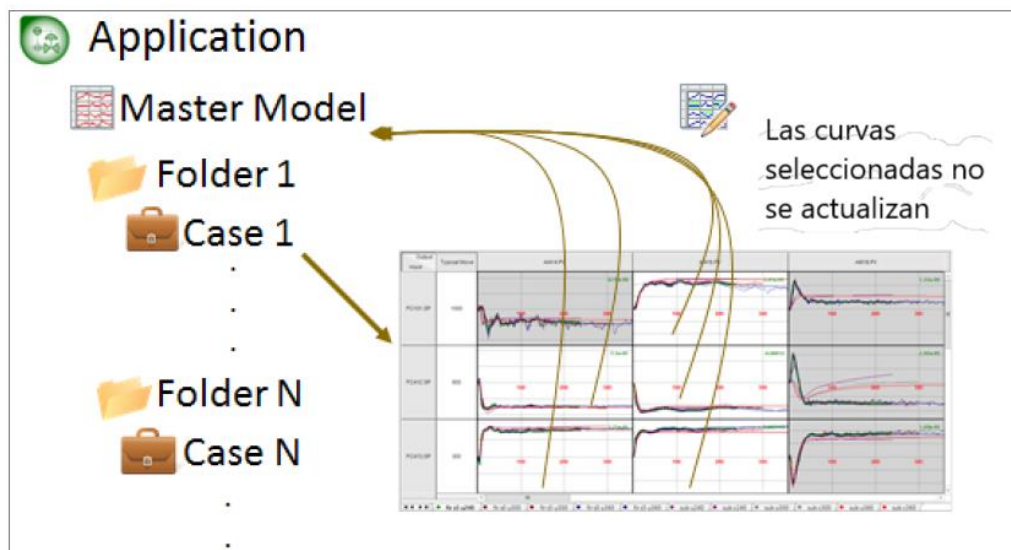


Figura 33 Las curvas del modelo son transferidas al Master Model

La vista *Master Model* para el modelo maestro de la aplicación muestra una matriz de curvas de respuesta escalón para cada relación entrada / salida de la variable en el modelo completo de la aplicación. Los comandos en el menú de la cinta (y en el menú contextual, menús de acceso directo) permite realizar varias actualizaciones de modelos y tareas de análisis. La vista *Constraints* para el modelo maestro de la aplicación habilita definir las restricciones de ganancia para el modelo maestro. La vista de resumen para el modelo maestro de la aplicación muestra lo siguiente:

- *Model summary*: informe del estado de todos casos de los modelos en el proyecto de identificación del modelo actual, con información organizada de acuerdo con las carpetas de casos que contienen los casos de identificación del modelo.
- *Configuration summary*: informe de cómo el filtro de retroalimentación, los subcontroladores y los grupos de prueba están configurados para la aplicación de controlador seleccionada, y si la aplicación es una participante en una aplicación *Composite*.

Después de crear una nueva aplicación, se muestra la matriz del modelo. Si se hace clic en el nodo de la aplicación, se puede ver el resumen para esa aplicación.

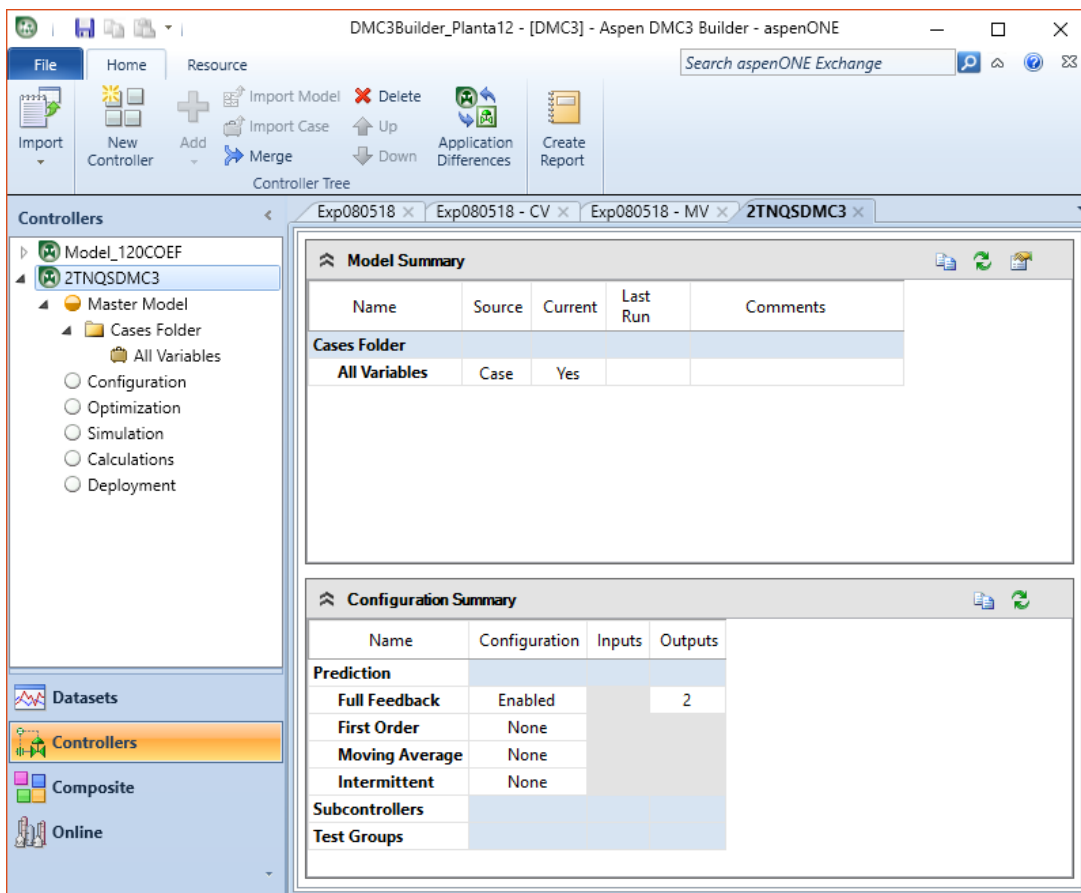


Figura 34 Model Summary

5.7.3. Vista *Master Model*

El nodo del modelo maestro mostrará el modelo ensamblado utilizado por el controlador en línea. Usando el árbol de navegación, hay que hacer clic en el nodo *Master Model*.

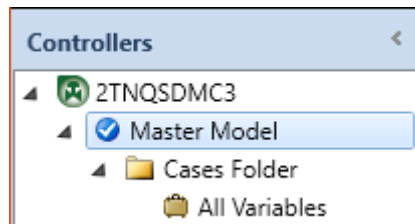


Figura 35 Master model node

Se ha de hacer clic en la vista *Master Model*. Hay varias funciones asociadas con la vista de modelo maestro (resaltada por el rectángulo rojo en la figura a continuación).

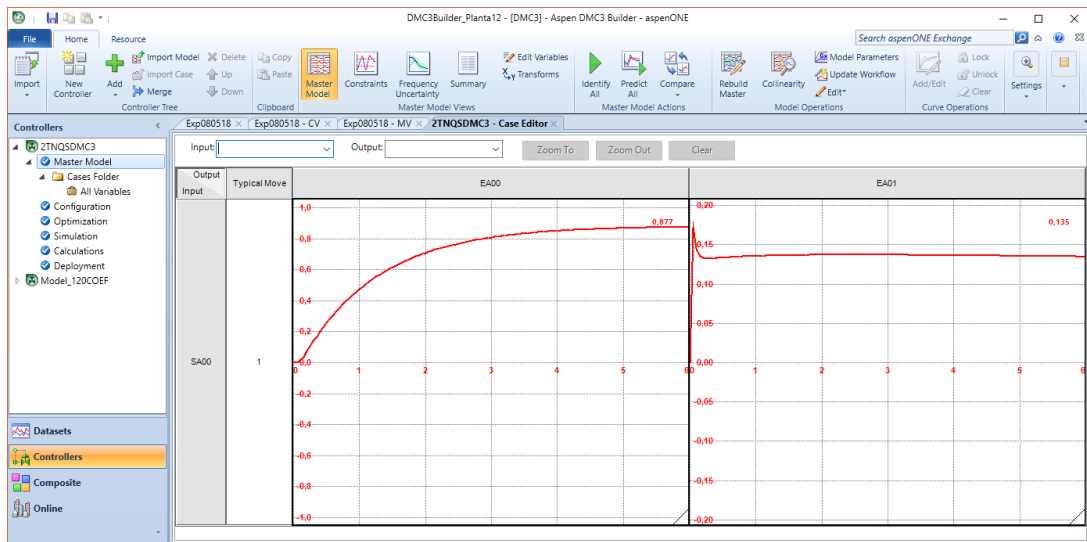


Figura 36 Vista Master Model

En la siguiente sección, se revisarán los pasos restantes del flujo de trabajo para compilar una aplicación en Aspen DMC3 Builder.

5.7.4. Árbol Controller

En este grupo de cinta, se tienen todas las acciones que tienen lugar en el Árbol de navegación. Este grupo también está disponible en otras vistas. La figura 37 a continuación muestra el grupo de cinta del Árbol del Controlador.

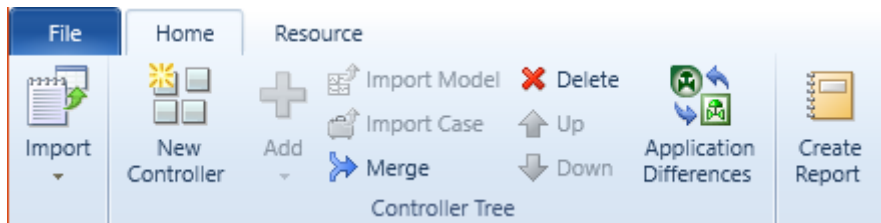


Figura 37 Grupo de opciones de la cinta Controller

5.7.5. Vista de los casos

La vista *Cases* se presenta cada vez que se hace clic en un caso. Los elementos de la cinta cambiarán en consecuencia y mostrarán toda la funcionalidad relacionada con la identificación del modelo.

La vista de casos se muestra en la figura a continuación. El caso *All Variables* se crea automáticamente después de especificar las entradas y salidas de la matriz modelo. La vista de *Variables* para un caso de ID de modelo permite designar qué conjunto de datos y vectores se desea usar para la identificación del modelo en el caso actual.

