

Simulación de Modulación Vectorial para un Conversor PWM Fuente de Voltaje

Roberto Cárdenas Dobson, DIE Universidad de Chile.

Curso de Electrónica de Potencia.

EL6009-1 EM722-1

Utilizando los programas MATLAB y SIMULINK es posible simular con bastante flexibilidad algoritmos SVM (acrónimo de Space Vector Modulation). Si se representan los IGBTs como interruptores ideales se logra un modelo adecuado para el estudio de algoritmos de control, pero que en general no se utiliza para el estudio de eficiencia, pérdidas o algunos fenómenos no lineales que están presentes en el inversor.

Fig.1 muestra un conversor PWM fuente de voltaje con un neutro ficticio en el enlace DC (símbolo “O”). Solo dos vectores se muestran en esta figura. En las Figs. 2 y 3 se muestra el resto de los ocho vectores que se pueden obtener considerando todas las combinaciones permitidas en los interruptores del conversor PWM.

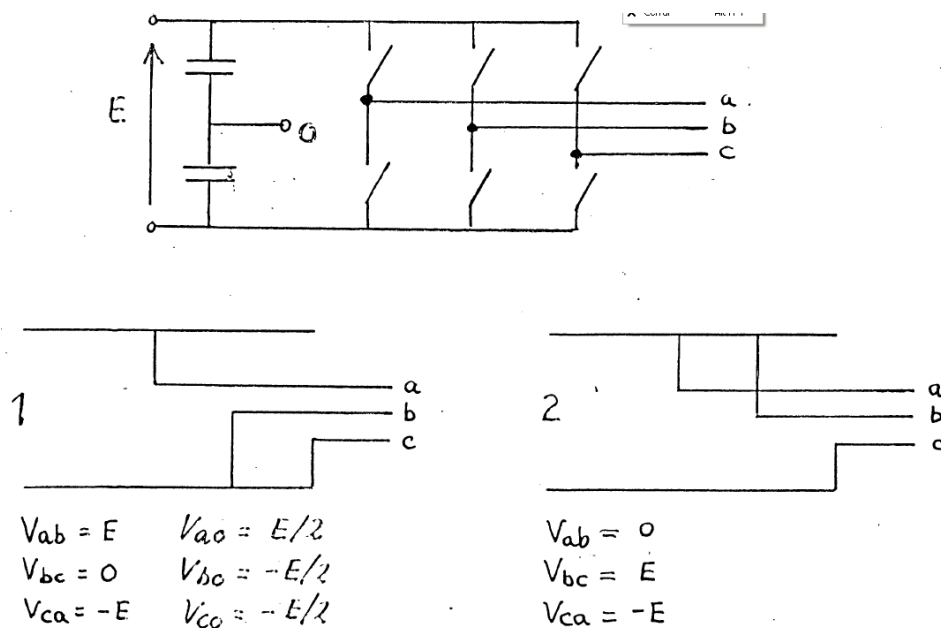
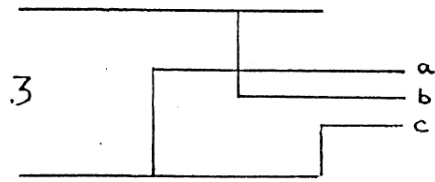


Figura 1. Conversor PWM con neutro ficticio y dos de los vectores activos.

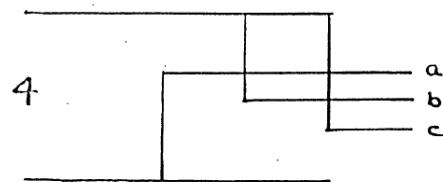
A. Vectores Disponibles

Utilizando los vectores activos y nulos que se encuentran en las Figs. 1-3, se puede implementar la Tabla 1. El módulo y ángulo de cada vector se calcula utilizando la siguiente transformada α - β :

