DESARROLLO DE UNA BIBLIOTECA DE DISPOSITIVOS ELÉCTRICOS NO LINEALES

Francisco Javier Torres Ramírez
Avda. Dr. Federico Rubio y Galí 1, 9º-2
28039 Madrid
fjavitorres@yahoo.es

Resumen

En el marco de un proyecto fin de carrera de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid se desarrolló una biblioteca de dispositivos eléctricos no lineales que aprovecha las posibilidades de EcosimPro en cuanto a modelos algebraico-diferenciales. Esta biblioteca posibilita modelar cualquier tipo de dispositivos no lineales, al permitir la introducción de relaciones constitutivas de toda índole. En el presente trabajo se describen los dispositivos incluidos, una descripción de la herramienta utilizada, EcosimPro, así como diversos ejemplos ilustrativos.

Palabras Clave: EcosimPro, simulación, circuitos eléctricos no lineales.

1 INTRODUCCIÓN

Se ha desarrollado una biblioteca de dispositivos eléctricos no lineales para el paquete informático EcosimPro de la empresa EA International. El presente trabajo está dedicado a la descripción de esta biblioteca. Se ha utilizado este programa informático por dos razones fundamentales. En primer lugar, se trata de una herramienta de modelado general orientada a objetos que permite construir una biblioteca de forma modular y con total libertad para modelar los dispositivos. Por otra parte, es capaz de simular modelos basados en ecuaciones algebraico-diferenciales, a través de un tratamiento simbólico de ecuaciones y de un integrador numérico específico para DAEs. Además, este desarrollo supone un complemento para dicho paquete.

El desarrollo del proyecto fin de carrera del que este trabajo forma parte dio lugar a las comunicaciones en congresos recogidas en [13] y [14].

En la sección 2 se describe brevemente el programa EcosimPro. No se detalla la sintaxis del mismo, sino los principios básicos del modelado en dicho programa, que permitan al lector entender el código fuente y la estructura de la biblioteca. Para una información más detallada de este paquete se remite al lector a los manuales del programa, o a la página web http://www.ecosimpro.com.

En la siguiente sección se explica la estructura de la biblioteca desarrollada, complementando las explicaciones anteriores sobre aspectos concretos. Se discuten las decisiones tomadas a la hora de modelar ciertas configuraciones (acoplos, cuadripolos, fuentes controladas, etc.) o almacenar ciertas variables.

La sección 4 está dedicada a la exposición de todos los componentes de la biblioteca. Se incluye el código fuente, así como una breve descripción de cada elemento. Para facilitar la lectura se agrupan los componentes en distintas subsecciones según su naturaleza.

En la quinta sección se exponen diversos ejemplos de modelado de circuitos eléctricos utilizando la biblioteca diseñada, con una doble finalidad. Por una parte, se pretende complementar la explicación de la biblioteca, facilitando su entendimiento. Por otra parte, algunos de los ejemplos señalados ilustran la naturaleza algebraico-diferencial del modelo, constituyendo verdaderos casos de interés. En este sentido, se han escogido algunos ejemplos de [11] y [12].

Finalmente, en la última sección, se extraen conclusiones y se proponen futuros trabajos de investigación.

2 DESCRIPCIÓN GENERAL DE ECOSIMPRO

EcosimPro es una herramienta de modelado orientado a objetos para sistemas continuos y discretos. Una característica del mismo de especial interés para este trabajo es su aplicabilidad a problemas físicos modelables mediante ecuaciones algebraico-diferenciales. No entraremos en su capacidad para modelar eventos discretos. Además de los circuitos eléctricos, EcosimPro tiene aplicación en otros ámbitos muy diversos, como los sistemas de control, termohidráulicos o de propulsión.

En cuanto a las características matemáticas de EcosimPro, cabe destacar que incluye ciertos algoritmos simbólicos, además de los métodos

numéricos. Se realiza una descripción del procesado matemático de EcosimPro en la sección 2.5.

El modelado orientado a objetos de este paquete tiene las siguientes características:

- **Encapsulación**. Se encapsulan datos y comportamiento en cada componente.

- **Ocultamiento de la información**. Se define una parte pública y otra privada en los componentes.

- **Composición o agregación**. Se pueden crear nuevos componentes formados por modelos ya desarrollados interconectados entre sí.

- **Herencia**. Se pueden definir nuevos componentes derivándolos de otros ya construidos. Además, existe la posibilidad de crear componentes abstractos.

Para implementar un modelo en EcosimPro y simularlo, deben seguirse cuatro pasos, si bien desaparece el primero de ellos cuando se utiliza una biblioteca predefinida. Este proceso se detalla en las secciones 2.1 – 2.4.

2.1 MODELOS DE COMPONENTES

En una biblioteca (LIBRARY) se encapsulan funciones (FUNCTION), puertos (PORT), componentes (COMPONENT), así como variables y tipos de enumeración globales, almacenándose en un archivo .el, de texto plano sin formato. Se pueden utilizar elementos de otras bibliotecas mediante la directiva USE. A continuación se ha compilar dicho archivo, generando un código binario con los componentes, sobre el que se podrán construir particiones y experimentos.

Nos centraremos en la descripción de puertos y componentes, por ser bs de mayor interés para la biblioteca desarrollada.

2.1.1 Puertos

Un puerto es la única vía de comunicación entre un componente y el exterior. Permite representar la interconexión de dos componentes, añadiendo las ecuaciones de conexión y encapsulando las variables que se intercambian. Además las variables que se encuentran dentro de esta estructura son públicas, lo que es de gran utilidad para el modelado de sistemas.

Deben declararse de manera independiente al comienzo de la biblioteca, para después poder incluirlos dentro del cuerpo de cada componente que los implemente.

2.1.2 Componentes

Son los elementos de modelado más importantes. Un componente representa un sistema o parte del mismo por medio de constantes, variables, ecuaciones algebraico-diferenciales, topología, puertos y eventos discretos.

Como ya se comentó anteriormente, los puertos representan el intercambio de variables con otros componentes. La topología (TOPOLOGY) permite comprobar que la conexión entre los componentes es correcta, al establecer un camino cerrado entre ellos, de manera que no quede ningún puerto sin conectar. Las variables (declaradas en DECLS) y constantes (DATA) locales son privadas. Las ecuaciones se incluyen en la sección CONTINUOUS, existiendo la posibilidad de modelos discretos (DISCRETE), así como secuencias de inicialización de componentes (INIT), en las que no entraremos.

Otro aspecto importante de los componentes es la herencia. Un componente se puede derivar de otro a partir de la directiva IS_A. Por otra parte, es posible la creación de componentes abstractos (directiva ABSTRACT).

2.2 MODELOS DE SISTEMAS

Dada la estructura modular de EcosimPro, se construyen los modelos de los sistemas a partir de los modelos de componentes de las bibliotecas existentes (o creadas por el usuario). Se crea, por tanto, un COMPONENT para cada sistema a simular. En primer lugar se declaran los componentes, en la sección TOPOLOGY, asignándoles un nombre y dando valores a los parámetros que los caracterizan. La interconexión de los mismos, a través de sus puertos, se realiza mediante la sentencia CONNECT, también en la sección TOPOLOGY.

2.3 DEFINICIÓN DE UNA PARTICIÓN

La partición (PARTITION) completa el modelo matemático de un componente. En este sentido, se pueden realizar hasta cuatro operaciones distintas en esta estructura:

- Determinar las variables sobre las que se impondrán las condiciones de contorno (el establecimiento de las mismas se realizará en el experimento).

