

Aspectos Particulares de la Simulación de Circuitos de Electrónica de Potencia.

RUBEN CHAER , member, IEEE.

Sumario- Se destacan aquellos inconvenientes que surgen al simular Circuitos de Electrónica de Potencia, tratando de resaltar la característica especial de dichos circuitos. Se muestra como se ha intentado solucionar dichos inconvenientes en el paquete de simulación SIMEEP. En particular, se describe la forma en que se obtienen las ecuaciones de Estado, cómo se determinan los instantes de conmutación y las perspectivas de ampliación del simulador.

I. INTRODUCCION

El objetivo de este trabajo es plantear aquellos aspectos particulares de la simulación de circuitos de Electrónica de Potencia, que la diferencian en complejidad y calidad de la simulación de circuitos en general.

Por simulación de CEPs (circuitos de Electrónica de Potencia), se entiende aquí la simulación del comportamiento en el tiempo de convertidores estáticos de energía, como ser inversores de tensión, rectificadores controlados, conversores DC-DC, etc.

Los CEPs se caracterizan por tener componentes que conmutan (tiristores, transistores, diodos, fets, etc.), con los cuales se implementa lo que se llama una llave. Según el componente usado, el cambio de estado de la llave (de corte conducción o a la inversa), será comandable en una de las transiciones, en ambas o en ninguna.

Tanto para la simulación como para el análisis de los CEPs, cabe distinguir entre diferentes aproximaciones, según sea el modelo utilizado para los componentes más o menos detallado.

En [4] se comenta comparativamente resultados de simulación de un chopper elevador, alimentado de la red (60 hz) rectificadas, con dos lazos de realimentación, utilizando, por un lado, modelos detallados para los componentes, y por otro, un simulador para circuitos conmutados, para el cual los modelos son idealizados (como llaves ideales). Con el modelo detallado, es posible analizar con precisión las transiciones en las conmutaciones. Esto es útil para el diseño de la llave (necesidad de snubbers, máximos a los que estarán sometidos los componentes, etc.). La simulación detallada de 400 ms (40 ciclos de la llave) utilizando SPICE en una VAXStation 3100, duró 3 horas, y la simulación con modelos idealizados de 8.33 ms (830 ciclos de la llave), llevó 375 s de CPU. Por supuesto, en este último caso, las formas de onda del convertidor son precisas, se ha perdido el detalle de las transiciones.

Hay propuestas muchas aproximaciones más [2..9], que intentan abstraer "lo importante" del comportamiento de las llaves, con fines de acelerar el análisis y permitir el modelado de los convertidores. En el mismo trabajo del que se extrajo el ejemplo anterior, se comenta tres modelos más

para el mismo ejemplo. Estos modelos son por ahora bastante particulares en cuanto a la aplicación; están pensados para modelar determinados tipos de convertidores y hasta ahora no son una herramienta directamente disponible en los simuladores.

Hasta aquí, la intención ha sido mostrar que:

- 1) La simulación de un convertidor (aunque sea de complejidad menor), no puede llevarse a cabo en un tiempo razonable utilizando modelos detallados para los componentes.
- 2) La simulación detallada tiene utilidad para el análisis de pequeños intervalos de tiempo (con fines probablemente de diseño y selección de los componentes).
- 3) Para el análisis global del convertidor (formas de ondas, régimen y estabilidad) y la obtención de modelos del mismo, es ineludible llevar a cabo abstracciones, que hasta el momento no son sistemáticas y por lo tanto difícilmente automatizables. No es descabellado pensar que los simuladores de CEPs necesiten técnicas de Inteligencia Artificial, para ir aprendiendo cómo realizar esas abstracciones no sistemáticas y fuertemente dependientes de la utilización.

Este trabajo, trata con simuladores de circuitos conmutados, es decir, simuladores diseñados para simular secuencias de circuitos lineales. Si bien en la mayoría de este tipo de simuladores es posible describir modelos tan complicados como se quiera de los componentes, se apuntará justamente en el otro sentido, es decir, idealizar los modelos.

El análisis de los CEPs, se basa en la suposición de un estado para el conjunto de las llaves y en resolver el CL (circuito lineal) equivalente mientras que las condiciones sean tales que se mantenga válido el estado de las llaves.

Cuando ocurre una "conmutación", (algunas de las llaves cambian de estado), es necesario obtener el nuevo CL válido y resolverlo. Este es el procedimiento manual de análisis de los CEPs, y es lo menos que un usuario pretende que un simulador haga por él. Esta característica de topología variable, es una diferencia cualitativa importante entre un simulador de CLs, y uno para CEPs. Este último requiere determinar el estado válido de las llaves (componentes que pueden tener más de un estado, como ser diodos, FETs, tiristores, etc), poder formar las ecuaciones para el CL resultante, y llevar a cabo la integración de las mismas, realizando en forma simultánea un chequeo de la validez del estado supuesto de las llaves.