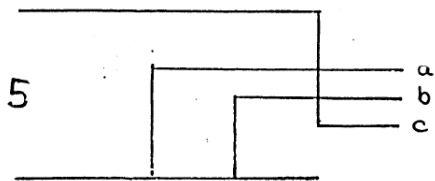
$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} V_{ao} \\ V_{bo} \\ V_{co} \end{bmatrix} \quad (1)$$



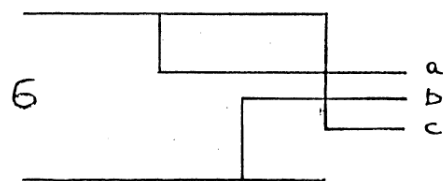
$$\begin{aligned} V_{ab} &= -E \\ V_{bc} &= E \\ V_{ca} &= 0 \end{aligned}$$



$$\begin{aligned} V_{ab} &= -E \\ V_{bc} &= 0 \\ V_{ca} &= E \end{aligned}$$

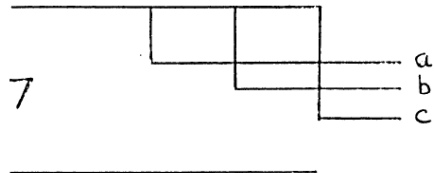


$$\begin{aligned} V_{ab} &= 0 \\ V_{bc} &= -E \\ V_{ca} &= E \end{aligned}$$

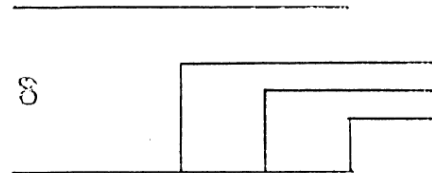


$$\begin{aligned} V_{ab} &= E \\ V_{bc} &= -E \\ V_{ca} &= 0 \end{aligned}$$

Fig. 2. Vectores activos 4 al 6.



$$\begin{aligned} V_{ab} &= 0 \\ V_{bc} &= 0 \\ V_{ca} &= 0 \end{aligned}$$



$$\begin{aligned} V_{ab} &= 0 \\ V_{bc} &= 0 \\ V_{ca} &= 0 \end{aligned}$$

Fig. 3 Vectores nulos.

Tabla I. Resumen de los vectores obtenidos desde las Figs. 1-3.

Vector	V_{ao}	V_{bo}	V_{co}	V_{ab}	V_{bc}	V_{ca}	Módulo	Ángulo (Grados)
1	$E/2$	$-E/2$	$-E/2$	E	0	$-E$	E	0
2	$E/2$	$E/2$	$-E/2$	0	E	$-E$	E	60
3	$-E/2$	$E/2$	$-E/2$	$-E$	E	0	E	120
4	$-E/2$	$E/2$	$E/2$	$-E$	0	E	E	180
5	$-E/2$	$-E/2$	$E/2$	0	$-E$	E	E	240
6	$E/2$	$-E/2$	$E/2$	E	$-E$	0	E	300
7	$E/2$	$E/2$	$E/2$	0	0	0	0	X
8	$-E/2$	$-E/2$	$-E/2$	0	0	0	0	X

Donde los valores de $V_{a0} \dots V_{c0}$ para cada combinación de interruptores corresponden a los que se muestran en la Tabla 1. El módulo y ángulo de cada vector se pueden calcular utilizando las ecuaciones:

$$|V| = \sqrt{V_{\alpha}^2 + V_{\beta}^2} \quad \theta_e = \tan^{-1}(V_{\beta} / V_{\alpha}) \quad (2)$$

La función $\tan^{-1}$, corresponde a la función arco-tangente de 4 cuadrantes, definida habitualmente como *atan2* en los sistemas basados en DSP Texas Instruments. El DSP (Digital Signal Processing) utilizado en los prototipos implementados por nuestro grupo investigación es habitualmente la tarjeta TI DSK6713, de alta velocidad, programable en lenguaje C.

Los vectores 1-8, se pueden mostrar en un eje estacionario α - β , tal como se observa en la Fig. 4.