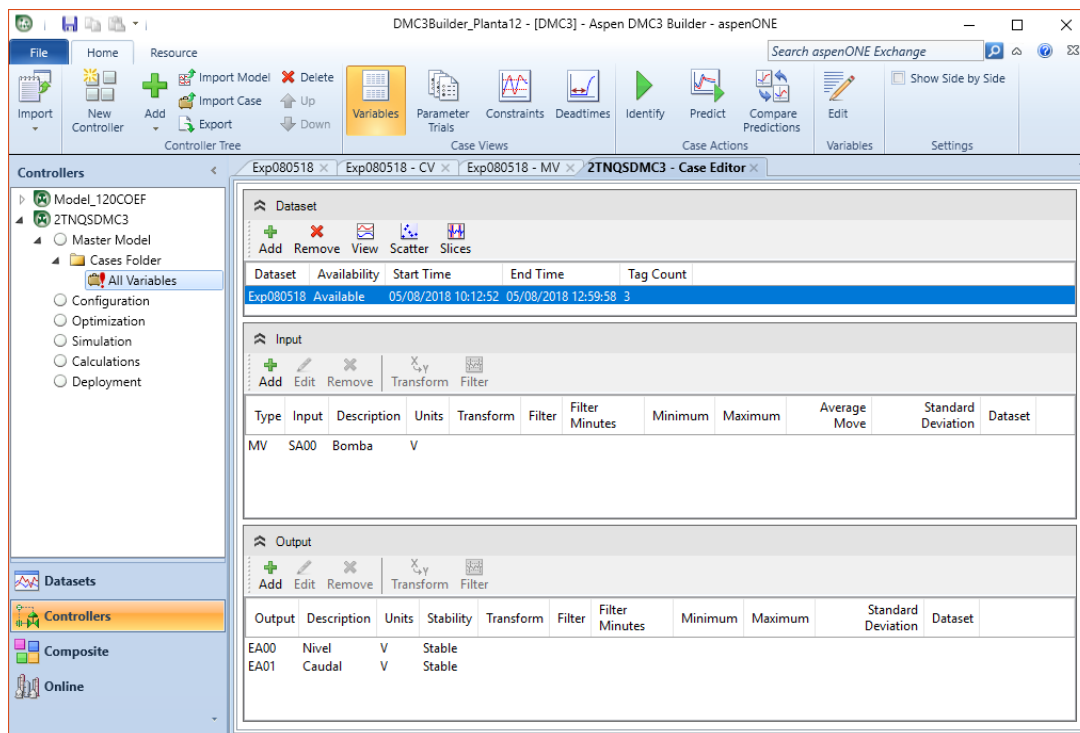


Figura 38 All Variables en la vista Cases

En la vista *Variables*, se tienen tres tablas, conjuntos de datos, entradas y salidas para especificar los conjuntos de datos que se utilizarán para la identificación del modelo en el caso, variables de entrada y variables de salida que se desea incluir en el modelo de caso, respectivamente. Se ha de seleccionar el conjunto de datos y hacer clic en el ícono *Scatter* para abrir el cuadro de diálogo Diagrama de dispersión. Hay que usar este cuadro de diálogo para ver un diagrama de dispersión para los vectores seleccionados en el conjunto de datos y luego crear transformaciones.

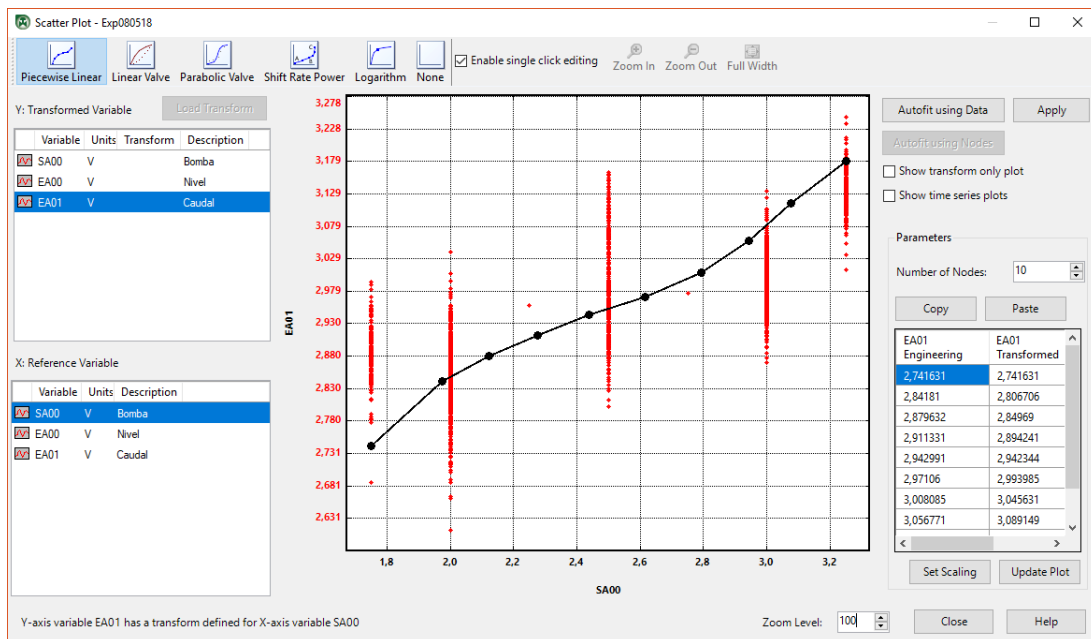


Figura 39 Scatter Plot

En la vista *Parameter Trials*, se pueden configurar uno o varios ensayos de identificación de modelos, con cada prueba que consiste en un conjunto de parámetros para definir cómo debe ocurrir una identificación de modelo para el caso actual. En la vista Pruebas de parámetros, se puede seleccionar la casilla maestra de una prueba. Esto designa la prueba como el resultado de identificación del modelo a usar para transferir las curvas del modelo de caso seleccionado al modelo maestro del proyecto.

La vista de *Constraints* para un caso de ID de modelo permite especificar restricciones de ganancia para el modelo del caso. La vista *Models* para una ID de modelo muestra una matriz de curvas de respuesta escalón para cada relación variable de entrada / salida en el modelo. Además, desde la vista ID de modelo, se pueden realizar varias tareas para analizar y editar propiedades y curvas del modelo del caso, y actualizar el modelo maestro de la solicitud.

La vista de *Deadtimes* para un caso de ID de modelo permite especificar valores de tiempo muerto para pares de variables de modelo, que se consideran durante el proceso de identificación del modelo.

Hay que hacer clic en *Identify* para ejecutar la identificación del modelo para el caso *All Variables*. Se muestra la identificación FIR del modelo, como se muestra en la siguiente figura:

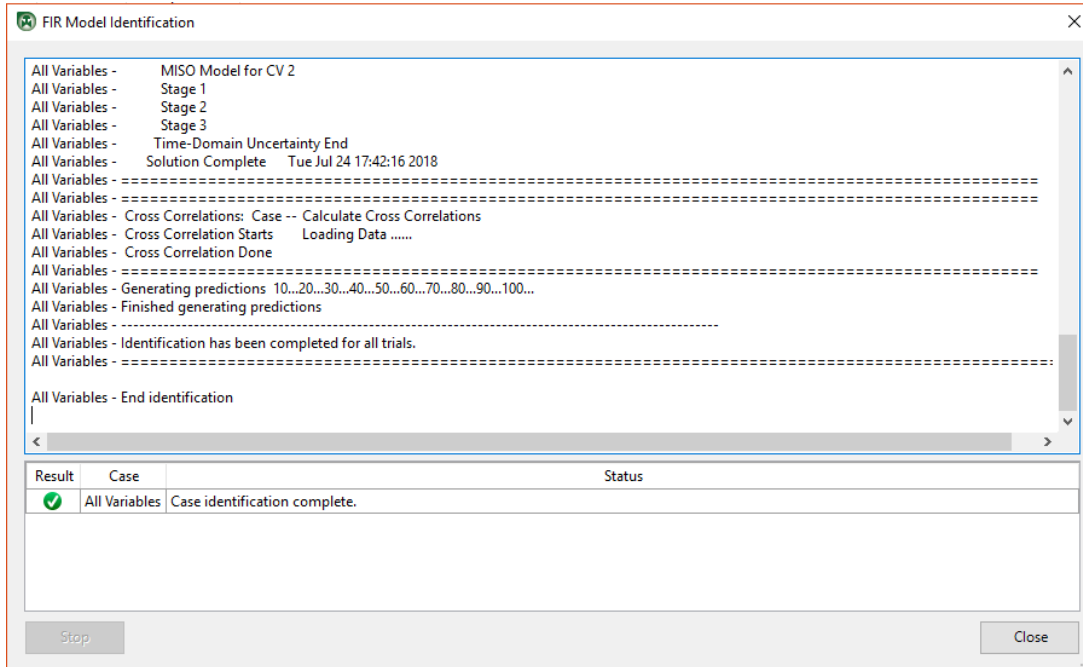


Figura 40 Identificación FIR del modelo

Una vez que se haya terminado, hay que hacer clic en *Close*. Se notará que el caso *All Variables* ahora se actualiza y el icono de la caja cambia de tener un signo de exclamación a no tenerlo. Se agregan otras cinco Vistas de Casos después de ejecutar la identificación del modelo: *Models*, *Predictions*, *Frequency Uncertainty*, *Time Uncertainty* y *Correlations*.

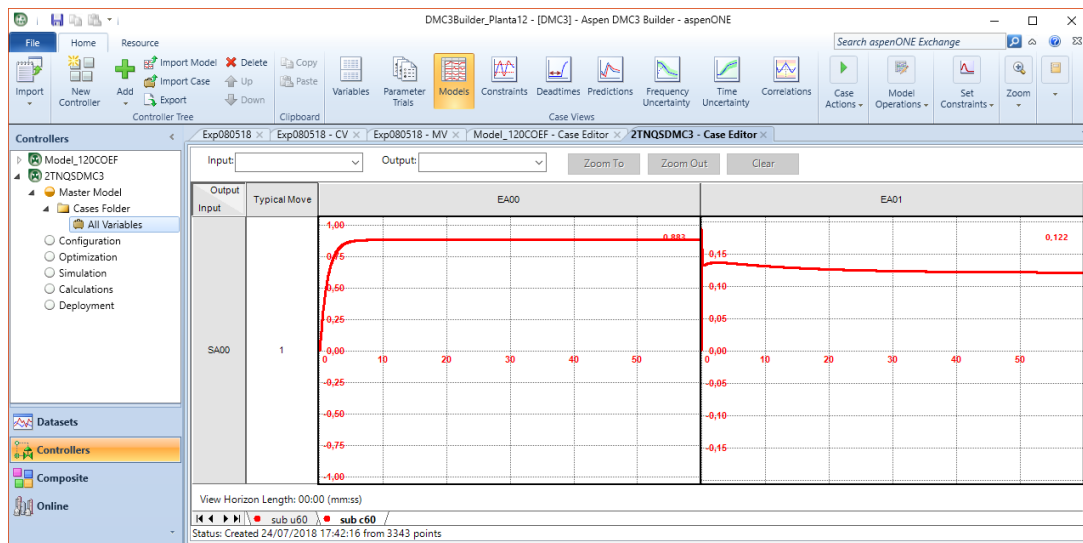


Figura 41 Vistas del caso adicionales disponibles tras ejecutar la identificación

La vista *Models* muestra los resultados de identificación del modelo. Aquí, la relación entre las variables manipuladas y las controladas está representada por curvas del modelo. Se observan las diversas pestañas que indican las pruebas en la parte inferior de la ventana. Se puede seleccionar cada una para investigar los resultados.

