

AUTOMATIZANDO EL MODELADO Y SIMULACION CON BOND GRAPHS DE SISTEMAS FISICOS CONMUTADOS

Matías A. Nacusse^{a,b}, Sergio J. Junco^a, Alejandro Donaire^a

^a*Departamento de Electrónica, Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Ríobamba 245 Bis – S2000EKE Rosario – Argentina.*

^b*CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas.
{nacusse,sjunco,adonaire}@fceia.unr.edu.ar*

Palabras Clave: modelos conmutados, bond graph, motor serie de corriente continua.

Resumen: Este trabajo aborda el problema del modelado de sistemas físicos conmutados utilizando el formalismo Bond Graph (BG) e introduce componentes de librería para el software de simulación 20sim®.

Las representaciones clásicas (BG continuos) de los modos continuos de un sistema conmutado son complementadas con los vínculos de potencia conmutados conocidos por sus siglas en inglés SPJ (Switch Power Junction), obteniéndose así una representación global y única del sistema conmutado llamado Bond Graph Conmutado (Switched Bond Graph SwBG).

El formalismo de los SPJ permite tratar con algunos problemas asociados con el modelado tales como causalidad variable de elementos conmutados, manejo de la estructura de interconexión, fallas al desconectar subsistemas, y otras inconsistencias. Otra característica significativa es que los SPJs permiten una completa representación gráfica de un sistema híbrido, vía el acoplamiento bidireccional del SwBG con cualquier gráfico de información usado para modelar explícitamente los subsistemas que determinan sus conmutaciones. Estas ventajas motivan a escoger, de entre las diferentes ideas y técnicas propuestas para el manejo de conmutaciones, a los SPJs como formalismo para el modelado y simulación (M&S) de sistemas conmutados.

Existen dos tipos de SPJ, los 0s y los 1s, estos vínculos son la generalización de los vínculos 1 y 0 estándar del formalismo BG y admiten una mayor cantidad de situaciones causales. Se proveen una nueva implementación de los SPJs para modelos 20sim®, que pueden ser agregados a la librería 20sim® original de componentes BG y usarlos como cualquier otro componente estándar, sin necesidad de programación adicional.

Además algunos problemas típicos que surgen al trabajar con sistemas conmutados son tratados y se proponen soluciones. La más importante es la re-inicialización de integradores cuando los modos del sistema originan discontinuidades o saltos en las variables de estado.

Los métodos y herramientas desarrolladas son aplicadas en el M&S de un motor serie de corriente continua equipado con un circuito de conmutación que posibilita el debilitamiento del campo de excitación, práctica usual en aplicaciones de ingeniería.

1 INTRODUCCIÓN

En muchos problemas de ingeniería cambios abruptos en sistemas físicos son considerados instantáneos. Esto es debido, principalmente al hecho que el ingeniero está interesado en el comportamiento del sistema en una escala más grande en relación a la del cambio abrupto. Por ende, los detalles del mismo dentro de esta ventana de tiempo no son relevantes frente al comportamiento en estudio. De este modo, ignorarlos resulta un ahorro de esfuerzo, análisis y tiempo de simulación. Sin embargo, esta suposición requiere un trato especial, porque el modelado del comportamiento de las discontinuidades puede involucrar muchas facetas, cambiar la estructura del modelo y de las variables, saltos instantáneo en valores de estado, etc. Para una discusión detallada de estos temas ver ([Mosterman and Biswas, 1998](#)).

Cuando las conmutaciones ocurren el sistema cambia su modo de operación; por ejemplo si se tiene n llaves podemos llegar a tener 2^n modos diferentes de operación. Un camino para representar todas las conmutaciones es generar 2^n juegos de ecuaciones, las cuales a lo sumo serán algebraicas diferenciales (DAE's), donde cada juego describe el comportamiento continuo del sistema para un modo en particular. En la práctica no todos los modos son posibles de realizar, es más la identificación de todos los modos relevantes y el desarrollo de sus respectivas ecuaciones puede ser una tarea muy complicada.

En particular, en este trabajo estas ecuaciones son representadas mediante la herramienta de modelado basada en energía llamada Bond Graph ([Karnoop et al., 2000](#)), ésta es una sólida forma de representar *sistemas físicos continuos*. Debido a que la teoría BG fue originalmente desarrollada para tratar solamente fenómenos continuos surgió la necesidad de incorporar algunas herramientas para manejar el problema de conmutación. Aquí el problema de la conmutación en BG se aborda mediante la utilización de los SPJ introducidos por ([Umarikar and Umanand, 2005](#)).

Este trabajo presenta una implementación computacional genérica, reutilizable de los SPJ como un componente de librería de 20sim® ([Controllab Products B.V](#)), permitiendo el modelado de un único sistema conmutado de n modos, tratando además el problema de la inicialización de los almacenadores de energía luego de la conmutación.

La [sección 2](#) presenta algunas definiciones y resultados previos. La [sección 3](#) desarrolla los resultados principales de este trabajo. En la [sección 4](#) se ilustran las técnicas propuestas sobre el motor de corriente continua de excitación serie con debilitamiento de campo ([Leonhard, 1997](#)). Posteriormente se le adiciona un control en cascada tipo todo-nada obteniéndose así el modelo de un esquema de aplicación real. En la [sección 5](#) se presentan los resultados de simulación de las aplicaciones desarrolladas en la [sección 4](#). Por ultimo la [sección 6](#) presenta las conclusiones del trabajo.

2 DEFINICIONES Y RESULTADOS PREVIOS

En esta sección se presentan definiciones y resultados previos que motivan el desarrollo de este trabajo.

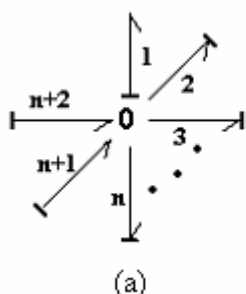
2.1 0 y 1 Clásicos/Estándar:

Los vínculos estructurales del formalismo clásico de BG por definición son elementos conservadores de energía. En particular todos los enlaces conectados a un vinculo 1 tienen el mismo flujo y la sumatoria de esfuerzos de todos los enlaces es cero. A su vez en un vinculo 0 sucede lo contrario, es decir, que todos los enlaces tienen el mismos esfuerzo y la sumatoria de flujos de todos los enlaces es igual a cero. Lo expresado anteriormente puede visualizarse en las ecuaciones [1](#), [2](#) y en la [figura 1](#).