- Indicar a EcosimPro cómo *romper* los lazos algebraicos, seleccionado la variable a utilizar en el algoritmo de rotura de lazos.

- Seleccionar las variables de estado que se utilizarán para realizar la reducción del modelo en el caso de índice superior.

- Cambiar una constante para realizar simulaciones paramétricas.

Existe un asistente que facilita la realización de la partición, así como la posibilidad de realizar particiones totalmente automáticas.

2.4 SIMULACIÓN DE UN EXPERIMENTO

En el experimento es donde se realiza la simulación en sí. Se fijan las condiciones iniciales (INIT) y de contorno (BOUNDS), consistentemente con la partición realizada. Se establece el paso de integración y valor final para cada variable (u otra información equivalente) sobre la que se realiza la integración. Se pueden definir experimentos complejos, utilizando funciones que proporciona el programa. Se puede seleccionar qué variables del modelo mostrar, ya sea en forma numérica o gráfica. Para el manejo de estos parámetros existe un interfaz gráfico que ayuda al usuario.

Se comentará por último, que existen funciones de excitación que permiten imponer como condiciones de contorno formas de onda típicas (pulso, rampa, etc.).

2.5 PROCESADO MATEMÁTICO EN ECOSIMPRO

Una de las principales características de EcosimPro es el procesado simbólico que realiza sobre las ecuaciones generadas antes de aplicar métodos numéricos. Además, permite observar al usuario las ecuaciones finales obtenidas. Parte de estos procesos se realizan durante la definición de la partición, por lo que se han enumerado anteriormente. Los pasos de este proceso simbólico-numérico son:

- Identificación y resolución simbólica de sistemas lineales de ecuaciones con coeficientes constantes.

- Detección y reducción de problemas de índice alto. El proceso anterior elimina algunos casos sencillos. Para el resto se utiliza el algoritmo de Pantelides [9].

- Selección de variables de contorno, en función del número de incógnitas y de ecuaciones. Para ello se aplica el algoritmo del máximo transversal [3], según el cual se asigna una variable a cada ecuación. En este algoritmo no se consideran como variables las que aparecen derivadas.

- Ordenación de ecuaciones mediante el algoritmo BLT (*Block Lower Triangularization*), [4]. Si el sistema de ecuaciones es explícito, se consigue una matriz triangular inferior y es posible despejar de cada ecuación la variable correspondiente, pasándose directamente a la integración numérica. En caso contrario, es necesario el paso siguiente.

- El algoritmo BLT identifica bloques de subsistemas de ecuaciones acoplados (lazos algebraicos). Para los lazos algebraicos lineales se utiliza un algoritmo de resolución de ecuaciones lineales. Para los lazos algebraicos no lineales se utiliza un algoritmo de *rotura* de lazos, basado en métodos heurísticos. No hay algoritmos que permitan encontrar una rotura óptima, por lo que el usuario puede cambiar esta selección al definir la partición.

- Integración numérica. Se puede utilizar método Runge-Kutta clásico de orden 4, o el código más sofisticado de resolución numérica de ecuaciones algebraico-diferenciales DASSL [1,10], basado en métodos BDF.

3 ESTRUCTURA DE LA BIBLIOTECA: PUERTOS Y COMPONENTES ABSTRACTOS

En esta sección se explican las características fundamentales de la biblioteca, así como las decisiones tomadas a la hora de implementar ciertas configuraciones circuitales. Esta exposición se desarrolla sobre el código fuente de los puertos y los componentes abstractos, a partir de los que se construyen los modelos de los dispositivos.

3.1 PUERTOS

En los puertos se deben integrar variables públicas asociadas a los componentes creados. En este sentido se definen puertos para flujo (FLUXPORT), carga (CHARGEPORT) o variables tensión/corriente de rama (VARIABLES). El puerto ELEC tiene una doble función. Por una parte, incluye las tensiones de nodo. En segundo lugar, es el utilizado para la interconexión de componentes, implementando la ley de Kirchhoff de las corrientes (de ahí la directiva SUM, que iguala a cero la suma de las corrientes incidentes a un nodo).

El puerto ANALOG_SIGNAL tiene un propósito radicalmente distinto a los anteriores. Permite la introducción de funciones o señales en los componentes. Se utilizará para introducir las funciones de excitación de las fuentes, y para

introducir las relaciones constitutivas de los dispositivos no lineales. En este sentido, se deja plena libertad al usuario para describir dicha relación constitutiva.

Se recomienda introducir funciones que relacionen las magnitudes teóricamente involucradas para no desvirtuar los componentes, si bien se pueden utilizar otras variables para modelar dispositivos no habituales. Así, por ejemplo, se puede hacer que la capacidad de un condensador dependa de otras variables del circuito distintas de la tensión entre sus terminales.

```
-------------------------------------------------------------
-- Port flux
-------------------------------------------------------------
PORT fluxport
        REAL phi
END PORT
-------------------------------------------------------------
-- Port charge
-------------------------------------------------------------
PORT chargeport
        REAL q
END PORT

-------------------------------------------------------------
-- Port variables
-------------------------------------------------------------
PORT variables
        REAL v
        REAL i
END PORT

-------------------------------------------------------------
-- Analog signal
-------------------------------------------------------------
PORT analog_signal SINGLE IN
        EQUAL OUT REAL signal
END PORT

-------------------------------------------------------------
-- Electrical port
-------------------------------------------------------------
PORT Elec
        SUM REAL i
        EQUAL REAL v
END PORT
```

3.2 COMPONENTES ABSTRACTOS

Se han diseñado dos componentes abstractos que describen dispositivos de 2 y 4 terminales. Cada terminal se representa mediante un puerto ELEC, correspondiente a un nodo. Asimismo, se incluye un puerto VARIABLES por cada rama considerada. Además, se implementa la ley de Kirchhoff de las tensiones, utilizando como variables tensiones de nodo y tensiones de rama.

El componente FOURPINS se utilizará tanto para representar dispositivos intrínsecamente de cuatro terminales, como para modelar acoplos y fuentes controladas. En cuanto a los componentes de tres terminales, se ha optado por un doble enfoque:

- En el caso de los transistores se han definido componentes que incluyen tres puertos de tipo ELEC. Estos componentes no se derivan de ningún componente abstracto, lo que permite utilizar los nombres habituales para dichos terminales.

- Para los amplificadores operacionales, se deriva un componente a partir del TWOPINS, añadiéndole un puerto ELEC que modela el terminal de salida.

```
-------------------------------------------------------------
-- Abstract component with an input and an output
-------------------------------------------------------------
ABSTRACT COMPONENT TwoPins
  PORTS
    IN  Elec e_p
    OUT Elec e_n
    OUT variables branch
  TOPOLOGY
    PATH e_p TO e_n
  CONTINUOUS
     e_p.i = e_n.i
     branch.v = e_p.v - e_n.v
     branch.i = e_p.i
END COMPONENT

-------------------------------------------------------------
-- Abstract component with two inputs and two outputs
-------------------------------------------------------------
ABSTRACT COMPONENT FourPins
  PORTS
    IN  Elec e1_p
    OUT Elec e1_n
    IN  Elec e2_p
    OUT Elec e2_n
    OUT variables branch1
    OUT variables branch2
  TOPOLOGY
    PATH e1_p TO e1_n
    PATH e2_p TO e2_n
  CONTINUOUS
    e1_p.i = e1_n.i
    e2_p.i = e2_n.i
    branch1.v = e1_p.v - e1_n.v
    branch2.v = e2_p.v - e2_n.v
    branch1.i = e1_p.i
    branch2.i = e2_p.i
END COMPONENT
```

4 DESCRIPCIÓN DE LOS COMPONENTES

A continuación se describen los componentes de la biblioteca. Se incluye el código fuente y una breve descripción, clasificados según la naturaleza de los dispositivos.