Hay dos grupos principales de cinta asociados a la Vista de Modelos: Operaciones de Modelo y Restricciones de Conjunto.

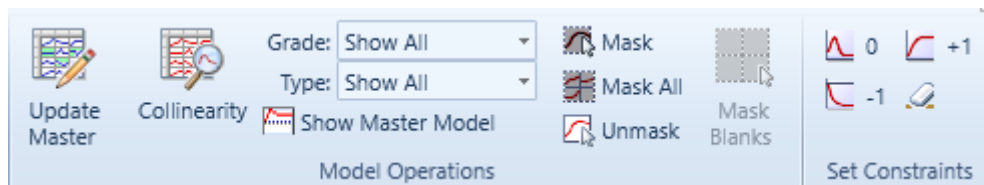


Figura 42 Grupo de opciones de operaciones del modelo y restricciones de conjunto

Se pueden usar las funciones de operación del modelo para actualizar las curvas identificadas al modelo maestro. También se puede inspeccionar la colinealidad dentro del caso de identificación. La opción de clasificación es una buena manera de mostrar sólo curvas con un cierto nivel de calidad. Esta clasificación se basa en Incertidumbre de la frecuencia y mira la incertidumbre de ganancia más específicamente.

La vista de Predicciones para un caso de ID de modelo muestra los datos de medición y las líneas de tendencia de predicción del modelo de caso para ayudar a evaluar el rendimiento del modelo. Se proporcionan herramientas para analizar los resultados de predicción en un diagrama de dispersión y, si corresponde, para editar las propiedades del modelo y la implementación de sectores en los datos de medición que se utilizan para identificar el modelo de caso. La vista *Predictions* ha sido mejorada en Aspen DMC3 Builder y ahora se pueden seleccionar las tendencias de entrada y salida, como se muestra en la figura a continuación:

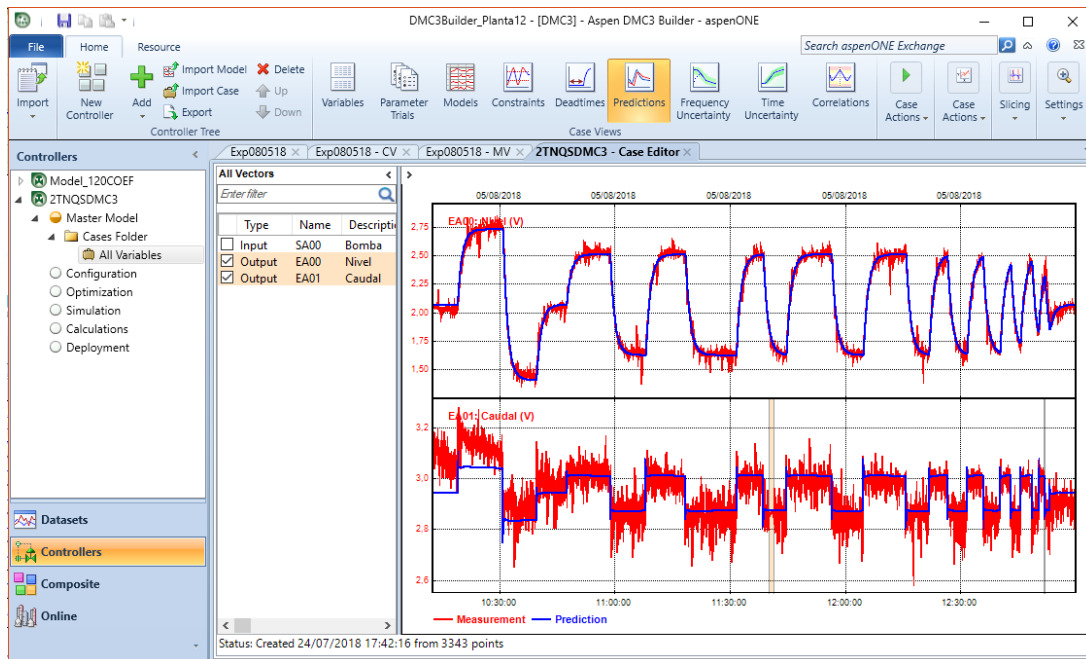


Figura 43 La vista Predictions permite seleccionar las tendencias de las entradas y las salidas

La vista de Incertidumbre de frecuencia para un caso de ID de modelo proporciona otro método para analizar el rendimiento del modelo de caso. Para cada par de variables entrada / salida, el diagrama de incertidumbre de Bode con una banda de incertidumbre 2-sigma se muestra en una vista de matriz. También se proporciona la clasificación del modelo desde A hasta D (Método de validación: clasificación de los modelos. Esto se hace comparando el tamaño relativo del límite con el modelo en las frecuencias baja y media. (Zhu, 1998) Más específicamente, las funciones de transferencia identificadas se clasifican en A (muy bueno, límite $\leq 30\%$ del modelo), B (bueno, límite $\leq 60\%$ del modelo), C (marginal, límite $\leq 90\%$ del modelo) y D (pobre, o, no existe ningún modelo, límite $> 90\%$ del modelo). Basado en extensas simulaciones y experiencia en proyectos, generalmente un grado A y los modelos de grado B se pueden usar en el controlador, siempre que el proceso no esté muy mal condicionado para CVs importantes. Los modelos de grado C y grado D no son relevantes para el control MPC y son tratados de la siguiente manera:

- 1) Hay que hacerlos cero cuando no haya transferencia entre los pares MV / CV. Esto puede determinarse utilizando el conocimiento del proceso y la verificación cruzada.
- 2) Si se espera y se necesita una función de transferencia en el control, hay que rediseñar una prueba para mejorar la precisión de estos modelos.

La relación señal / ruido de la CV (variable controlada o salida) se superpone en el diagrama de Bode.

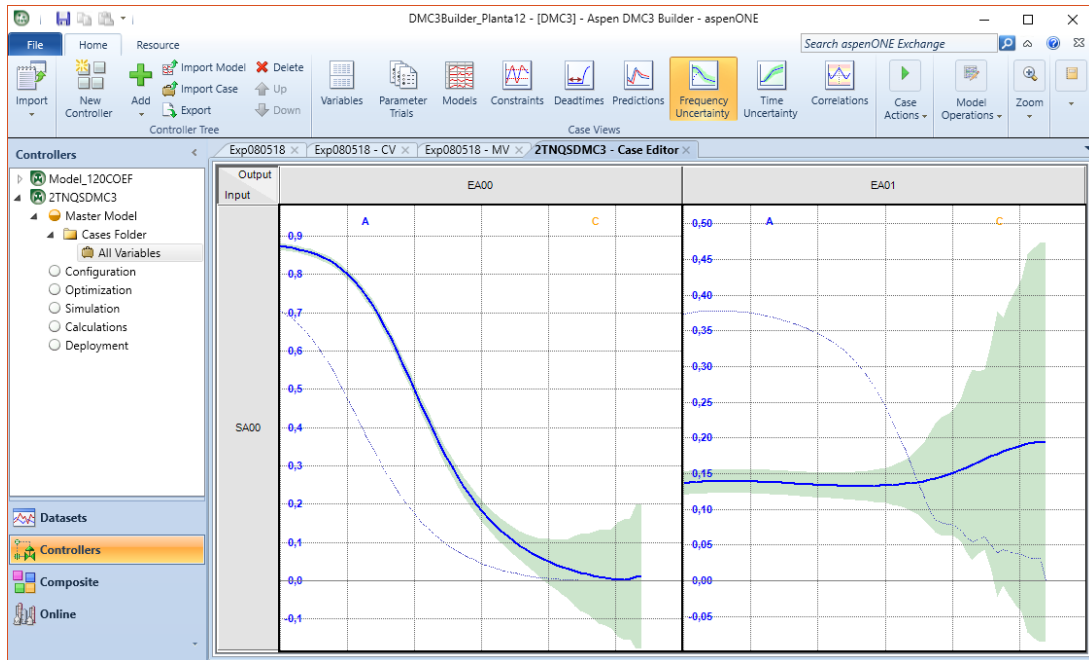


Figura 44 Gráfica de la incertidumbre del modelo

Nota importante: la selección de calificación se basa sólo en la incertidumbre de ganancia. El gráfico de Incertidumbre de tiempo con una banda de incertidumbre 2-sigma se muestra en una vista de matriz. Las calidades del modelo de A a D también se proporcionan. Hay que hacer clic en la vista *Correlations* para verificar las correlaciones de entrada.

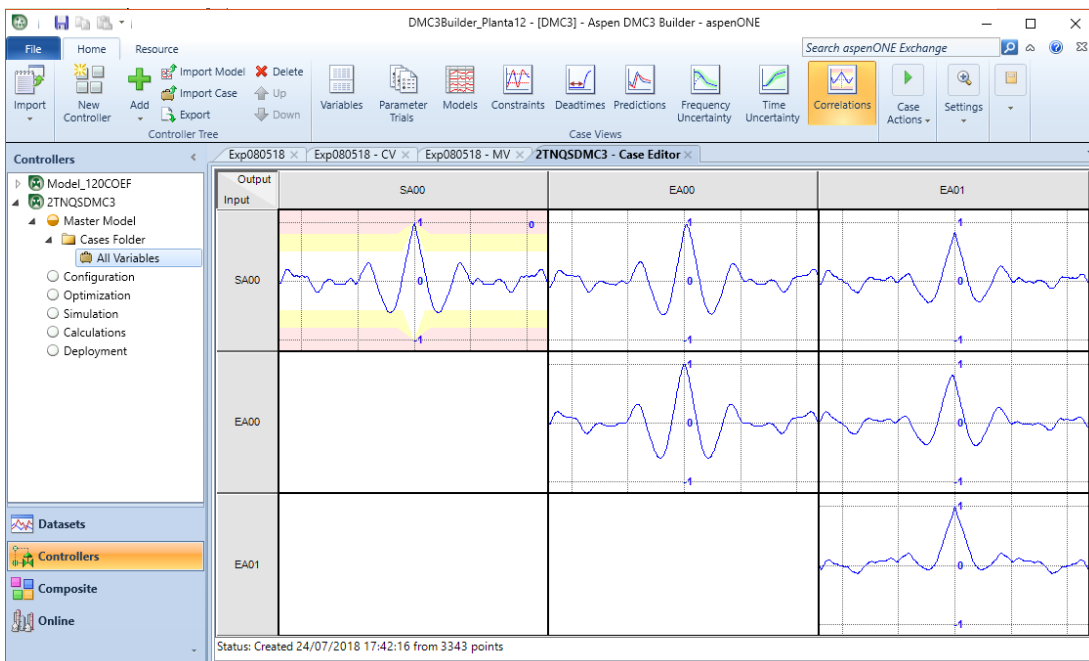


Figura 45 Vista Correlations

La vista *Correlations* para un caso de ID de modelo muestra gráficos para mostrar los niveles de correlación entre las variables del modelo de caso. Las parcelas en la vista de correlaciones de casos muestran los resultados de la evaluación matemática de cada variable para otras variables correlacionadas en el modelo.

