

# ATP

# USO DE MODELS

IIE - FING - UDELAR

# TACS

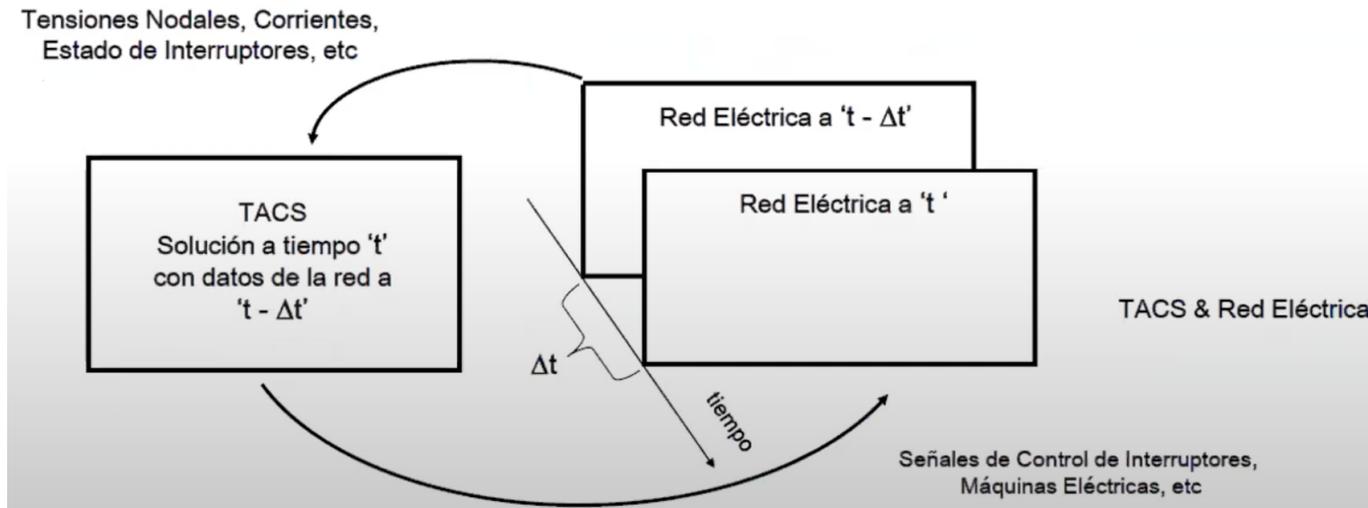
El ATP/EMTP realiza tres tipos de simulaciones en el dominio del tiempo :

- ▶ Red Eléctrica sin sistemas de control (sin TACS)
  - ▶ TACS STAND ALONE (sólo TACS)
  - ▶ TACS HYBRID (TACS + Red Eléctrica)
- 
- ▶ En estas dos últimas, en los archivos con datos para la simulación se incorpora una especie de "Computador Analógico", al cual suele hacerse referencia con el nombre de su acrónimo: TACS (Transient Analysis of Control Systems).
  - ▶ Esta funcionalidad permite simular Sistemas de Control de la Excitación y/o velocidad de Generadores, ángulos de disparo de convertidores, FACTS, comportamientos no lineales(cámaras de extinción de interruptores, cebado de aisladores), etc.

# TACS

## La Red Eléctrica & TACS:

- ▶ Es posible el intercambio de información entre la Red Eléctrica (elementos “de Potencia”) y los elementos de control (TACS). Advertir que existirá un retardo de tiempo 'delta', igual a un paso de integración, en el cálculo entre ellos.



# MODELS

## DEFINICIÓN:

- ▶ Es un lenguaje técnico de uso general apoyado por un conjunto de herramientas de simulación para la representación y estudio de sistemas variables en el tiempo.