4.1 DISPOSITIVOS REACTIVOS

4.1.1 Condensadores y bobinas de dos terminales

Se implementan versiones lineales y no lineales de dispositivos de esta naturaleza. Para componentes no lineales se proponen las configuraciones más habituales, debiendo introducirse las relaciones constitutivas a través de puertos del tipo ANALOG_SIGNAL.

El código fuente de estos dispositivos se muestra a continuación.

```
-------------------------------------------------------------
-- Nonlinear capacitor
-------------------------------------------------------------
COMPONENT C_NONLINEAR IS_A TwoPins
  PORTS
    IN analog_signal f_q_v
    OUT chargeport charge
  CONTINUOUS
    charge.q'= branch.i
    0 = f_q_v.signal
END COMPONENT

-------------------------------------------------------------
-- Nonlinear charge controlled capacitor
-------------------------------------------------------------
COMPONENT C_NONLINEAR_Q IS_A TwoPins
  PORTS
    IN analog_signal f_q
    OUT chargeport charge
  CONTINUOUS
    charge.q'= branch.i
    branch.v = f_q.signal
END COMPONENT

-------------------------------------------------------------
-- Nonlinear voltage controlled capacitor
-------------------------------------------------------------
COMPONENT C_NONLINEAR_V IS_A TwoPins
  PORTS
    IN analog_signal f_v
    OUT chargeport charge
  CONTINUOUS
   charge.q'= branch.i
   charge.q = f_v.signal
END COMPONENT

-------------------------------------------------------------
-- Nonlinear capacitor with C=C(v)
-------------------------------------------------------------
COMPONENT CAPACITOR_C_V IS_A TwoPins
  PORTS
    IN analog_signal C_v
  CONTINUOUS
    branch.i = C_v.signal*branch.v'
END COMPONENT

-------------------------------------------------------------
-- Linear capacitor
-------------------------------------------------------------
COMPONENT CAPACITOR IS_A TwoPins
  DATA
    REAL C= 1.e-9  RANGE 0, Inf
  CONTINUOUS
    branch.v'= branch.i / C
END COMPONENT


-------------------------------------------------------------
-- Nonlinear inductance
-------------------------------------------------------------
COMPONENT L_NONLINEAR IS_A TwoPins
  PORTS
    IN analog_signal f_phi_i
    OUT fluxport flux
  CONTINUOUS
    flux.phi'= branch.v
    0 = f_phi_i.signal
END COMPONENT

-------------------------------------------------------------
-- Nonlinear flux controlled inductance
-------------------------------------------------------------
COMPONENT L_NONLINEAR_FLUX IS_A TwoPins
  PORTS
    IN analog_signal f_phi
    OUT fluxport flux
  CONTINUOUS
    flux.phi'= branch.v
    branch.i = f_phi.signal
END COMPONENT

-------------------------------------------------------------
-- Nonlinear current controlled inductance
-------------------------------------------------------------
COMPONENT L_NONLINEAR_I IS_A TwoPins
  PORTS
    IN analog_signal f_i
    OUT fluxport flux
  CONTINUOUS
    flux.phi'= branch.v
    flux.phi = f_i.signal
END COMPONENT

-------------------------------------------------------------
-- Nonlinear inductance with L=L(i)
-------------------------------------------------------------
```

```
COMPONENT INDUCTOR_L_I IS_A TwoPins
   PORTS
      IN analog_signal L_i
   CONTINUOUS
      branch.v = L_i.signal*branch.i'
END COMPONENT

-----------------------------------------------------------------
-- Linear inductance
-----------------------------------------------------------------
COMPONENT INDUCTOR IS_A TwoPins
   DATA
      REAL L = 1.e-3 RANGE 0, Inf
  CONTINUOUS
      branch.v = L * branch.i'
END COMPONENT
```

4.1.2 Condensadores y bobinas acoplados. Transformadores

Los dispositivos reactivos acoplados se modelan utilizando un componente abstracto FOURPINS. Se incluyen, de nuevo, tanto dispositivos lineales como no lineales. No obstante, en este caso, se han reducido las configuraciones no lineales a las más habituales.

Se han desarrollado componentes que incluyen acoplos *directos* e *inversos* lo que, unido a que el coeficiente de acoplamiento mutuo M puede admitir ambos signos, permite modelar cualquier configuración.

