

Modelling and Simulation Software

EcosimPro/PROOSIS · Newsletter Nº 14 · Marzo 2018

EDITORIAL

Queremos presentar en esta nueva edición de nuestra newsletter algunas de las novedades que hemos introducido en las nuevas versiones de nuestros productos a finales de 2017. Creemos que algunas de ellas serán muy bien recibidas por nuestros usuarios pues las llevaban pidiendo mucho tiempo. Por ejemplo la nueva capacidad para editar todos los datos de un esquemático en un nuevo editor de tipo spreadsheet. Ahora el usuario puede de un vistazo tener acceso a todos los datos de todos los objetos y cambiarlos de una manera muy sencilla. Esto supone una gran mejora en la edición de esquemáticos de simulación.

Cada vez más usuarios usan de forma intensiva el uso de las clases en EL (similares a C++ pero muy simplificadas). Cada nueva versión introduce mejoras en este área que mejoran su uso. Por ejemplo en esta nueva versión se han introducido unas nuevas clases predefinidas para permitir el cálculo matricial y algebra lineal en el lenguaje EL. Hemos optado por usar las conocidas librerías Eigen3 por debajo, pero simplificando mucho el uso de las matrices y vectores. Les invito a ver algunos ejemplos en este newsletter.

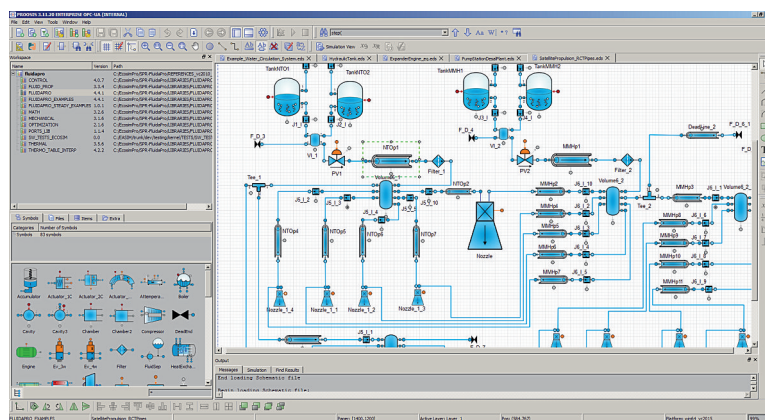
Hemos terminado la capacidad de importación modelos FMI/FMU en EcosimPro/PROOSIS. Un año antes habíamos implementado la capacidad de exportar modelos usando este estándar para co-simulación. Con esta nueva capacidad ahora también un usuario de EcosimPro puede usar modelos FMI/FMU dentro de un modelo en EcosimPro/ PROOSIS de una manera muy sencilla. Pueden ver la descripción en este número.

Estamos finalizando un toolkit de simulación aeronáutica que creemos va a ser muy importante. Permitirá modelar, además del motor que ya existía con el toolkit TURBO, los sistemas neumáticos (control ambiental del avión), termo-fluidos (combustible, aceite, etc) o los ciclos de vapor (sistema de refrigeración de cabina, etc.). Este nuevo toolkit permitirá el diseño integrado de los múltiples sistemas del avión unificados en una única herramienta.

Además presentamos en este newsletter aplicaciones muy interesantes (y complejas) en el mundo de la criogenia y el modelado de rendimiento de una turbina axial en PROOSIS. Esperamos que sean del interés de nuestros lectores.

Pedro Cobas (pce@ecosimpro.com)

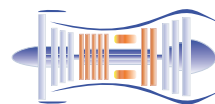
**Jefe del Equipo de Desarrollo EcosimPro/PROOSIS
EA Internacional**



Sistema de Propulsión en EcosimPro/FluidaPro

ÍNDICE

PROOSIS: Aircraft Systems Simulation Toolkit	2	Nuevos Resolvedores de Ecuaciones	8
Simulador Sistema Criogénico ITER	2	Clases para Operaciones Matriciales	8
Modelado de Bombas Criogénicas	3	Sistemas de Ecuaciones Lineales	11
Simulación de Rendimiento de una Turbina Axial en PROOSIS	4	Cálculo de Autovalores y Autovectores	11
Exportación de Modelos Usando FMI	5	Mejoras en la Comparación de Ficheros	12
Nuevo Editor de Datos	6	Documentación de Librerías	13
		Visualización Gráfica de Dependencias en un Workspace	14



1. PROOSIS: AIRCRAFT SYSTEMS SIMULATION TOOLKIT (ASYST)

FERNANDO RODRÍGUEZ, ECOSIMPRO/PROOSIS

Ya está disponible la nueva versión beta del “Aircraft systems simulation toolkit” para PROOSIS, que estará comercialmente disponible en breve, que permite la simulación de los sistemas termo-fluidos del avión como: el sistema neumático (control ambiental o ECS), hidráulico (fuel, oil) o ciclos de vapor (refrigeración, bottoming cycles, ...).

Las librerías incluidas en el toolkit son compatibles con la librería TURBO para la simulación de turbinas de gas, siendo posible la simulación conjunta del motor junto con los sistemas con los que interacciona: sistema de alimentación de combustible, lazo de refrigeración de aceite, sistema de sangrado de aire y ECS así como las interacciones térmicas entre ellos.

El toolkit comparte la filosofía de modelado y alcance de la librería TURBO. Es útil por lo tanto como herramienta de análisis de ciclo termodinámico, siendo posible realizar tanto estudios de punto de diseño (incluyendo todos los cálculos típicos en estos análisis: paramétricos, optimización, MonteCarlo...), como simulaciones off-design de un sistema en concreto (cálculos estacionarios y simulaciones dinámicas). Para las simulaciones dinámicas, las librerías incluyen los efectos transitorios más importantes (dinámicas lentas).

La flexibilidad de modelado de cualquier sistema termo-fluido, junto con la variedad de cálculos posibles hacen de PROOSIS-Aircraft Systems una herramienta muy útil para el diseño integrado de los distintos sistemas del avión (thermal management).

Aunque es útil para cualquier sistema fluido del avión, el toolkit está especialmente indicado para la simulación del ECS, siendo posible modelar cualquier configuración convencional o innovadora (more electric aircraft) basada en ciclos de aire (ACM) o en ciclos de vapor (VCM). Además del ciclo de refrigeración, es posible simular tanto el subsistema de sangrado de aire, como el subsistema de distribución de aire en cabina.

Cualquier usuario interesado puede ponerse en contacto con nosotros para obtener más información sobre este nuevo toolkit (info@ecosimpro.com).

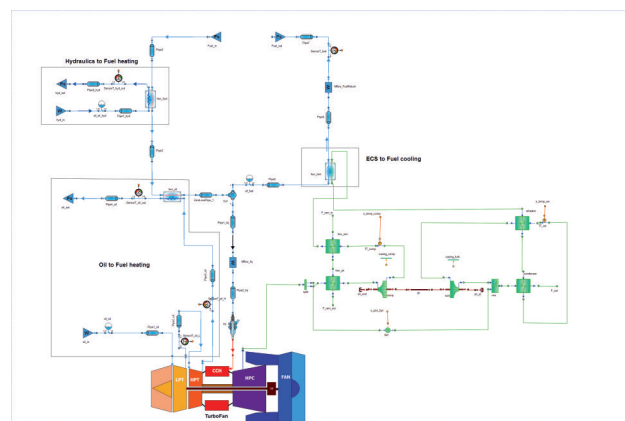


Figura 1-1 PROOSIS Aero Toolkit

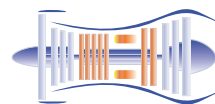
2. SIMULADOR SISTEMA CRIOGÉNICO ITER

ANA MARÍA VELEIRO, ECOSIMPRO/PROOSIS

Iter Organization tiene entre sus metas desarrollar un simulador integrado de los distintos sistemas que conforman el reactor experimental ITER que se está construyendo en Cadarache (Francia). El simulador pretende aglutinar los simuladores individuales en desarrollo de los distintos sistemas e integrarlos. El objetivo final del simulador integrado es apoyar el comisionado de ITER y el entrenamiento de operarios.