La evaluación de correlación es más útil para entradas o variables manipuladas (MV), pero también está disponible para variables de salida si selecciona la casilla de verificación *Show Outputs*.

Se hace clic en el botón *Update Master* para abrir el cuadro de diálogo *Model Update Report*. Este diálogo ayudará a transferir las curvas del caso *All Variables* para el modelo maestro.

5.7.6. Configuración

La sección de configuración muestra las entradas de configuración de la CCF 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.

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

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.

En el nodo *Configuration*, la primera vista se selecciona de manera predeterminada. En este caso, se selecciona la vista de resumen. Hay cuatro puntos de vista en el nodo de configuración: *Summary*, *Feedback Filters*, *Subcontrollers* y *Test Groups*.

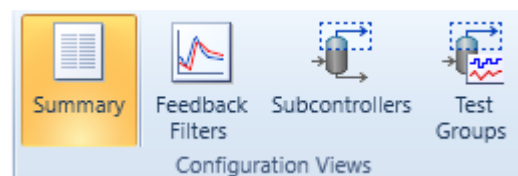


Figura 46 Grupo de opciones en la vista *Configuration*

La vista de resumen se muestra en la figura a continuación. Cuando se visita por primera vez esta página, se puede ver una descripción general rápida de la configuración del controlador y los pasos del flujo de trabajo involucrados. Por ejemplo, se puede ver que el controlador tiene dos salidas usando la predicción

predeterminada filtro de error (*Full Feedback*), no hay, no hay grupos de prueba y no está participando en una aplicación compuesta.

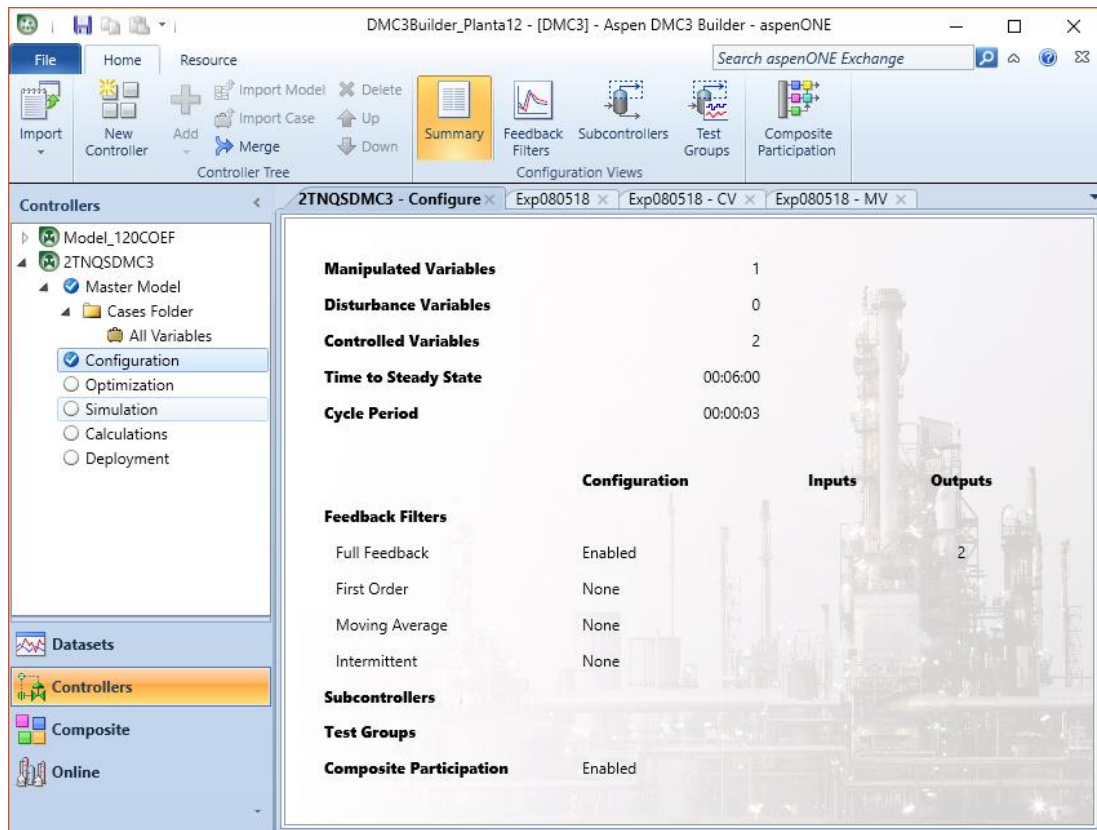


Figura 47 Vista Summary

La vista *Feedback Filters* para *Configuration* es donde se pueden configurar y ajustar los componentes de filtro de la aplicación actual. El filtro de error de predicción determina cómo el controlador actualiza la predicción en función de la retroalimentación recibida de la planta. En esta tabla, se puede elegir qué tipo de filtro usar (*Full Feedback*, *First order* o *Moving Average*) y especificar si una variable es intermitente.

La vista de *Subcontrollers* para *Configuration* (sólo aplicable a las aplicaciones tipo FIR) permite organizar variables de modelo en grupos de subcontroladores. Ésta también es una nueva vista en la sección de configuración para Aspen DMC3 Builder. Utiliza los mismos controles y elementos de la interfaz de usuario desarrollados para *Smart Tune*.

La vista de grupos de prueba para la configuración (sólo aplicable a las aplicaciones tipo FIR) permite definir los grupos de prueba Aspen SmartStep en una aplicación. En esta vista, se puede lograr lo siguiente:

- Agregar o eliminar grupos de prueba.

- Configurar el nombre del grupo de prueba y las propiedades de iteraciones de cada grupo de prueba.
- Asignar variables a cada grupo de prueba.
- Especificar entradas emparejadas dentro de cada grupo de prueba.

5.7.7. Optimización

Setpoint frente a Steady State Target

Se ha de distinguir en la configuración del controlador los conceptos de *Setpoint* y *Steady State*. Como *Setpoint* (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 este 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.

Los rankings creados sirven para relajar las restricciones del problema de control en un orden determinado, para asegurar así 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.

Una vez que el paso de *Configuration* se haya completado para la aplicación, ya está listo para pasar al paso de Optimización. Un optimizador de estado estacionario correctamente configurado es clave para entregar los beneficios asociados con un controlador APC. El optimizador de estado estacionario calcula objetivos para variables manipuladas (MV) y variables controladas (CV) y es

responsable de la parte "¿dónde se quiere ir?" del problema de control. En esta sección, se muestra cómo configurar el optimizador de un controlador Aspen DMC3. Se ha de seleccionar el nodo de flujo de trabajo *Optimization* para ingresar a la vista de optimización.

Cuando se selecciona por primera vez el nodo *Optimization*, se notará que su estado cambiará de no visitado  a parcialmente configurado .

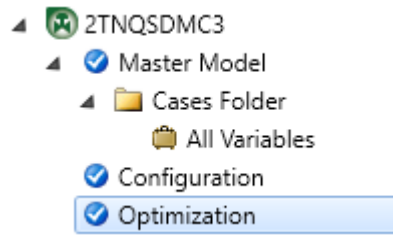


Figura 48 Indicación de estado totalmente configurado

En la figura a continuación, se pueden ver los principales grupos en el nodo *Optimization*:

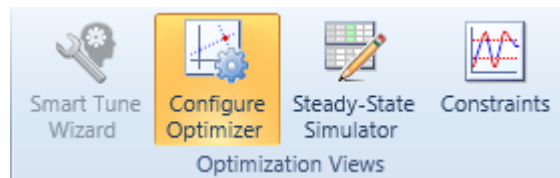


Figura 49 Grupos de opciones principales en el nodo de optimización: Method y Optimization Views

En la sección *Method* de la cinta de opciones, en Configurar optimizador, se puede elegir qué estrategia de optimización para la aplicación actual se quiere usar. Se puede seleccionar *Smart Tune* o *Traditional*.

La opción Tradicional omite el uso del método *Smart Tune* para construir un optimizador. Cuando se selecciona Tradicional, la secuencia de vistas para configurar y ajustar el optimizador se presenta como comandos de menú, de izquierda a derecha, en el menú principal del grupo de configuración del optimizador. Si el proyecto actual es un proyecto Aspen DMC3, se puede elegir utilizar el método *Smart Tune* en cualquier momento, incluso si se comienza a configurar el optimizador utilizando la opción Tradicional.