Vínculo 0

$$e_i = e_1 \quad ; \quad i = 2, \dots, n+2$$

$$f_1 = f_{n+1} + f_{n+2} - f_2 - \dots - f_n \quad (1)$$



Vínculo 1

$$f_i = f_1 \quad ; \quad i = 2, \dots, n+2$$

$$e_1 = e_{n+1} + e_{n+2} - e_2 - \dots - e_n \quad (2)$$

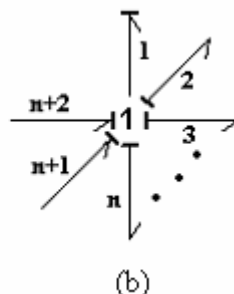


Figura 1. Vínculo estructural estándar con asignación de causalidad (a) 0 (b) 1

2.2 0s y 1s Switcheados:

Los vínculos de potencia conmutados, conocidos en el lenguaje de Bond Graph como Switched Power Junction (SPJ) fueron introducidos por (Umarikar and Umanand, 2005) como una generalización de los vínculos estructurales (0 y 1) del Bond Graph. Esta generalización admite para el caso de tener un vínculo estructural 0 por ejemplo; que dos o más enlaces le fijen esfuerzo. En el formalismo clásico de BG esto es imposible debido a que se produce un conflicto de causalidad. Sin embargo, imponiendo la restricción de que el vínculo estructural 0 elija uno de los enlaces que fijan esfuerzo se resuelve el conflicto de causalidad obteniendo un nuevo elemento BG llamado switched power 0-junction, cuyo mnemónico es 0s. Sólo el enlace seleccionado entregará flujo, en tanto que el resto de los enlaces tendrán flujo cero. Las restricciones son señales de entrada al vínculo conmutado, de las cuales sólo una ellas valdrá uno para cada momento. La misma seleccionará el esfuerzo correspondiente mientras que, para un mismo instante de tiempo, las restantes señales tendrán valor cero. Análogamente se define el 1s.

<p>Vínculo 0s:</p> $Esfuerzo = U_1 e_1 + U_2 e_2 + \dots + U_n e_n$ $f_i = U_i (f_{n+1} + f_{n+2}) \quad ; \quad i = 1, \dots, n \quad (3)$	<p>Vínculo 1s:</p> $Flujo = U_1 f_1 + U_2 f_2 + \dots + U_n f_n$ $e_i = U_i (e_{n+1} + e_{n+2}) \quad ; \quad i = 1, \dots, n \quad (4)$
<p>Figura 2 Elemento 0S con n enlaces fijando esfuerzo.</p>	<p>Figura 3 Elemento 1S con n enlaces fijando flujo.</p>

Tabla 1: Esquemas de los de los vínculos SPJ con n enlaces fijando causalidad y ecuaciones correspondientes

La [tabla 1](#) contiene las ecuaciones [3](#) y [4](#) que gobiernan la estructura de los SPJ así como también las figuras [2](#) y [3](#) que muestran las asignaciones de causalidad para los SPJ. En la [tabla 2](#) se muestra la representación de los SPJ con componentes estándares del formalismo BG.

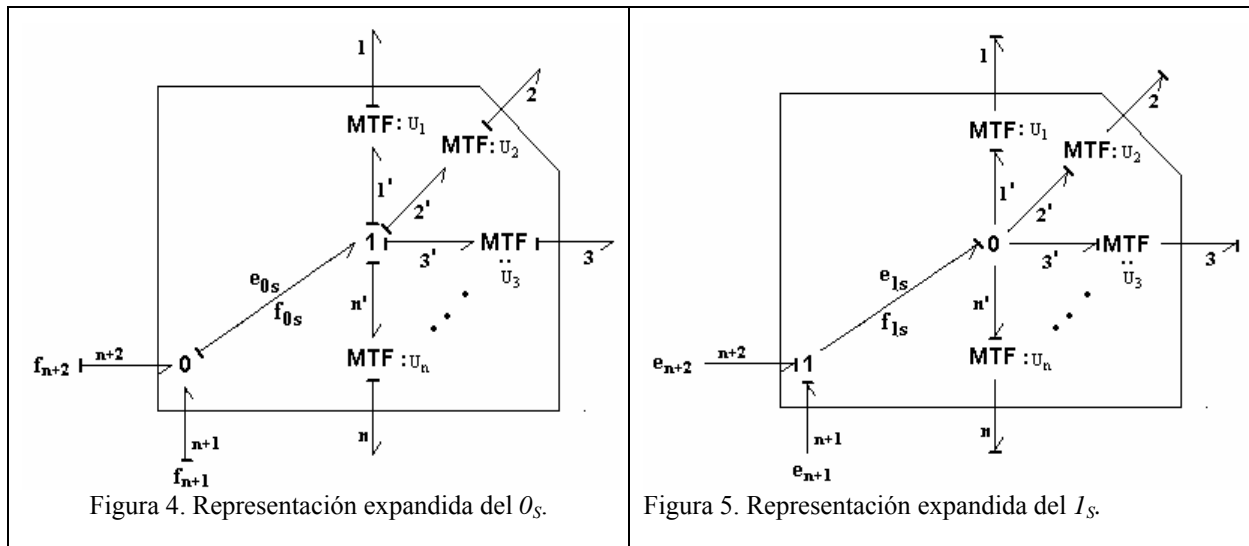


Tabla 2: Representación de los SPJ con componentes estándares del formalismo BG

2.3 Implementaciones ad-hoc de los SPJ, Versiones Compactas y Expandidas

Aquí se muestra uno de los resultados de ([Junco et al 2007](#)) donde se obtuvieron dos implementaciones de los SPJ en el software de M&S 20sim. Estas implementaciones son ad-hoc porque deben ser programadas de acuerdo a la conmutación que se desee modelar. El resultado presentado en ([Junco et al 2007](#)) resuelve el problema de los vínculos en los cuales dos enlaces fijan causalidad. En la [tabla 3](#) se muestra el código que corresponde a la versión compacta del vínculo 0s y la versión expandida del mismo. La [figura 6](#) se presenta el caso del 0s.

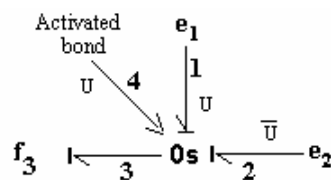


Figura 6 el vínculo 0s descrito en la tabla 1.

Implementación Compacta del 0 _s .	Implementación expandida del 0 _s
<p>Variables real U_inv;</p> <p>inialequations port3.e=0;</p> <p>equations U_inv = (1 - U); port3.e = U*port1.e +U_inv*port2.e; port1.f = U*port3.f; port2.f = U_inv*port3.f;</p>	

Tabla 3: Implementación compacta y expandida de los SPJ en 20sim

2.4 Crítica a las Implementaciones ad-hoc de los SPJ.

El resultado mostrado en la sección anterior soluciona el problema de conmutación sólo cuando dos enlaces fijan causalidad al vínculo. Esto obliga al modelador a desarrollar un esquema particular para un vínculo que contenga más de dos enlaces que fijan causalidad.

3 RESULTADO PRINCIPAL: HERRAMIENTAS PARA SIMULACION AUTOMATICA DE SWBG.

En esta sección se presenta una implementación en 20sim genérica y compacta de los SPJ, que resuelve el problema de múltiples enlaces que fijan causalidad al vínculo. Dicha implementación puede adicionarse a la librería de 20sim.

Además se describe como puede implementarse en 20sim la solución del problema de inicialización de los almacenadores de energía luego de la conmutación.

3.1 Implementación en 20sim Genérica del 0s y 1s.

En esta subsección se detalla la programación en lenguaje SIDOPS+ del 0s genérico, siendo análogo el desarrollo del 1s.