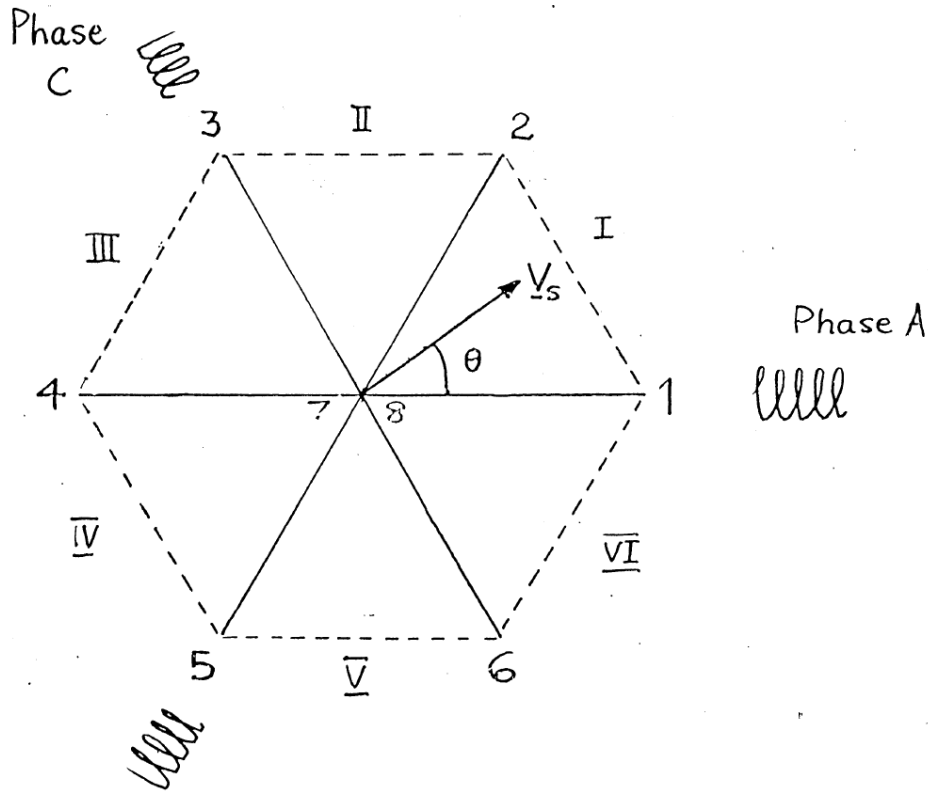


Fig. 4. Vectores de la Tabla I en coordenadas α - β .

B. Cálculo de los Ciclos de Trabajo de los Vectores.

En esta sección se asume que se desea sintetizar un vector V_s que se encuentra entre los vectores V_1 y V_2 tal como se muestra en la Fig. 5. Los ciclos de trabajo son δ_0 , δ_1 y

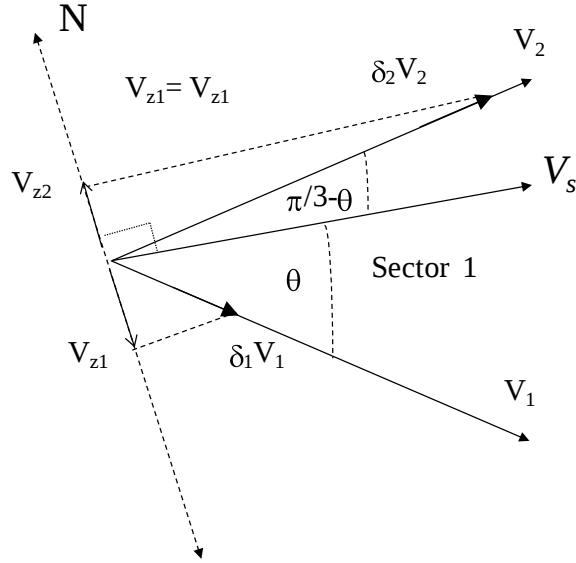


Fig. 5. Diagrama de vectores en el sector 1, sintetizado utilizando los vectores V_1 y V_2