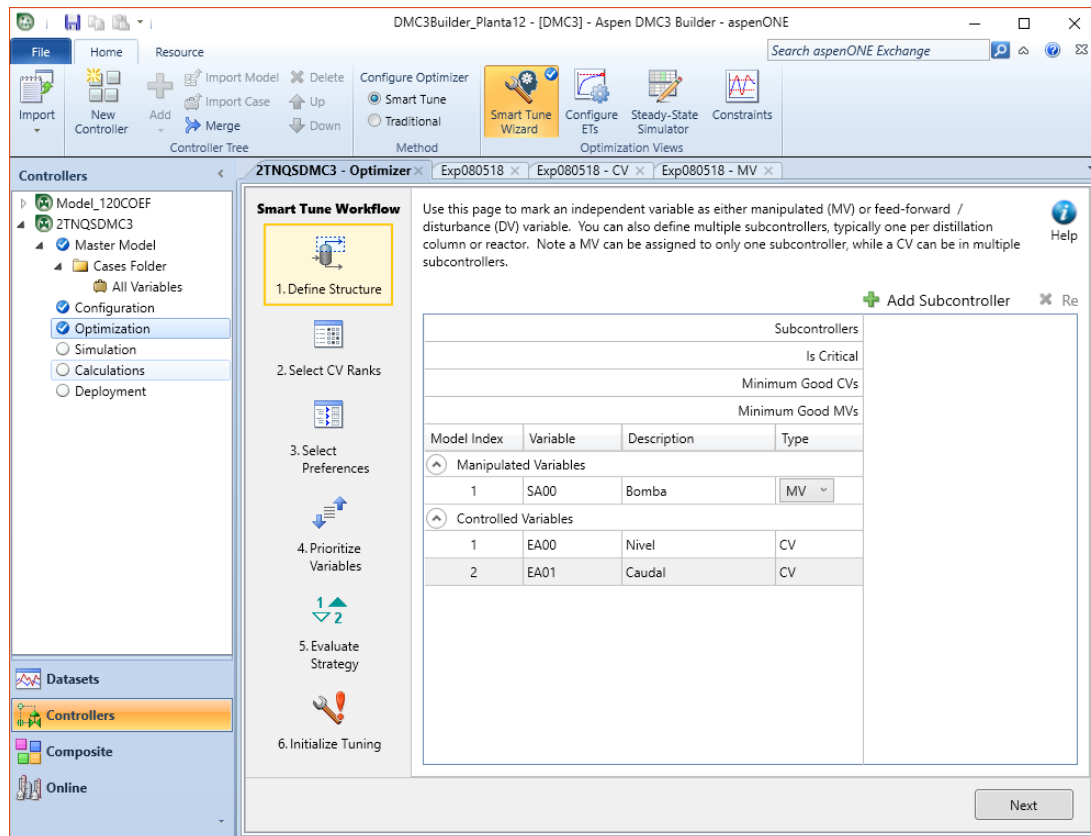


Figura 50 Asistente de configuración Smart Tune

El *Smart Tune Wizard* guiará al usuario por los pasos necesarios para sintonizar un controlador Aspen DMC3. En el lado izquierdo de la vista principal con pestañas, se puede ver el flujo de trabajo *Smart Tune* (pasos 1 a 7). La parte superior de la vista con pestañas tiene las instrucciones para el paso del flujo de trabajo del *Smart Tune* seleccionado y un icono de ayuda en la esquina superior derecha. Una vez que se haya terminado con un paso, se puede hacer clic en *Next* en la esquina inferior derecha o seleccionar el siguiente paso del flujo de trabajo.

5.7.8. Simulación

La vista de simulación es un buen entorno para las aplicaciones de ajuste que incluyen grupos de variables, como el subcontrolador o los grupos de prueba en una aplicación de prueba (Aspen SmartStep). Se usa la vista *Data* y la vista *Groups* para configurar los parámetros como se desee, y luego validar los cambios a través de ejecuciones de simulación. Cuando se hace clic en el nodo Simulación, el cuadro de diálogo emergente que se muestra en la figura 51 es desplegado.

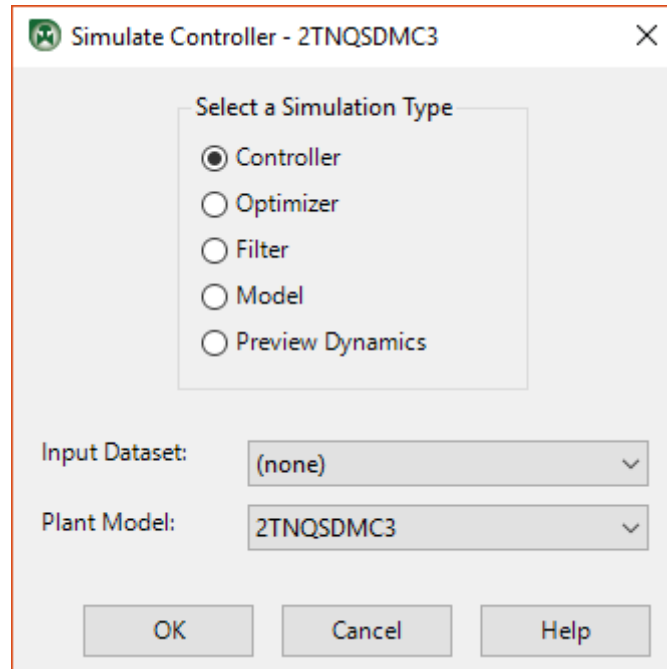


Figura 51 Cuadro de diálogo de la simulación

Se tiene la opción de simular ciertas partes de la aplicación que se está construyendo. El escenario más común es simular el controlador.

La figura 52 muestra la vista *Simulation* tras seleccionar *Controller* en el cuadro de diálogo emergente anterior.

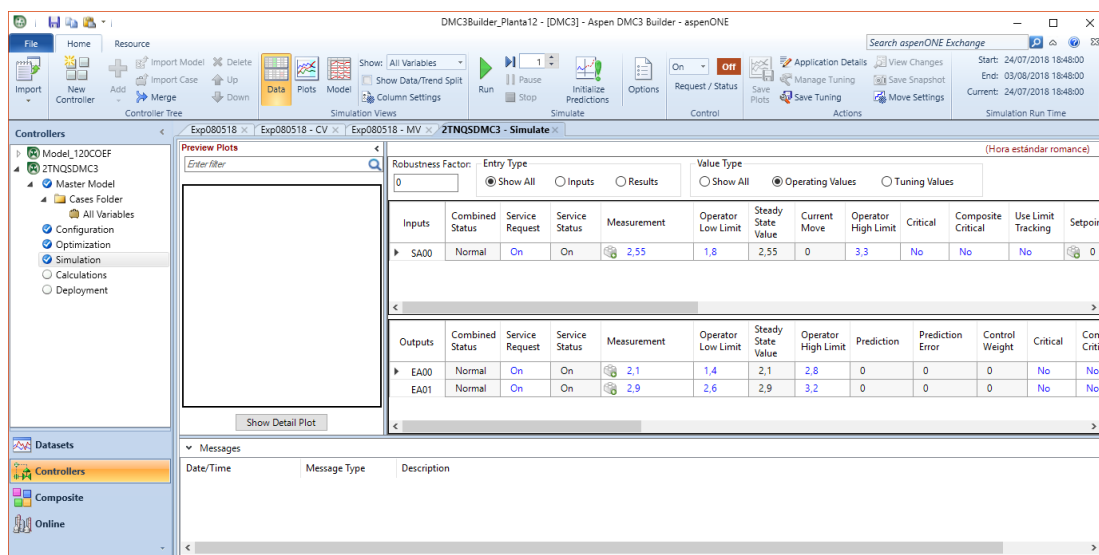


Figura 52 Vista Simulation

La vista *Data* muestra los valores actuales y el estado de las variables de la aplicación y las entradas asociadas. Los valores y los indicadores gráficos (iconos y colores de fondo de celdas de tabla) se actualizan durante el transcurso de una ejecución de simulación. Cuando la simulación se detiene, se pone en pausa, o si se está realizando una simulación interactiva de bucle cerrado, se pueden editar los valores de ciertas entradas, específicamente aquellas que se muestran con texto azul.

A continuación, se expone una explicación de los valores más importantes a modificar en la simulación.

PROGRAMACIÓN LINEAL Y RESTRICCIONES

Anteriormente en este documento se ha visto cómo se genera la ley de control en base a un modelo dinámico y a un vector de errores previstos. A partir de estos elementos, mediante el uso de una función de optimización basada en el principio de mínimos cuadrados, se calcula el vector de movimientos $\Delta\mathbf{mv}$. A continuación, en esta sección se expone de dónde provienen los puntos de consigna usados para calcular el vector de errores y cómo se incluye en el controlador el tratamiento de las restricciones del proceso, mediante un algoritmo de programación lineal con restricciones.

La filosofía a seguir consiste en intentar encontrar un conjunto de puntos de consigna en estado estacionario, de forma que todas las variables controladas y manipuladas estén dentro de los límites especificados. Es decir, que se cumplan todas las restricciones.

Si es posible encontrar una solución a este problema, entonces el conjunto de puntos de consigna representará una solución posible a este problema de optimización en estado estacionario. Si además de encontrar una solución posible, existen varias soluciones, entonces habrá que usar criterios económicos para determinar cuál es la solución posible más rentable.

El conjunto de restricciones especificadas juntamente con el proceso, determinan los grados de libertad del conjunto Proceso+Controlador. Luego, si las restricciones son muy severas y el proceso no disponía de muchos grados de libertad, entonces es posible que no exista ninguna solución al problema de optimización. En este caso habrá que determinar qué restricción no se va a poder cumplir y relajarla temporalmente, hasta que las condiciones del proceso permitan volver a cumplirla, o bien modificar el conjunto de restricciones para de esta forma poder encontrar una solución posible.

RESTRICCIONES EN LAS VARIABLES MANIPULADAS

En primer lugar, se consideran las restricciones impuestas en las variables manipuladas. Sería deseable poder predecir el comportamiento de todas las variables restringidas del sistema, para de esta manera asegurar que en el estado estacionario se satisfagan todas las restricciones. Si se decide no modificar ninguna variable manipulada en un transitorio, entonces $\Delta\mathbf{mv}$ permanecerá constante durante todo el camino hasta que se alcance un estado estacionario en el proceso. Es decir:

$$MVi_{ACTUAL} = MVi_{SS}$$

donde:

- MVi_{ACTUAL} denota el valor de la variable manipulada 'i' en el momento actual
- MVi_{SS} denota el valor de la variable manipulada 'i' en el estacionario

Si se desea cambiar el valor de una MV en el estacionario, entonces se puede expresar el nuevo valor en función del actual más un incremento:

$$MVi_{DESEADO} = MVi_{SS} + \Delta MVi_{SS}$$

En cada intervalo de ejecución del controlador hay que decidir el orden de magnitud de cambio en MVi_{SS} , en función de las restricciones actuales, los criterios económicos y el conocimiento de la dinámica del proceso. **La primera limitación es que nunca y bajo ningún concepto se tiene que permitir que las variables manipuladas se salgan de los límites preestablecidos.** Definiendo el límite superior como MVi_{HIGH} y el límite inferior como MVi_{LOW} , se puede expresar lo anterior de la siguiente forma:

$$MVi_{SS} + \Delta MVi_{SS} \leq MVi_{HIGH}$$

$$MVi_{SS} + \Delta MVi_{SS} \geq MVi_{LOW}$$

RESTRICCIONES EN LAS VARIABLES CONTROLADAS

Al igual que con las variables manipuladas se puede aplicar el mismo tratamiento con las variables controladas. En primer lugar, se limitarán los valores superiores e inferiores que puedan tomar las variables controladas. Usando la misma nomenclatura que con las variables manipuladas, se obtiene:

$$CVj_{SS} + \Delta CVj_{SS} \leq CVj_{HIGH}$$

$$CVj_{SS} + \Delta CVj_{SS} \geq CVj_{LOW}$$

Siendo CVj_{SS} el valor estacionario de la predicción en lazo abierto de la variable controlada j-ésima.

A diferencia de las variables manipuladas, las variables controladas no se pueden mover directamente, para provocar un ΔCVj_{SS} determinado, es preciso generar un cambio en las variables manipuladas que afecten a CVj . Como ya tiene un modelo del proceso, entonces se puede determinar qué variables afectan a CVj y se puede calcular qué Δmv_{SS} es necesario para provocar un ΔCVj_{SS} deseado. La parte del modelo que se tiene que usar es exclusivamente la parte estacionaria. Se define 'Gji' como la ganancia estacionaria que relaciona la variable manipulada MVi con CVj .