Para lograr un componente genérico, es decir que puedan conectarse al mismo cualquier número de enlaces de potencia, es necesario que el algoritmo que define al componente contenga alguna instrucción que permita generar un vector cuya dimensión varíe a medida que se conecten los enlaces. El software 20sim ([Controllab Products B.V](http://www.controllab.com)) utiliza las instrucciones *collect(x)* y *direct(x)* para realizar esta operación. A continuación se detallan los pasos que se utilizaron para la programación del 0s genérico y compacto, y las correspondientes líneas en SIDOPS+.

Primero se definen dos tipos de puertos de potencia. El puerto *pin*, que se corresponde con los enlaces que presentan conflicto de causalidad según el SPJ en cuestión, para el caso del 0s tienen causalidad fija de esfuerzo entrante al vínculo. Por el contrario al puerto *pout* se asocian los enlaces que no presentan conflicto causal, para el caso del 0s fija de esfuerzo como variable saliente del vínculo.

Además se definieron dos puertos de señal, uno de entrada y otro de salida. La señal de entrada *U* permite seleccionar entre los enlaces *pin* el deseado, y la señal de salida *effort* permite monitorear el esfuerzo seleccionado del vínculo 0s ([figura 7](#)).

Las líneas de código SIDOPS+ detallan como se definen los distintos puertos y como se restringen sus causalidades.

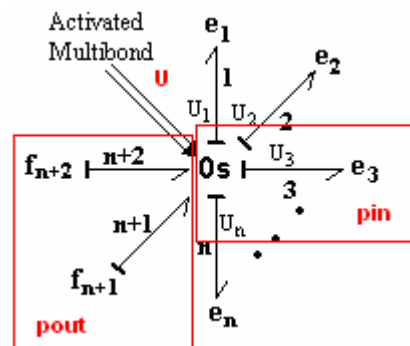


Figura 7. Elemento 0s con n enlaces *pin* y dos enlaces *pout*

ports

```
power knot duplicatable none pout [1];
power knot duplicatable none pin [1];
```

```

signal knot out effort [1];
signal knot duplicatable in U [1];
restrictions
causality fixed out pout;
causality fixed in pin;
end;

```

Una vez definidos los puertos, el segundo paso es la selección de un puerto del conjunto *pin* mediante las señales de selección *U*. El programa desarrollado identifica a la señal de selección *U1* como la primera señal que se conecta al SPJ, y se la asocia con el primer puerto de potencia *pin* que se conecte. Por lo tanto hay que tener en cuenta el orden de conexión para generar las correctas relaciones entre enlaces de potencia y señales selectoras.

A continuación se detallan las líneas de programa restantes utilizadas para el *Os*:

implementation eq

```

variables                                     /*En esta sección se definen las variables a utilizar
integer i;
integer k;
integer S;
integer large;
integer largeO;
real Uin[any];
real Effin[any];
real Floin[any];
real Floout[any];
real Effout[any];
real E;

```

initialequations

```

/*En esta sección de definen las ecuaciones que deben ser calculadas al inicio de la
simulación, estas se calculan solo una vez.*/
large=columns(collect(U));           // Se define el largo de los vectores U y pout
largeO=columns(collect(pout.e));

```

code

```

/*Esta sección contiene todas las ecuaciones que deben ser calculadas de manera secuencial,
es decir las ecuaciones y el orden de las mismas no se rescriben durante la compilación del
modelo. El código secuencial resultante es calculado en cada paso de integración.*/

```

```

for i = 1 to large do
  if Uin[i]==1 then
    S=i;           //S es el índice del vector U que presenta un 1
  Else           //y por lo tanto define que puerto pin esta activo
    Floin[i]=0;  //Fuerzo a los otros puertos de pin en valores de
  end;           //flujo 0
end;

```

equations

/ Esta sección contiene todas las ecuaciones estándares. Estas no tienen que ser ingresadas de manera secuencial. Estas son rescritas durante la compilación para obtener el correcto orden de ejecución. El código resultante es calculado en cada paso de integración.*/

```

Effin=collect(pin.e);           //Asigno los puertos a vectores
Floin=collect(pin.f);
Uin=collect(U);
Effout=collect(pout.e);
Floout=direct(pout.f);

E=Effin[S];                    //Asigno el valor del esfuerzo activado según U a E
effort=E;                      //Asigno el valor E a la señal de esfuerzo de Os
Floin[S]=msum(Floout);        //Calculo el valor del flujo del enlace activado
for k=1 to largeO do          //Asigno el valor de E a todos los puertos de
    Effout[k]=E;              //pout
end;

```

implementation_end;

De esta manera se logra una versión compacta y genérica del *Os*, la cual puede añadirse a la librería de 20sim para ser reutilizada. Utilizando estos componentes se puede modelar un sistema conmutado de n modos como un único sistema donde estén presentes todos los modos. Se destaca que el problema de la simulación de un sistema conmutado necesita la inicialización de sus almacenadores de energía en las transiciones entre los n modos del sistema.

3.2 Inicialización de Modos.

El problema de la inicialización de los almacenadores de energía luego de cada conmutación puede no resultar trivial debido a que el vector de estado puede variar su dimensión en cada modo de operación, siendo 2^n el número posible de diferentes modos de operación.

Cada modo de operación tiene su propio conjunto de variables de estado, cuyo número puede variar en la transición de un modo a otro. Se distinguen los tres casos siguientes:

- El número de variables de estado permanece constante luego de la transición. Esto puede suceder, por ejemplo si cambia la causalidad en un elemento R.
- El número de variables de estado aumenta luego de la transición. Esto sucede si algún almacenador de energía pasa de causalidad derivativa a causalidad integral luego de una transición.
- El número de variables de estado disminuya luego de la transición. Esto ocurre cuando un almacenador de energía con causalidad integral pasa a causalidad derivativa luego de la transición.

Para los primeros dos casos la inicialización de los estados cuando el sistema ingresa a un nuevo modo no presenta problemas significativos. Esto se debe a que el número de estados se mantiene constante o aumenta, es decir, que el sistema no reduce sus grados de libertad. Por lo que los estados del sistema permanecerán continuos después del cambio de modo. Para el último caso la inicialización se resuelve desde un punto de vista físico como una generalización del principio de conservación del momento ([Mosterman 1997](#)). En ([Edström, K. 1999](#)) pueden encontrarse los algoritmos de inicialización para los tres casos.

Para utilizar los SPJ en 20sim deben establecerse las condiciones iniciales de los modos actuantes. Para su implementación se deben reemplazar en todos los almacenadores de

energía con causalidad integral la instrucción de integración $int(x, init)$ por la instrucción $resint(x, newoutp, reset, init)$. Esta instrucción permite establecer el valor de la variable asociada al integrador cuando se pasa de un modo a otro. La variable $reset$, utilizada como argumento de la instrucción $resint$, debe ser generada por la instrucción $event(input)$ que fuerza al algoritmo de integración a encontrar el punto exacto de conmutación de modos. Para mas información acerca de las instrucciones mencionadas ver [20sim-Help](#).

Para el cálculo de las condiciones iniciales se deben conocer todos los estados del sistema (independientes y dependientes) y las relaciones constitutivas de los almacenadores de energía. Si bien el software de simulación 20sim ([Controllab Products B.V](#)) reconoce, cuando compila el modelo, la existencia de estados dependientes e identifica sus dependencias no se pudo realizar una aplicación general de los algoritmos presentados en ([Edström, K. 1999](#)) que resuelvan el problema del modelado de sistemas conmutados de una manera automática como un componente de librería. Por lo tanto la inicialización de los modos de un sistema conmutado debe ser desarrollada para cada caso particular.