La topología cambiante de los CEPs, hace que técnicas para la obtención de las ecuaciones de estado de un CL, como ser grafos de incidencia o matrices de bucles [12] , se vuelva un poco pesada para el tratamiento de los CEPs. El uso de estos descriptores significa pensar en el circuito como un objeto que posee, en forma separada, una topología y un conjunto de componentes con sus valores. Sin embargo, la manera natural

de realizar la entrada de datos que describen al circuito, es definiendo, para cada componente, su valor y cómo está conectado.

Basados en los comentarios anteriores, pensaremos en un circuito como un conjunto de componentes poseedores de la información que originan (su valor y cómo están conectados). La topología (como información) está ahora repartida entre los componentes.

Este enfoque tiene ventajas significativas cuando tratamos con circuitos cuya topología es cambiante debido a cambios en los componentes mismos, como es el caso de los CPEs.

Son comunes los CPEs con "cortes inductivos" o "bucles capacitivos"[12] , fuentes de corriente o tensión ideales y circuitos no amortiguados. Encontrar una manera general de encarar estos casos particulares no es sencilla, y podría pensarse en decir, "las fuentes no son nunca ideales" o "las inductancias y los condensadores tienen fugas" y por lo tanto es más "real" utilizar modelos que las tengan en cuenta.

El argumento no es falso, pero si se usan modelos idealizados es porque los efectos secundarios no son necesarios en la simulación. Se quiere el comportamiento del circuito ideal.

Si son agregadas las no idealidades para tratar los casos particulares descritos, igual que el caso más sencillo de circuitos lineales con suficiente amortiguamiento, el condicionamiento del sistema puede resultar débil [13..16] (definición de Número de Condición de un sistema), dando como resultado un mayor tiempo de integración de las ecuaciones para evitar la propagación explosiva de los errores numéricos[11]. Por esta razón, las ecuaciones formadas por SIMEEP para un circuito lineal dado, son un variación de las ecuaciones de nudos, que permiten contemplar de forma natural los casos particulares nombrados.

II. DEFINICIONES

Se consideran circuitos que tienen llaves, pero que para cada estado del conjunto de llaves, son circuitos lineales, invariantes en el tiempo y de parámetros concentrados.

Para simular cualquier sistema, es necesario describir su dinámica por intermedio de ecuaciones. Suponemos que el lector está familiarizado con el concepto de "Estado" de un sistema y a expresar la dinámica del sistema en la forma genérica:

$$\dot{X} = f(X, t) , \quad (1)$$

donde X es el "vector de estado" y t es el "tiempo". Cuando el sistema es lineal, invariante en el tiempo y de parámetros concentrados, la ecuación de

estado se escribe de la forma:

$$\begin{aligned} \dot{X} &= A X + B r(t) \\ Y &= C X + D r(t) \end{aligned} \quad (2)$$

donde:

$B : [nr, nr]$; $A : [nx, nx]$; $C : [ny, nx]$; $D : [ny, nr]$;
son matrices constantes,

$$\begin{aligned} X &= X(t) : [nx, 1] && \text{"Estado"} \\ r(t) &: [nr, 1] && \text{"Entrada"} \\ Y &= Y(t) : [ny, 1] && \text{"Salida"} \end{aligned}$$

Un circuito del tipo considerado tendrá, para cada estado del conjunto de llaves, una ecuación del tipo de la (ec. 2). El estado del circuito queda determinado dando el estado de las llaves y el estado del circuito lineal asociado a dicho estado de las llaves. Como el conjunto de los estados posibles de las llaves es un conjunto discreto, por lo tanto es codificable en forma binaria y se hará referencia a sus elementos como "estados lógicos del circuito".

Se representará el estado lógico con un vector de variables booleanas, DX de dimensión ndx igual a la necesaria para codificar todas las combinaciones de las llaves. Supongase, para ejemplificar, un circuito con seis diodos. Cada diodo puede estar cortado o conduciendo, y resulta por lo tanto natural asociarle a cada uno una de las componentes de DX. Si el circuito no tiene otros componentes que puedan conmutar, su estado lógico (DX) estará representado por un vector de seis booleanas y por lo tanto el circuito tiene en total $2^6 = 64$ estados lógicos posibles.

