

# SIMULADOR DIDACTICO DE CIRCUITOS DE ELECTRÓNICA DE POTENCIA

Casaravilla G., Chaer R.

Universidad de la República Oriental de Uruguay.  
Facultad de Ingeniería.  
Instituto de Ingeniería Eléctrica.  
J. Herrera y Reissig 565. Montevideo (URUGUAY)

Tel: (598-2) 71.09.74

Fax: (598-2) 71.54.46

e-mail: gcp@iie.edu.uy, rch@iee.edu.uy

## RESUMEN

*Este trabajo presenta el programa SiMEEP, el cual está especialmente diseñado para colaborar en la enseñanza del funcionamiento de circuitos de electrónica de potencia. Se presenta la filosofía básica del simulador, la metodología de trabajo de simulación por intermedio de ejemplos.*

## 1. INTRODUCCIÓN.

Este trabajo, presenta un simulador desarrollado especialmente para analizar circuitos de electrónica de potencia. El simulador modela los circuitos como un conjunto de llaves y un conjunto de componentes lineales. El conjunto de llaves es una representación interna de los componentes semiconductores utilizados normalmente en electrónica de potencia. Este enfoque, implica un mal modelado físico de los componentes en si, pero tiene la virtud de ser similar al que usamos normalmente para analizar un convertidor, y por consiguiente es, a nuestro entender, el punto de vista válido para una herramienta que intente *colaborar* en el análisis de la función de los convertidores y no en el detalle de la física de sus componentes. Por esta característica, SiMEEP, es una herramienta útil para la enseñanza de electrónica de potencia. Le permitirá "ver" al estudiante frente a un circuito dado, las diferentes topologías por las que evoluciona el convertidor, la fatiga a la que se someten los componentes y hasta diseñar estrategias de control completas.

En este trabajo, se dará solo una idea rápida del método de representación interna utilizado en SiMEEP, una descripción completa se publicó en [2]. El método de integración y el cálculo automático del paso de tiempo fue publicado en [1].

La mayoría de los convertidores de electrónica de potencia, está formado por un conjunto de componentes lineales pasivos (resistencias, inductancias, condensadores, etc.), más un conjunto de llaves (diodos, transistores, tiristores, etc) más un conjunto de fuentes (de tensión o corriente). La posición del conjunto de llaves determinará, en cada instante, el conexionado del conjunto de componentes pasivos y del conjunto de fuentes. Por la función misma de los circuitos simulados (convertidores de potencia) exige que algunas partes del circuitos representen subsistemas de muy baja amortiguación y alta complejidad que vuelven la solución utilizando simuladores complicada por problemas de estabilidad de los métodos de integración numéricos. Actualmente SiMEEP, cuenta con librerías de componentes que cubre los componentes de circuitos lineales clásicos, varios modelos de diodos, tiristores, transistores y llaves comandables. La definición de nuevos componentes, como combinación, de los ya existentes es relativamente sencilla.

Analizar el funcionamiento de un convertidor, sin la ayuda de un simulador, es posible en la mayoría de los circuitos de una sola variable de estado real, es difícil para circuitos con dos variables reales exigiendo en este último caso que las fuentes de excitación del circuito sean constantes a tramos y prácticamente imposible para complejidades mayores.

Entender el funcionamiento de un convertidor significa, poder predecir en todo momento en que estado lógico se encuentra, como evolucionan sus variables de estado real y en que instante cambiará de estado lógico y para cual.

En cursos avanzados de electrónica de potencia, donde se estudia el control de los convertidores inmersos en sistemas tales como accionamientos electromecánicos, el simulador puede prestar apoyo en dos etapas diferentes. Primero, dado el convertidor y el sistema en lazo abierto, utilizando el simulador se puede obtener el modelo muestreado del sistema, y la linealización del mismo para diferentes puntos de operación. Con estos modelos se diseña, por algún método, un controlador para cerrar el lazo de realimentación. En la segunda etapa, se implementa el control diseñado en el simulador, y se valida así las aproximaciones realizadas tanto en la linealización de los modelos discretos como aquellas asumidas durante el diseño del controlador.

## 2. ASPECTOS GENERALES DEL SIMULADOR.

En esta sección se describe brevemente la filosofía adoptada en SiMEEP. Como cualquier simulador, para hacer posible la simulación es necesario suministrar una descripción del circuito, de las entradas y de las salidas deseadas de la simulación. Esta etapa se logra mediante la escritura de un programa fuente (PASCAL) el cual será vinculado al simulador constituido por un conjunto de librerías precompiladas.