Al igual que con las variables manipuladas,

$$CVj_{DESEADO} = CVj_{SS} + \Delta CVj_{SS}$$

PROGRAMACIÓN LINEAL

Se pueden dar varias posibilidades como solución, desde tener un área de infinitas soluciones posibles, a tener un único punto como solución e incluso puede darse el caso que no exista ninguna solución posible. El problema de optimización aparece cuando hay más de una solución posible, en ese caso se busca la solución atendiendo a criterios económicos y usando la programación lineal.

Puede tratarse de minimizar la función de coste. El algoritmo de programación lineal puede resolver funciones de coste, que representa cualquier función de optimización que sea razonablemente lineal con respecto a cambios en el vector de movimientos de las variables manipuladas. Si se desea introducir en dicha función dos términos, por un lado, se tiene el beneficio por producir más cantidad de un producto valioso y, por otro, el desembolso que supone en energía el producir más del producto anterior. Es decir:

Función de coste = $-(\Delta SS \text{ cambio en producción}) \cdot \text{valor de los productos}(\text{€}) + (\Delta SS \text{ cambio en energía}) \cdot \text{precio de la energía}(\text{€})$

Ahora hay que relacionar los productos de valor añadido con las variables controladas y los costes energéticos con las variables manipuladas. Normalmente los costes energéticos se asocian directamente con las variables manipuladas ya que se puede determinar fácilmente lo que cuesta en términos económicos el aumentar o disminuir cada variable manipulada. Para las variables de valor añadido el análisis no es tan simple, ya que hay que relacionar cada una de estas variables con las variables controladas y posteriormente mediante el uso del modelo obtener la equivalencia con respecto a las variables manipuladas.

A la ganancia de la cantidad de producto con respecto a la variable manipulada, multiplicada por el valor de dicho producto, si se le suma el coste de la variable manipulada, se le denomina **LPCOST** en DMC3.

Una vez resuelta la función de coste sujeta a restricciones mediante el algoritmo de programación lineal, falta por ver cómo se incluye esto dentro del controlador. La forma más habitual consiste en forzar al módulo de cálculo de movimientos la solución de estado estacionario calculada por el módulo de programación lineal. El módulo de programación lineal genera el vector $\Delta \mathbf{MV}_{ss}$ que es el vector de movimientos de las variables manipuladas en el estacionario. Pero aún falta por determinar cómo van a ser cada uno de los movimientos $\Delta \mathbf{mv}$ hasta que se alcance el estacionario. La solución consiste en hacer $\sum \Delta \mathbf{mv} = \Delta \mathbf{MV}_{ss}$.

Por lo tanto, lo más importante para la operación de un controlador multivariable es:

- Qué variables se quieren controlar.
- Los rangos en que se quieren controlar las variables controladas (límites de las CVs).
- Qué variables manipuladas están disponibles para el controlador.
- Los rangos en que pueden moverse las variables manipuladas (límites de las MVs).

Y, los parámetros de ajuste a destacar son:

- Importancia relativa de las variables controladas (Equal Concern, véase sección 4.4.3 *Equal Concern*, página 55).
- Penalización al movimiento excesivo de las variables manipuladas (Move suppression factor, véase sección 4.4.2 *Move suppression*, página 55).
- Coste asociado al movimiento de cada variable manipulada (LPCOST).

La vista *Trends* para simulación tiene un panel *Preview Plots* y un panel *Detail Plots*. El panel Vista previa de gráficos proporciona un lugar para seleccionar las variables que se quiere mostrar en el panel de Detalle de gráficos. El panel Diagramas de detalle proporciona un lugar para mostrar y comparar los resultados graficados de la simulación y el ajuste que se realiza al construir aplicaciones de controlador.

La vista de grupos para simulación permite simular y ajustar aplicaciones que tienen grupos de variables, como grupos de subcontroladores o grupos de prueba.

5.7.9. Calculations

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, 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 se tiene un apartado para las entradas de cada variable MV (variable manipulada) o FF (variable independiente pero no manipulada) que se vaya a utilizar.
- *Dependent*: ídem al anterior, pero con las variables CV (variables controladas).
- *Calculations*: dividido en dos, *Declarations*, donde se declaran nuevas entradas de la .ccf que cree el usuario, y *Equations*, donde se podrán crear fórmulas para su uso en el archivo.
- *Subcontrollers*: sólo se muestra si se permite que en el controlador puedan participar subcontroladores.

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 se definirán las entradas que se deseen crear para completar la configuración de la CCF y que Builder no proporciona de forma predeterminada. Para crear una nueva entrada basta con pulsar el botón de la barra de herramientas *User Entries*. Entonces el programa pedirá un nombre para la nueva entrada, que debe ser distinto de los nombres que están reservados para las entradas que Builder proporciona de forma predeterminada, y además se deberá elegir el tipo de datos que utilizará la entrada. Una vez validado lo anterior, aparecerá en la sección la nueva entrada y se podrá editar. Al editar se podrá observar que se tienen a disposición del usuario todas las palabras clave o *Keyword* de que dispone Builder, esto es: *None, CONS, LOCAL, INIT, READ, WRITE, LWRITE, PWRITE, AWRITE* y *RDWRT*. Así pues, se podrá 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 Builder, sin diferencias con las que vienen predeterminadas en Builder.

EQUATIONS

Builder ofrece la posibilidad de utilizar una serie de fórmulas sencillas, lo que 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 se van a controlar, por ejemplo. Para crear una ecuación se ha de seguir una serie de pasos:

1. Pulsar el botón de herramientas *Formulas, Input Calculation* o *Output Calculation*.
2. Seleccionar el campo *Result* y seleccionar del desplegable *Grp|Var* la variable o 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 se desee, se pulsa el botón *Insert*.
3. Posicionarse luego en el campo *Formula*. En este campo se podrán insertar también entradas de variables o grupo de variables, así como algunas de las funciones que Builder trae predeterminadas (ver el anexo correspondiente).
4. Repetir los pasos para seguir insertando nuevas ecuaciones.

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

Aunque la etapa *Calculations* sigue a la etapa de Simulación en el árbol de navegación Controladores, se pueden agregar cálculos a una aplicación durante cualquier etapa de su desarrollo.

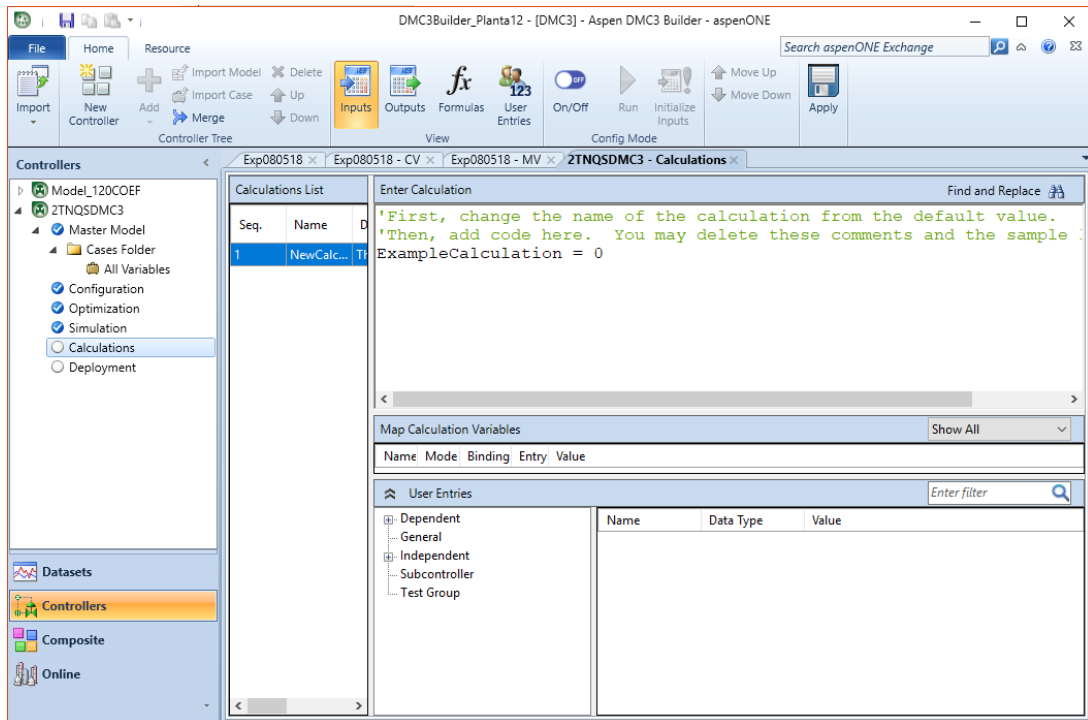


Figura 53 Añadir cálculos a una aplicación durante cualquier etapa de su desarrollo

5.7.10. Deployment

En la vista *Deployment*, se pueden especificar parámetros de conexión de entrada / salida (IO) para una aplicación, de modo que la aplicación esté lista para su implementación en un servidor de aplicaciones en línea. Esto incluye especificar los nombres de las etiquetas fuente IO, los límites de validez y la medición de tiempos de espera que son necesarios para que la aplicación se comunice con una interfaz para los datos de planta, como un servidor Aspen Cim-IO, servidor OPC o servidor de datos de proceso Aspen. Hay dos vistas principales: *IO Tags* y *Online Settings*.

El objetivo final en la vista *IO Tags* para *Deployment* es completar los campos en la tabla inferior *Variable Detail*, vinculando las etiquetas IO (desde interfaces para los datos de planta) a cada parámetro que está asociado con una variable de aplicación a medida que selecciona cada variable de la tabla en el panel del Generador de Etiquetas del medio.