Ahora el problema es encontrar las matrices A, B, C y D para cada estado lógico. Intentar calcular las matrices para todos los estados antes de iniciar la simulación no es una buena idea. En el ejemplo de los seis diodos, implicaría calcular 64 conjuntos de matrices, y almacenarlas. En la práctica, casi nunca aparecen todos los estados lógicos durante la simulación. Resulta por lo tanto una solución más razonable ir calculando las matrices a medida que aparecen los distintos estados durante la simulación. Esto es más económico en tiempo y memoria. Por otro lado, tener la facilidad de formar las ecuaciones correspondientes a un DX en cualquier momento, es una herramienta potente que nos permite, por ejemplo, realizar variación de parámetros.

La filosofía seguida en la implementación software del simulador, está orientada a agrupar toda la información sobre cada componente en un sólo objeto. De esta manera, el componente es quien sabe cómo está conectado en el circuito, y qué modelo presenta al circuito según su entorno y estado anterior. Por ejemplo, es el diodo que decide si está cortado o conduciendo.

De esta manera, se logra independizar al simulador del modelo de los componentes, y además, da gran flexibilidad, ya que en última instancia, son los modelos de los componentes los que gobiernan la ejecución del simulador. Por ejemplo, puede ser útil hacer un modelo de un diodo, que en función de la energía que disipa, de la temperatura ambiente y de cierta resistencia y capacidad térmica, decida actualizar los valores de su resistencia y caída en directa o de sus fugas, y ordene al simulador actualizar las matrices. Esto le da completa libertad al usuario.

III. DETERMINACION DE LA ECUACION DE ESTADO.

Se considera como variables de estado a la tensión de los condensadores y la corriente en los inductores del circuito. Este conjunto de variables puede ser redundante si en el circuito ocurren cortes inductivos o bucles capacitivos.

Se numeran todos los nudos del circuito del 0 en adelante,

0, 1, 2, , NN

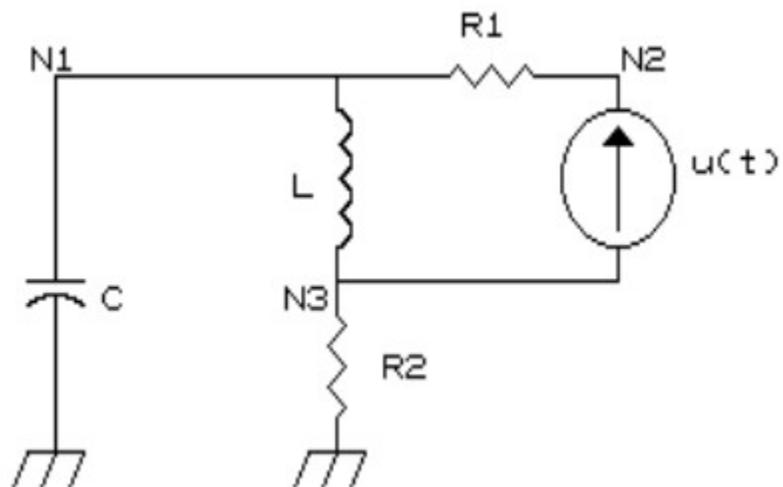
El nudo numerado como "0" será considerado la TIERRA del circuito, y es respecto a él que se mide el potencial de los demás nudos.

III.1. ECUACIONES DE NUDOS.

Con en_k): se indica la ecuación correspondiente al nudo k y en palabras, dicha ecuación expresa que la suma de corrientes salientes del nudo k debe ser nula.

Para determinar el circuito, son suficientes las ecuaciones de todos los nudos menos uno. La ecuación correspondiente al nudo 0, no se escribe.

Figura 1: Ejemplo



A continuación, se describe el método utilizado para formar las ecuaciones por intermedio de un ejemplo, cuyo circuito se muestra en la fig.1.

Supóngase el circuito inicialmente vacío y que los componentes son agregados uno a uno.

Las ecuaciones para el circuito vacío de 3 nudos son:

$$\begin{aligned} \text{en1):} & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V[1] \\ V[2] \\ v[3] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ \text{en2):} & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V[1] \\ V[2] \\ v[3] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ \text{en3):} & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V[1] \\ V[2] \\ v[3] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

donde $V[k]$ es el potencial del nudo k .

Al agregar la resistencia $R1$, una corriente de valor $y_1*(V[1]-V[2])$ sale del nudo 1 y entra en el 2, pasando por la resistencia. (y_1 es la admitancia de $R1$). Las ecuaciones resultantes son :

$$\begin{aligned} \text{en1):} & \begin{bmatrix} y_1 & -y_1 & 0 \end{bmatrix} \begin{bmatrix} V[1] \\ V[2] \\ v[3] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ \text{en2):} & \begin{bmatrix} -y_1 & y_1 & 0 \end{bmatrix} \begin{bmatrix} V[1] \\ V[2] \\ v[3] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ \text{en3):} & \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V[1] \\ V[2] \\ v[3] \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

Para simplificar la notación, no se escribirá más los enk): y el vector de tensiones. De esta manera, cuando se agrega $R2$ las ecuaciones son:

$$\begin{bmatrix} y_1 & -y_1 & 0 \\ -y_1 & y_1 & 0 \\ 0 & 0 & y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Este cambio indica que ahora una corriente $y_2*V[3]$ sale del nudo 3 hacia el nudo 0 (la ecuación del nudo 0 no se escribe).