El equipo de ITER responsable del sistema de criogenia viene trabajando desde hace tiempo en el desarrollo de modelos que permitan verificar el diseño, diseñar algoritmos de control avanzado y probar el control con simulaciones hardware in the loop.

Con este objetivo el departamento de simulación de EAI ha desarrollado modelos dinámicos de los circuitos que refrigeran los imanes de ITER y modelos iniciales para las bombas criogénicas y su distribución. Uno de los problemas



Modelling and Simulation Software

EcosimPro/PROOSIS · Newsletter Nº 14 · Marzo 2018

que se plantean con estos modelos es la complejidad de los mismos pues el número de fenómenos que se simulan es enorme y presentan un reto para cualquier herramienta de simulación.

Dada la complejidad del sistema, IO se ha propuesto crear una plataforma de simulación distribuida que pueda integrar modelos de los distintos subsistemas y simularlos conjuntamente. Para lograr este objetivo EAI ha desarrollado en 2017 ciertas funcionalidades en EcosimPro para permitir la simulación distribuida de estos modelos tan complejos

En el marco de este proyecto EAI creó en EcosimPro la capacidad de generar servidores OPC UA desde la herramienta. Esto permite conectar modelos desarrollados en EcosimPro con otras herramientas que tengan una interfaz OPC UA y crear una plataforma de simulación distribuida muy potente. Asimismo se ha desarrollado un mecanismo de sincronización entre los distintos modelos que permite enlazar múltiples sistemas para realizar una simulación conjunta de todos ellos y visualizar los resultados en EcosimPro de una manera unificada.

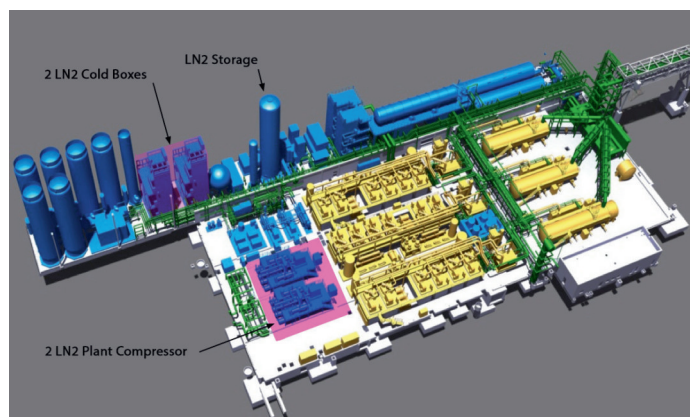


Figura 2-1 Planta Criogénica ITER

3. MODELADO DE BOMBAS CRIOGÉNICAS

ANA MARÍA VELEIRO, ECOSIMPRO/PROOSIS

El equipo de simulación de Empresarios Agrupados ha continuado realizando trabajos para el reactor experimental ITER dentro del área de fusión. En este periodo se ha realizado un prototipo del sistema de distribución criogénico de ITER (FECDS) que incluye la coldbox auxiliar (ACB) de las bombas

criogénicas, las líneas de distribución que llevan el refrigerante desde la ACB hasta la Cold Valve Box (CVB), la CVB con su control y una de las bombas criogénicas del toroide. El modelo final incluirá todas las bombas criogénicas y persigue dos objetivos fundamentales:

- La verificación del sistema de distribución (FECDS) de ITER que suministra el refrigerante a las bombas criogénicas.
- Ayudar en el desarrollo del sistema de control proporcionando información sobre la dinámica del sistema.

Las bombas criogénicas se usan en aplicaciones de alto vacío y consisten en una superficie interna (paneles criogénicos) enfriada a bajas temperaturas donde los gases y vapores se condensan. Las moléculas de gas son inmovilizadas en estas superficies, lo cual disminuye la presión del sistema. En el caso particular de ITER las bombas criogénicas tienen una doble misión, por una parte se usan tras las bombas mecánicas para reducir la presión en la vasija de vacío hasta las condiciones de presión adecuadas y por otra parte durante el funcionamiento del reactor adsorberán los gases provenientes del plasma.

Dada la dificultad intrínseca para atrapar las partículas de helio, las superficies utilizadas deben ser refrigeradas a una temperatura muy baja, 4.5K. Por otra parte, cada cierto tiempo las partículas de deuterio y tritio adsorbidas junto con las de helio serán recuperadas y tratadas en la planta de tritio para ser reintroducidas en el sistema. Por este motivo las bombas trabajan en un ciclo continuo de enfriamiento-bombeo-calentamiento moviéndose en un rango de temperaturas que van desde los 4.5K durante bombeo hasta los 470K alcanzados durante uno de los escenarios de regeneración.

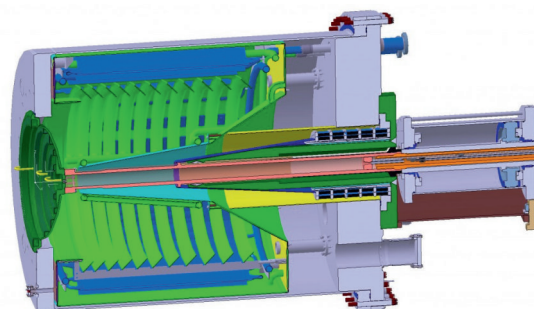
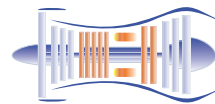


Figura 3-1 Bomba criogénica

El sistema de control del FECDS debe operar dinámicamente de forma que se cumplan los requerimientos de operación del plasma teniendo en cuenta al mismo tiempo la correcta



regeneración de las bombas. El disponer de un modelo dinámico permitirá por una parte verificar el comportamiento de los elementos del FECDS trabajando en conjunto y por otra parte diseñar y verificar el complejo sistema de control necesario para su correcto funcionamiento.

4. SIMULACIÓN DE RENDIMIENTO DE UNA TURBINA AXIAL EN PROOSIS

I. KOLIAS & A. ALEXIOU, NTUA (NATIONAL TECHNICAL UNIVERSITY OF ATHENS)

La fase de diseño preliminar de un motor de turbina de gas implica una variedad de disciplinas de ingeniería como la termodinámica y la aerodinámica (entre otras) y es un procedimiento altamente iterativo. El análisis del ciclo del motor generalmente comienza asumiendo, entre otros parámetros, la eficiencia de los componentes de turbomaquinaria. Basado en las elecciones de diseño realizadas, los cálculos de diseño aerodinámico que siguen pueden afectar las supuestas eficiencias de turbomaquinaria. De esta forma, se establece un procedimiento iterativo entre las dos disciplinas que concluye cuando las eficiencias asumidas al inicio del análisis del ciclo coinciden con los valores de cálculo de diseño aerodinámico. Entonces se pueden obtener las dimensiones y los pesos del componente básico (y en consecuencia del motor).

En el contexto del proyecto de investigación e innovación Horizon 2020 Clean Sky 2 de la UE: DEMOS (Developing advanced Engine Multi-disciplinary Optimization Simulations), el Laboratorio de Turbomáquinas Térmicas de la Universidad Técnica Nacional de Atenas desarrolló un proceso de modelización robusto para realizar esta labor. El proceso integra, al mismo nivel de modelado, el diseño aerodinámico 1D, la generación de geometría del flujo y las capacidades de estimación de peso en los modelos de rendimiento 0D para compresores multietapa y de flujo axial y turbinas.

Los nuevos componentes de turbomaquinaria que integran el diseño aerodinámico, generación de flujo y funcionalidades de estimación de peso heredan el modelo matemático (atributos, variables, puertos, ecuaciones) de la biblioteca PROOSIS-TURBO. Los procedimientos iterativos que se emplean para estimar la relación de presión de la etapa y la eficiencia isentrópica se ejecutan a nivel de etapa. De esta forma, se establece un procedimiento consistente de diseño de un solo paso donde no hay necesidad de intercambio continuo de datos entre las diferentes disciplinas a nivel de componente,

mientras que las formulaciones matemáticas existentes (número de sistemas de ecuaciones no-lineales y/o variables algebraicas) para las simulaciones de rendimiento de los componentes y todo el motor no se ven afectados y obtienen el mismo nivel de robustez y velocidad de ejecución que para los cálculos de rendimiento típicos.