### Definición del circuito.

Las variables que fijan el estado del circuito son: *el estado lógico*, dado por la posición del conjunto de llaves con que internamente se representó cada componente más *el estado real* constituido por las tensiones en los condensadores más las corrientes en las inductancias (menos los nodos inductivos y bucles capacitivos).

La definición se realiza en seis procedimientos separados a saber: *Cirt1*, *Cirt2*, *funcsr*, *Inicio*, *pesca*, *fin*.

En el procedimiento *Cirt1*, se debe describir los componentes del circuito que no tienen estados lógicos (resistencias, inductancias, fuentes, etc.), en *Cirt2* se incluyen aquellos

que pueden cambiar de estado lógico (diodos, llaves, etc.). En *funcsf* se definen aquellas funciones dependientes del tiempo que se usarán como entradas asociadas alguna de las fuentes (de tensión o corriente) declaradas en *Cirt1*. El procedimiento *Inicio* debe incluir el conjunto de inicializaciones necesarias al inicio de la simulación, tales como definir valores para las variables de estado, indicar las pantallas que se quiere desplegar durante la simulación y las escalas, etiquetas y títulos de las mismas, definir instante inicial, final, paso máximo de tiempo, precisión con la que se deberá calcular los instantes de conmutación entre estados lógicos, etc.

El procedimiento *pesca* sirve para literalmente "pescar" los resultados. Cada vez que se avanza un paso de tiempo, el simulador llama a este procedimiento para darle la oportunidad al usuario de capturar los valores de cualquiera de las variables. El conjunto de variables disponible es:

- Estados lógicos de los componentes declarados en *Cirt2*.
- Estados reales. Tensión en condensadores y corrientes en inductancias. Estos componentes normalmente están definidos en *Cirt1*, (salvo casos especiales como inductancias con saturación que se deben incluir en *Cirt2*).
- Potencial de todos los nodos del circuito con respeto del nodo 0 (tierra).

Es en este procedimiento, *pesca*, en el cual se debe ordenar graficar las variables que se desee observar en tiempo de simulación. También es posible en este procedimiento ir guardando las variables de interés en un archivo para su posterior análisis.

### **Modelo de los componentes**

En esta sección se dará una descripción rápida, por intermedio de un ejemplo, de la filosofía seguida para el modelado didáctico de los componentes cuando el objetivo es la enseñanza de los circuitos de electrónica de potencia como convertidores de energía y de los componentes sirviendo a tal propósito.

De la estructura del programa de simulación y de los métodos de cálculo utilizados, surge que al definir el modelo de un componente hay que determinar el circuito equivalente, a conectar entre los nodos del componente para representarlo, según el estado en que se encuentre el mismo.

En general, cuando se quiere "ver" el funcionamiento de un componente en determinado circuito, es primordial comprender la función que desempeña el componente en el circuito y dejar el modelado de las características detalladas, del componente, para una etapa separada de análisis (talvez otro curso y otras herramientas) ya que estas, no resultan prácticas ni para las simulaciones del funcionamiento de los convertidores ni para explicar la función que desempeñan los componentes en un circuito determinado.

Para fijar ideas con respecto al enfoque a utilizar para un modelado didáctico de las funciones de un componente, utilizaremos como ejemplo un tiristor. Como se verá, en la propia definición del modelo se pone en práctica el razonamiento fundamentalmente lógico asociado con la propia dinámica del componente, por lo que representa un muy buen ejercicio que ayudará al estudiante a comprender mejor el funcionamiento del componente en el contexto del circuito.