4 APLICACIÓN

Se presenta como ejemplo de aplicación el motor de corriente continua (MCC) con excitación serie y debilitamiento de campo ([Leonhard, 1997](#)). Se analizan los modelos a campo pleno y a campo debilitado que permiten la descripción del sistema en un único modelo que presenta ambos modos de operación. En una segunda instancia se presenta el motor de corriente continua con excitación serie en cascada con un control tipo todo-nada.

4.1 MCC Serie con Circuito para Debilitamiento de Campo

La [Figura 8](#) muestra el equivalente electromecánico del sistema físico del motor de corriente continua con excitación en serie, mientras que las ecuaciones [5](#) expresan las expresiones que rigen la conversión electromecánica. En esta configuración el devanado de armadura se encuentra en serie con el devanado de campo. Este último puede ser debilitado mediante la conexión de una resistencia en paralelo. Dicha conexión permite operar al motor en dos modos, una a campo pleno, es decir con la llave abierta, y el otro a campo debilitado, llave cerrada. El primero de los modos se utiliza cuando la velocidad de giro del rotor se encuentra entre cero y la velocidad de base, mientras que el segundo modo se alcanza cuando la velocidad supera a la velocidad de base. ([Leonhard, 1997](#)).

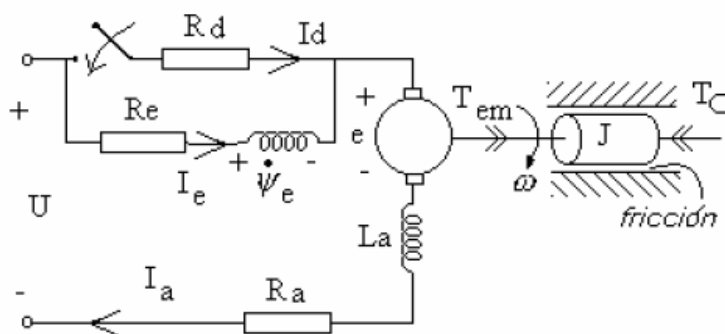


Figura 8. MCC excitación serie

$$\begin{aligned}
 T_{em} &= K\psi_e I_a \\
 e &= K\psi_e \omega \\
 \psi_e &= g(I_e)
 \end{aligned}
 \tag{5}$$

4.1.1 Modelo del MCC de Excitación Serie a Campo Pleno (modo 1).

La [figura 9](#) muestra el modelo Bond Graph del motor de corriente continua de excitación serie a campo pleno (Full Field FF). Este modo de operación contiene dos variables de estado, la corriente de armadura y la velocidad del rotor. El sistema de ecuaciones [6](#) describe la dinámica en el espacio de estados del modelo de la [figura 9](#). En ellas, $g_{\Sigma}()$ representa la suma de la característica magnética de campo $g()$ más la característica lineal de armadura ($L_a I_a$), mientras que $g'_{\Sigma}()$ representa la derivada de la función anterior respecto a su argumento, es decir, se trata de la inductancia incremental de la conexión serie de ambas bobinas, la de excitación y la de armadura.

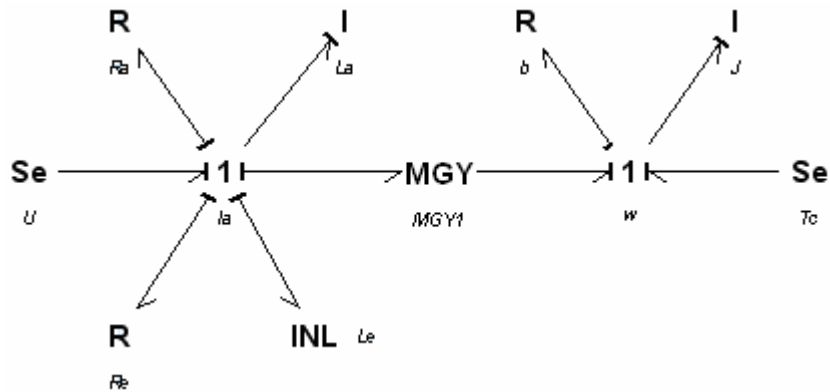


Figura 9. Modelo BG del MCC excitación serie FF

$$\begin{aligned} \dot{\omega} &= -\frac{b}{J}\omega + \frac{K}{J}g(I_a)I_a - \frac{T_c}{J} \\ \dot{I}_a &= -\frac{K}{g'_{\Sigma}(I_a)}g(I_a)\omega - \frac{R_a + R_e}{g'_{\Sigma}(I_a)}I_a + \frac{U}{g'_{\Sigma}(I_a)} \\ \psi_e &= g(I_e) = g(I_a) \end{aligned} \tag{6}$$

4.1.2 Modelo del MCC de Excitación Serie a Campo Debilitado (modo 2).

La [figura 10](#) muestra el modelo Bond Graph del motor de corriente continua de excitación serie a campo debilitado (field weakening, FW). Este modo de operación a diferencia del anterior presenta una tercera variable de estado, que corresponde al flujo de excitación. Dicha variable de estado es introducida al cerrar la llave volviendo dinámica la relación existente entre la corriente de armadura y la corriente de excitación. El sistema de ecuaciones [7](#) muestra las ecuaciones de estado del modelo de la [figura 10](#).

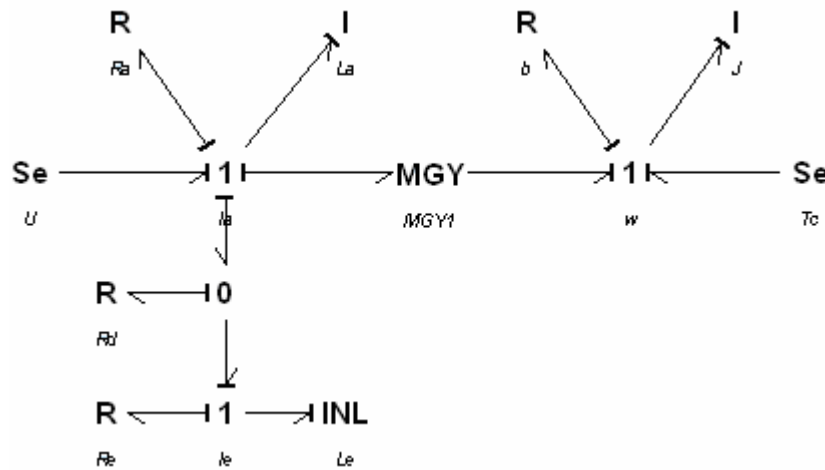


Figura 10. Modelo BG del MCC excitación serie FW

$$\begin{aligned}
 \dot{\omega} &= -\frac{b}{J} \omega + \frac{K}{J} \psi_e I_a - \frac{T_c}{J} \\
 \dot{I}_a &= -\frac{K}{L_a} \psi_e \omega - \frac{R_a + R_d}{L_a} I_a + \frac{R_d}{L_a} g^{-1}(\psi_e) + \frac{U}{L_a} \\
 \dot{\psi}_e &= R_d I_a - (R_d + R_e) g^{-1}(\psi_e)
 \end{aligned} \tag{7}$$

Para conocer el valor de las condiciones iniciales en de cada uno de los modos, esto es a campo pleno y a campo debilitado, es necesario conocer la dependencia entre los estados.