δ_2 , correspondientes a los vectores nulos y vectores V_1 y V_2 respectivamente. Los ciclos de trabajo están sujetos a la siguiente restricción:

$$\delta_0 + \delta_1 + \delta_2 = 1 \quad (3)$$

En general el ciclo de trabajo de un vector se entiende como el porcentaje (o tanto por uno) del periodo de switching que dicho vector ocupa. Por ejemplo si $\delta_1=0.5$ y la frecuencia de switching es 2kHz entonces el vector V_1 ocupa la mitad del tiempo o $250\mu s$ de ese período T_s en particular.

Para sintetizar el vector V_s se aplican fracciones $\delta_1 V_1$ y $\delta_2 V_2$ de los vectores activos. La ecuación utilizada para sintetizar V_s es:

$$V_s = \delta_1 V_1 \cos \theta + \delta_2 V_2 \cos \left(\frac{\pi}{3} - \theta \right) \quad (4)$$

recordando que los módulos de todos los vectores son iguales a E (ver Tabla 1), (4) se puede escribir:

$$V_s = \delta_1 E \cos \theta + \delta_2 E \cos \left(\frac{\pi}{3} - \theta \right) \quad (5)$$

se puede obtener otra ecuación considerando que los componentes que se encuentran en el eje perpendicular (eje N) al vector V_s deben anularse (por lo tanto $V_{z1}=V_{z2}$ en Fig. 5). Esto se expresa como:

$$\delta_1 E \sin(\theta) = \delta_2 E \sin\left(\frac{\pi}{3} - \theta\right) \Rightarrow \delta_1 = \delta_2 \frac{\sin\left(\frac{\pi}{3} - \theta\right)}{\sin(\theta)} \quad (6)$$

Utilizando esta expresión de δ_1 en (5) se tiene:

$$\delta_2 = \left[\sin\left(\frac{\pi}{3} - \theta\right) \cos\left(\frac{\pi}{3}\right) + \cos\left(\frac{\pi}{3} - \theta\right) \sin\left(\frac{\pi}{3}\right) \right] \frac{V_s}{E} \sin(\theta) \quad (7)$$

Reemplazando las identidades de (8):

$$\begin{aligned} \sin(\alpha + \beta) &= \sin\alpha \cos\beta + \cos\alpha \sin\beta \\ \cos(\alpha + \beta) &= \cos\alpha \cos\beta - \sin\alpha \sin\beta \end{aligned} \quad (8)$$

En la ecuación 7, se obtiene la siguiente expresión:

$$\delta_2 \left[\left[\sin\left(\frac{\pi}{3}\right) \cos\left(\frac{\pi}{3}\right) - \cos\left(\frac{\pi}{3}\right) \sin\left(\frac{\pi}{3}\right) \right] \cos(\theta) + \left[\cos\left(\frac{\pi}{3}\right) \cos\left(\frac{\pi}{3}\right) + \sin\left(\frac{\pi}{3}\right) \sin\left(\frac{\pi}{3}\right) \right] \sin(\theta) \right] = \frac{V_s}{E} \sin(\theta) \quad (9)$$

Finalmente el ciclo de trabajo δ_2 se puede obtener como:

$$\delta_2 = \frac{2V_s}{\sqrt{3}E} \sin(\theta) \quad (10)$$

Utilizando (10) en (6) se obtiene δ_1 como:

$$\delta_1 = \frac{2V_s}{\sqrt{3}E} \sin\left(\frac{\pi}{3} - \theta\right) \quad (11)$$

$$\delta_0 = 1 - \delta_1 + \delta_2 \quad (12)$$

Las ecuaciones pueden utilizarse de la forma mostrada en (10-12). Sin embargo se debe recordar que los componentes α - β de V_s , el vector a sintetizar, al igual que los componentes α - β de los vectores V_1 a V_2 se obtuvieron utilizando la transformada mostrada en (1). Esta transformada entrega un módulo de V_s , que se calcula de acuerdo a (2) que no es igual al valor rms del voltaje trifásico a sintetizar. Como se ha demostrado anteriormente en clases, el módulo del vector V_s se puede obtener como:

$$|V_s| = \sqrt{V_\alpha^2 + V_\beta^2} = \frac{3}{2} V_m \quad (13)$$

Por lo tanto se debe recordar que el módulo de V_s es 1.5 veces el valor máximo (V_m) de la onda sinusoidal a sintetizar (valores fase-neutro ficticio en este caso).