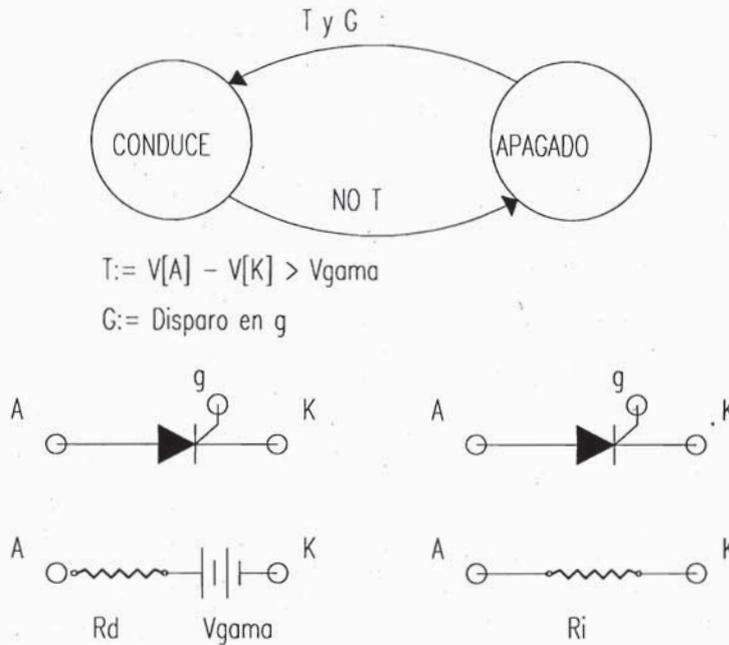


Figura 1. Modelo de Tiristor

Tal como se representa en la figura 1 utilizaremos dos estados "lógicos" (Apagado y prendido) para representar al tiristor. Cuando el tiristor se encuentre encendido (estado prendido) el circuito equivalente que utilizaremos para representarlo es el del lado inferior izquierdo de la figura, corresponde a una fuente de tensión de valor  $V_{gama}$  en serie con una resistencia de valor  $R_d$ . Cuando se encuentre en estado apagado por el contrario lo representaremos por una única resistencia, de valor  $R_i$ , simulando las corrientes de fugas (más que un valor exacto de las fugas, importa modelar el hecho de que aunque el tiristor esté apagado, el ánodo y el cátodo no están galvánicamente aislados situación que podría resultar en topologías inconexas y en potenciales de nodos indeterminados). El circuito equivalente para el estado "apagado" es el mostrado en el lado inferior derecho de la figura 1.

En el propio modelo del tiristor se debe implementar los métodos necesarios para determinar el estado en el que se encuentra. El siguiente algoritmo (en PASCAL) determinará el Estado Actual del tiristor en función de la tensión entre ánodo y cátodo, de si está siendo disparado o no y del estado anterior:

```

if EstadoAnterior = PRENDIDO then
  if  $V[a] - V[k] > V_{gama}$  then EstadoActual := EstadoAnterior
  else EstadoActual := APAGADO
else {Apagado}
  if  $(V[a] - V[k] > V_{gama})$  and (DisparoActivo) then
    EstadoActual := PRENDIDO
  else EstadoActual := EstadoAnterior;

```

Las variables  $V[a]$  y  $V[k]$ , contienen los valores del potencial del ánodo y del cátodo respectivamente respecto del nodo de tierra. La variable *DisparoActivo* será verdadera si el tiristor está siendo disparado y falsa si no.

Una abstracción considerable en este modelo es considerar que el disparo en el gate es una función lógica (boolean en PASCAL).

Además de determinar su estado, el modelo debe especificar cómo modelar cada uno de los estados dando para ello, el circuito equivalente a utilizar. Los circuitos equivalentes, se especifican como combinación de componentes de las *librerías* de componentes ya definidos. Para el caso del tiristor, el siguiente código PASCAL sería el válido:

```
if EstadoActual = PRENDIDO then
    rvg(a, k, Rd, -Vgama)
else
    r(a, k, Ri);
```

En palabras, si el tiristor se encuentra prendido, se coloca el componente *rvg* (resistencia en serie con generador de tensión) de la *Librería Básica* de componentes entre los nodos (a) y (k) siendo (Rd) el valor de la resistencia y (-Vgama) el del generador de tensión. El signo negativo es para darle en sentido correcto al generador de tensión. Por el contrario si el tiristor está apagado, se coloca un componente *r* (resistencia simple) también de la *Librería Básica* de valor Ri entre los nodos (a) y (k).

Con los dos procedimientos de los recuadros anteriores, el modelo del tiristor está completo. Ya sabe cómo determinar su estado lógico y que circuito lineal "mostrar" como equivalente según el estado lógico en que se encuentre.

### 3. EJEMPLOS DIDÁCTICOS.

En esta sección se muestran algunos ejemplos, elegidos por didácticos.

El primero de ellos es un rectificador monofásico de media onda. Es un circuito muy sencillo, pero que servirá para mostrar como puede colaborar la simulación.