Considerando el instante de conmutación entre el modo 1 y el modo 2 en el instante de tiempo t_0 , el sistema gana un grado de libertad al debilitarse el campo, por lo que la inicialización de los nuevos estados se elijen de manera de que estos sean continuos. Quedando las condiciones iniciales para los almacenadores como indican las ecuaciones [8](#).

$$\begin{aligned}
 \psi_e(t_0^+) &= g(I_e(t_0^+)) = g(I_e(t_0^-)) \\
 I_a(t_0^+) &= I_a(t_0^-) \\
 \omega(t_0^+) &= \omega(t_0^-)
 \end{aligned} \tag{8}$$

Para el caso de la transición desde el modo 2 al modo 1, el sistema pierde un grado de libertad, por lo que el cálculo de la condición inicial de la corriente de armadura se basa en el principio de conservación del estado ([Mosterman 1997](#)). Para este caso se interpreta como la conservación del flujo magnético total. Considerando la conmutación en instante de tiempo t_1 se obtiene las ecuaciones [9](#) y [10](#) que representan el principio de invariancia del estado y las condiciones iniciales del modo 1, respectivamente.

$$\psi_\Sigma(t_1^+) = \psi_a(t_1^+) + \psi_e(t_1^+) = \psi_a(t_1^-) + \psi_e(t_1^-) \tag{9}$$

$$\begin{aligned}
 I_a(t_1^+) &= g_\Sigma^{-1}(\psi_\Sigma(t_1^+)) \\
 \psi_a(t_1^+) &= L_a * I_a(t_1^+) \\
 \omega(t_1^+) &= \omega(t_1^-)
 \end{aligned} \tag{10}$$

4.2 Modelo Bimodo Conmutado del MCC Excitación Serie

La [figura 11](#) muestra el modelo bimodo conmutado desarrollado en ([Junco et al 2007](#)) utilizando la versión compacta y genérica del *Os* presentada en este trabajo, se incorporan además dos bloques llamados *CIGenerator* y *Switch_Condition*. El primero es un bloque de ecuaciones encargado de hacer el cálculo de las condiciones iniciales para cada transición. Al mismo le ingresan los estados y modifica las condiciones iniciales que son tratadas como variables globales del sistema. Mientras que el segundo indica cuando debe hacerse la conmutación entre modos.

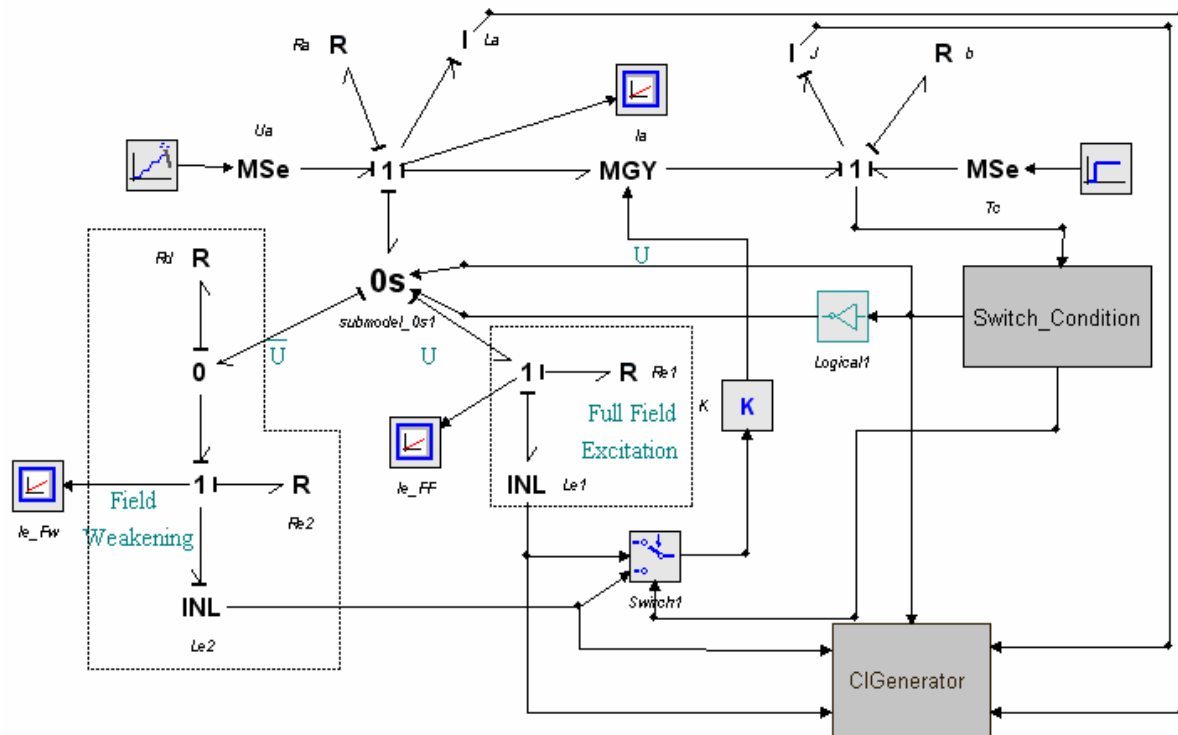


Figura 11. Modelo BG del MCC de dos modos

El control de la llave que permite el debilitamiento de campo se implementa con el bloque *Switch_Condition* mostrado en [figura 12](#).

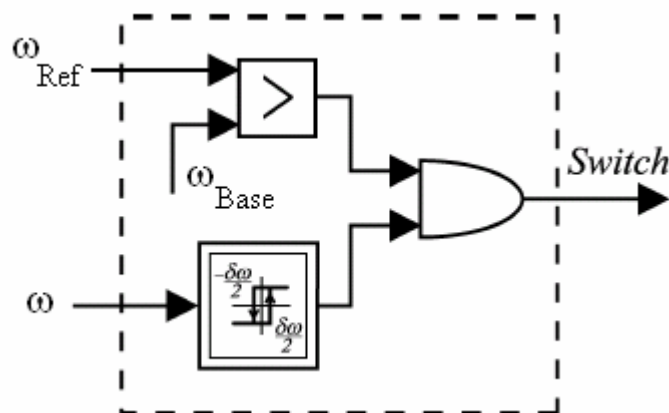


Figura 12 Esquema del circuito de control de llave

Para este circuito la condición de conmutación a campo debilitado se da solo si, simultáneamente la velocidad de referencia es mayor a la velocidad de base ($\omega_{ref} > \omega_{base}$) y la velocidad del motor es superior a una histéresis centrada en la velocidad de base. Esta histéresis es necesaria para evitar conmutaciones indeseadas a causa de posibles fluctuaciones en la velocidad del motor.