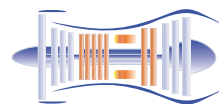
El diseño aerodinámico de los componentes del compresor y la turbina se lleva a cabo a través de una línea de valores medios. La eficiencia isoentrópica se calcula empleando correlaciones semiempíricas o de pérdida de carga. También se obtiene la eficiencia del punto de diseño general de los componentes de turbomaquinaria. Esto evita la necesidad de costosas iteraciones de componentes al asumir la eficiencia del componente al comienzo del análisis del ciclo, al tiempo que garantiza la coherencia entre el rendimiento y los cálculos de diseño aerodinámico.

A partir de los cálculos de diseño aerodinámico, también se calculan los radios del anillo de flujo por etapas y se usan para dimensionar axialmente las etapas de los componentes y para generar la geometría general del flujo de componentes del componente. Las correlaciones empíricas que utilizan parámetros de ciclo y geometría se utilizan para estimar el peso de la turbomaquinaria y otros componentes relacionados con el motor.

La metodología desarrollada en este proyecto se utiliza para realizar estudios de diseño multipunto (MPD) y optimización paramétrica de motores del tipo Ultra-High Bypass Ratio Geared Turbofan (UHBR) con Variable Pitch Fan (VPF) y tobera de área variable (VAN). Este tipo de motor tiene prevista una entrada en servicio (EIS) en 2025. Este estudio se ha hecho teniendo en cuenta consideraciones de rendimiento, aerodinámicas, estructurales y térmicas para las 3 fases más importantes del vuelo: despegue, ascenso y crucero.

Para el modelado VPF, se mejoró el componente Fan de la librería TURBO para poder simular la variación del ángulo de paso de pala y su efecto en el rendimiento del fan, mientras que para la VAN, el componente Nozzle se mejoró para permitir la simulación de toberas de área fija y variable.

La Figura 4-1 muestra el diagrama esquemático de un motor UHBRGT que incorpora los componentes recientemente desarrollados:



Modelling and Simulation Software

EcosimPro/PROOSIS · Newsletter Nº 14 · Marzo 2018

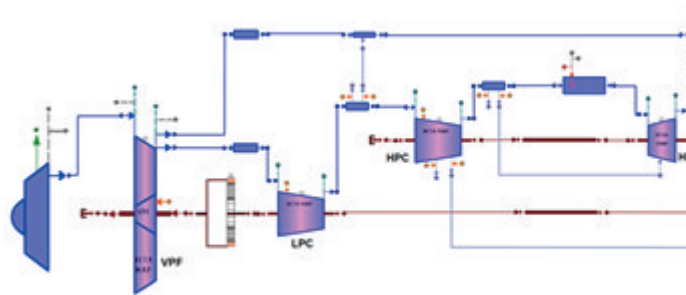


Figura 4-1 - Esquemático de un motor aeronáutico UHBRGT en PROOSIS

El rendimiento instalado del motor se calcula corrigiendo el empuje neto para la resistencia de la góndola y el peso del motor utilizando un enfoque simplificado. La Figura 4-2 es un ejemplo de un diseño 3D que se obtiene cuando se utilizan los tres métodos diferentes para establecer la eficiencia de los componentes de la turbomaquinaria en condiciones de crucero:

- Método-1: Usar el nuevo enfoque de diseño aerodinámico recientemente desarrollado;
- Método-2: El uso de correlaciones que relacionan la eficiencia politrópica y el tamaño de los componentes;
- Método-3: Uso de valores de eficiencia constantes representativos de un turbofan con un EIS previsto en 2020.

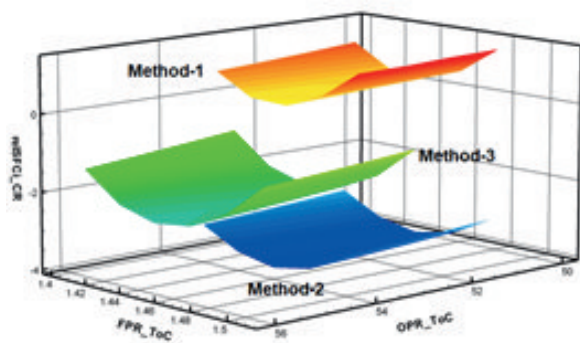


Figura 4-2 - Espacio de diseño calculado con el método multipunto

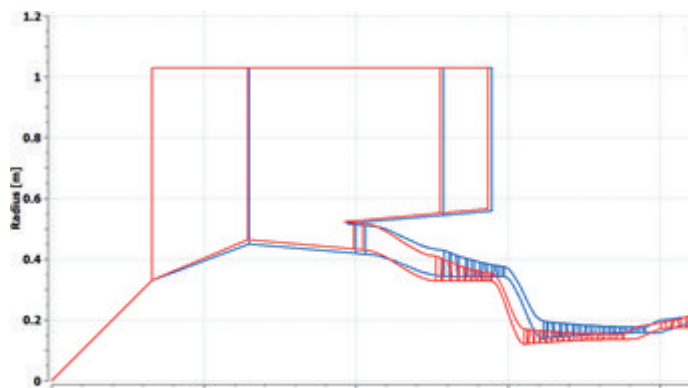


Figura 4-3 - Comparación de los flujos del motor en los métodos 1 y 3

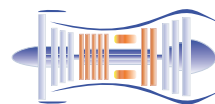
PROOSIS permite comparar el flujo en el motor utilizando los valores óptimos de los parámetros de diseño obtenidos del Método-1 y del Método-3 (Figura 4-3).

5. EXPORTACIÓN DE MODELOS USANDO FMI

FERNANDO PUECH, ECOSIMPRO/PROOSIS

Functional Mockup Interface (FMI) es un estándar para el intercambio de modelos dinámicos entre herramientas de simulación. En este estándar un FMU, o Functional Mockup Unit, es la unidad de intercambio, es decir, el modelo que implementa la interfaz FMI, sus binarios o código fuente y un fichero que describe las capacidades implementadas del estándar FMI así como información sobre el modelo tal como las variables de entrada y de salida. En su versión 2.0 el estándar define dos maneras de exportar los modelos para que sean consumidos por herramientas capaces de entender la interfaz FMI:

- Model Exchange: se exportan las ecuaciones y eventos para que un tercero los resuelva utilizando un integrador global. Esta parte del estándar no es soportada por EcosimPro ni por PROOSIS.
- Co-simulation: el modelo se exporta como una caja negra que tiene una serie de entradas y salidas y su propio integrador capaz de sincronizarse con un maestro encargado de gestionar la simulación e intercambiar los datos entre los diferentes FMU. Esta parte del estándar es la que soportan tanto EcosimPro, como PROOSIS.



EcosimPro 5.6 y PROOSIS 3.8 introdujeron la capacidad para exportar los modelos siguiendo el estándar “FMI 2.0 for Co-simulation”, lo cual permite a cualquier usuario modelar en EcosimPro y enviar el modelo para su uso por terceras herramientas que implementen el estándar FMI.

La novedad introducida por EcosimPro 5.10 y PROOSIS 3.10 es la capacidad para usar modelos FMI 2.0 para co-simulación ya estén generados utilizando EcosimPro 5.6 o superior, PROOSIS 3.8 o superior o cualquier otra herramienta capaz de exportar modelos con la interfaz FMI 2.0 para co-simulación. Con esta nueva capacidad se puede completar un esquema de co-simulación similar al siguiente utilizando PROOSIS o EcosimPro como simuladores maestros:

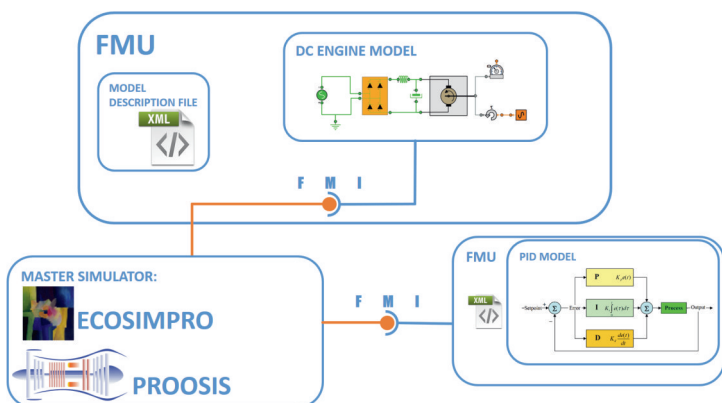


Figura 5-1 Uso de FMI en EcosimPro

Generar un FMU en EcosimPro o PROOSIS implica crear el modelo, generar un experimento sencillo en el que se integre hasta el siguiente intervalo de comunicación (CINT) y después generar un deck con la interfaz FMI 2.0 para co-simulación.