El condensador dejará circular una corriente $C_s * V[1]$ del nudo 1 hacia la tierra. Las ecuaciones resultan:

$$\begin{bmatrix} y_1 + Cs & -y_1 & 0 \\ & -y_1 & y_1 & 0 \\ & 0 & 0 & y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

La corriente por la bobina será $(V[1] - V[3]) / Ls$ y las ecuaciones:

$$\begin{bmatrix} y_1 + Cs + \frac{1}{Ls} & -y_1 & -\frac{1}{Ls} \\ & -y_1 & y_1 & 0 \\ & -\frac{1}{Ls} & 0 & y_2 + \frac{1}{Ls} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Falta agregar la fuente de corriente. Como la corriente por la fuente no es expresable en función de la tensión entre sus bornes, debe aparecer entre los términos independientes de las ecuaciones. Las ecuaciones completas son:

$$\begin{bmatrix} y_1 + Cs + \frac{1}{Ls} & -y_1 & -\frac{1}{Ls} \\ & -y_1 & y_1 & 0 \\ & -\frac{1}{Ls} & 0 & y_2 + \frac{1}{Ls} \end{bmatrix} = \begin{bmatrix} 0 \\ u(t) \\ -u(t) \end{bmatrix}$$

III.2 ECUACIONES DE NUDOS EXTENDIDAS.

En la sección anterior, se mostró cómo formar las ecuaciones de nudos del circuito, agregando los componentes uno a uno. Como se puede observar del resultado del ejemplo, las variables de estado no aparecen en forma directa. Además, algunos coeficientes tienen la variable s y por lo tanto no son un número. Intentar trabajar computacionalmente esta matriz implicaría realizar cálculo simbólico o realizar los cálculos para valores concretos de s .

Para lograr separar las variables de estado y además tener sólo coeficientes numéricos, se extenderán las ecuaciones agregando en forma explícita las variables de estado. Al agregar variables, será necesario agregar también ecuaciones. En el ejemplo agregaremos x_1 y x_2 , tensión en el condensador y corriente en la bobina respectivamente.

Las ecuaciones a agregar son:

$$ex_1): x_1 = V[1]$$

$$ex_2): L x_2 s = V[1] - V[3]$$

Con estos agregados, el conjunto de ecuaciones del circuito es:

$$\begin{matrix} ex_1): \\ ex_2): \end{matrix} \begin{bmatrix} y_1 & -y_1 & 0 & Cs & 1 \\ -y_1 & y_1 & 0 & 0 & 0 \\ 0 & 0 & y_2 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & Ls \end{bmatrix} \bullet \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ u(t) \\ -u(t) \\ 0 \\ 0 \end{bmatrix}$$

donde se ha marcado explícitamente las filas y columnas agregadas.

Para separar los coeficientes que contienen s de los que no, lo se duplicaran las columnas correspondientes a las variables de estados. Las ecuaciones resultantes se muestran a continuación.

$$\begin{matrix} en_1): \\ en_2): \\ en_3): \\ ex_1): \\ ex_2): \end{matrix} \begin{bmatrix} y_1 & -y_1 & 0 & C & 0 & 0 & 1 \\ -y_1 & y_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & y_2 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & L & 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ x_1s \\ x_2s \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ u(t) \\ -u(t) \\ 0 \\ 0 \end{bmatrix}$$

Aplicando doble escalerización a este sistema de ecuaciones, se logra despejar $V(t)$ (vector de tensiones) ,y X (derivada del vector de estado) en función de X (vector de estado) y la entrada del circuito. De esta manera se obtiene la ecuación de estados del circuito, y quedan determinadas las matrices A, B, C, D considerando como salidas al vector de tensiones. Si X es un vector de estado dado un valor de X y uno de la entrada, es posible determinar en forma única los valores de X y de V por definición de estado, y

por lo tanto la escalerización del sistema de ecuaciones no presenta problemas. Si por el contrario X es redundante como vector de estado (es decir que se puede encontrar un conjunto de variables de dimensión menor para representar el estado del sistema), el sistema es de rango inferior al sistema de ecuaciones hallado, y sobran tantas ecuaciones como grados de redundancia tiene X . La solución es eliminar algunas de las ecuaciones y algunas de las variables de X , pero el procedimiento no se discutirá aquí.