El segundo ejemplo, es comparativamente más complejo y mostrará como la simulación permite mostrar y explicar de una manera natural conceptos que normalmente resultan de difícil comprensión por parte de los estudiantes.

#### **Rectificador monofásico de media onda.**

El circuito a simular es el que se representa en la figura 2.

La primer tarea a los efectos de determinar la descripción de entrada para poder simular el sistema es identificar los nodos de interconexión. Cada componente a incluir en la descripción del circuito debe conocer sus puntos de inserción en el circuito (entre que nodos

está ubicado) y los parámetros intrínsecos como el valor resistivo en el caso de una *resistencia* etc.

Por ejemplo, en el presente circuito la descripción del circuito queda resumida al el siguiente código de programa.

```

const
  vp = 311.0; {tensión de pico del generador de tensión}
  r0 = 10; { resistencia de carga }
  rdon = 0.01; {resistencia en conducción del tiristor }
  RShunt = 50.0e-3; { resistencia para medir la corriente por la fuente }
  NCiclos = 2.5; { número de ciclos de red a simular }

{definición de componentes}
procedure Cirt1;
begin
  vg(1,0,1, vp);{ Generador de tensión, dependiente de la fuente rf[1], entre los nodos 0 y 1
                con el positivo en el nodo 1. El parámetro vp es un multiplicador de rf[1]
                para obtener el valor de la fuente.}
  r(1,3, RShunt); { resistencia entre los nodos 1 y 3 de valor RShunt }
  r(2,0, r0); {resistencia entre los nodos 2 y 0 de valor r0 }
  c(2,0, c0); { condensador entre los nodos 2 y 0 de valor Co }
end;

procedure Cirt2;
begin
  d2(3, 2, rdon); {diodo del tipo (D2) con el cátodo en el nodo 2 y el ánodo en el nodo3
                  con valor de resistencia serie en conducción rdon }
end;

procedure funcrf(t: NReal);
begin
  rf[1]:= sin(w50*t);
end;

```

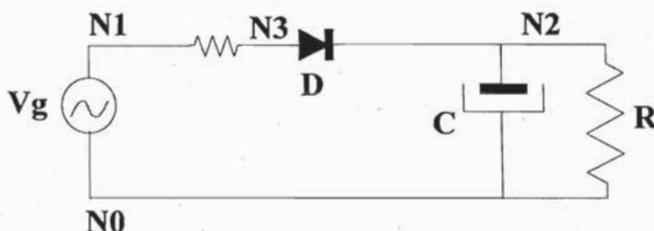


Figura 2. Rectificador monofásico de media onda.

En la descripción de componentes se incluye una resistencia en serie con el diodo a los efectos de poder visualizar la corriente por el mismo.

La siguiente tarea es definir qué variables del sistema se quiere visualizar o almacenar.

A tal efecto el usuario escribirá el siguiente código:

```
definición de salidas;
begin
    trazo(canal_0,v[1]);
    trazo(Canal_0,v[2]);
    ifuente:=(v[1]-v[3])/RShunt;
    trazo(canal_1,ifuente);
    trazo(canal_2, estado_dx[1])
    trazo(Canal_3, ifuente*v[1]);
end;
```

Por ejemplo en la primera línea estamos solicitando al simulador que muestre durante la simulación la tensión del nodo 2 en uno de las ventanitas gráficas identificada como canal\_0.

```
procedure inicio;
begin
    x0[1]:=0;
    tinicial:=0;
    tfinal:=2*pi/w*NumeroDeCiclos;
    puntos:=250;
    dtmax0:=(tfinal-tinicial)/puntos;
    PasoDivisor:=10;
    TmaxTmin:=1e3;
    {cantidad de ventanitas gráficas de visualización}
    subplot(2,2); {2x2}

    {sección que define títulos, etc de una ventanita gráfica}
    gridY:= 8;
    definaY(0,-400, 100);
    titulo('Ug, Uc');
    xlabel('5 ms/Div');
    ylabel('100 V/DIV');
    grid;
    inea0;
end;
```

La figura 3 muestra la pantalla desplegada durante la simulación en donde el estudiante puede visualizar las diferentes estados (compuesto por las variables reales y lógicas) por los que evoluciona el circuito.

En la ventanita superior izquierda se grafica la tensión de entrada, en respuesta a la línea *trazo(canal\_0,v[1])* del procedimiento *pesca*. En esta misma ventana se grafica evolución de la tensión en el condensador.