El asistente de creación del deck permite, además, elegir las variables que serán entradas y salidas del FMU. Una vez completado el asistente habrá creado un FMU listo para enviar a un tercero.

Para controlar un FMU desde EcosimPro o PROOSIS estos incluyen la librería COMM_FMI que permite crear componentes que interactúen con el FMU y utilizar estos a su vez en otros modelos más complejos o crear potentes experimentos de co-simulación utilizando el lenguaje EL, como por ejemplo el siguiente experimento en el que se cargan cuatro FMUs que contienen el modelo de un turbojet y se simulan en paralelo:

```

BODY
-
-
// Define starting integration time and time step
TIME = 0
CINT = 0.1
TSTOP = 100

// Load engine models and specify TSTOP and communication interval
FOR(eId = 1; eId <= nEngines; eId = eId + 1)
  hModel[eId] = cfmLoadModel("engine.fmu", TRUE)
  cfmSetCINT(hModel[eId], CINT)
  cfmSetStopTime(hModel[eId], TSTOP)
END FOR

// Integration loop
WHILE((TIME + CINT) < TSTOP)

  // Send to INPUTS the engines
  FOR(eId = 1; eId <= nEngines; eId = eId + 1)
    cfmSetValueReal(hModel[eId], "FN", FN[eId]) != 0)
  END FOR

  // Run a step
  FOR(eId = 1; eId <= nEngines; eId = eId + 1)
    cfmRun(hModel[eId])
  END FOR

  // Wait for the steps to complete
  FOR(eId = 1; eId <= nEngines; eId = eId + 1)
    cfmWait(hModel[eId])
  END FOR

  // Read engine outputs
  WRITE("TIME: %g\n", TIME)
  FOR(eId = 1; eId <= nEngines; eId = eId + 1)
    FN[eId] = computeFNBasedOnWF(cfmGetValueReal(hModel[eId], "WF"))
  END FOR

  // Advance time step
  TIME = TIME + CINT
END WHILE
  
```

Figura 5-2 Ejemplo de Cosimulación usando FMUs en EcosimPro

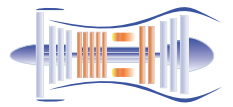
Para obtener más detalles sobre cómo generar FMUs o consumir FMUs con EcosimPro y/o PROOSIS os animamos a leer el capítulo sobre FMI contenido en el manual de EcosimPro y PROOSIS.

6. NUEVO EDITOR DE DATOS

ANTONIO RIVERO, ECOSIMPRO/PROOSIS

Una de las principales novedades de la nueva versión de EcosimPro 5.10 y PROOSIS 3.10 es el nuevo Editor de Atributos que se incluye en los esquemáticos.

Esta nueva herramienta sustituye al antiguo editor de atributos y permite al usuario final mucha más flexibilidad y rapidez en el manejo de los datos de los componentes, ya que su funcionamiento se asemeja en gran medida a los estándares que siguen las modernas hojas de cálculo.



Modelling and Simulation Software

EcosimPro/PROOSIS · Newsletter Nº 14 · Marzo 2018

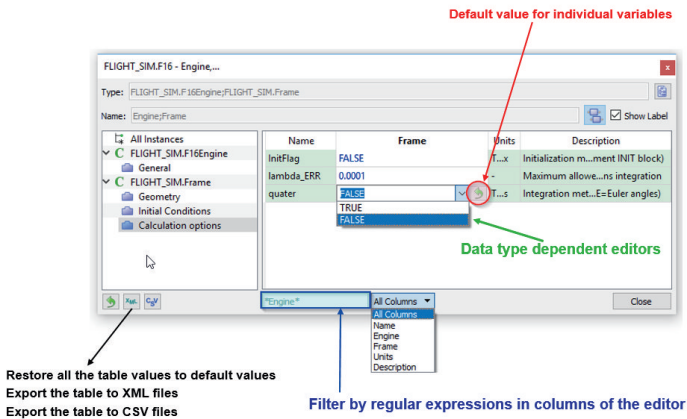


Figura 6-1 Aspecto del nuevo editor

Enumerar todas las capacidades del nuevo editor es una tarea que implica mucho detalle, por lo que aquí vamos a centrarnos en los tres aspectos esenciales, que son:

- Posibilidad de visualizar uno o varios editores a la vez
- Editores no bloqueantes que permiten trabajar en el esquemático o en la aplicación
- Edición de múltiples objetos de múltiples tipos

Efectivamente, el nuevo EcosimPro/PROOSIS permite trabajar con varios editores a la vez, incluso de distintos esquemáticos. El usuario final fácilmente puede comparar datos, copiarlos de un editor a otro, exportarlos... y todo mientras se realizan operaciones como compilar un esquemático, editar un símbolo, o lanzar el monitor para simular un experimento.

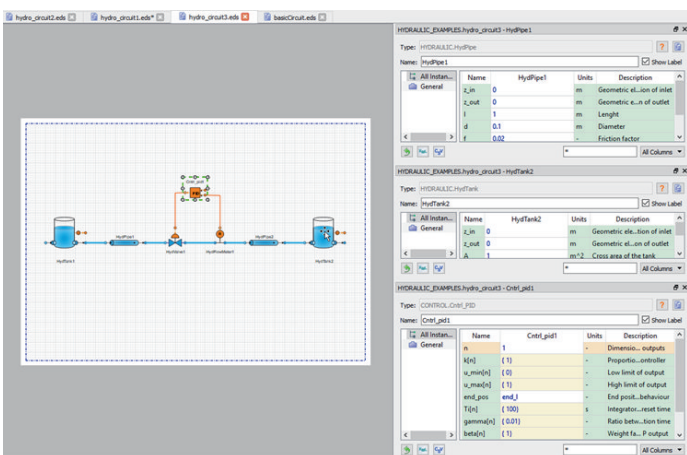


Figura 6-2 Múltiples editores en un mismo esquemático

Sin embargo, la principal característica del nuevo editor es la posibilidad de poder gestionar en una misma tabla distintos objetos de un mismo esquemático. Es decir, el usuario puede seleccionar los objetos que considere oportunos y editarlos todos a la vez. El sistema, automáticamente, se encargará de gestionar la información y agrupar los datos según su nombre, tipo y unidad. El resultado final es una hoja de cálculo donde es muy sencillo buscar, comparar y editar los valores de decenas de componentes.

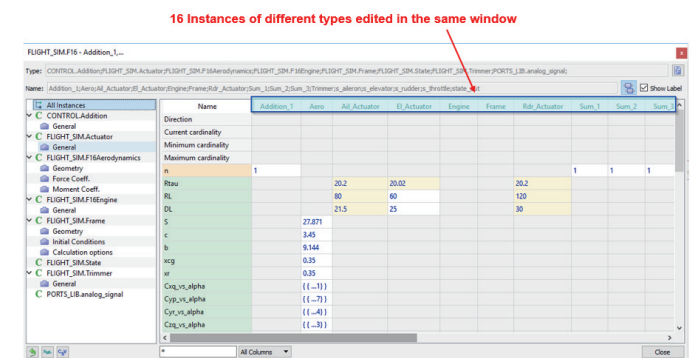


Figura 6-3 Múltiples componentes en un mismo editor

Y todo con múltiples ayudas en la edición y en el filtrado de los datos, para hacer más cómodas y rápidas las búsquedas de variables en base a su categoría, su valor, su tipo de componente, su nombre, ...