III.3. EXTENSION DE LAS ECUACIONES DE NUDOS PARA INCLUIR FUENTES DE TENSION CON RESISTENCIA NULA.

Como la corriente que circula por una fuente de tensión con resistencia nula no es expresable en función de la tensión en sus bornes, no es posible incluir tal componente en las ecuaciones de nudos. Una solución es no permitir el uso de tales fuentes obligando siempre a usar fuentes con cierta resistencia interna, pues de esta manera podemos hacer el equivalente NORTON de la fuente con su resistencia, sustituyéndolas por una admitancia en paralelo con una fuente de corriente (ya mostró en el ejemplo cómo incluir dichos componentes). Esta solución tiene el inconveniente, de bajar drásticamente el tamaño del paso de tiempo del integrador numérico, para obtener estabilidad del mismo tal como se puede desarrollar en [11] para el método de Euler.

En la biblioteca básica de componentes del simulador se incluye con el nombre de "rvg" (generador de voltaje con resistencia).

Esta solución funciona bien en la medida en que la resistencia incluida sea lo "suficientemente grande". Al achicar la resistencia, disminuye el condicionamiento entre la tensión y la corriente de la fuente, acercándose el sistema de ecuaciones a ser no invertible. Esto lleva, en el momento de la inversión del sistema, a divisiones por números muy pequeños que amplifican los errores numéricos y pueden causar desbordamiento numérico en alguna de las operaciones.

Para evitar este tipo de condicionamientos débiles, se propone a continuación un cambio en las ecuaciones de nudos, para permitir incluir generadores de tensión ideales.

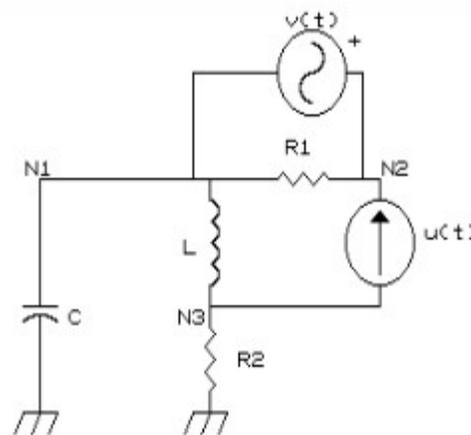


fig. 2.

Figura 2: Ejemplo 2

Se agrega al circuito ejemplo una fuente de tensión $v(t)$ entre los nudos 1 y 2 tal como se indica en la figura 2.

No es posible expresar la corriente por el generador en función de la tensión en sus bornes, consideramos la ecuación del "corte" del circuito que encierra sólo a los nudos que son bornes del generador. Dicha ecuación se puede obtener como la suma de las ecuaciones de nudo de los bornes, ya que la corriente por el generador (que es desconocida), se cancela al hacer la suma.

De esta manera, se sustituye una de las ecuaciones de nudos de los bornes por la ecuación del corte, y la ecuación de nudo correspondiente al otro borne la se sustituye por la ecuación que impone el generador incluido entre ambos nudos. En el ejemplo, las operaciones de inclusión del generador expresadas en forma simbólica son:

$$\begin{aligned} e1) &<- \{ e1) + e2) \}; \\ e2) &<- \{ V[2] - V[1] = v(t) \}; \end{aligned}$$

y el sistema de ecuaciones resultante es el que se muestra a continuación:

$$\begin{array}{l} en_1 + en_2) : \\ *) : \\ en_3) : \\ ex_1) : \\ ex_2) : \end{array} \begin{bmatrix} 0 & 0 & 0 & C & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & y_2 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & L & 0 & 0 \end{bmatrix} \bullet \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ x_1 s \\ x_2 s \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} u(t) \\ v(t) \\ -u(t) \\ 0 \\ 0 \end{bmatrix}$$

IV.4. DETALLES DE LA IMPLEMENTACION

IV.4.1. La ecuación correspondiente al nudo 0, nunca se forma, y por lo tanto no está disponible para sumarla a otra en caso de ser necesario. Por esta razón, cuando uno de los nudos del generador es el 0, se realiza la operación simbólica:

$e0) <- \{ e0) + ek) \}$, siendo k el otro nudo del generador, y se sustituye $ek)$ por la relación de tensiones que impone el generador.

IV.4.2. Cada vez que se introduce al circuito un generador de tensión, se afectan dos ecuaciones, las cuales dejan de representar "la ecuación de nudo". Por lo tanto, si se intenta agregar posteriormente un componente que afecte dicha ecuación, hay que tener ciertas precauciones. Se realiza anotando, cuando se suman dos ecuaciones, en qué ecuación

queda la suma. De esta manera si se agrega un componente que afectaría a la ecuación original, se afecta en realidad a la ecuación suma.