4.3 Control en Cascada del MCC Serie Bimodo con Lazo Conmutado de Corriente.

El control en cascada tipo todo-nada presentado en (Leonhard, 1997) es un esquema clásico para el control de velocidad del motor de corriente continua. El mismo se presenta en la figura 13 y consta de dos lazos de realimentación, uno que estima el torque electromagnético del motor y otro que compara la velocidad del rotor con una velocidad de referencia. El error de velocidad ingresa a un PI que obtiene como resultado un torque de referencia. La diferencia entre el torque de referencia y el torque electromagnético estimado ingresa a un comparador con histéresis que conmuta la alimentación del motor entre el valor nominal de alimentación y 0.

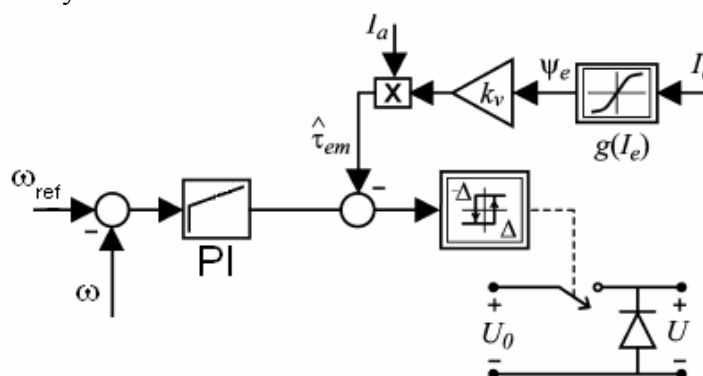


Figura 13 Esquema del control todo nada

En la figura 14 se muestra la implementación en 20sim (Controllab Products B.V) del sistema de control del motor. El bloque *controller* es un PI donde el integrador tiene limitadores de amplitud cuyos valores se eligieron según el torque nominal y una realimentación antiwindup. La reconstrucción del flujo de excitación se realizó a través del bloque *function1* que obtiene los datos de la característica magnética del devanado de excitación desde un archivo .txt.

Por último el bloque llamado MCC contiene el diagrama mostrado en la figura 11.

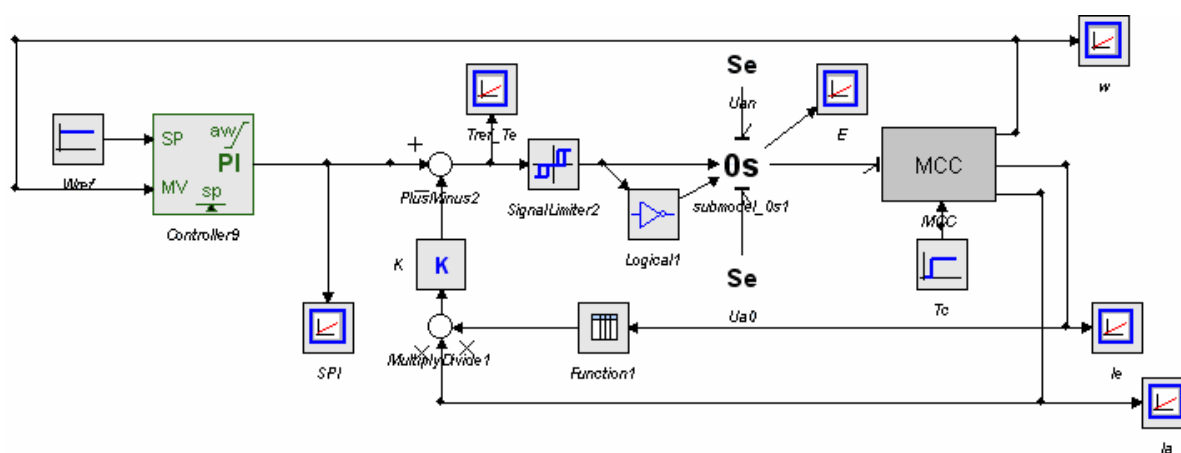


Figura 14 Esquema del control todo nada implementado en 20sim

Destacamos que esta aplicación tiene la conmutación ya vista en el modelo del motor y además otras dos conmutaciones: una en la electrónica de potencia de la alimentación del motor, y otra en la ley de control por histéresis del lazo de corriente.

5 RESULTADOS DE SIMULACIÓN:

En esta sección se presentan los resultados de simulación para la aplicación desarrollada en la [sección 4](#). Los parámetros del MCC utilizados para realizar las simulaciones fueron obtenidos de ([Chiason 1994](#)) y se presentan en el [apéndice](#).

5.1 MCC Excitación Serie de 2 Modos

Se simuló el modelo de la [figura 11](#) donde se aplicó una señal de entrada, tensión de armadura, de tipo escalón suavizado cuyo crecimiento va desde el tiempo de simulación 0 al tiempo de simulación 1 segundo, los resultados se muestran en la [figura 15](#). Las conmutaciones entre modos fueron forzadas, iniciándose a campo pleno y conmutando a campo debilitado a los 0.42 segundos de simulación. A los 0.6 segundos se vuelve a conmutar a campo pleno. El proceso se repite a los 0.8 segundos y a los 1.2 segundos.

En la simulación puede observarse la continuidad del estado I_a en la conmutación de campo pleno a campo debilitado, y la discontinuidad del mismo estado en la conmutación inversa.

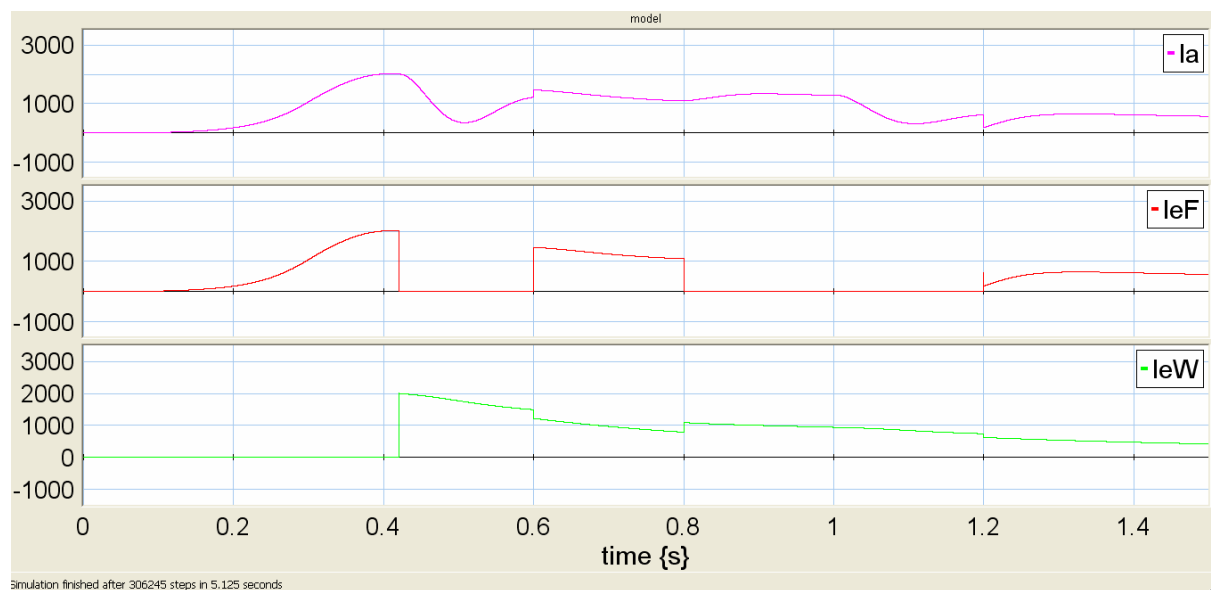


Figura 15 Evolución de la corriente de armadura y corrientes de excitación a FF y FW.