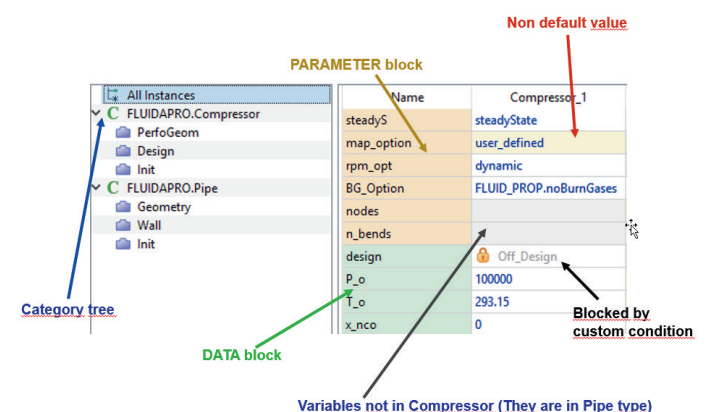
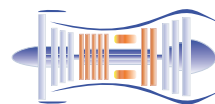


Figura 6-4



Como resumen final, podemos decir que el nuevo Editor de Atributos es una herramienta totalmente nueva, extraordinariamente versátil y potente que está diseñada y construida según de los requisitos que nuestros usuarios nos han transmitido durante estos años. Y seguiremos mejorándolo.

7. NUEVOS RESOLVEDORES DE ECUACIONES

FERNANDO PUECH, ECOSIMPRO/PROOSIS

EcosimPro y PROOSIS están en continua evolución y cuidan especialmente que sus resolvers numéricos funcionen de la manera más eficiente y robusta posible. Las últimas versiones de EcosimPro y PROOSIS incluyen varias mejoras de rendimiento, entre las que destaca la mejora en la resolución de cajas lineales que tiene el modelo. El gráfico a continuación muestra que el nuevo resolver de cajas lineales es hasta un 50% más rápido que la implementación utilizada en versiones anteriores mientras que mantiene los mismos resultados numéricos. Evidentemente, no todo el modelo son cajas lineales y, en función de la cantidad de cajas lineales y su tamaño, la influencia de esta mejora será más o menos palpable en la reducción del tiempo de simulación. Los modelos eléctricos suelen verse muy beneficiados por esta mejora.

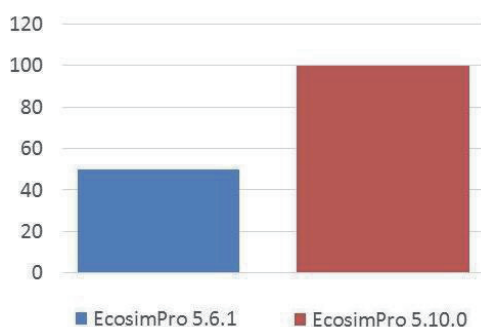


Figura 7-1 Linear box solver performance (higher is better)

Además de la mejora en la resolución de cajas lineales se ha mejorado el manejo de memoria de los resolvers CVODE e IDAS incluidos en EcosimPro y PROOSIS además de optimizar el cálculo del Jacobiano que realizan durante el proceso de resolución, obteniendo mejoras de rendimiento sustanciales en ciertas simulaciones:

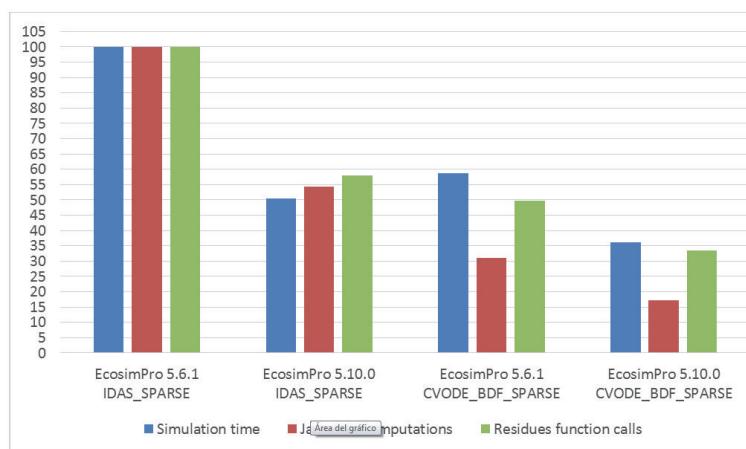


Figura 7-2 Performance of the Printing_complex ESPSS experiment using IDAS and CVODE.

EcosimPro 5.6.1 vs EcosimPro 5.10.0 (Lower relative values are better)

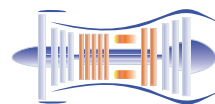
EcosimPro y PROOSIS no sólo han mejorado los resolvers existentes, sino que han añadido los siguientes para dar mayor flexibilidad al usuario a la hora de simular:

- RK45: resolver de ODE explícito y de paso variable equivalente al ode45 de Matlab. Converge más rápido que RK4.
- BACKEULER: resolver de ODE implícito y de paso fijo. Útil para resolver problemas "stiff" en situaciones en las que se requiera paso fijo.
- BACKEULER_SPARSE: resolver de ODE implícito y de paso fijo. Útil para resolver problemas "stiff" en situaciones en las que se requiera paso fijo. Este resolver es más eficiente que BACKEULER y está pensado para aprovechar la estructura del modelo reduciendo el número de veces que se llama a la función de residues cuando se calcula el Jacobiano.

8. CLASES PARA OPERACIONES MATRICIALES

PEDRO COBAS, ECOSIMPRO/PROOSIS

Las nuevas clases EigenMatrix, EigenVector y EigenRowVector permiten usar matrices y vectores de números reales en EL de una manera muy intuitiva. Por ejemplo nos permitirán operar con ellos usando los operadores típicos como +, -, *, /, etc. tal como si operaríamos con números.



Modelling and Simulation Software

EcosimPro/PROOSIS · Newsletter Nº 14 · Marzo 2018

Están basadas en una librería open source C++ llamada Eigen3 (eigen.tuxfamily.org). Es una librería muy robusta y optimizada y que usan otras herramientas como por ejemplo Google's TensorFlow for Machine Intelligence. En EL se ha implementado un subconjunto de Eigen3 pero con una sintaxis bastante más sencilla.

Además la clase auxiliar EigenHandler nos permite hacer cálculos más avanzados con estas matrices para resolver sistemas lineales y cálculo de autovalores/autovectores. Esto se explica en otra sección de este Newsletter.

La clase EigenMatrix es dinámica, es decir, no se conoce su dimensión cuando se declara, pero EL también permite trabajar con matrices estáticas. Por ejemplo se predefinen todas las matrices cuadradas hasta tamaño 10 si el nombre va seguido por su dimensión, por ejemplo la clase EigenMatrix5 define una matriz estática de tamaño 5x5. Es el usuario el que decide si usarlas de un tipo o de otro en cada caso. Nos remitimos a la documentación de EL donde se explican las ventajas e inconvenientes de cada una.

Veamos un primer ejemplo que suma dos matrices e imprime el resultado:

```
FUNCTION NO_TYPE testInitial()
DECLS
  REAL aInit[2,3]= {{1,4,1},{2,2,2}}
  REAL bInit[2,3]= {{4,6,8},{2,5,2}}
OBJECT
  EigenMatrix A, B, Result
BODY
  A.setFromArray2D(aInit,2,3)
  B.setFromArray2D(bInit,2,3)
  Result = A + B
  WRITE("A=\n%s \n", A.asString())
  WRITE("B=\n%s \n", B.asString())
  WRITE("Result= A + B =\n%s \n", Result.
    asString())
END FUNCTION
```

La salida de la función es:

```
A=
[1, 4, 1]
[2, 2, 2]
B=
[4, 6, 8]
[2, 5, 2]
Result= A + B =
[ 5, 10, 9]
[ 4, 7, 4]
```

Vemos como hemos usado el operador '+' para sumar dos matrices como haríamos con dos números.