IV.4.3. El ir anotando qué nudos fueron incluidos en un corte, permite detectar cuándo los generadores de tensión forman un bucle. Como esta situación es o redundante o errónea, el programa se para escribiendo el mensaje: "BUCLE DE GENERADORES DE TENSION".

IV.4.4. En el caso de existir en el circuito "bucles capacitivos" o "cortes inductivos", el conjunto de tensiones de los condensadores y corrientes en los inductores es redundante como vector de estado del circuito. Esto se detecta en SIMEEP, pues es imposible llevar a cabo la eliminación Gaussiana del sistema de ecuaciones, pero a partir del mismo sistema (a medio resolver), se determina cuál o cuáles de las variables de estado son redundantes y cómo dependen de las demás elegidas para la simulación como verdaderas variables de estado. Esto es totalmente transparente para el usuario.

IV. DETERMINACION DE LOS INSTANTES DE CONMUTACION

Supondremos que el circuito, dado un estado del conjunto de llaves, es lineal, y por lo tanto, mientras no cambie ninguna llave, es representable por una ecuación de estado lineal. El estado del circuito queda entonces determinado por el estado lógico, que determina la posición del conjunto de llaves, y por el estado real, que corresponde a las variables de estado elegidas para el estado lógico actual. Con cada estado lógico tenemos asociada una ecuación de estado lineal que determina la dinámica del circuito.

Notación: DX = "estado lógico"
X = "estado real"
(DX,X) = "estado"

Dado un estado inicial del circuito y conocidas las entradas del mismo, es posible realizar la simulación numérica del mismo, integrando la ecuación de estado obtenida como se describió en la sección III. Pero dicha ecuación será válida sólo mientras no ocurran conmutaciones, es decir mientras el estado lógico permanezca siendo el mismo. Para determinar la validez o no del DX actual habría que haber obtenido el conjunto de condiciones necesarias para su validez junto con las ecuaciones de estado.

Sin embargo, en lugar de obtener dicho conjunto de condiciones, hemos decidido, "preguntarle" a los componentes del circuito si están de acuerdo con la topología actual, o si no están de acuerdo, cuál es el cambio que proponen. Este enfoque, puede pensarse, como el conjunto de condiciones repartidas entre los componentes que las originan. Como ejemplo pondremos el modelo más simple de diodo disponible en la biblioteca de modelos de SIMEEP.

{-----}

```
procedure diodoIdeal(k,j:word; rd:real);
```

```

begin
  nd:=nd+1;
  if check then DX[nd] := v[k] > v[j]
  else
    if DX[nd] then r(k,j,rd) (* resistencia rd entre *)
                                (* los nudos k y j *)
    else (* circuito abierto *)
end;

{-----}

```

Como se observa, los modelos se incluyen en las librerías como fuentes PASCAL, que describen el funcionamiento del componente. El modelo del componente, será invocado muchas veces durante una simulación, algunas veces para formar las ecuaciones de estado (check = FALSE) y otras, en modo "CONSULTA" (check = TRUE), para determinar la validez del estado lógico actual.

En el ejemplo, el modelo es de un diodo, conectado entre los nodos (k) y (j) y de resistencia en estado de conducción (rd). Observar, que si (check = TRUE), entonces se ejecuta nada más que la instrucción $DX[nd] := v[k] > v[j]$, en la cuál, $DX[nd]$ corresponde a la variable booleana del estado lógico asociada al estado de este diodo en particular y $v[j]$ y $v[k]$, son los potenciales de los terminales del diodo. Así, cuando el modelo es invocado en modo CONSULTA, la variable lógica asociada con él es devuelta con el estado en que tendría que estar según la tensión que el diodo "ve" en sus terminales.

Antes de invocar a todos los componentes en modo consulta, el contenido de DX, es guardado en un vector auxiliar DX0, de esta manera, comparando el resultado de realizar la consulta con el estado lógico actual, podemos concluir si son iguales ($DX = DX0$), que el estado lógico actual es válido y que podemos continuar con la integración de su ecuación de estados.

Si el resultado de la consulta difiere del estado lógico almacenado ($DX \neq DX0$), se debe determinar cuál será el nuevo estado lógico. Para ello vale la pena comentar que es muy importante en los CEPs determinar con mucha precisión el instante en que tiene lugar la conmutación.