Entre los 0.6 y 0.8 segundos de simulación, que corresponde al intervalo en el cual el motor se encuentra excitado a campo pleno, la corriente de excitación a campo debilitado sigue evolucionando. Esto es debido a que el $0s$ desconecta el modelo de campo debilitado pero este sigue evolucionando por si mismo. No obstante notar que se produce una discontinuidad en dicha corriente cuando el motor pasa al modo de campo debilitado.

5.2 MCC Excitación Serie controlado.

A continuación se presentan los resultados de simulación del modelo de la [figura 14](#) para evaluar como responde el modelo conmutado al seguimiento de una velocidad de referencia. Además se aplica un torque de carga del valor nominal de motor a los 3 segundos de simulación. La velocidad de base es de 200 rad/seg. (Ver [apéndice](#)).

En la [figura 16](#) se muestra que el sistema controlado sigue a la velocidad de referencia. La reducción en la velocidad a los 3 segundos es producto de la aplicación del torque de carga.

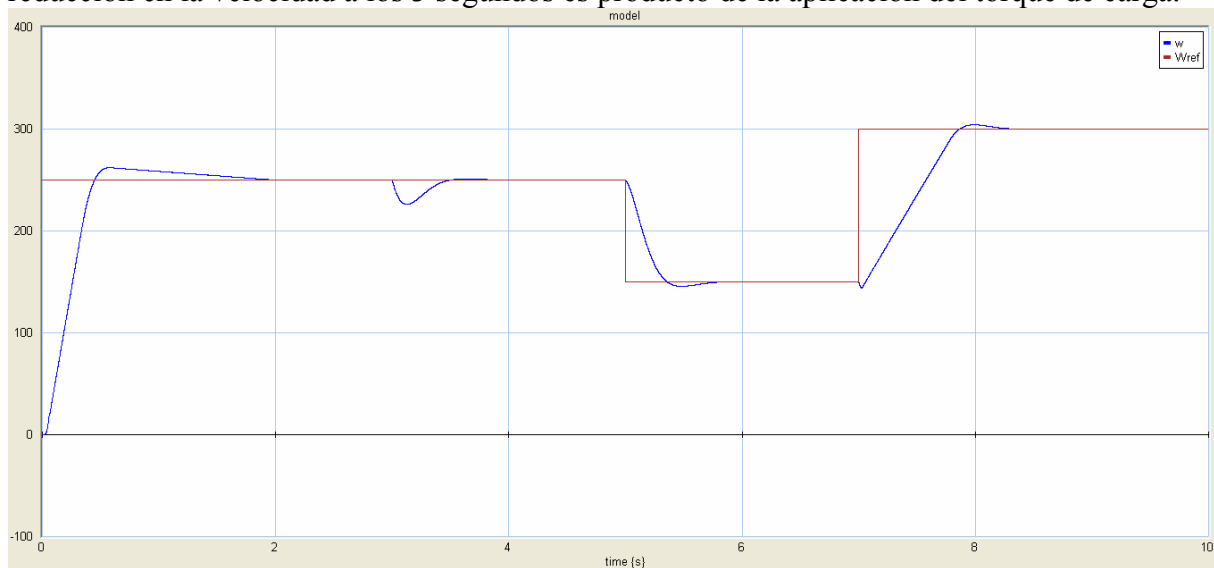


Figura 16 En rojo la velocidad de referencia, en azul la velocidad del motor

En la [figura 17](#) se muestra la evolución de la corriente de armadura y de la corriente de excitación a campo pleno y campo debilitado. Puede verse la conmutación a campo debilitado que se produce a lo 0,35 segundos de simulación aproximadamente, manteniéndose en este modo hasta los 5 segundos.

El sobrevalor que presenta la corriente de armadura a los 3 segundos se debe a la aplicación del torque de carga. Mientras que la discontinuidad que se observa a los 7 segundos es producto de la conmutación de campo debilitado a campo pleno. El sistema conmuta nuevamente a campo debilitado al sobrepasar la velocidad de base a los 7,35 segundos.

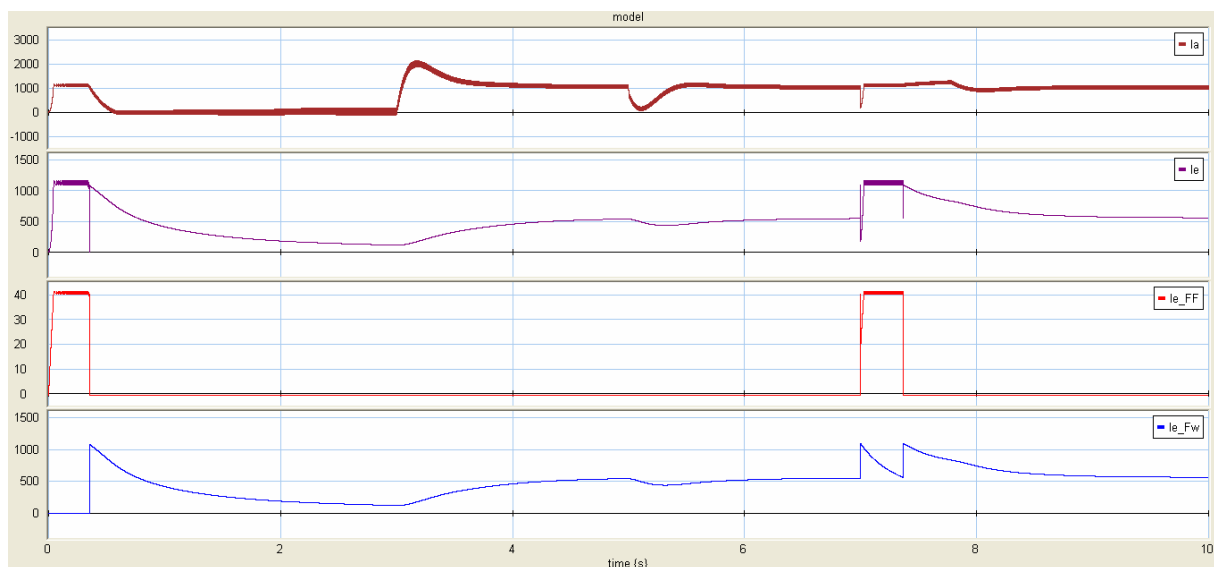


Figura 17 Evolución de la Corriente de armadura (I_a), corriente de excitación (I_e), Corrientes a campo pleno (I_{e_FF}) y campo debilitado (I_{e_Fw})