En último lugar se ha incluido un transformador ideal. Nótese que se pueden modelar transformadores muchos más complejos utilizando bobinas no lineales acopladas.

```
-----------------------------------------------------------------
-- Two Linear Coupled Inductors
-----------------------------------------------------------------
COMPONENT COUPLED_INDUCTORS IS_A FourPins
   DATA
      REAL L1 = 1.e-3 RANGE 0, Inf
      REAL L2 = 1.e-3 RANGE 0, Inf
      REAL M = 1.e-3 RANGE -Inf, Inf
   CONTINUOUS
      branch1.v = L1*branch1.i' + M*branch2.i'
      branch2.v = L2*branch2.i' + M*branch1.i'
END COMPONENT

-----------------------------------------------------------------
-- Two Linear Inversely Coupled Inductors
-----------------------------------------------------------------
COMPONENT INV_COUPLED_INDUCTORS IS_A FourPins
      DATA
      REAL L1 = 1.e-3 RANGE 0, Inf
      REAL L2 = 1.e-3 RANGE 0, Inf
      REAL M = 1.e-3 RANGE -Inf, Inf
   CONTINUOUS
      branch1.v = L1*branch1.i' + M*branch2.i'
      branch2.v = L2*branch2.i' - M*branch1.i'
END COMPONENT

-----------------------------------------------------------------
-- Two Nonlinear Coupled Inductors with L=L(i)
-----------------------------------------------------------------
COMPONENT COUPLED_INDUCTORS_L_I IS_A FourPins
   PORTS
      IN analog_signal L1_i1
      IN analog_signal L2_i2
      IN analog_signal M_i1_i2
   CONTINUOUS
      branch1.v = L1_i1.signal*branch1.i' + M_i1_i2.signal*branch2.i'
      branch2.v = L2_i2.signal*branch2.i' + M_i1_i2.signal*branch1.i'
END COMPONENT

-----------------------------------------------------------------
-- Two Nonlinear Inversely Coupled Inductors with L=L(i)
-----------------------------------------------------------------
COMPONENT INV_COUPLED_INDUCTORS_L_I IS_A FourPins
   PORTS
      IN analog_signal L1_i1
      IN analog_signal L2_i2
      IN analog_signal M_i1_i2
   CONTINUOUS
      branch1.v = L1_i1.signal*branch1.i' + M_i1_i2.signal*branch2.i'
      branch2.v = L2_i2.signal*branch2.i' - M_i1_i2.signal*branch1.i'
END COMPONENT

-----------------------------------------------------------------
-- Two Linear Coupled Capacitors
-----------------------------------------------------------------
COMPONENT COUPLED_CAPACITORS IS_A FourPins
   DATA
      REAL C1= 1.e-9  RANGE 0, Inf
      REAL C2= 1.e-9  RANGE 0, Inf
      REAL M= 1.e-9  RANGE -Inf, Inf
   CONTINUOUS
      branch1.i = C1*branch1.v' + M*branch2.v'
      branch2.i = C2*branch2.v' + M*branch1.v'
END COMPONENT

-----------------------------------------------------------------
-- Two Linear Inversely Coupled Capacitors
-----------------------------------------------------------------
COMPONENT INV_COUPLED_CAPACITORS IS_A FourPins
   DATA
      REAL C1= 1.e-9  RANGE 0, Inf
      REAL C2= 1.e-9  RANGE 0, Inf
```

```
    REAL M= 1.e-9 RANGE -Inf, Inf
  CONTINUOUS
    branch1.i = C1*branch1.v' + M*branch2.v'
    branch2.i = C2*branch2.v' - M*branch1.v'
END COMPONENT

-------------------------------------------------------------
-- Two Nonlinear Coupled CAPACITORS with C=C(v)
-------------------------------------------------------------
COMPONENT COUPLED_CAPACITORS_C_V IS_A FourPins
  PORTS
    IN analog_signal C1_v1
    IN analog_signal C2_v2
    IN analog_signal M_v1_v2
  CONTINUOUS
    branch1.i = C1_v1.signal*branch1.v' + M_v1_v2.signal*branch2.v'
    branch2.i = C2_v2.signal*branch2.v' + M_v1_v2.signal*branch1.v'
END COMPONENT

-------------------------------------------------------------
-- Two Nonlinear Inversely Coupled CAPACITORS with C=C(v)
-------------------------------------------------------------
COMPONENT INV_COUPLED_CAPACITORS_C_V IS_A FourPins
  PORTS
    IN analog_signal C1_v1
    IN analog_signal C2_v2
    IN analog_signal M_v1_v2
  CONTINUOUS
    branch1.i = C1_v1.signal*branch1.v' + M_v1_v2.signal*branch2.v'
    branch2.i = C2_v2.signal*branch2.v' - M_v1_v2.signal*branch1.v'
END COMPONENT

-------------------------------------------------------------
-- Ideal transformer
-------------------------------------------------------------
COMPONENT TRANSFORMER IS_A FourPins
  DATA
    REAL n = 10 RANGE -Inf, Inf "turns ratio"
  CONTINUOUS
    branch1.v = n*branch2.v
    branch2.i = -n*branch1.i
END COMPONENT
```

4.2 DISPOSITIVOS RESISTIVOS

4.2.1 Resistencias de dos terminales

De manera análoga a los dispositivos reactivos, se incluyen resistencias lineales y no lineales de todo tipo. Para las no lineales se debe introducir la relación que liga tensión y corriente a través de los puertos de tipo ANALOG_SIGNAL.

```
-------------------------------------------------------------
-- Nonlinear Resistor
-------------------------------------------------------------
COMPONENT R_NONLINEAR IS_A TwoPins
  PORTS
    IN analog_signal f_v_i
  CONTINUOUS
    0 = f_v_i.signal
END COMPONENT

-------------------------------------------------------------
-- Nonlinear voltage controlled Resistor
-------------------------------------------------------------
COMPONENT R_NONLINEAR_V IS_A TwoPins
  PORTS
    IN analog_signal f_v
  CONTINUOUS
    branch.i = f_v.signal
END COMPONENT

-------------------------------------------------------------
-- Nonlinear current controlled Resistor
-------------------------------------------------------------
COMPONENT R_NONLINEAR_I IS_A TwoPins
  PORTS
    IN analog_signal f_i
  CONTINUOUS
    branch.v = f_i.signal
END COMPONENT

-------------------------------------------------------------
-- Linear Resistor
-------------------------------------------------------------
COMPONENT RESISTOR IS_A TwoPins
  DATA
    REAL R = 1000 RANGE -Inf, Inf "Resistance (Ohms)"
  CONTINUOUS
    branch.v = branch.i * R
END COMPONENT
```

4.2.2 Cuadripolos

Se incluyen distintos modelos de cuadripolos resistivos. Es posible la descripción de un cuadripolo lineal a partir de los matrices de parámetros más habituales: Z, Y, H y T. Para el caso no lineal, el componente TWOPORTS_NONLINEAR permite introducir dos ecuaciones cualesquiera que modelen su comportamiento. Se puede aprovechar incluso para modelar dispositivos reactivos no lineales, introduciendo ecuaciones diferenciales en las relaciones constitutivas.

```
-----------------------------------------------------------------
-- Linear Two Ports component described by Z parameters
-----------------------------------------------------------------
COMPONENT TWOPORTS_Z IS_A FourPins
  DATA
    REAL Z11 = 1000  RANGE -Inf, Inf
    REAL Z12 = 1000  RANGE -Inf, Inf
    REAL Z21 = 1000  RANGE -Inf, Inf
    REAL Z22 = 1000  RANGE -Inf, Inf
  CONTINUOUS
    branch1.v = Z11*branch1.i + Z12*branch2.i
    branch2.v = Z21*branch1.i + Z22*branch2.i
END COMPONENT

-----------------------------------------------------------------
-- Linear Two Ports component described by Y parameters
-----------------------------------------------------------------
COMPONENT TWOPORTS_Y IS_A FourPins
  DATA
    REAL Y11 = 0.01  RANGE -Inf, Inf
    REAL Y12 = 0.01  RANGE -Inf, Inf
    REAL Y21 = 0.01  RANGE -Inf, Inf
    REAL Y22 = 0.01  RANGE -Inf, Inf
  CONTINUOUS
    branch1.i = Y11*branch1.v + Y12*branch2.v
    branch2.i = Y21*branch1.v + Y22*branch2.v
END COMPONENT

-----------------------------------------------------------------
-- Linear Two Ports component described by H parameters
-----------------------------------------------------------------
COMPONENT TWOPORTS_H IS_A FourPins
  DATA
    REAL H11 = 1  RANGE -Inf, Inf
    REAL H12 = 1  RANGE -Inf, Inf
    REAL H21 = 1  RANGE -Inf, Inf
    REAL H22 = 1  RANGE -Inf, Inf
  CONTINUOUS
    branch1.v = H11*branch1.i + H12*branch2.v
    branch2.i = H21*branch1.i + H22*branch2.v
END COMPONENT

-----------------------------------------------------------------
-- Linear Two Ports component described by T parameters
-----------------------------------------------------------------
COMPONENT TWOPORTS_T IS_A FourPins
  DATA
    REAL T11 = 1  RANGE -Inf, Inf
    REAL T12 = 1  RANGE -Inf, Inf
    REAL T21 = 1  RANGE -Inf, Inf
    REAL T22 = 1  RANGE -Inf, Inf
  CONTINUOUS
    branch1.v = T11*branch2.v - T12*branch2.i
    branch1.i = T21*branch2.v - T22*branch2.i
END COMPONENT
```

```
-----------------------------------------------------------------
-- Nonlinear Two  Ports component described by 2 general equations
-----------------------------------------------------------------
COMPONENT TWOPORTS_NONLINEAR IS_A FourPins
  PORTS
    IN analog_signal eq1
    IN analog_signal eq2
  CONTINUOUS
    eq1.signal=0
    eq2.signal=0
END COMPONENT
```

4.3 FUENTES DE TENSIÓN Y CORRIENTE

En esta biblioteca se incluye todo tipo de fuentes de tensión y corriente. Las fuentes no controladas pueden ser de continua, sinusoidales o descritas por cualquier otra relación temporal. Se pueden utilizar estas últimas (V_VAR e I_VAR) para modelar fuentes controladas no lineales, sin más que introducir una función que dependa de otras variables circuitales. Las fuentes controladas lineales se modelan como cuadripolos, donde se ha utilizado la notación de [2].