Para la creación de las matrices hay varias formas, por ejemplo podemos copiar una matriz de EL a EigenMatrix y viceversa:

```
FUNCTION NO_TYPE testCopyFromArray2D()
DECLS
  REAL mat1[3,3] = {{1,2,3},{4,5,6},
    {7,8,9}}
  REAL mat2[3,3]
OBJECTS
  EigenMatrix a, b, c -- dynamic matrix
BODY
  a.setFromArray2D(mat1,3,3)
  b.setFromArray2D(mat1,3,3,1,2,3,1)
  WRITE("a=\n%s\n",a.asString())
  WRITE("b=\n%s\n",b.asString())
  a.copyToArray2D(mat2,3,3)

  FOR(i IN 1,3)
    FOR(j IN 1,3)
      WRITE("mat2[%d,%d]= %g\n",
        i,j,mat2[i,j])
    END FOR
  END FOR
END FUNCTION
```

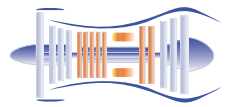
Cuando se copia con setFromArray2D() hay que definir la dimensión, pues el objeto a lo desconoce y por tanto es responsabilidad del usuario darle las dimensiones correctas. La salida de esta función es:

```
a=
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]

b=
[2]
[5]
[8]

mat2[1,1]= 1
mat2[1,2]= 2
mat2[1,3]= 3
mat2[2,1]= 4
mat2[2,2]= 5
mat2[2,3]= 6
mat2[3,1]= 7
mat2[3,2]= 8
mat2[3,3]= 9
```

Hemos copiado en 'a' la matriz completa de EL mat1, sin



embargo en 'b' hemos copiado sólo una submatriz (la columna 2). Luego hemos copiado 'a' en mat2 (otra matriz de EL) y hemos impreso el resultado.

También podríamos haber creado la matriz desde un string haciendo:

```
a.setFromStr("1,2,3;4,5,6;7,8,9")
```

Se usan los separadores ';' para separar filas y ',' para separar columnas. El resultado es el mismo que hacerlo con setFromArray2D().

También lo podríamos haber hecho de manera particular para cada elemento, pero primero dimensionando la matriz (los métodos anteriores lo hacen automáticamente):

```
a.resize(3,3)
a.set(1,1,1)
a.set(1,2,2)
a.set(1,3,3)
...
a.set(3,3,9)
```

El resultado es el mismo que los casos anteriores.

También se pueden cambiar submatrices con el método setChunk(), por ejemplo:

```
FUNCTION NO_TYPE testSetChunk()
OBJECT
  EigenMatrix a, b, c
BODY
  a.resizeInit(5,5,0)
  b.resizeInit(2,2,8)
  WRITE("a=\n%s\n",a.asString())
  WRITE("b=\n%s\n",b.asString())
  a.setChunk(3,3,2,2,b)
  WRITE("a after a.setChunk(3,3,2,2,b)=
    \n%s\n",a.asString())
END FUNCTION
```

Produce la salida

```
a=
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
```

```
b=
[8, 8]
[8, 8]
```

```
a after a.setChunk(3,3,2,2,b) =
[0, 0, 0, 0, 0]
[0, 0, 0, 0, 0]
[0, 0, 8, 8, 0]
[0, 0, 8, 8, 0]
[0, 0, 0, 0, 0]
```

La matriz 'b' ha reemplazado una submatriz de 'a' en la posición (3,3).

También se pueden hacer las típicas operaciones matriciales tales como inversa, determinante, traspuesta, etc. Por ejemplo:

```
FUNCTION NO_TYPE testGetTransf()
DECLS
  REAL v1[3,3]= {{0,1,1},{1,0,0},{0,0,1}}
OBJECTS
  EigenMatrix p1,plinv,plad,pltrans
BODY
  p1.setFromArray2D(v1,3,3)
  WRITE("p1=\n%s\n", p1.asString())
  WRITE("p1.determinant()=g\n",
    p1.determinant())
  plinv= p1.inverse()
  pltrans= p1.transpose()
  plad= p1.adjoint()
  WRITE("p1.inverse()=\n%s\n",
    plinv.asString())
  WRITE("p1.transpose()=\n%s\n",
    pltrans.asString())
  WRITE("p1.adjoint()=\n%s\n",
    plad.asString())
END FUNCTION
```

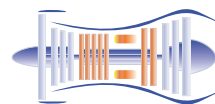
La salida es:

```
p1=
[0, 1, 1]
[1, 0, 0]
[0, 0, 1]

p1.determinant()= -1

p1.inverse()=
[ 0, 1, 0]
[ 1, 0, -1]
[ 0, 0, 1]

p1.transpose()=
[0, 1, 0]
[1, 0, 0]
[1, 0, 1]
```



Modelling and Simulation Software

EcosimPro/PROOSIS · Newsletter Nº 14 · Marzo 2018

```
p1.adjoint()=
[0, 1, 0]
[1, 0, 0]
[1, 0, 1]
```

La documentación de EL contiene muchos más ejemplos que permiten hacerse una idea de la versatilidad de estas clases.

9. SISTEMA DE ECUACIONES LINEALES

PEDRO COBAS, ECOSIMPRO/PROOSIS

La nueva clase EigneHandler tiene un método solve() para resolver sistemas lineales de ecuaciones. Usa un método de la librería C++ Eigen3 del tipo "Householder rank-revealing QR decomposition" con pivotación de columna. Los parámetros que requiere el método son:

A: una matriz cuadrada de tamaño NxN
B: un vector columna de tamaño N
sol: vector con la solución final

Este método devuelve TRUE si ha encontrado correctamente la solución. Veamos un ejemplo, supongamos que queremos resolver el sistema ecuaciones:

```
4*x + 5*y - 12*z = 98
36*x - 8*y + 3*z = 23
-3*x + 5*y - 2*z = 45
```

Habría que crear un matrix A de tamaño 3x3 con los coeficientes de "x", "y" y "z". Asimismo crearíamos un vector columna de tamaño 3 con los valores independientes y luego se los pasaríamos al método solve. La siguiente función plantea, resuelve e imprime el resultado:

```
FUNCTION NO_TYPE testSolve()
DECLS
    REAL relative_error
OBJECTS
    EigenMatrix A, sol
    EigenVector B
    EigenHandler eh
BODY
    A.setFromStr("4,5,-12;36,-8,3;
                -3,5,-2")
    B.setFromStr(" 98; 23; 45")
    WRITE("A=\n%s\n",A.asString())
    WRITE("B=\n%s\n",B.asString())
    eh.solve(A, B, sol)
    WRITE("Solution\n%s\n",
          sol.asString())
END FUNCTION
```

Produce la salida con la solución del sistema:

```
A=
[ 4, 5, -12]
[ 36, -8, 3]
[ -3, 5, -2]
```

```
B=
[98]
[23]
[45]
```

```
Solution
[ 3.02447]
[ 9.54153]
[-3.18287]
```

10. CALCULO DE AUTOVALORES Y AUTOVECTORES

PEDRO COBAS, ECOSIMPRO/PROOSIS

La nueva clase EigenHandler tiene un par de métodos que nos devuelven los autovalores y los autovectores de una matriz cuadrada. Como pueden ser complejos, lo que se hace es pasar siempre 2 matrices para cualquiera de los dos casos (autovalores y autovectores) de tal manera que si la solución es compleja, la parte real se escribe en una matriz y la parte compleja en la otra.

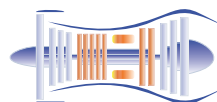
Además la clase EigenHandler dispone de un método llamado complexAsString() que pasándole las dos matrices nos escribe el resultado en formato complejo.