Los ciclos de trabajo también pueden ser representados en función del índice de modulación (m), definiéndose este como:

$$m = 2 \frac{V_m}{E} \quad (14)$$

Donde V_m es el valor máximo de la componente fundamental de la señal a sintetizar.

Utilizando (13,14) en (10,11), los ciclos de trabajo δ_1 y δ_2 se calculan como:

$$\begin{aligned}\delta_1 &= \frac{\sqrt{3}}{2} m \sin \left(\frac{\pi}{3} - \theta \right) \\ \delta_2 &= \frac{\sqrt{3}}{2} m \sin \theta\end{aligned}\tag{15}$$

El máximo valor de V_s que se puede modular sin distorsión de baja frecuencia, corresponde al radio del círculo circunscrito en el hexágono formado por los seis vectores. Esto se muestra en la Fig. 6. Utilizando esta figura, el máximo vector V_s puede ser calculado como:

$$E \cos \left(\frac{\pi}{6} \right) = V_s^{\max} \Rightarrow \frac{\sqrt{3}}{2} E = \frac{3}{2} V_m^{\max} \Rightarrow m_{\max} = \frac{2}{\sqrt{3}}\tag{16}$$

Al igual que en el caso del PWM convencional con inyección de terceros armónicos, el valor máximo de índice de modulación alcanzado con SVM (sin distorsión de baja

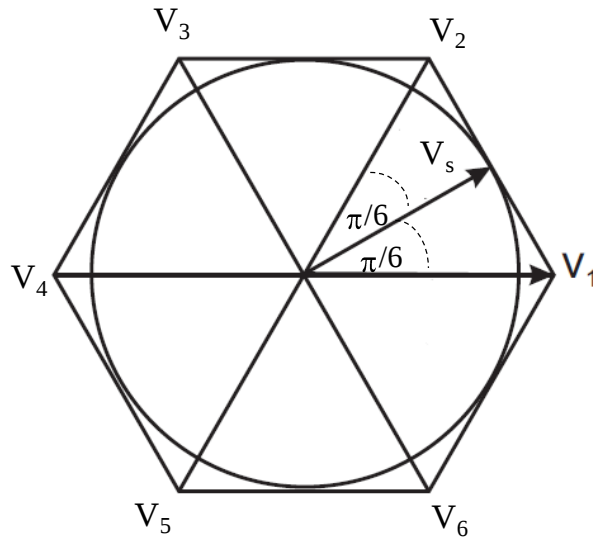


Fig. 6. Círculo de radio V_s circunscrito en el hexágono formado por los seis vectores.

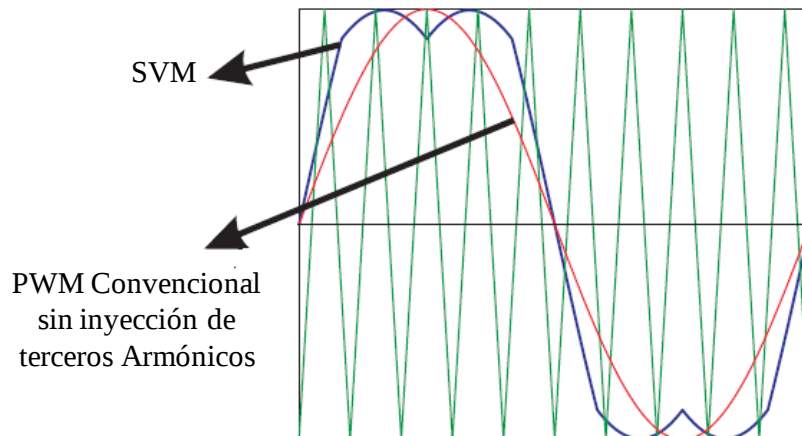


Fig. 7. Diferencias entre PWM convencional y Modulación SVM.