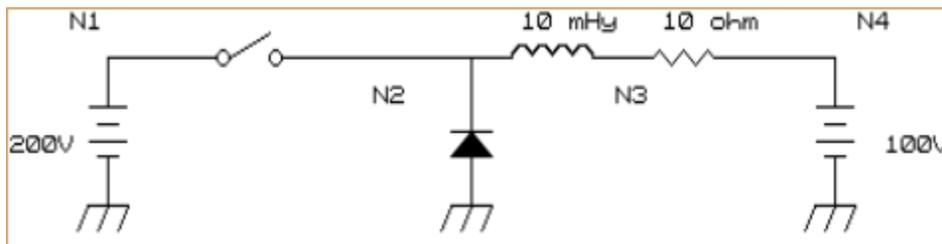


Figura 3: Cambio brusco en las constantes de tiempo

El circuito de la figura 3 es muy sencillo, pero basta para presentar el problema. Cuando la llave está cerrada, la corriente crece, al abrir la llave decrece hasta anularse, instante en que se corta el diodo.

Si no determinamos con precisión el instante de conmutación, cambiaremos de topología con una corriente por la bobina (inversa por el diodo), que no corresponde a la realidad y que altera totalmente los resultados. En este caso particular, estamos cortando una bobina, con corriente no nula, lo que llevara a transitorios de tensión importantes en bornes del diodo.

Este comportamiento que se asemeja mucho, a lo que sucede en los diodos reales, sometidos a conmutaciones violentas debido al fenómeno de recuperación inversa, aquí es producto de determinar en forma imprecisa el instante de conmutación.

En SIMEEP, se especifica en el mismo archivo de descripción del circuito a simular, la precisión con que se quiere afinar las conmutaciones, especificando cuantas veces más chico debe ser el paso de integración (con respecto al paso con que se venía integrando) para aceptar el cambio de topología.

De esta manera, cuando nos acercamos a una conmutación, el paso de integración se divide por 2 en instancias sucesivas, hasta que se logra cruzar el punto de conmutación con un paso de tiempo menor o igual al especificado como máximo aceptable.

Luego que tiene lugar una conmutación, debemos determinar cuál es la nueva conmutación. Para lo cuál, en lugar de probar todas las combinaciones posibles de DX, que además de ser muy costoso en tiempo, podría llevarnos a un estado lógico válido que no sea necesariamente aquel al que iría el circuito real que estamos simulando, se procede a resolver el circuito con el estado lógico resultante de la última consulta, calculando la tensión de los nudos en tal caso (sin avanzar en el tiempo), hacemos una nueva pasada en modo consulta y así hasta que dos pasadas consecutivas den el mismo valor para DX.

Cuando esto sucede, es porque todos los componentes están de "acuerdo" con el estado lógico DX.

Si el circuito, para un determinado valor de X puede tener más de un valor de DX válido, frente a una determinada conmutación, el procedimiento indicado, nos lleva a aquel valor que parece más natural, por ser aquel al que los componentes han querido evolucionar. Esta descripción (poco rigurosa), se basa en que las conmutaciones reales tienen cierta dinámica y causística, que las lleva evolucionar en cadena, (unas primero que las otras) hasta un estado final compatible.

Por otro lado, es sencillo encontrar ejemplos (biestable con transistores ideales), para los cuales, frente a una determinada conmutación, el procedimiento descrito, no encadena el estado lógico de partida con el definitivo y por lo tanto, el simulador entra en lo que llamamos "loop lógico".

Detectar que se entró en loop lógico no difícil, basta con anotar los estados que aparecen en el proceso de determinación de la conmutación y

observar si se repiten. Lo difícil, es una vez detectado, determinar cuál es el estado lógico por otro procedimiento. SIMEEP, no tiene implementada ninguna solución a este problema.

Una alternativa es en tal caso, iniciar una búsqueda exhaustiva entre aquellos estados que no han sido probados, pero debido a la complejidad combinatoria que significa una búsqueda de este estilo (y la falta de fundamento físico) este es un punto en el que esperamos alguna idea mejor.

V. CONCLUSION Y PERSPECTIVAS.

La simulación de CEPs, es complicada, llena de particularidades, que lleva a que la mayoría de los usuarios, se vean decepcionados cuando intentan usar un simulador pensado para simular circuitos analógicos en general para tal propósito.

Hasta el momento, no conozco, ningún simulador, que sin "parches" introducidos por el propio usuario, (según el problema enfrentado), logre simular de manera eficiente cualquier CEP.

Los circuitos en que es más difícil, obtener una simulación exitosa (y eficiente) suelen ser los más sencillos e ideales, no estando la complejidad en el número de componentes, sino, en lo poco o nada amortiguados que son, o en particularidades tales, como bobinas que se abren (con un diodo o llave), condensadores que se conectan en paralelo con una fuente de tensión etc.