## APLICACIONES GENERALES:

- ▶ Es un lenguaje apuntado a la simulación en el dominio del tiempo, y es usado como una herramienta para describir el comportamiento dinámico de sistemas físicos complejos.
- ▶ Se puede utilizar para describir la operación tanto de componentes del circuito como de componentes de control.
- ▶ Puede ser usado para generar señales o para analizar medidas del circuito.
- ▶ Puede ser usado como una interfase entre ATP y programas externos.

# MODELS

## APLICACIONES GENERALES:

- ▶ Un modelo consiste en procedimientos que describen cómo opera, cuáles elementos almacenarán el estado y la historia de su operación para intercambiarla con el exterior, y las directivas que influyen en la operación del modelo a lo largo de la simulación.

# MODELS

## CARACTERÍSTICAS GENERALES:

- ▶ Tiene una sintaxis “habitual” para el uso de variables, vectores, matrices, expresiones y funciones.
- ▶ Admite varios tipos de variables, almacenadores de valor modificados por los procedimientos del modelo; las variables pueden ser numéricas o lógicas, bien sea en escalares o vectores.
- ▶ Maneja funciones:
  - ▶ Numéricas y lógicas predefinidas
  - ▶ Funciones adicionales que pueden ser parametrizadas por el usuario
  - ▶ Funciones programadas íntegramente por el usuario.

# MODELS

## CARACTERÍSTICAS GENERALES:

- ▶ Permite el uso de condiciones y bucles típicos (if, while, for, etc.)
- ▶ Admite incorporar el uso de comentarios para una descripción del código. (TACS no lo permite)
- ▶ Admite el uso de nombres largos y arbitrarios para facilitar la documentación. (TACS limita a 6 caracteres)
- ▶ Manejo de variables internas individuales a cada modelo.
- ▶ Cada modelo es nombrado en forma individual, permitiendo instanciar un mismo código.

# MODELS

## CARACTERÍSTICAS GENERALES:

- ▶ Separación de la descripción de un modelo de su uso en una simulación.
- ▶ Permite el uso de múltiples instancias del mismo componente, con la posibilidad de darles diferentes parámetros, diferentes entradas o diferentes directivas de simulación.
- ▶ Posibilidad de especificar operaciones que son ejecutadas sólo cuando algunas condiciones se satisfacen, por ejemplo para la inicialización de modelos, ante errores de simulación, etc.
- ▶ El tiempo automáticamente se incrementa durante una simulación.
- ▶ Las instancias individuales de los modelos de un sistema se actualizan automáticamente de acuerdo a su operación.

# MODELS

## TACS vs MODELS

- ▶ En TACS todas las variables del sistema están en el mismo nivel, no existen variables locales. Esto es una dificultad para utilizar varios modelos desarrollados en TACS en un mismo archivo de simulación ya que no se pueden repetir los nombres de las variables.
- ▶ En MODELS el usuario no está restringido a seleccionar dentro de un conjunto fijo de elementos, ni a un flujo simple de señales de entrada/salida, puede definir cualquier elemento utilizando el lenguaje de programación que provee MODELS, y efectuar cualquier manipulación numérica y/o lógica de las variables.
- ▶ En MODELS se distingue claramente entre la definición del modelo y su uso. La definición contiene declaraciones de variables, constantes, señales de entrada y salida, procedimientos, funciones y la forma en que se ejecuta el algoritmo que describe al modelo.

# MODELS

## TACS vs MODELS

- ▶ En MODELS los modelos pueden usarse repetidamente en un mismo caso, y cada vez con sus propios valores de variables, y aún también con diferentes pasos de tiempo. Esto se hace instanciando el modelo.
- ▶ En MODELS es posible detectar cualquier error que lleve a un mal funcionamiento del modelo, pudiéndose forzar la interrupción de la simulación en cualquier momento determinado.

# MODELS: CONSTANTES GLOBALES

Novarían durante la simulación y están predefinidas.