frecuencia) es de $m = 2/\sqrt{3}$ equivalente a $m=1.1547$ aproximadamente. Fig. 7 muestra las modulaciones obtenidas con el algoritmo SVM y la obtenida con el PWM convencional donde se compara la onda modulante con una señal triangular.

Como se ha discutido anteriormente en este curso, la modulación SVM puede sintetizar una onda fundamental de amplitud 15.47% mayor a la obtenida con un PWM convencional.

C. Implementación de PWM SimétricoDoble.

Al efectuar el cálculo de los ciclos de trabajo, se obtienen los tiempos totales de los vectores cero y vectores activos de acuerdo a:

$$T_0 = \delta_0 T_s \quad T_1 = \delta_1 T_s \quad T_2 = \delta_2 T_s \quad (17)$$

La forma habitual de implementar el PWM es ordenar los vectores de acuerdo a un patrón simétrico. Por ejemplo para el sector 1, un patrón adecuado es el que se muestra en la Fig. 8.

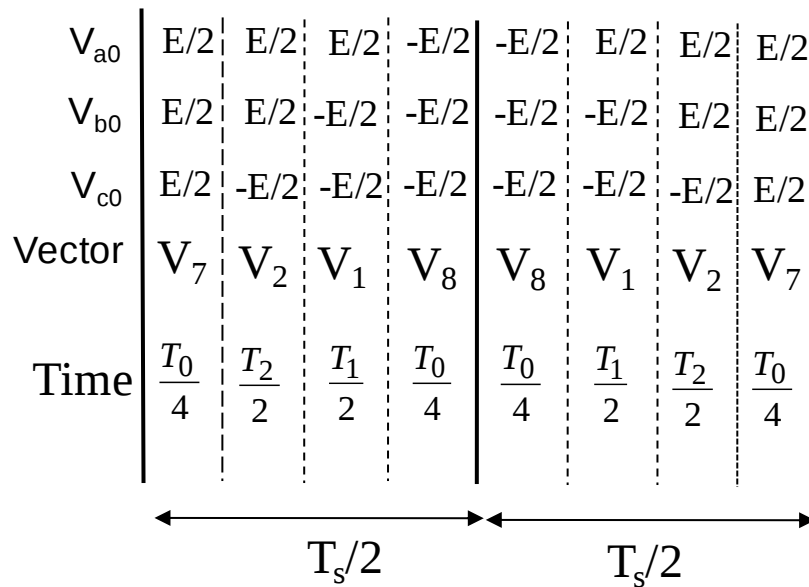


Fig. 8. Ejemplo de un PWM simétrico doble.

El patrón de la Fig. 8 podría implementarse como $V_8 V_1 V_2 V_7 V_7 V_2 V_1 V_8$. Es decir utilizar como primer vector cero V_8 en lugar de V_7 .

Nótese en la Fig. 8 que en cada conmutación de vectores, por ejemplo de V_7 a V_2 , solo un switch IGBT cambia de estado. Por ese motivo el patrón simétrico doble utilizado en esta simulación cumple con dos objetivos. Minimizar las pérdidas por

switching en el conversor y mejorar el espectro de la onda sintetizada al utilizar un “espejo” del primer semiperíodo de T_s en la segunda mitad (ver Fig. 8).

D. Bases para la Modelación del SVM utilizando MATLAB y SIMULINK

El objetivo de este documento es ayudar a los estudiantes con la programación necesaria para implementar un modulador de espacio de estado en MATLAB. Por lo tanto lo que se entrega a continuación no es a nivel de detalle ni cubre cada aspecto del trabajo ya que se espera un aporte personal significativo de los alumnos para obtener un algoritmo SVM funcionando adecuadamente.