Apuntando hacia el uso de simuladores diseñados para simular circuitos conmutados como base para el desarrollo de ambientes de simulación de CPEs potentes con suficientes herramientas para el análisis y modelado de CEPs (con diferentes grados de abstracción), se describió (bastante al detalle) el módulo de formación de ecuaciones usado en SIMEEP.

En cuanto a las perspectivas de desarrollo del tema, señalaré dos problemas:

- a) Dado un CEP, encontrar el estado de régimen.
- b) Dado un CEP en régimen, obtener un modelo de pequeñas perturbaciones para el mismo, con el fin de poder diseñar lazos de realimentación, y estudiar la influencia de pequeñas perturbaciones externas.

Hamill en [6], indica su solución al punto a). Esta solución adolece de suponer que el CEP, funciona a frecuencia fija conocida.

En [5] se indica una solución para b), pero no se supone el estado de régimen conocido.

Ambas ideas, se basan, en utilizar un simulador "bruto", como pueden ser SPICE, MICROCAP, o SIMEEP, como herramienta base.

Nuestra idea, es desarrollar (usando como base SIMEEP), un simulador "astuto", con el cuál, se pueda determinar si un CEP, tiene estado de régimen, y la frecuencia del mismo.

Teniendo la manera de calcular el estado estacionario, (sin la

necesidad de efectuar la simulación bruta hasta el), intentaremos, implementar un Modelador, que nos permita obtener modelos de pequeña señal de los convertidores, y realizar estudios de estabilidad y comportamiento frente a pequeñas perturbaciones.

La posibilidad de obtener modelos para grandes señales en forma automática es muy atractiva, pero por el momento muy lejana.

VI. REFERENCIAS BIBLIOGRAFICAS.

MODELADO DE CONVERTIDORES:

[1]- R.D. Middlebrook, S. Cuk, "A General Unified Approach To Modelling Switching-Converter Power Stages", *IEEE, PESC*, Jun. 1976, Cleveland, OH.

[2]- S. Cuk, R.D. Middlebrook, "A General Unified Approach To Modelling Switching DC-To-DC Converters in Discontinuous Conduction Mode", *IEEE, PESC*, Jun. 1977, Palo Alto, CA.

[3]- R.J. King, "A Fourier Analysis for Fast Simulation Algorithm", *IEEE Trans. on PE*, vol. 4, No. 4, Oct. 1989.

[4]- V. Thottuvelil, D. Chin, G. Verghese, "Hierarchical Approaches to Modeling High-Power-Factor AC-DC Converters", *IEEE Workshop on Computers in Power Electronics*, Aug. 1990 PA-USA.

[5]- P. Maranesi, "PWM Converter Characterisation in the Frequency-Domain Through Fast Time-Domain Computer Simulation", *IEEE Workshop on Computers in Power Electronics*, Aug. 1990 PA-USA.

[6]- D. Hamill, "Improved Time-Domain Simulation of Resonant and Other DC-DC Converters", *IEEE Workshop on Computers in Power Electronics*, Aug. 1990 PA-USA.

[7]- S.Y.R. Hui, C. Christopoulos, "A Discrete Approach to the Modeling of Power Electronic Switching Networks", *IEEE, Trans. PE*, no. 4, pp. 398-403, Oct. 1990

[8]- P. Krein, J. Bentsman, R. Bass, B. Lesieutre, "On the Use of Averaging for the Analysis of Power Electronic Systems", *IEEE, Trans. PE*, no. 2, pp. 182-190, Apr. 1990.

[9]- V. Vorpérian, R. Tymerski, C.Y. Lee, "Equivalent Circuit Models for Resonant and PWM Switches", *IEEE, Trans. on PE*, vol. 4, no. 2, Apr. 1989.

APROXIMACIONES DE LA MATRIZ DE TRANSICION:

[10]- A. Luciano, A. Strollo, "A Fast Time-Domain Algorithm for the Simulation of Switching Power Converters", *IEEE, Trans. PE*, no. 3, pp. 363-370, Jul. 1990

[11]- R. Chaer, "On The Numerical Integration of the State Equation", *IEEE Workshop on Computers in Power Electronics*, Aug. 1990 PA-USA.

LIBROS BASICOS:

[12]- S. Seshu, N. Balabanian, "Linear Network Analysis", J. Wiley and Sons.

[13]- Noël Gastinel, "*Analisis Numérico Lineal*", Reverté 1975

[14]- D. Kahaner, C. Moler, S. Nash, "Numerical Methods and Software", Prentice-Hall 1989.

[15]- G. Dahlquist, A. Björck, "Numerical Methods", Prentice Hall 1974.

[16]- Atkison, Harley, "an itroduction to Numerical Methods with PASCAL", Addison-Wesley.