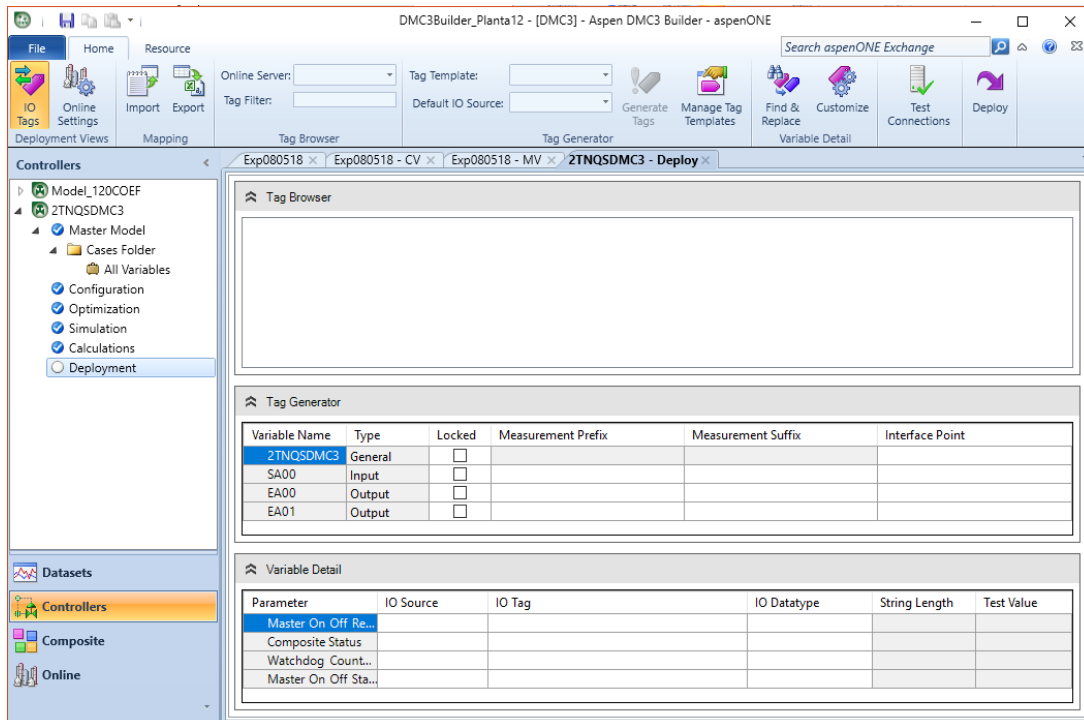


Figura 54 Vista Deployment

Una vez está configurado el controlador y listo para ser implementado, se ha de hacer clic en el botón *Deploy* para transferir la aplicación al entorno online.

5.8. Composite

En el uso de *Composite* se indica cuándo se van 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 se incluye el controlador en un conjunto gestionado por *Composite* para participar en un objetivo en común.

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.

Validación de la CCF

Una vez que se tenga la CCF configurada, Builder permite realizar un chequeo de ésta 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.

5.9. Online

En el caso concreto de este proyecto, no se puede realizar la implementación en línea con el servidor OPC que comunica con la planta de laboratorio. Existe un problema de comunicación que hasta ahora no se ha podido resolver, tras conferencias y conversaciones con autoridades norteamericanas de la materia, una posible solución pasaría por instalar un Windows Server de 64 bits que ejerciera de intermediario.

Aunque aquí no se realice esta parte de la implementación, por tener como objetivo la redacción de los manuales de uso, a la hora de trabajar en una planta y diseñar un controlador predictivo con este software, el usuario se deberá cerciorar que puede haber comunicación con el servidor.

Capítulo 6. CONCLUSIONES Y TRABAJO FUTURO

6.1. ¿Qué se ofrece en este capítulo?

En este capítulo se desarrollarán las conclusiones que se han sacado de la realización de este proyecto, así como posibles mejoras que pueden realizarse en el proyecto.

6.2. Conclusiones

Tras un largo periodo de investigación y trabajo en la elaboración de este proyecto y tras haber superado problemas de configuración y diseño, el resultado obtenido cumple con muchos de los objetivos planeados en un principio, puesto que los manuales redactados se constituyen como una herramienta eficaz a la hora de realizar el diseño del control y la supervisión de una planta específica de laboratorio.

Además, en el transcurso de este trabajo se ha tomado contacto y ampliado los conocimientos en materias que no se conocían con tanta profundidad. Por un lado, se ha enfrentado a la tarea de diseñar un manual de uso utilizando un paquete software totalmente orientado a plantas industriales, como es el caso del paquete aspenONE v10, con sus aplicaciones incorporadas y se ha podido comprobar su potencia y grado de complejidad. Por otro lado, se ha podido ampliar conocimiento en otras áreas como las comunicaciones OPC, el control estadístico de datos, modelado y simulación de procesos.

Asimismo, se ha aplicado todo lo aprendido en materia de control automático para el control de la planta. Con el uso del DMC3, se obtiene un buen modelo matemático del sistema experimental que se quiere controlar, y se diseña un sistema de control teórico que permite controlar las variables más importantes del proceso.

Una vez se diseña el controlador del sistema, se ejecuta la planta para simular diferentes reacciones según se establezcan los parámetros correspondientes. Además, el controlador predictivo se conecta con un programa de optimización que determine los puntos de consigna con tal de minimizar una función de costo determinada.

Después de hacer este trabajo, viendo las simulaciones y los resultados obtenidos, los manuales de uso pueden resultar una herramienta útil para el diseño de controladores.

Las estrategias de control diseñadas también pueden ser interesantes, se podría haber usado alguna estrategia de control más puesto que al tener un sistema MIMO se pueden diseñar muchos sistemas de control, los expuestos aquí son sistemas de control bastante usados en la realidad y, como se puede observar, en se pueden obtener buenos resultados.

6.3. Trabajo futuro

Todo proyecto, en especial si se trata de una aplicación informática, siempre es apto para futuras ampliaciones o modificaciones. Por ello se adjuntan aquí las posibles mejoras que se pudieran acometer con mayor interés:

En cuanto a posibles esquemas de planta a controlar, se puede añadir mucha instrumentación y numerosas variables, en este proyecto, la planta de laboratorio es muy simple, pero el software tiene unas prestaciones muy superiores.

Para el control base propuesto los controladores se han sintonizado mediante un método de los opcionales. Sería interesante realizar una sintonía óptima de los mismos de manera que se pudieran conocer el rendimiento máximo de las estructuras de control propuestas.

Como trabajo futuro se desea integrar esa capa de optimización en línea con el sistema de control de la planta, implementando una capa superior de optimización con la que comandar el sistema de control para lograr una operación óptima de la planta. Sin embargo, para poder abordar problemas de un nivel superior, los niveles inferiores han de funcionar correctamente, en concreto, la optimización económica exige que el nivel de control funcione adecuadamente.

REFERENCIAS

Aspen Technology, Inc., 2018. *Aspen DMC3*. [Online] Available at: <https://www.aspentech.com/en/products/msc/aspens-dmc3> [Accessed Septiembre 2018].

Aspen Technology, Inc., 2018. *AspenTech*. [Online] Available at: <https://www.aspentech.com/> [Accessed Septiembre 2018].

Aufderheide, B. & Wayne, B., 2003. Extension of dynamic matrix control to multiple models. Issue 27, pp. 1079-1096.

Badgwell, S. & Qin, T., 1997. An overview of industrial model predictive control technology. *AIChE symposium series*.

Badgwell, S. & Qin, T., 2003. A survey of industrial model predictive control technology. *Control Engineering Practice* 11.

Bentes, F., Aparecido, N., Guimares, R. & Teixeira, J., 2006. Comparison among PI, DMC, QDMC, and GPC Algorithms In The Control Of The Spouted Bed Dryer. Issue 193, pp. 24-37.

Bordons, E. F. & Camacho, C., 2010. Control Predictivo: Pasado, Presente y Futuro. *Revista Iberoamericana de Automatica e Informatica Industrial (RIAI)*, 3(1).

Bordons, E. F. & Camacho, C., Octubre 2004. Control Predictivo: Pasado, Presente y Futuro. *Revista Iberoamericana de Automática e Informática Industrial, Vol. 1, Núm.3*.

Cutler, C. & Johnston, C., 1985. Comparison of the Quality Criterion for PID and Predictive Controllers. *American Control Conference*, p. 214 – 219.

Cutler, C. R. & Ramaker, B. L., n.d. Dynamic matrix control. A computer control algorithm. *Joint Automatic Control Conference*, Volume 17.

Demircan, M., Amurdan, C. & Postlethwaite, E., 1999. On-Line Learning Fuzzy Relational Model Based Dynamic Matrix Control of an Open loop Unstable Process. *Trans IChemE*, Issue 77, pp. 421-428.

Dubay, R., Kember, G., Lakshminarayan, C. & Pramujatic, B., 2005. Development of characteristic equations and robust stability analysis for MIMO move suppressed and shifted DMC. *ISA Transactions*, Issue 44, p. 465–479.

Fischer, M., Nelles, O. & Isermann, R., 1998. Predictive control based on local linear fuzzy models. *International Journal of Systems Science*, 7(29), pp. 679-697.

González, R. & Cifuentes, F., 1999. *Curso DMCPlus*. Madrid: s.n.

JavaRegula, 2018. *JavaRegula*. [Online] Available at: <http://bit.ly/JavaRegula> [Accessed Septiembre 2018].

Kember, G., Dubay, R. & Mansoura, S., 2005. Continuous analysis of move suppressed and shifted DMC. *ISA Transactions*, Issue 44, p. 69–80.

Richalet, J., 1993. Industrial Applications of Model Based Predictive Control. *Automatica*, Vol. 29, No. 5.

Richalet, J., Rault, A., Testud, J. L. & Papon, J., 1978. Model Predictive Heuristic Control: Applications to Industrial Processes. *Automatica*, Vol. 14.

Ruihua, X. & Lihong, W., 2009. Adaptive Paralleled DMC-PID Controller Design on System with Uncertainties. *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pp. 720-724.

Sanjuán, M., Smith, C. & Iglesias, E., 2006. Tuning equation for dynamic matrix control in siso loops. *Ingeniería y desarrollo*, Issue 19, pp. 88-110.

Serra, M., Perrier, M., Espuña, A. & Puigjaner, L., 2001. Computers and Chemical Engineering. Issue 25, p. 859-866.

Softing AG, 2018. *Softing OPC*. [Online] Available at: <https://company.softing.com/en/startpage.html> [Accessed Septiembre 2018].

Vasconcelos, R. & Maciel, R., 1998. Development of a Supervisory Control Strategy For The Optimal Operation Of Grain Dryers. *Drying Technology*, 9&10(16), pp. 2017-2031.

Zhu, Y., 1998. Multivariable process identification for MPC: the asymptotic method and its applications. *Journal of Process Control*, 8(2), pp. 101-115.

ANEXOS

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. Existe una serie de etiquetas reservadas para entradas estándar del controlador DMC3, 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 (*Distributed Control System*: 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 permite encontrar el punto a través de la conexión CIM-IO.
 - *CIM-IO information*: sólo si se utiliza la interfaz CIM-IO.

- *ET (External Targets)*: las *External Targets* son objetivos de control especificados desde el exterior de DMC3. Esta sección no se muestra a no ser que se especifique dicha posibilidad. Las entradas de todas las secciones pueden encontrarse en el anexo correspondiente.

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

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 se le elija a la entrada se podrá realizar una acción u otra y no todas las entradas disponen de todas las palabras clave.

Las *keywords* 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 Builder solamente.
- *CALGET*: cálculo 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 DMC3. 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.

Configure

- *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 Builder.
- *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 Builder, son los minutos establecidos para el *Steady State*.

General

- *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 manipuladas (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*.