## ▶ Numéricas:

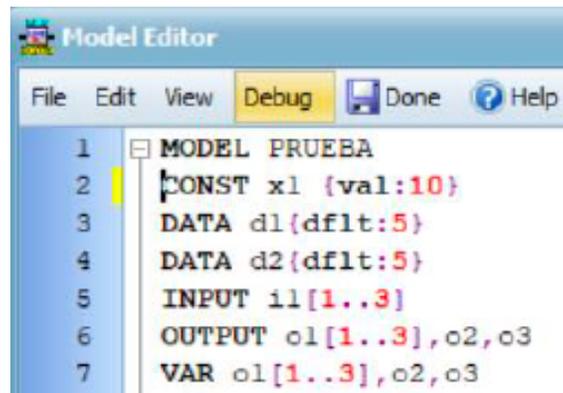
- pi=elnúmeropi,3.141592...
- inf=elvalornuméricomásgrandequepuedemanejar suPC
- undefined=88888.88888

## ▶ Lógicas:

- Igualesacero(0):false,no,open,off
- Igualesauno(1):true,yes,close,on

# MODELS: CONSTANTES GLOBALES

- ▶ Los nombres de los elementos CONST, sólo son visibles dentro del modelo donde son declaradas.
- ▶ Sus valores son asignados dentro del modelo, en la declaración.
- ▶ El mismo valor es usado en todas las instancias del modelo.
- ▶ Definición de nombres de parámetros y su valor por defecto (esto último es opcional).
- ▶ Todo dato de entrada que no provenga de una fuente externa al modelo (por ejemplo del EMTP u otro modelo) deben ser declaradas aquí.



```
Model Editor
File Edit View Debug Done Help
1 MODEL PRUEBA
2 CONST x1 {val:10}
3 DATA d1{dflt:5}
4 DATA d2{dflt:5}
5 INPUT i1[1..3]
6 OUTPUT o1[1..3],o2,o3
7 VAR o1[1..3],o2,o3
```

# MODELS: VARIABLES GLOBALES Y LOCALES

Variables globales iguales para todos los Models:

- ▶ Startime= valor del tiempo al comienzo de la simulación.
- ▶ Stoptime= tiempo en el cual termina la simulación.
- ▶ Startstep= paso de tiempo asociado a la simulación

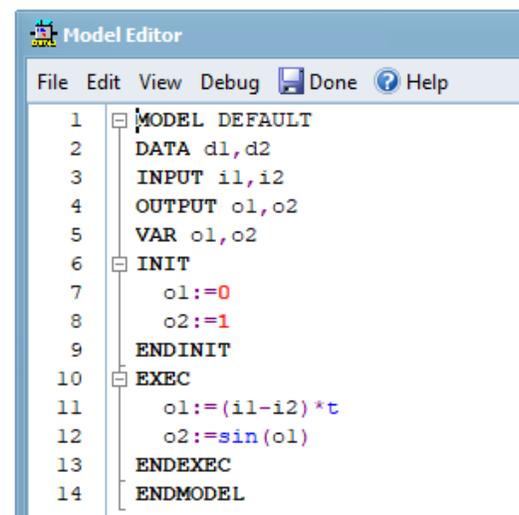
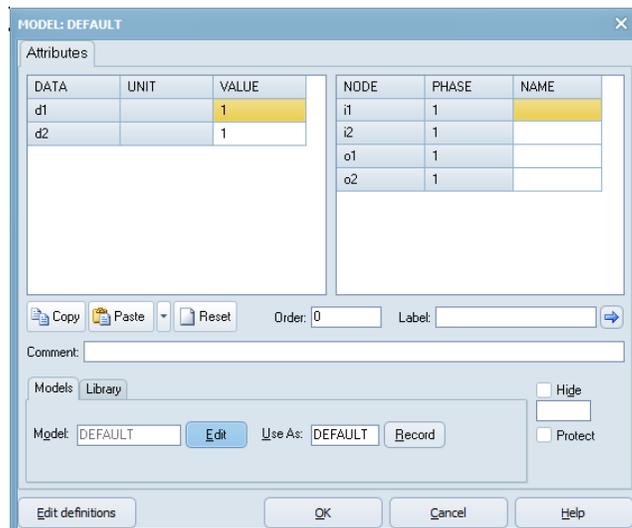
Variables locales:

- ▶  $t$ = valor presente del tiempo de simulación en cada modelo
- ▶ prevtime= valor previo del tiempo de simulación del modelo
- ▶ timestep= el paso de tiempo de simulación en el modelo
- ▶  $timestep = t - prevtime$
- ▶ fullstep= intervalo de tiempo total sobre el cual el modelo será actualizado

# MODELS: INPUT

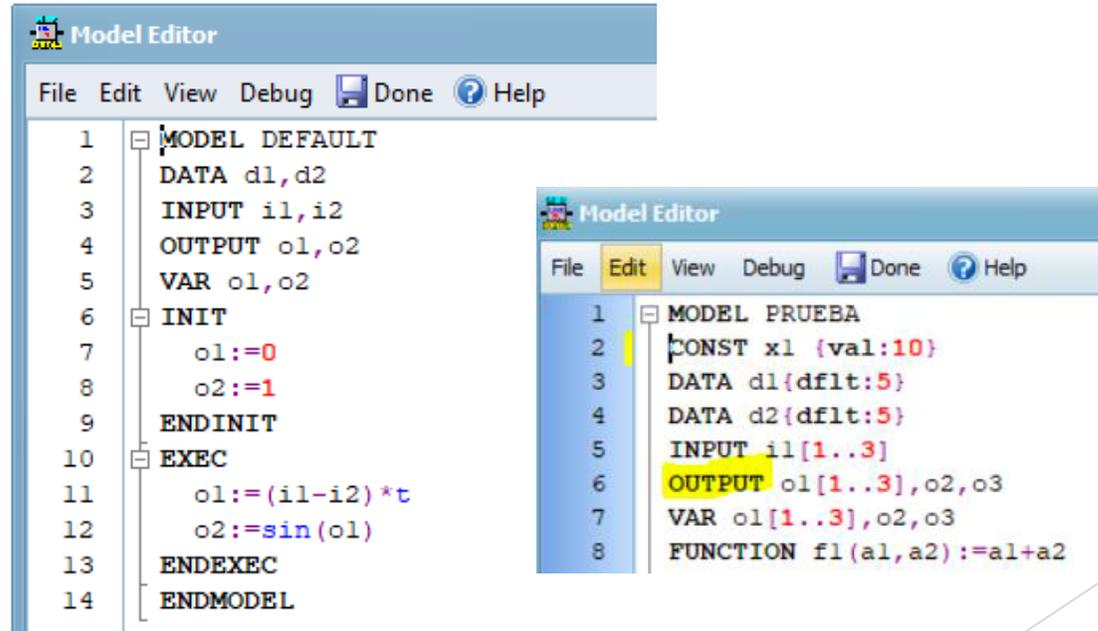
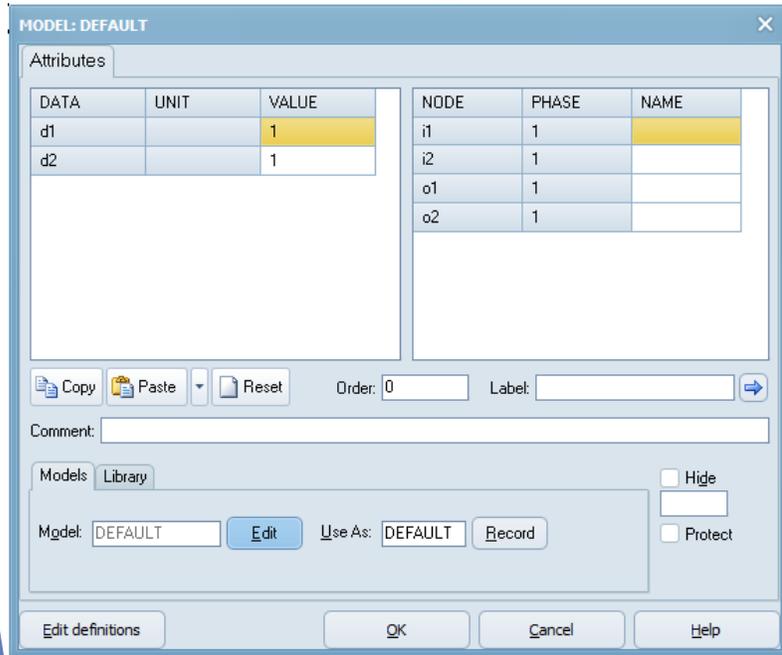
Indica las variables que serán pasadas desde el circuito eléctrico hacia MODELS, en cada paso de tiempo. Las mismas pueden ser:

- ▶ Variables pasadas desde el circuito (ATP)
- ▶ Variables de la sección TACS
- ▶ Valores extraídos de un archivo tipo PL4
- ▶ Valores internos del ATP (como timestep, tmax, etc.; que MODELS puede modificar)



# MODELS: OUTPUT

- ▶ Asigna nombre a las variables del circuito eléctrico que serán recibidas desde el modelo.
- ▶ Las salidas deben ser definidas como variables (VAR) dentro del modelo.



# MODELS: FUNCTION

## ► Las funciones pueden ser divididas en:

- Definidas por el usuario

```
FUNCTION name ( argument-name ) := expr
```

FUNCTION nombre(var1,...,varn) = f (var1,...,varn)

y :=f (x1,...,xn)

- Los argumentos de las funciones son locales y no deben ser declarados bajo la directiva VAR.
- La función debe estar definida dentro del modelo donde se utiliza o declararla EXTERNAL.

FUNCTION nombre EXTERNAL

# MODELS: FUNCTION

- ▶ Las funciones pueden ser divididas en:
  - Definidas en un lenguaje distinto a MODELS (FOREIGN FUNCTIONS)
    - MODELS tiene una interface predefinida que permite utilizar funciones escritas en otros lenguajes. Estas funciones deben ser compiladas en el EMTP, generando un nuevo archivo “TPBIG.exe”

# MODELS: FUNCTION

▶ Las funciones pueden ser divididas en:

➤ Definidas punto a punto (POINTLIST)

- Es una lista de coordenadas ordenadas en forma creciente en función de la coordenada "X",  $\{(x_k, y_k), k=1..n\}$
- La función es definida de -inf a +inf, pudiendo elegirse entre tres opciones: interpolación lineal, interpolación cuadrática o mantener el valor anterior.

➤ Residentes

- Numéricas y lógicas
- Funciones de simulación (funciones relacionadas con el tiempo de simulación)

# MODELS: DIRECTIVAS DE SIMULACIÓN

- ▶ Las directivas que afectan al modelo son:
  - Manejo del *tamaño del time step* usado por el modelo.
  - El método de *interpolation* a ser usado sobre los input en pasos de tiempo internos intermedios, asociados a la ejecución interna del MODEL. (lineal, cuadrática o constante entre pasos de integración)
  - El número de *delay cells* a ser reservado para guardar los valores pasados de variables e inputs afectadas por la función **delay()**.
  - Las expresiones de **history** describiendo analíticamente el valor de algunos de los inputs y las variables de un modelo antes de su simulación.

# MODELS: DIRECTIVAS DE SIMULACIÓN

- ▶ Las directivas que afectan al modelo son:
  - Manejo del *tamaño del time step* usado por el modelo.

El paso de tiempo externo que utiliza el modelo para actualizar su estado puede no estar en una escala adecuada para el MODEL.

Especificando un límite mínimo y/o un máximo se puede mantener en un rango adecuado.

Si el paso de tiempo externo es mayor que el máximo especificado, el modelo subdividirá el paso externo en menores pasos internos. Si, por el contrario, el paso de tiempo externo es menor al mínimo especificado, el modelo ignorará la actualización requerida.