Su uso es muy sencillo, por ejemplo queremos calcular los autovalores y autovectores de la matriz:

```
[-3, 0, 2]
[ 1, -1, 0]
[-2, -1, 0]
```

Lo podríamos hacer en la siguiente función en EL:

```
FUNCTION NO_TYPE testEigenValues()
DECLS
    REAL Ainitial[3,3]= {{-3, 0, 2},
                        {1, -1, 0},{-2, -1, 0}}
OBJETO
    EigenMatrix A, var, vai,ver,vei
    EigenHandler eh
BODY
    A.setFromArray2D(Ainitial,3,3)
```

```
WRITE("A=\n%s\n",A.asString())
eh.eigenvalues(A,var,vai)
WRITE("Autovalores:\n%s\n",
eh.complexAsString(var,vai))
eh.eigenvectors(A,ver,vei)
WRITE("Autovectores:\n%s\n",
eh.complexAsString(ver,vei))
END FUNCTION
```

La salida que produce nos da ambos: autovalores y autovectores

```
A=
[-3, 0, 2]
[ 1, -1, 0]
[-2, -1, 0]

Autovalores:
[-1+1.41421i]
[-1-1.41421i]
[-2]

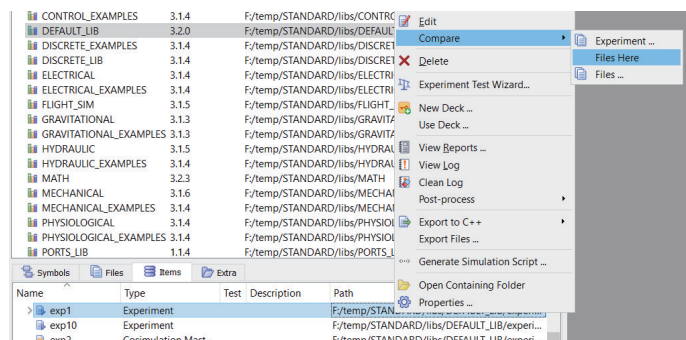
Autovectores:
[-0.540449-0.203096i,-0.540449+0.203096i,
-0.666667]
[-0.143611+0.382155i,-0.143611-0.382155i,
0.666667]
[-0.396839-0.585251i,-0.396839+0.585251i,
-0.333333]
```

En este caso son valores complejos y por tanto se escriben con la parte imaginaria. Si fueran soluciones reales la matrix imaginaria saldría a valor 0.

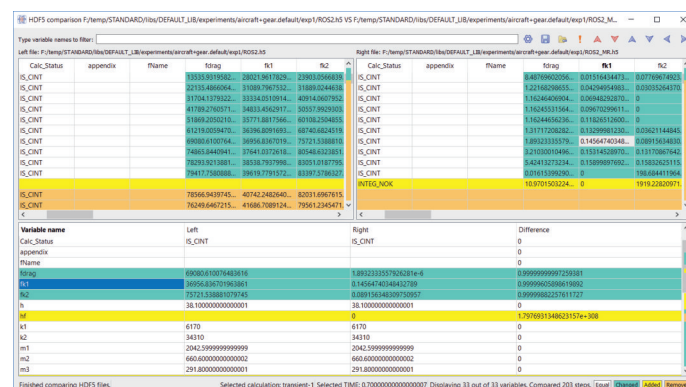
11. MEJORAS EN LA COMPARACIÓN DE FICHeros

FERNANDO PUECH, ECOSIMPRO/PROOSIS

EcosimPro y PROOSIS han unificado las opciones de comparación de ficheros para facilitar la localización de diferencias entre resultados de simulaciones, experimentos, particiones, etc. Este cambio muestra las opciones de comparación al lanzar el menú contextual de los diferentes elementos de la pestaña "Files" y la pestaña "Items":

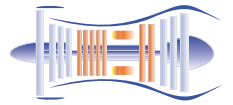


Cuando se comparan ficheros en EcosimPro y PROOSIS, estos tratan de realizar la mejor comparación que conocen en función del tipo de fichero, es decir, comparan los ficheros como si fueran texto, salvo que sean ficheros de post-proceso con extensión h5 en los que realizan una comparación binaria y detallada:



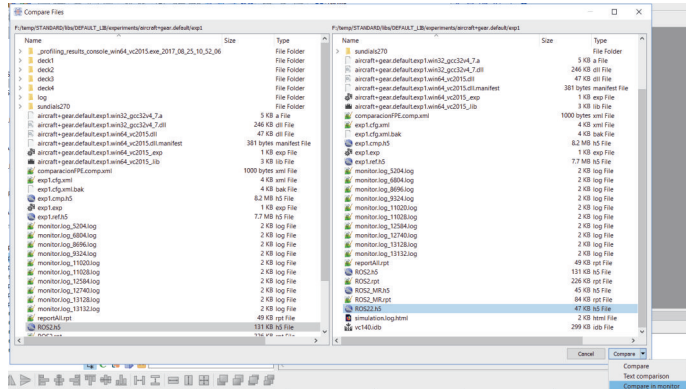
La comparación binaria de los ficheros de post-proceso permite analizar las diferencias entre los resultados de dos simulaciones comparando cada intervalo de comunicación (CINT) o cada paso de integración (STEP). Es una comparación flexible y potente que indica exactamente lo que ha pasado.

Además, la comparación de ficheros permite, en vez de detectar la mejor comparación, realizar una comparación gráfica de los ficheros de post-proceso o forzar una comparación textual de los ficheros elegidos. El siguiente diálogo muestra los ficheros que cuelgan de los elementos que se desean comparar para elegir qué ficheros comparar seleccionando uno de la parte izquierda, otro de la parte derecha y cómo comparar desplegando las opciones del botón "Compare":

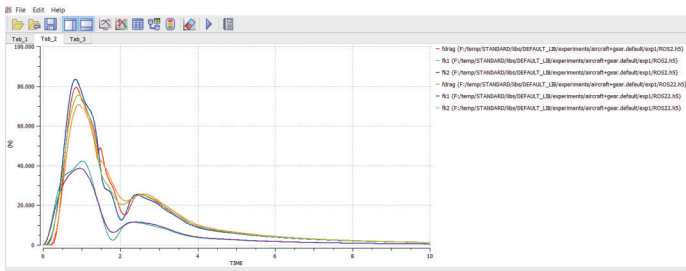


Modelling and Simulation Software

EcosimPro/PROOSIS · Newsletter Nº 14 · Marzo 2018



Si se comparan en el monitor dos ficheros de post-proceso, este permitirá comparar gráficamente aquellas variables que existen en ambos ficheros y tienen el mismo tipo, como por ejemplo se muestra en el pantallazo a continuación:



12. DOCUMENTACIÓN AUTOMÁTICA DE LIBRERÍAS

EcosimPro/PROOSIS permite generar la documentación automática de librerías, en este Newsletter nos vamos a concentrar en mostrar las distintas visualizaciones que podemos tener de los componentes. Para hacerlo vamos a usar la librería TURBOJET del workspace DEFAULT. Podemos generar la documentación automática de esta librería si seleccionamos la librería y la opción: "Documentation->Generate Documentation", nos abrirá un navegador con el siguiente cabecero:

TURBOJET Library | **Components** | **Ports** | **Classes** | **Functions** | **Enumeratives** | **Global Variables**

TURBOJET Library Documentation

Desde aquí podremos navegar para ver los distintos componentes, puertos, clases, funciones, tipos enumerados y variables globales de la librería TURBOJET. Vamos a ver un poco en detalle la información que nos da de Components. Si pinchamos en Components nos saldrá otra página con este cabecero:

TURBOJET Library | **Components** | **Ports** | **Classes** | **Functions** | **Enumeratives** | **Global Variables**

Alphabetical | **Inheritance Tree** | **Inheritance Graph**

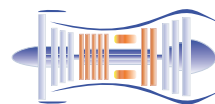
Nos da 3 vistas distintas para visualizar los componentes, una en orden alfabético, otra con la jerarquía de herencia en modo texto y otra con la jerarquía de herencia en modo gráfico. En la primera vemos una tabla con los componentes, su icono y su descripción:

Components

Note: click [+] for accessing the source code file

Symbol	Name	Description
	AfterBurner [+]	Afterburner
	Burner [+]	Burner
	Compressor [+]	Compressor

Si pinchamos en el signo [+] accederemos directamente al código fuente de ese component (sólo si está disponible). Si seleccionamos la opción "Inheritance Tree" veremos:

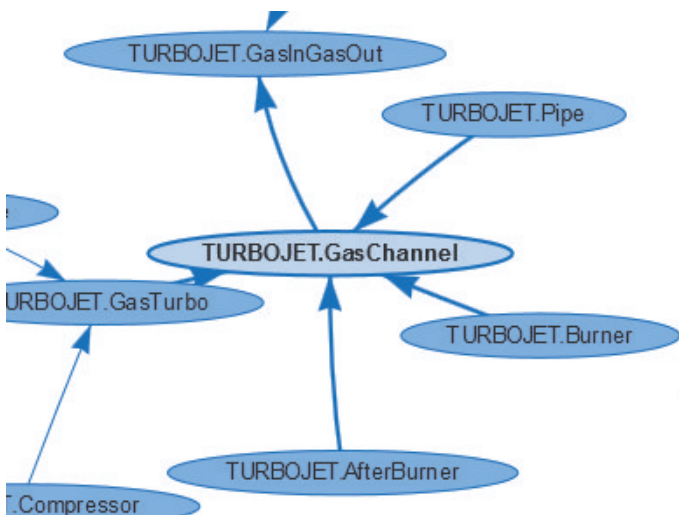


Component Inheritance Tree

Note: click [+] for accessing the source code file

- FuelTank [+]
- FuelValve [+]
- Fuelmeter [+]
- GasInGasOut [+]
 - GasChannel [+]
 - AfterBurner [+]
 - Burner [+]
 - GasTurbo [+]
 - Compressor [+]
 - Fan [+]
 - Turbine [+]
 - Pipe [+]
 - Plane [+]

Se muestra la jerarquía de la herencia entre componentes. Por ejemplo GasChannel está heredado de GasInGasOut y AfterBurner está heredado a su vez de GasChannel. Si queremos ver esta información en modo gráfico podemos seleccionar la vista "Inheritance Graph" y nos saldrá en pantalla un grafo como:



Cada burbuja representa un componente, cuando una burbuja tiene una flecha hacia otra burbuja significa que está heredado de ese componente. Por ejemplo vemos como GasChannel apunta a GasInGasOut indicando que está

heredado de este y así sucesivamente con los demás componentes. Es una vista bastante interesante pues de un vistazo tenemos toda la relación de herencia entre todos los componentes de la librería.

Además el usuario puede interactuar con el grafo, pues puede reposicionar las burbujas con el ratón para tener otra disposición de las mismas, al hacerlo las demás burbujas se moverán automáticamente para dejar paso a esta burbuja. Recomendamos al usuario llegar a esta visualización y "jugar" con esta visión de alto nivel de la librería.

13. VISUALIZACIÓN GRÁFICA DE DEPENDENCIAS EN UN WORKSPACE

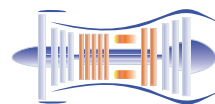
Muchas veces trabajando en un workspace estamos interesados en conocer las dependencias entre librerías, es decir cuando una librería usa elementos de otras librerías. Esta información nos indica que necesitamos tener todas ellas para compilar la librería final. Además se debe compilar en el orden adecuado para que todos los componentes, puertos, clases y funciones se compilen en orden cronológico de uso.

Existe una utilidad gráfica en la herramienta que permite una visualización gráfica de alto nivel de estas dependencias entre librerías. Sólo tenemos que seleccionar la opción "Properties" asociada a un workspace y en el tab "Information" nos saldrá un grafo con las dependencias. Por ejemplo si lo hacemos en el workspace TURBO nos muestra la ventana de abajo:

En este caso vemos como la librería TURBO usa las librerías MATH, PORTS_LIB y CONTROL. Otras librerías como GTE_TURBOSHAFT y GTE_EXAMPLES usan a su vez la librería TURBO.

El usuario puede mover las burbujas con el ratón y reposicionarlas para tener otra vista del grafo, esto permite crearse vistas más a medida para cada usuario.

Hemos visto de una manera rápida y visual la dependencia entre librerías que en muchos casos puede venir bien para tener una idea más clara de su orden de compilación y de sus dependencias.



Modelling and Simulation Software

EcosimPro/PROOSIS · Newsletter Nº 14 · Marzo 2018

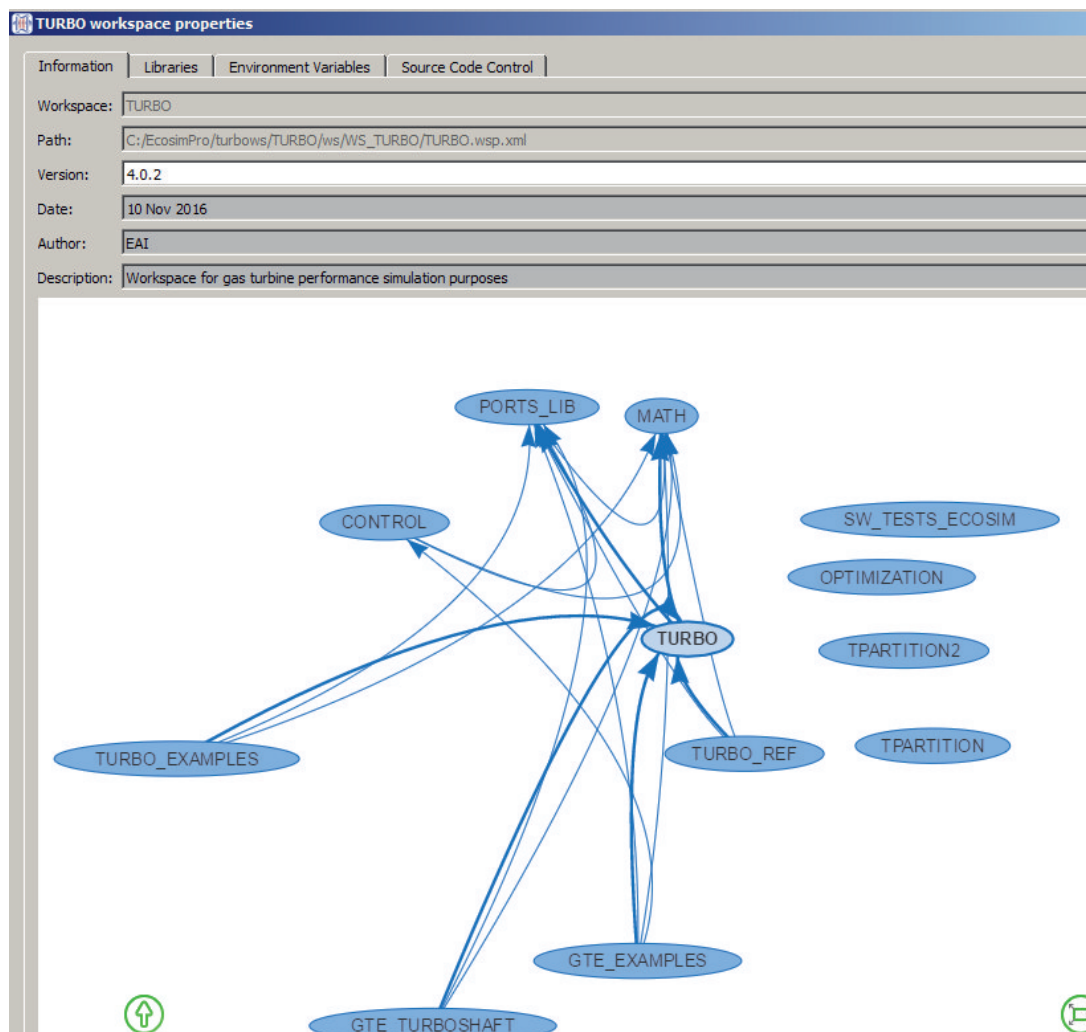


Figura 13-1 Vista gráfica de dependencias entre Librerías en un Workspace

EA Internacional S.A.
Magallanes, 3 Madrid
28015 Spain
E-mail: info@ecosimpro.com
URL: <http://www.ecosimpro.com>
Phone: +34 91 309 81 42
Fax: +34 91 591 26 55



MAQUETADO POR: INMACULADA REYES
REVISADO POR: ÁNGEL BARRASA