Independent

- *CRIND*: 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 se incurre 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 límites, que se podrían llamar límites de operación. Se trata 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 *setpoint* 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 *setpoint* 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 se sumaba al valor que tuviera en ese momento la variable el valor de *CMOV* mediante ecuaciones, y el total era el valor que se le proporcionaba 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 se estén 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 se definieron anteriormente. Así, si se escribe antes de leer el valor de la variable entonces en *VIND* no se tendrá 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 se puede crear una nueva variable en la base de datos de la planta, *V8_VIND* y *RESISTENCIA_VIND*, que sea una copia del valor que se quiere leer y escribir, evitando de esta manera que la escritura borre los datos que se quieren leer.

Dependents

- *DEP*: se trata del valor actual de la variable leído directamente del proceso. Puede ser *READ* o *LOCAL*.
- *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 (*LDPEN*, *UDPEN*). Con estos límites lo que se delimita es una zona válida en la que el programa debe encontrar una solución en régimen permanente para esta variable. Si se quiere forzar la variable para que se dirija a un valor concreto se deberá tomar el mismo límite inferior y superior, con lo cual se restringe 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*, *LOCAL* o *READ*.
- *SRVDEP*: es la entrada que 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 guarda el error acumulado en la predicción. Esta entrada 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 se pueden 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*: indica el objetivo *Steady State* o de régimen permanente para la variable con la que se esté 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*.

DMC3 Model

BARRA DE MENÚS

File

New Project: Create a new Model project (with the extension *.dpp).

Open Project: Open an existing project.

Close Project: Close the current project but leave Model open.

Save Project: Store the project.

Save Project As: Save the project under a new name.

Import: Import various objects.

Export (Slices and Formulas): Export various objects.

Page Setup: For printing, change page options: margins, layout, numbers, and headers/footers,

Print: Create a hard copy.

Print Preview: View a copy on-screen before printing it.

Print Setup: Change printer settings.

Exit: Close Model.

Edit

Copy: Create a duplicate of a selected object.

Paste: Paste the duplicated object.

Select All: Highlight all available objects.

Find Vectors: Locate a vector.

Tag Properties: View/modify tag properties.

Bad Slices: View/modify bad data slices.

Go To: Go to the indicated independent/dependent variable pair.

Next Page: Move to the next screen in a view.

Previous Page: Move to the previous screen in a view.

Table of Contents: View the table of contents for a report view.

Make List: Create a new list using the selected object(s).

Report/List Properties: View report or list properties.

Font: Change the font for the active view.

Mark Bad Slice: Identify bad data slices.

Edit Bad Slice: Edit bad data slices.

Get Curve: Get a curve from a source model and place it in the target model.

Put Curve: Put a curve from a source model into a target model.

Curve Operations: Perform various operations on a curve.

Delete Curve: Remove a selected curve from a model.

Refresh: Regenerate the selected object (usually recalculates a model's curves)

Copy Steady-State Gain: Copy the steady-state gain matrix onto the clipboard.
Available from

the Model view only.

View

New Project Outline: Display a copy of the project view.

New Vector Plot: Display a vector plot view.

New Model Plot: Display a model plot view.

New Report: Display a report view.

Show/hide: Display or hide tool bars (Main, Project) and the status bar.

Project

Expand/Collapse: Open or close the selected part of the project outline tree.

New: Create a new object.

Copy: Duplicate an existing object.

Delete: Remove an existing object.

Make New List: Create a new list using the selected object(s).

View: Display a view of the selected object(s).

Run: Run/recalculate the selected object.

Reassemble: Update (recalculate) the selected assembled model.

Export: Create an external file of the selected object(s), for import into another

project.

Properties of: Display the properties of the selected object(s).

Format

Colors and Fonts: Change colors/fonts for a view.

Show Legend: Display a legend for a view.

Show Grid: Display a grid for a view.

Show Description: Display a description for a view.

Show X-Scale: Display the X-Scale for a view.

Show Y-Scale: Display the Y-Scale for a view.

Show Tooltips: Display information under the cursor point for a view.

Show Time Coordinates: Display the time coordinates for a view. Toggles with Show Sample

Number Coordinates.

Show Sample Number Coordinates: Show the sample number coordinates for a view. Toggles

with Show Time Coordinates.

Set Plots Per Page: Change the number of plots shown in a view.

Set Transform Flag: Show a vector as transformed or untransformed.

Range

Set: Indicate a new range.

Full Width: Show the full range.

Zoom In: View a smaller time range in more detail.

Zoom Out: View a larger range in less detail.

Scale

Auto: Set the scale so that the tick marks along the Y-axis are reasonably whole, easily readable numbers.

Common: Set the scaling limits near the min and max vector values in the displayed sample range; this function imposes the same scale on all vectors in the display and allows for direct comparison of similar vectors.

Distributed: Spread the vectors in multi-plot displays vertically across the screen so that each vector is displayed in a separate region of the screen with

no overlap.

Min/Max: Set the scaling range at the minimum and maximum values in the vector.

Set: Display the Set Scale dialog box. The current tag is selected; enter a Minimum and Maximum value.

Typical Moves: Display the model plot using the typical moves indicated for each tag name.

Move Sizes: Show a Move Size dialog box for selecting a tag name on the plot and changing its typical move value.

Tools

Formula Editor: Display the Formula Editor.

Options: Change various project-related options.

Window

CascadeTileArrange Icons: Select standard display functions for windows in the work area; see

your Windows documentation for details. Also, choose a window from the list of open windows to activate one in the work area.

Help

Help Topics

DMCplus Reference

Entry Dictionary

Technical Support

About DMCplus Model

ANEXO II

FÓRMULAS PREDETERMINADAS DE MODEL

A continuación, se listan las fórmulas que Model proporciona para realizar cálculos con los vectores. Las fórmulas pueden combinarse entre ellas para obtener una nueva fórmula que sea de nuestro interés.

Todas las fórmulas realizan sus cálculos utilizando doble precisión de punto flotante, incluso con funciones Booleanas, en las cuales el cero es Falso y distinto de cero Verdadero.

Para introducir valores del tipo Bad Value, debe utilizarse la palabra reservada BAD en lugar del valor -9999.

El rango de valores permitidos para las fórmulas está comprendido entre -10^{20} y 10^{20} .

@Log(value): Natural logarithm of a positive real value

@Log10(value): Common logarithm of a positive real value

@Exp(value): Natural antilogarithm of a value

@Sqrt(value): Square root of a positive real value

@Abs(value): Absolute value of a value

@Interp(value,x1, y1, x2, y2[, x3, y3...]): Piece-wise linear interpolation: (x,y pairs define function

@InterpBad(vector): Interpolate over bad values in a vector

@Cav(vector, nsamples): Central average of a vector. nsamples must be odd

@EFilt(vector, factor): Exponential filter: $out(1) = in(1)$,

$out(i) = out(i-1) * factor + in(i) * (1 - factor)$

@Diff(vector): Difference a vector: $out(1) = 0$,

$out(i) = in(i) - in(i-1)$

@Integ(vector, bias): Integrate a vector: $out(1) = in(1) + bias$,

$out(i) = in(i) + out(i-1)$

@Shift(vector, nsamples): Shift a vector. Negative nsamples means shift earlier.

@Valim(vector, vallow, valhigh): Mark samples outside range vallow and valhigh as BAD

@MkBad(vector, nfirst1, nlast1[, nfirst2, nlast2...]): Mark sample ranges as BAD. nfirst and nlast are

sample numbers

@LoClip(vector, valclip, valnew): Set values less than valclip to valnew

@HiClip(vector, valclip, valnew): Set values greater than valclip to valnew

@Max(value1[, value2, value3 ...]): Maximum of a set of values

@Min(value1[, value2, value3 ...]): Minimum of a set of values

@IF(test, value1, value2): value1 if test is "true" or value2 if test is "false"

@EQ(value1, value2): "true" if value1 is equal to value2

@NE(value1, value2): "true" if value1 is not equal to value2

@GT(value1, value2): "true" if value1 is greater than value2

@GE(value1, value2): "true" if value1 is greater than or equal to value2

@LT(value1, value2): "true" if value1 is less than value2

@LE(value1, value2): "true" if value1 is less than or equal to value2

@OR(value1, value2[, value3...]): "true" if any value in a list of values is true

@AND(value1, value2[, value3...]): "true" if all values in a list of values are true

@NOT(value): "true" if value is "false"

@BADVAL(value): "true" if value is a BAD value

@PCT2(Temp<degF>, Pres<psig>, Rpres<psig>, B, C, Bias<degF>, LogInd<1=ln, 2=log10>): Pressure

Compensated Temperature

@PFPCT(Temp<degF>, Pres<psig>, StdP<psig>, WatK): Petroleum Fraction Pressure Compensated

Temperature

@REFPCT(Temp<degF>, Pres<psig>, StdP<psig>, Bias<degF>): Refinery Pressure Compensated

Temperature

@SELECT(N, V1, V2, V3, ...): Returns Vn based on value of N

@MAPRANGE(X, L1, I1, L2, I2, ..., Ln, In, In+1): Returns range indicator In if X <= Ln

ANEXO III

FÓRMULAS PREDETERMINADAS DE BUILDER

A continuación, se listan las fórmulas que Builder proporciona para realizar cálculos con los vectores. Las fórmulas pueden combinarse entre ellas para obtener una nueva fórmula que sea de nuestro interés.

Todas las fórmulas realizan sus cálculos utilizando doble precisión de punto flotante, incluso con funciones Booleanas, en las cuales el cero es Falso y distinto de cero Verdadero.

Para usar una variable CCF en una fórmula su nombre debe satisfacer los siguientes requerimientos:

- El nombre de las variables de ser una combinación de caracteres, números y símbolos, pero deben comenzar por una letra.
- No debe sobrepasar los 15 caracteres.

Funciones matemáticas:

@MAX(value1, value2[, value3...]): maximum of a list of values

@MIN(value1, value2[, value3...]): minimum of a list of values

@LOG10(value): base 10 logarithm of value

@LOG(value): natural logarithm of value

@EXP(base): natural antilogarithm of base

@POW(base, exponent): base to the power exponent

@INTERP(value, x1, y1, x2, y2[, x3, y3...]): piece-wise linear interpolation

@SQRT(value): square root of value

@ABS(value): absolute value of value

@RAND(seed): uniformly distributed random number between the range of 0.0 and 1.0

Funciones lógicas:

@IF(test1, value1, value2): value1 if test1 is "true" or value2 if test1 is "false"

@EQ(value1, value2): "true" if value1 is equal to value2

@NE(value1, value2): "true" if value1 does not equal value2

@GT(value1, value2): "true" if value1 is greater than value2

@GE(value1, value2): "true" if value1 is greater than or equal to value2

@LT(value1, value2): "true" if value1 is less than value2

@LE(value1, value2): "true" if value1 is less than or equal to value2

@OR(value1, value2[, value3...]): "true" if any value in a list of values is true

@AND(value1, value2[, value3...]): "true" if all values in a list of values is true

@NOT(value): "true" if value is "false"

@BADVAL(value): "true" if value is less than -9998.0