El hecho de que aparezca un signo negativo en las fuentes de corriente independientes se debe a que en estos dispositivos se considera la corriente con sentido positivo saliente en lugar de entrante, como es habitual en dispositivos activos.

```
-----------------------------------------------------------------
-- Continuous voltage generator
-----------------------------------------------------------------
COMPONENT VDC IS_A TwoPins
  DATA
    REAL VDC
  CONTINUOUS
    branch.v = VDC
END COMPONENT

-----------------------------------------------------------------
-- Alternating voltage generator
-----------------------------------------------------------------
COMPONENT VAC IS_A TwoPins
  DATA
    REAL VAC = 10
    REAL freq = 50.
    REAL phase= 0.
    REAL VDC =  0.
  CONTINUOUS
      branch.v =  VAC * sin(2 * PI * (freq * TIME + phase/360)) - VDC
END COMPONENT
```

```
-------------------------------------------------------------
-- Variable voltage generator
-------------------------------------------------------------
COMPONENT V_var IS_A TwoPins
   PORTS
      IN analog_signal s_in
   CONTINUOUS
      branch.v = s_in.signal
END COMPONENT

-------------------------------------------------------------
-- Continuous current source
-------------------------------------------------------------
COMPONENT IDC IS_A TwoPins
   DATA
      REAL IDC
    CONTINUOUS
      branch.i = - IDC
END COMPONENT

-------------------------------------------------------------
-- Alternating current source
-------------------------------------------------------------
COMPONENT IAC IS_A TwoPins
   DATA
      REAL IAC = 10
      REAL freq = 50.
      REAL phase= 0.
      REAL IDC =   0.
   CONTINUOUS
       branch.i = - IAC * sin(2 * PI * (freq * TIME +
phase/360)) + IDC
END COMPONENT

-------------------------------------------------------------
-- Variable current source
-------------------------------------------------------------
COMPONENT I_var IS_A TwoPins
   PORTS
      IN analog_signal s_in
   CONTINUOUS
      branch.i = - s_in.signal
END COMPONENT

-------------------------------------------------------------
-- Linear current-controlled voltage source
-------------------------------------------------------------
COMPONENT CCVS IS_A FourPins
   DATA
      REAL rm = 1000 RANGE -Inf, Inf
"Transresistance (Ohms)"
   CONTINUOUS
      branch1.v = 0
      branch2.v = rm*branch1.i
END COMPONENT

-------------------------------------------------------------
-- Linear voltage-controlled current source
-------------------------------------------------------------
COMPONENT VCCS IS_A FourPins
   DATA
      REAL gm = 0.01 RANGE -Inf, Inf
"Transconductance (Mhos)"
   CONTINUOUS
      branch1.i = 0
      branch2.i = gm*branch1.v
END COMPONENT

-------------------------------------------------------------
-- Linear current-controlled current source
-------------------------------------------------------------
COMPONENT CCCS IS_A FourPins
   DATA
      REAL alfa = 1 RANGE -Inf, Inf "Current
transfer ratio"
   CONTINUOUS
      branch1.v = 0
      branch2.i = alfa*branch1.i
END COMPONENT

-------------------------------------------------------------
-- Linear voltage-controlled voltage source
-------------------------------------------------------------
COMPONENT VCVS IS_A FourPins
   DATA
      REAL mu = 1 RANGE -Inf, Inf "Voltage
transfer ratio"
   CONTINUOUS
      branch1.i = 0
      branch2.v = mu*branch1.v
END COMPONENT
```

4.4 DISPOSITIVOS SEMICONDUCTORES

El modelado de dispositivos semiconductores es una disciplina muy amplia en la que no se ha entrado en el presente trabajo. No obstante, se han incluido modelos de los dispositivos más habituales de esta naturaleza [2, 7, 15] para completar la biblioteca. Aprovechando la modularidad de EcosimPro es posible construir a medida modelos más complejos utilizando el resto de componentes de la biblioteca. En este sentido, cabe destacar que los componentes BJT y MOS se han construido de forma modular, a partir de circuitos equivalentes.