De acuerdo a la experiencia del profesor de esta asignatura en la simulación de algoritmos de modulación para convertidores matriciales de tres piernas, convertidores matriciales de cuatro piernas y convertidores PWM de cuatro piernas, la mejor manera de modelar un conversor para evaluar sistemas de control es utilizar archivos y funciones tipo M, con interruptores ideales.

Uno de estos archivos M contiene una rutina de inicialización de variables públicas globales. Esto significa que el valor de la variable es visible en todo el ambiente MATLAB/SIMULINK y puede ser modificada en cualquier subrutina del programa. Un ejemplo del uso y ventajas de las variables globales se muestra en la rutina “*initial.m*” :

(no se utilizan tildes porque en un programa tipo “m” no es conveniente utilizar estos símbolos)

initial.m

```
global cuenta;          % Cuenta el tiempo de cada vector.
global vectors;         % Almacena información de todos los vectores
global T1 T2 T3 T4 T5 T6 T7; %Tiempos utilizados.
global V1 V2 V3 V4 V5 V6 V7; % Vectores utilizados en cada Ti
global sectors;         % orden de los vectores por sector.
global Ts;              % 1/(switching frequency)
global Tr ;             % resolución del contador (mínimo 500uS)
global CountF;          % Cuenta Final;

fs=1500; % Frecuencia de frecuencia solicitada. Se puede cambiar arbitrariamente
Ts=1/fs; % Periodo de switching del modulador
Tr=0.5e-6;%Resolucion minima solicitada.Se puede cambiar arbitrariamente
CountF=round(Ts/Tr)% Numero de pulsos de período Tr para lograr el periodo de muestreo Ts. Se
% redondea CountF al entero mas cercano
%Sugerencia
%Para una mejor resolucion intente que Ts/Tr sea un número entero o muy cercano a entero antes de
% aplicar el operador round.

% Definicion de vectores
```

```

V1=[1 -1 -1]; % como multiplos de E/2
V2=[1 1 -1];
V3=[-1 1 -1];
V4=[-1 1 1];
V5=[-1 -1 1];
V6=[1 -1 1];
V7=[1 1 1];
V8=[-1 -1 -1];
vectors=[V1;V2;V3;V4;V5;V6;V7;V8]; % En la variable publica vectors quedan los vectores
% almacenados, un vector por fila.
% Definicion de sectores
% Valores propuestos no necesariamente optimizados, consulte bibliografía acerca del tema.

S1=[ 8 1 2 7 2 1 8]; % Arreglo a utilizar en el sector 1 para obtener minimas conmutaciones.
S2=[ 7 2 3 8 3 2 7]; % Arreglo a utilizar en el sector 2 para obtener minimas conmutaciones.
S3=[ 8 3 4 7 4 3 8]; % Arreglo a utilizar en el sector 3 para obtener minimas conmutaciones.
S4=[      ];
S5= []; %Trabajo Personal,
S6=[      ];
Sectors=[S1;S2;S3;S4;S5;S6] % Vectores que se deben utilizar en cada sector, ahora estan almacenados
                             % en la variable publica sectors.
cuenta=1;                    % variable publica "cuenta" inicializada .

```

En esta subrutina se han colocado las variables básicas que deben ser inicializadas antes de comenzar la simulación. Una vez que se han inicializado estas variables públicas, se debe programar en Simulink el bloque donde se calcularán los ciclos de trabajo y también se obtendrán las salidas (V_{ao} , V_{bo} , V_{co}) del modulador en cada período de switching. Para esto debe implementar un sistema como el mostrado en la Fig. 9. El bloque "Mux" toma tres entradas, voltajes α - β y tensión en el enlace dc, y los coloca en un formato que puede