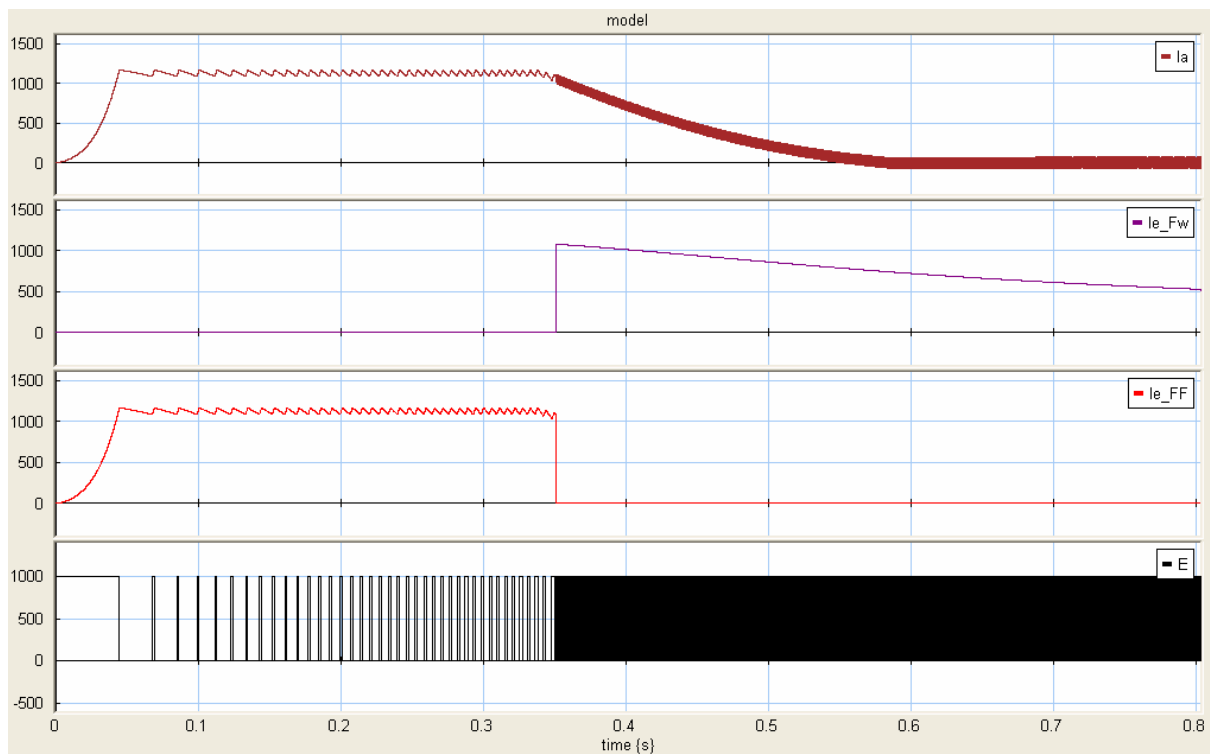


Figura 18 Evolución de la corriente de armadura, de excitación a campo debilitado y campo pleno, y tensión armadura.

La [figura 18](#) muestra la evolución temporal de la corriente de armadura la corriente de excitación, a campo pleno y campo debilitado, y las conmutaciones de la tensión de alimentación del motor. Con el propósito de observar con mayor claridad las conmutaciones de la tensión de alimentación se muestran el período inicial de la simulación comprendido entre 0 y 0.8 segundos.

6 CONCLUSIONES

Se presentó la implementación en el software 20-sim del formalismo de los SPJ para el modelado de sistemas conmutados descritos en el lenguaje bond graph. Se obtienen componentes genéricos integrables a la librería estándar de elementos BG y reutilizables para cualquier modelo conmutado.

Se discute el problema de inicialización de los modos en sistemas conmutados. Se consideran conmutaciones en las cuales el número de variables de estado de los modos involucrados pueden ser iguales, mayores o menores. Se presentó la forma de proceder para inicializar los modos así como también las instrucciones 20-sim que permiten realizarlo.

Las técnicas desarrolladas se utilizan para simular un sistema conmutado de aplicación en ingeniería. En efecto, se considera el motor de corriente continua con excitación serie y debilitamiento de campo. Los resultados de simulación muestran que los modelos conmutados propuestos responden satisfactoriamente.

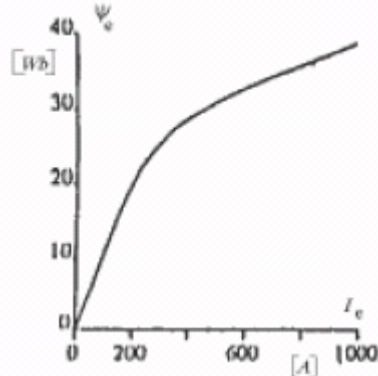
REFERENCIAS

- Chiason J. 1994. Nonlinear Differential-Geometric Techniques for Control of a Series DC Motor, *IEEE trans. Of Control System Technology* 2, No 1,35-45
 Controllab Products B.V., 20-sim™, <http://www.20sim.com/>

- Edström, K. Switched Bond Graphs Simulation and Analysis, *Department of Electrical Engineering, Linköpings University, Linköping, Sweden*. 1999
- Help 20sim 3.6 Controllab Products B.V., 20-sim™, <http://www.20sim.com/webhelp.html>
- Junco, S. Diéguez, G. Ramirez F 2007 On Conmutation Modeling in Bond Graphs. *International Conference on Bond Graph Modeling and Simulation*.2007.
- Karnopp, D. Margolis, D. Rosenberg, R. System dynamics: Modeling and simulation of mechatronic systems, *Wiley Interscience, 3rd ed.*, New York, pp. 297-336, 2000
- Leonhard, W. Control of Electrical Drives. *Springer-Verlag*, 2nd ed. New York. 1997.
- Mosterman, P. and G. Biswas. A theory of discontinuities in physical systems models. *J. Franklin Institute*, 335B (3), 401 - 439. 1998
- Mosterman, P. Hybrid Dynamic Systems; a hybrid bond graph modelling paradigm and its application in diagnosis. *Ph.D. Thesis, Vanderbilt University, Nashville, Tennessee, USA*. 1997.
- Umarikar, A. and L. Umanand. Modelling of switching systems in bond graphs using the concept of switched power junctions”, *J. Franklin Institute*, 342, 131-147. 2005,

APENDICE: PARAMETROS DEL MCC

Tensión Nominal	$U_{MAX} = 1000 \text{ V};$
Corriente Máxima	$I_{a \text{ MAX}} = 1000 \text{ A}.$
Velocidad Nominal	$\omega_{MAX} = 6860 \text{ rpm};$
Velocidad Base	$\omega_B = 1910 \text{ rpm};$
Resistencia de armadura	$R_a = 9.89 \text{ m}\Omega;$
Inductancia de armadura	$L_a = 1.4 \text{ mH};$
Resistencia de excitación	$R_e = 14,85 \text{ m}\Omega;$
Resistencia de Shunt	$R_d = 16,96 \text{ m}\Omega.$
Constante Electromagnética	$K = 0.04329 \text{ Nm/WbA};$
Inercia	$J = 3 \text{ Kg}\cdot\text{m}^2.$
Torque de carga máximo	$T_{MAX} = 1370 \text{ Nm}.$
Curva de magnetización	$\psi_e \text{ vs. } I_e$



Característica magnética de la inductancia de excitación.