Los transistores son dispositivos de tres terminales, por lo que se han construido directamente y no se derivan de un componente abstracto TWOPINS o FOURPINS. Otra posibilidad sería partir de un TWOPINS y añadir otro puerto de tipo ELEC. No obstante, se ha preferido la implementación directa para mantener la nomenclatura habitual en los terminales de este tipo de dispositivos.

```
-------------------------------------------------------------
-- Electrical diode
-------------------------------------------------------------
COMPONENT DPN IS_A TwoPins
   DATA
```

```
    REAL n = 1       "emission coefficient"
    REAL is = 10.e-9  "reverse saturation current"
    REAL tau = 1.e-12 "transit time"
    REAL vj = 1.     "built-in junction potential"
    REAL fc = .5     "emp. constant for capacitance"
    REAL Cjo = 1.e-12 "zero bias capacitance"
    REAL m  = 0.5    "junction gradient coefficient"
    REAL Rs = 10     "series resistance"
    REAL vt = 0.025  "termical potential"
  DECLS
    REAL id
    REAL vpn
    EXPL REAL Cdepl
    EXPL REAL Cdiff

  CONTINUOUS
    vpn = branch.v - Rs * branch.i
    id = is * (exp(branch.v /(n*vt)) - 1.)

      Cdiff = tau * branch.v * exp(branch.v /(n*vt))
/(n*vt)

    Cdepl = ZONE (branch.v < fc* vj)
          Cjo /(1 - (branch.v/vj))**m
        OTHERS \
          Cjo * (1 - fc*(1+m) + m*branch.v/vj) / (1
- fc)**(m+1)
    vpn' =  (branch.i - id) / (Cdiff + Cdepl)
END COMPONENT

-----------------------------------------------------------------
-- Static diode
-----------------------------------------------------------------
COMPONENT EDPN IS_A TwoPins
  DATA
    REAL is = 10.e-9  "reverse saturation current"
    REAL vt = 0.025   "termical potential"

  CONTINUOUS
    branch.i = is * (exp(branch.v / vt) - 1.)
END COMPONENT

-----------------------------------------------------------------
-- Bipolar transistor
-----------------------------------------------------------------
COMPONENT BJT
  PORTS
    IN  Elec C
    IN  Elec B
    OUT Elec E
  DATA
    REAL nBC = 1       "emission coefficient"
    REAL isBC = 10.e-9  "reverse saturat. current"
    REAL tauBC = 1.e-12 "transit time"
    REAL vjBC = 1.     "built-in junction potential"
    REAL fcBC = .5 "emp.constant for capacitance"
    REAL CjoBC = 1.e-12  "zero bias capacitance"
    REAL mBC  = 0.5   "junction gradient coef."
    REAL RsBC = 10     "series resistance"
    REAL vtBC = 0.025   "termical potential"
    REAL nBE = 1       "emission coefficient"
    REAL isBE = 10.e-9  "reverse sat.current"
    REAL tauBE = 1.e-12 "transit time"
    REAL vjBE = 1.     "built-in junction potential"
    REAL fcBE = .5  "emp.constant for capacitance"
    REAL CjoBE = 1.e-12 "zero bias capacitance"
    REAL mBE  = 0.5    "junction gradient coef."
    REAL RsBE = 10     "series resistance"
    REAL vtBE = 0.025   "termical potential"
    REAL betaF = 100        "forward current gain"
    REAL betaR = 100        "reverse current gain"

  TOPOLOGY
      DPN D1 (n = nBC, is = isBC, tau = tauBC, vj =
vjBC, fc = fcBC, Cjo = CjoBC, m = mBC, Rs =
RsBC, vt = vtBC)
      DPN D2 (n = nBE, is = isBE, tau = tauBE, vj =
vjBE, fc = fcBE, Cjo = CjoBE, m = mBE, Rs =
RsBE, vt = vtBE)

    CONNECT C TO D1.e_n
    CONNECT D1.e_p TO B, D2.e_p
    CONNECT D2.e_n TO E

  CONTINUOUS
    E.i - C.i = betaF * D2.branch.i - betaR *
D1.branch.i
END COMPONENT

-----------------------------------------------------------------
-- MOS transistor
-----------------------------------------------------------------
COMPONENT MOS
  PORTS
    IN  Elec G
    IN  Elec D
    OUT Elec S
  DATA
    REAL isGD = 10.e-9  "reverse sat. current"
    REAL vtGD = 0.025   "termical potential"
    REAL isGS = 10.e-9  "reverse saturation current"
    REAL vtGS = 0.025   "termical potential"
    REAL cgd = 0
    REAL cgs = 0
    REAL beta = 0.0001
    REAL vt0 = 0.025

  TOPOLOGY
    EDPN D1 (is = isGD, vt = vtGD)
    EDPN D2 (is = isGS, vt = vtGS)
    CAPACITOR Capgd (C=cgd)
    CAPACITOR Capgs (C=cgs)

    CONNECT G TO Capgd.e_p, D1.e_p,
Capgs.e_p, D2.e_p
    CONNECT D TO Capgd.e_n, D1.e_n
    CONNECT S TO Capgs.e_n, D2.e_n

  CONTINUOUS
    D.i - S.i = ZONE (D.v - S.v < G.v - S.v - vt0)
```

```
        beta * ((G.v  - S.v  - vt0)*(D.v  - S.v)  - 0.5 *
(D.v - S.v)**2)
      OTHERS \
        0.5 * beta * (G.v - S.v - vt0)**2
END COMPONENT
```

4.5 OTROS COMPONENTES

Finalmente, se incluyen en esta sección componentes que no tienen cabida en la anterior dasificación. En primer lugar se encuentra la referencia de tierra o masa, que fija el nodo de referencia. Será necesario crear un componente de este tipo en cada circuito a simular.

A continuación se exponen dos modelos de amplificador operacional. El primero de ellos es el modelo ideal, que supone impedancia de entrada infinita, impedancia de salida nula, y ganancia infinita (hipótesis de cortocircuito virtual). El segundo de ellos considera tres parámetros y es una aproximación más adecuada a los modelos reales, si bien no se tienen en cuenta diversos efectos de segundo orden. Cabe destacar que para el modelado de estos dispositivos se ha partido de un TWOPINS tal que se ha añadido un puerto de tipo ELEC, en lugar de construir directamente un componente de tres terminales, como se hacía en el caso de los transistores. La razón para ello es que en estos dispositivos no se ha considerado importante mantener la nomenclatura habitual de los terminales, como ocurría con los transistores.

```
-------------------------------------------------------------
-- Ground component
-------------------------------------------------------------
COMPONENT G
  PORTS
    IN Elec e_p
  TOPOLOGY
    PATH e_p TO e_p
  CONTINUOUS
    e_p.v= 0
END COMPONENT

-------------------------------------------------------------
-- Ideal Operational Amplifier component
-------------------------------------------------------------
COMPONENT IdOpAmp IS_A TwoPins
  PORTS
    OUT Elec e_out   "Electrical Port for the
amplified signal"
  CONTINUOUS
    branch.v = 0.
    branch.i = 0.
END COMPONENT

-------------------------------------------------------------
-- Non-Ideal Operational Amplifier component
-------------------------------------------------------------
COMPONENT OpAmp IS_A TwoPins
  PORTS
    OUT Elec e_out    DATA
    REAL ri = 1.e9  "input resistance"
    REAL r0 = 0.1   "output resistance"
    REAL A0 = 1.e6  "open loop gain"
  CONTINUOUS
    branch.v = ri * branch.i
    branch.i = 0.
    e_out.v = A0 * branch.v - e_out.i * r0
END COMPONENT
```

5 EJEMPLOS

En esta sección se incluyen algunos ejemplos de modelado de circuitos utilizando la biblioteca diseñada para EcosimPro. La principal finalidad de estos ejemplos es ilustrar el uso de la biblioteca y facilitar su comprensión. En este sentido, no se mostrarán los resultados numéricos o gráficas proporcionados por el integrador de EcosimPro sino, en todo caso, se comentará su consistencia con los desarrollos teóricos.

A pesar de esta finalidad didáctica, entre los ejemplos analizados se incluyen algunos casos particulares de interés, como circuitos con diodo túnel [2] o con unión de Josephson [11]. El interés de estos circuitos radica en el modelo algebraico-diferencial que adoptan. En estos casos, se compararán los resultados obtenidos en EcosimPro con los desarrollados en [11] y [12].

5.1 EJEMPLO 1. CIRCUITO CON DIODO TÚNEL

Se considera el circuito mostrado en la figura 1.

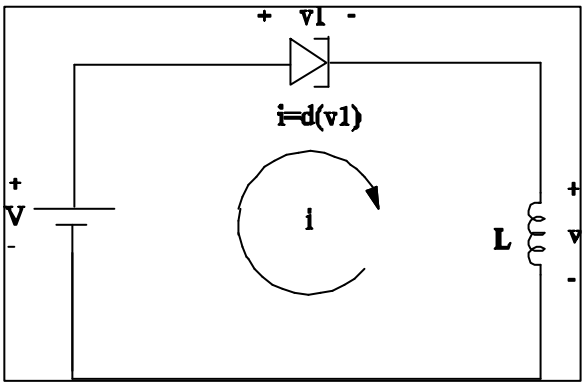

Figura 1: Ejemplo con diodo túnel

El diodo túnel viene descrito por la expresión

$$i = d(v_1 - V) = (v_1 - V)^3 + \mu(v_1 - V) + 1 \qquad (1)$$

Donde μ es un parámetro. Se considera como variable dinámica del circuito la corriente i, que coincide con la corriente de la bobina. Dado que L = 1, un sencillo cálculo resulta

$$i' = V - v_1 \qquad (2)$$

No es complicado comprobar que ($v_1 = V$, $i = I$) es un punto de equilibrio del sistema algebraico-diferencial (1)-(2), correspondiente a un punto de trabajo en continua del circuito. No se ha utilizado este concepto en el presente trabajo, si bien es una de las nociones básica del análisis circuital. Pueden encontrarse resultados que relacionan puntos de equilibrio de sistemas dinámicos con puntos de trabajo en continua en [5]. Según el desarrollo seguido en [11] dicho punto de equilibrio será estable si $\mu>0$, e inestable si $\mu<0$.