La ventana superior derecha muestra el estado lógico (on/off) de conducción del diodo. La ventana inferior izquierda la corriente por el diodo y la inferior derecha la potencia instantánea entregada por la fuente.

Así como se despliega estos valores durante la simulación, es posible almacenarlos en archivos para su posterior análisis.

## Rectificador trifásico onda completa con carga L-E

En la figura 4, se muestra la pantalla desplegada durante la simulación del un circuito rectificador de seis tiristores. El ejemplo de simulación se implemento con control arco-coseno cuya consigna es posible variarla en tiempo de simulación mediante el uso del teclado. La gráfica de la consigna y de las referencias de comparación para generar los disparos se muestran en el extremo superior izquierdo. En la gráfica superior derecha se observa la tensión en bornes de continua del puente junto con la tensión de la batería, conectada como carga. El valor de la batería de carga también es variable en forma dinámica durante la simulación. Esto permite al estudiante observar y experimentar la relación causa-efecto en una forma dinámica e inmediata.

En la simulación se observan los modos de operación "conducción continua", "conducción discontinua", los efectos perturbadores sobre la red de alimentación del puente (v67 de la figura).

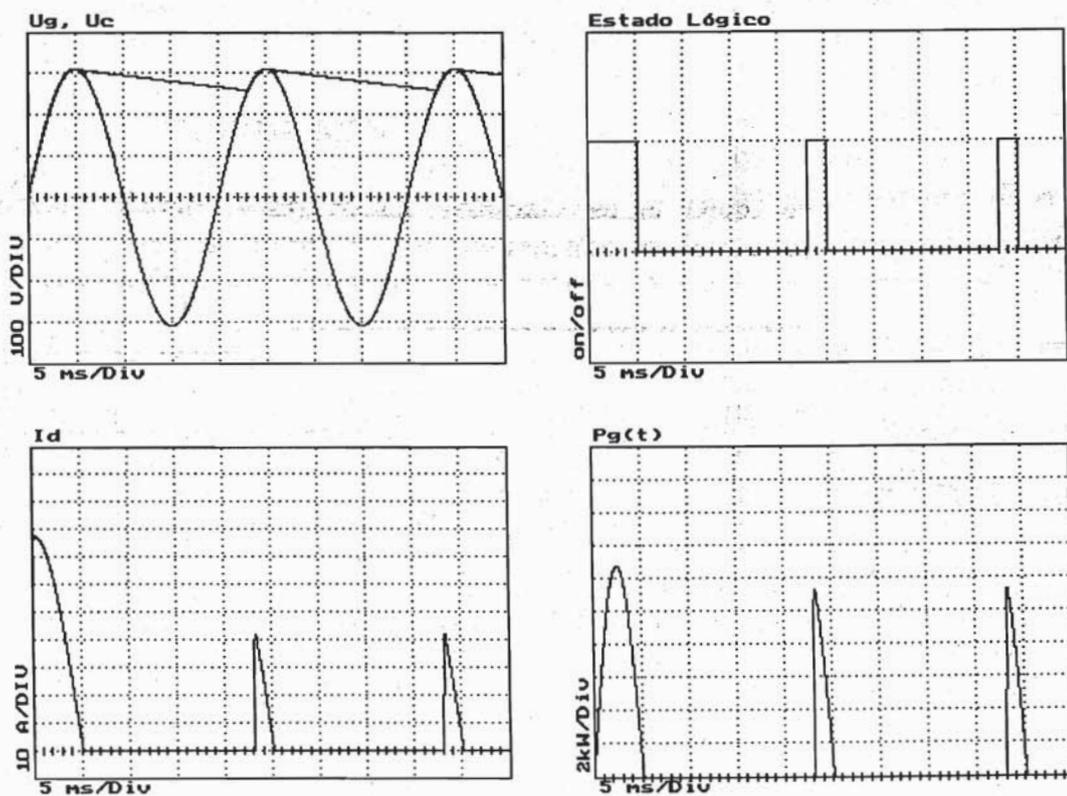


Figura 3. Resultado de la simulación del circuito de la figura 2.

## 4. CONCLUSIONES

Como principal conclusión de este trabajo, se quiere recalcar la importancia de un software de simulación "conceptual" de las funciones de los componentes y los convertidores de electrónica de potencia para el propósito de la enseñanza de la disciplina.