Las expresiones usadas para describir estos límites pueden hacer referencia a los valores de alguna variable del modelo, así se pueden haber ajustes dinámicos de estos límites.

Típicamente el tamaño del timestep se mantendrá en 1/100 y 1/10 veces el tamaño de la constante de tiempo más pequeña del modelo.

# MODELS: DIRECTIVAS DE SIMULACIÓN

- ▶ Las directivas que afectan al modelo son:
  - Método de *interpolation*

Si el timestep es limitado a un valor máximo y el paso externo excede ese valor, no habrá valores para los inputs intermedios. Por ello se usa una directiva de interpolación para especificar el método: 0, 1 ó 2, para paso, lineal e interpolación cuadrática.

Se pueden seleccionar diferentes métodos para cada uno de los inputs. También el método puede ser modificado posteriormente en USE para cada modelo.

# MODELS: DIRECTIVAS DE SIMULACIÓN

- ▶ Las directivas que afectan al modelo son:
  - El *delay cells es* el tamaño del espacio reservado necesario para grabar los valores pasados de una variable o entrada accesada por un función delay().

El tamaño del espacio necesario para grabar los valores pasados de una variable accesada por un función delay() depende de que tamaño tenga el retardo utilizado, y el tamaño del paso de tiempo al cual las muestras son tomadas. Hacer esta estimación no es simple.

Debe considerarse que tanto el tamaño del retardo como el del paso de tiempo pueden ser expresados como expresión simbólica y este valor puede variar dinámicamente durante una simulación.

Típicamente, la expresión del número de celdas incluirá una referencia al tamaño del paso de tiempo usado en el modelo. Esta expresión es evaluada sólo una vez cuando cada instancia del modelo es creada y por ello el tamaño no podrá ser ajustado luego. Una referencia a la variable timestep se hará usando el valor de esa variable en el tiempo en el que la instancia del modelo es creada e inicializada.

# MODELS: DIRECTIVAS DE SIMULACIÓN

- ▶ Las directivas que afectan al modelo son:
  - Las expresiones de **history** describen analíticamente el valor de algunos de los inputs y las variables de un modelo antes de su simulación.

Son posibles dos tipos de referencias a una expresión HISTORY de una variable o input. Una es hecha internamente por el resolutor cuando necesita acceder a un valor pasado no-existente de un valor  $t$  previo a la simulación del modelo.

El otro tipo de referencia es el que se escribe explícitamente en la descripción del modelo.

Las expresiones **HISTORY** pueden ser usadas también para obtener los valores iniciales. Se pueden definir valores históricos como expresiones regulares en función del tiempo.

Es posible especificarlas en tres lugares del modelo: como directiva de simulación en la definición, como directiva de simulación en USE (para cada instancia) y en INIT.

# MODELS: DIRECTIVAS DE SIMULACIÓN

- ▶ Las directivas que afectan al modelo son:

---

```
HISTORY y1  
      y2 {dflt:0}  
      y3 {dflt:A*sin(omega*t)}
```

```
HISTORY integral(y) {dflt:express.}  
HISTORY y[n1..n2] {dflt:express.}  
HISTORY y[n1..n2] {dflt:array exp.}
```

---

# MODELS: DIRECTIVAS DE SIMULACIÓN

MODEL <NombreNodo> -- Indica el comienzo de la definición de un modelo.

-- Sección de declaraciones CONST ...

DATA ...

INPUT ...

OUTPUT ...

FUNCTION ...

-- Directivas de simulación

TIMESTEP

INTERPOLATION

HISTORY

DELAY CELLS

-- Procedimientos

INIT -- {  
... -- } Este procedimiento se ejecuta solo una vez para

ENDINIT -- inicializar las variables.

EXEC -- {  
... -- } Este procedimiento se ejecuta cada paso de tiempo.

ENDEXEC --

ENDMODEL -- Fin de la definición del modelo.