El modelo de dicho circuito utilizando la biblioteca diseñada, para ciertos valores de μ, V, e I resulta de la siguiente forma:

```
LIBRARY
NON_LINEAR_CIRCUITS_EXAMPLES

USE NON_LINEAR_CIRCUITS

COMPONENT circuit1

DATA
     REAL mu = 1
     REAL V_0 = 1
     REAL I_0 = 1

TOPOLOGY
     NON_LINEAR_CIRCUITS.INDUCTOR          L1
(L=1)
     NON_LINEAR_CIRCUITS.VDC               V0
(VDC=V_0)

NON_LINEAR_CIRCUITS.R_NONLINEAR_V
TUNNEL_D
     NON_LINEAR_CIRCUITS.G G1

CONNECT V0.e_n TO G1.e_p, L1.e_n
CONNECT V0.e_p TO TUNNEL_D.e_p
CONNECT TUNNEL_D.e_n TO L1.e_p

CONTINUOUS
    TUNNEL_D.f_v.signal = (TUNNEL_D.branch.v
- V_0)**3 + mu * (TUNNEL_D.branch.v - V_0) +
I_0

END COMPONENT
```

Según [11] el punto de equilibrio $v_1 = V = 1$, $i= I = 1$ sería estable, al ser $\mu = 1 > 0$. Se simula en EcosimPro el siguiente experimento:

```
EXPERIMENT exp1 ON circuit1.p0

DECLS

INIT      L1.branch.i = -1

BOUNDS

BODY
  REPORT_TABLE("reportAll", " * ")
  TIME = 0
  TSTOP = 30
  CINT = 0.1
  INTEG()
END EXPERIMENT
```

Se parte de la condición inicial i(0) = -1. Los resultados de la simulación numérica con EcosimPro son coherentes con lo establecido en [11]. Las variables circuitales tienden al punto de equilibrio en un corto periodo de tiempo.

No obstante, si hacemos MU = -1, la simulación diverge del punto de equilibrio, de nuevo, en consonancia con [11].

5.2 EJEMPLO 2. CIRCUITO CON UNIÓN DE JOSEPHSON

Se considera el circuito de la figura 2.

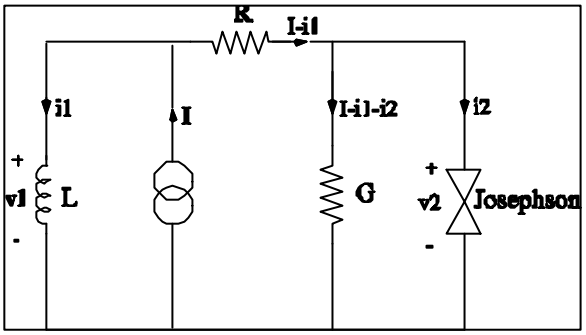

Figura 2: Circuito Josephson

Este circuito puede ser modelado por la siguiente ecuación algebraico-diferencial semiexplícita:

$$\begin{aligned} \phi_1' &= v_1 \\ \phi_2' &= v_2 \\ 0 &= i_1 - \phi_1 \\ 0 &= i_2 - I_0 \sin(k\phi_1) \\ 0 &= -v_1 + RI - Ri_1 + v_2 \\ 0 &= -Gv_2 + I - i_1 - i_2 \end{aligned} \qquad (3)$$

Donde $I_0 > 0$ y se ha supuesto L = 1. Posee puntos de equilibrio [12] definidos por las condiciones $v_1 = v_2 = 0$, $i_1 = I$, $\phi_1 = L\, I$, $i_2 = 0$, $\phi_2 = n\, p$, con n entero. Siguiendo el razonamiento de [12], se puede probar que la DAE tiene índice uno si G > 0. Además, en tal caso los puntos de equilibrio anteriores con n par son asintóticamente estables, mientras que, para n impar, son inestables.

El siguiente componente modela el circuito anterior, en el caso particular G = 1, I = I_0 = 1.

```
COMPONENT circuit2

DATA
    REAL G_0 = 1
    REAL I_0 = 1

TOPOLOGY
  NON_LINEAR_CIRCUITS.INDUCTOR L1 (L=1)
  NON_LINEAR_CIRCUITS.IDC I0 (IDC=I_0)
NON_LINEAR_CIRCUITS.L_NONLINEAR_FLU
X josephson
  NON_LINEAR_CIRCUITS.RESISTOR R1 (R=1)
  NON_LINEAR_CIRCUITS.RESISTOR RG1
(R=1/G_0)
  NON_LINEAR_CIRCUITS.G G1

CONNECT I0.e_n TO G1.e_p, L1.e_n,
josephson.e_n, RG1.e_n
CONNECT I0.e_p TO L1.e_p, R1.e_p
CONNECT R1.e_n TO RG1.e_p, josephson.e_p

CONTINUOUS
 josephson.f_phi.signal = I_0*sin(josephson.flux.phi)

END COMPONENT
```

En la simulación con EcosimPro se observa un comportamiento estable para los puntos de equilibrio anteriormente señalados con n par, e inestable si n es impar. Utilizando como condición inicial $\phi_2(0)= \phi_2{}^0$, $\phi_2(t)$ tiende al valor de la forma np con n par más cercano a $\phi_2{}^0$. Partiendo de puntos muy cercanos a np (con n impar) no se llega dichos puntos de equilibrio, sino a los más próximos que tengan n par, si bien la tendencia es más lenta que en otros casos.

5.3 EJEMPLO 3

La finalidad de este ejemplo es ilustrar el uso de dispositivos no lineales de diversa naturaleza con sus relaciones constitutivas introducidas en la forma más general. Sea el circuito de la figura 3.

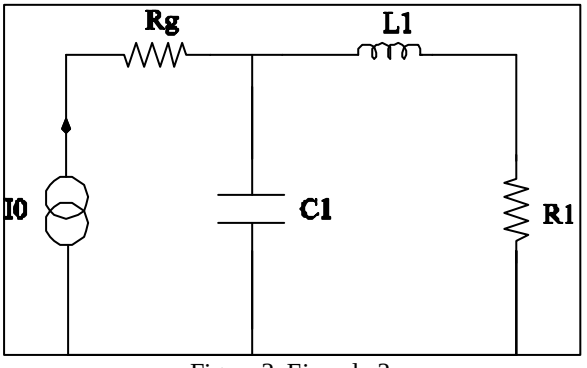

Figura 3: Ejemplo 3.

Los dispositivos presentan la siguiente caracterización:

- La bobina L1 es no lineal, descrita por la relación $\phi^2 + 5\,i + 1 = 0$.
- La resistencia Rg es lineal de parámetro resistencia igual a 50 O.
- La resistencia R1 es no lineal, descrita por la relación $v^2 - 0{,}5\,i^2 + 2 = 0$.
- El condensador C1 es no lineal, estando descrito por la relación $q^2 - 2\,v^c = 0$.
- La fuente I0 es de corriente constante e igual a 1 A.