Finalmente se recalca la importancia de poder experimentar, por parte del estudiante, las relaciones causa-efecto y analizar diferentes topologías. Sobre todo en electrónica de potencia, donde la construcción de prototipos para estudio es la mayor parte de las veces costoso y peligroso.

En la actualidad, esta herramienta es utilizada para entrenar a los estudiantes previa la realización práctica de prototipos físicos en el laboratorio. Así se logra un mejor aprovechamiento de las clases de laboratorios, optimizando los esfuerzos docentes y uso de los materiales.

### 5. BIBLIOGRAFÍA.

- [1] R. Chaer. "On the Numerical Integration of the State Equation". *IEEE 2nd Anual Workshop on Computers in Power Electronics*. USA-PA, 1990.
- [2] R. Chaer. "Aspectos particulares de la Simulación de circuitos de electrónica de potencia". *IEEE II Encuentro de Especialistas en Potencia, Instrumentación y Medida*. Montevideo-Uruguay, 1991

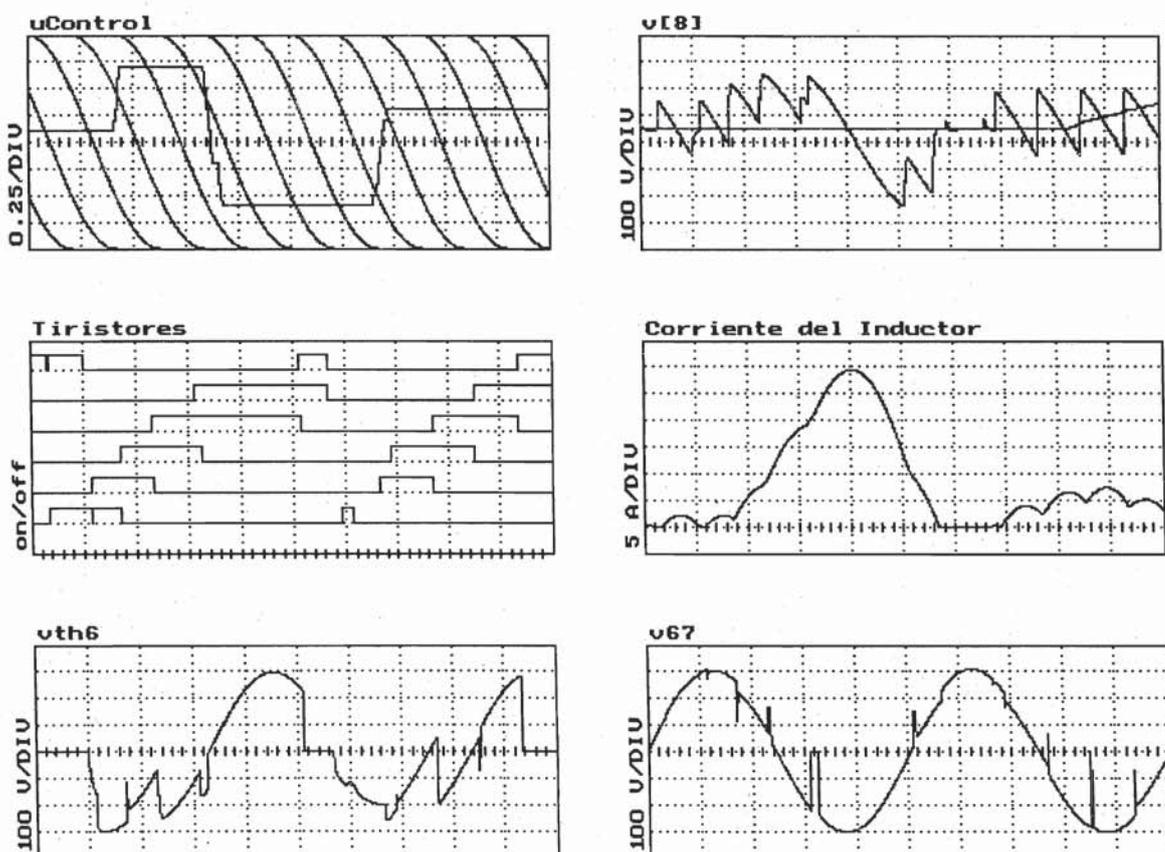


Fig. 4. Pantalla durante la simulación de un puente trifásico de tiristores.