Dicho circuito puede modelarse por el siguiente componente:

```
COMPONENT circuitqphi

DATA
    REAL I_0 = 1
    REAL R_g = 50

TOPOLOGY
    NON_LINEAR_CIRCUITS.L_NONLINEAR L1
    NON_LINEAR_CIRCUITS.C_NONLINEAR C1
    NON_LINEAR_CIRCUITS.RESISTOR Rg
(R=R_g)
    NON_LINEAR_CIRCUITS.R_NONLINEAR R1
    NON_LINEAR_CIRCUITS.IDC I0 (IDC=I_0)
    NON_LINEAR_CIRCUITS.G G1

CONNECT I0.e_n TO G1.e_p, C1.e_n, R1.e_n
CONNECT I0.e_p TO Rg.e_p
CONNECT Rg.e_n TO C1.e_p, L1.e_p
CONNECT L1.e_n TO R1.e_p

CONTINUOUS
 C1.f_q_v.signal=C1.charge.q**2 - 2*C1.branch.v
 L1.f_phi_i.signal=L1.flux.phi**2+5*L1.branch.i +1
 R1.f_v_i.signal=R1.branch.v**2-  .5*R1.branch.i**2
+ 2

END COMPONENT
```

5.4 EJEMPLO 4

De nuevo se incluye un ejemplo para ilustrar la utilización de la biblioteca. En este caso se incluye un dispositivo semiconductor y un componente de cuatro terminales para modelar un acoplo entre dos bobinas. El circuito de la figura 4 puede ser modelado por el código en EcosimPro que aparece a continuación.

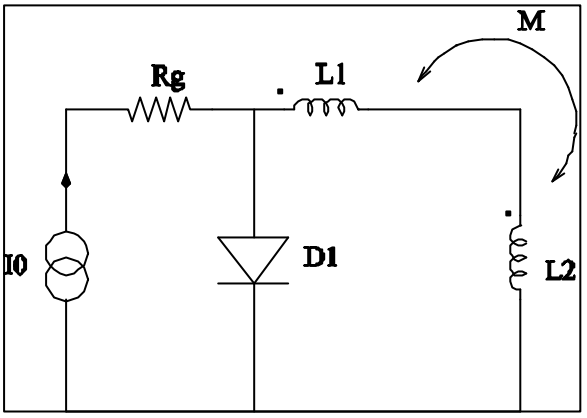

Figura 4: Ejemplo 4.

```
COMPONENT circuit4

DATA
    REAL I_0 = 1
    REAL R_g = 50
    REAL L01 = 0.01
    REAL L02 = 0.02
    REAL M0 = 0.001

TOPOLOGY

NON_LINEAR_CIRCUITS.COUPLED_INDUCTO
RS L0 (L1=L01, L2=L02, M=M0)
    NON_LINEAR_CIRCUITS.RESISTOR Rg
(R=R_g)
    NON_LINEAR_CIRCUITS.DPN D1
    NON_LINEAR_CIRCUITS.IDC I0 (IDC=I_0)
    NON_LINEAR_CIRCUITS.G G1
CONNECT I0.e_n TO G1.e_p, D1.e_n, L0.e2_n
CONNECT I0.e_p TO Rg.e_p
CONNECT Rg.e_n TO D1.e_p, L0.e1_p
CONNECT L0.e1_n TO L0.e2_p

END COMPONENT
```

Nótese que, al no definir los parámetros del diodo, se utilizan los valores por defecto.

6 CONCLUSIONES

La biblioteca de circuitos eléctricos no lineales desarrollada para EcosimPro permite modelar circuitos de diversa índole. En este sentido, complementa dicho paquete informático, facilitando al usuario la simulación de modelos de semiestados de circuitos eléctricos.

La posibilidad de definir libremente las relaciones constitutivas de los dispositivos hace idónea esta biblioteca para la simulación de casos no lineales de interés, como se vio en secciones anteriores.

En cuanto a futuras líneas de investigación, resulta de especial interés el estudio de los algoritmos simbólicos de EcosimPro. De esta manera, se podrían analizar las transformaciones que experimentan los modelos algebraico-diferenciales introducidos hasta que se aplican los métodos numéricos. Entre ellos, cabe destacar el algoritmo de Pantelides [9].

Por otra parte, se podrían considerar aspectos de tipo numérico. La adecuación de métodos numéricos de integración para su aplicación a DAEs ha sido estudiada exhaustivamente [1,6,8]. No obstante, la adaptación de distintas técnicas numéricas (como los métodos BDF implementados en DASSL [1,10]) a las ecuaciones obtenidas en modelos circuitales posiblemente sería de utilidad.

Agradecimientos

A la empresa EA International, por ceder una versión completa del paquete EcosimPro para la realización de este trabajo. A Ricardo Riaza Rodríguez, tutor del proyecto fin de carrera en el que se ha incluido este trabajo, y al Departamento de Matemática Aplicada a las Tecnologías de la Información de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid.

Referencias

[1] K. E. Brenan, S. L. Campbell and L. R. Petzold (1996), *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Classics in Applied Mathematics, SIAM.

[2] L. O. Chua, C. A. Desoer and E. S. Kuh (1987), *Linear and Nonlinear Circuits*, McGraw-Hill.

[3] I. S. Duff (1978), "On algorithms for obtaining a maximum transversal", *Computer Science and System Division-49*, A.E.R.E. Harwell, Didcot, Oxon.

[4] I. S. Duff and J.K. Reid (1976), "An implementation of Tarjan's algorithm for the block triangularization of a matrix", *Computer Science and System Division-49*, A.E.R.E. Harwell, Didcot, Oxon.

[5] M. M. Green and A. N. Wilson Jr (1995), "An algorithm for identifying unstable operating points using SPICE", *IEEE Trans. on Computer-Aided Design of Circuits and Systems* **14**, pp. 360-370.

[6] E. Hairer and G. Wanner (1996), *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer-Verlag.

[7] N. R. Malik (1996), *Circuitos electrónicos. Análisis, simulación y diseño*, Prentice Hall.

[8] R. März (1992), "Numerical methods for differential algebraic equations", *Acta Numerica*, pp. 141-198.

[9] C. C. Pantelides (1988), "The consistent initialization of differential-algebraic Systems", *SIAM J. Sci. Stat. Comput.* **9** pp. 213-231.

[10] L. R. Petzold (1984), *A description of DASSL: A differential/algebraic system solver*, Sandia National Laboratories, Livermore.

[11] R. Riaza (2003), "Double SIB points in differential-algebraic systems", *IEEE Trans. Aut. Control* **48,** pp. 1625-1629.

[12] R. Riaza, "A matrix pencil approach to the local stability analysis of nonlinear circuits", *Internat. J. Circuit Theory Appl.* **32**, pp. 23-46.

[13] R. Riaza and J. Torres-Ramírez (2004), "State and semistate models of lumped circuits", póster aceptado, *Scientific Computation in Electrical Engineering (SCEE'04)*, Capo D'Orlando (Italia).

[14] R. Riaza y J. Torres-Ramírez (2003), "Estabilidad en modelos de semiestados de circuitos eléctricos no lineales", Actas CD-ROM, *XVIII Congreso Ecuaciones Diferenciales y Aplicaciones / VIII Congreso Matemática Aplicada (CEDYA/CMA'03)*, Tarragona.

[15] J. Vlach and K. Singhal (1994), *Computer Methods for Circuits Analysis and Design*, Van Nostrand Reinhold ITP.