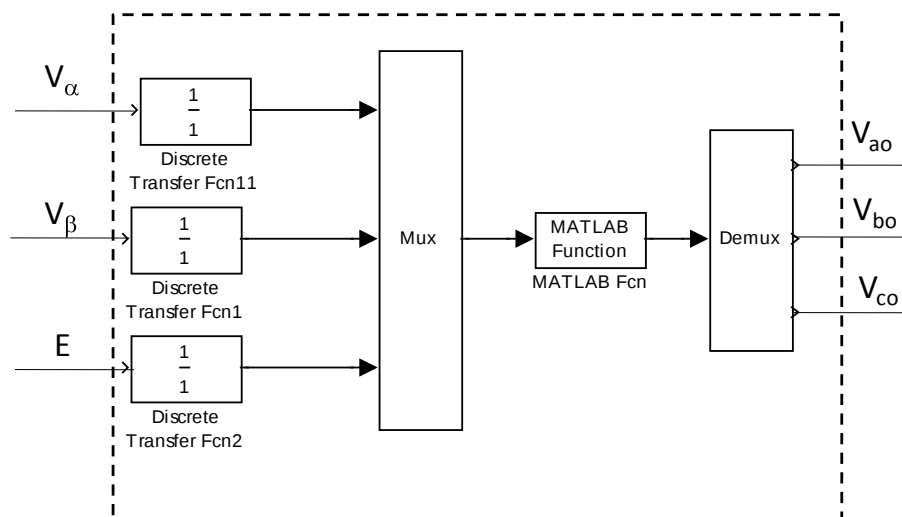


Fig. 9. Modulador de espacio de estado implementado en MATLAB/SIMULINK

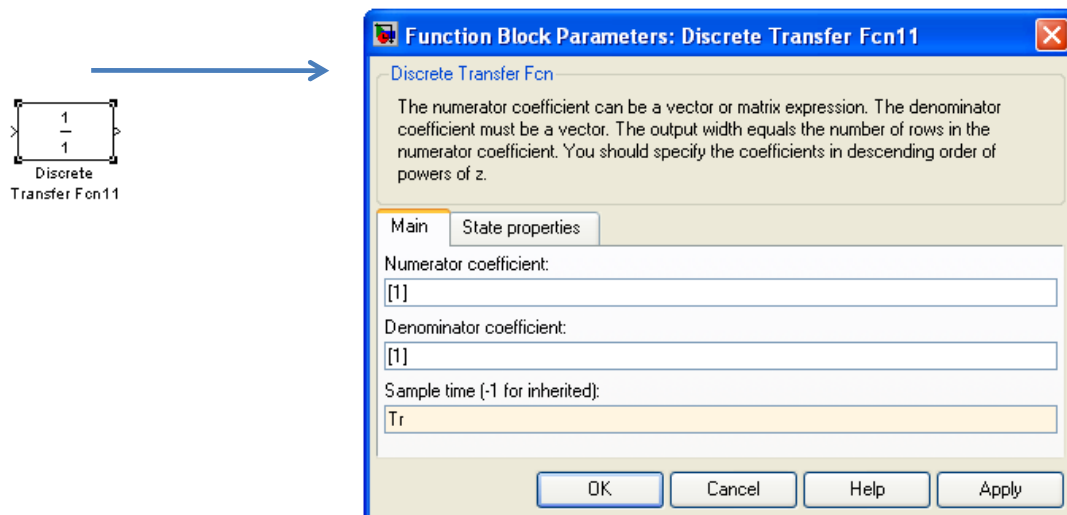


Fig. 10. Muestreador implementado.

ser ingresado a una función de Matlab. El bloque “demux” toma las tres salidas de voltaje de la función y las coloca en paralelo. El bloque “1/1” corresponde a un muestreador. Cada T_r segundos se ingresan los valores requeridos a la función de MATLAB. Fig. 10 muestra los parámetros del muestreador implementado utilizando el bloque “función de transferencia discreta” de Simulink

El bloque MATLAB Function, llama a una función externa, en este caso la función SVM. El bloque se muestra en mayor detalle en la Fig. 11.

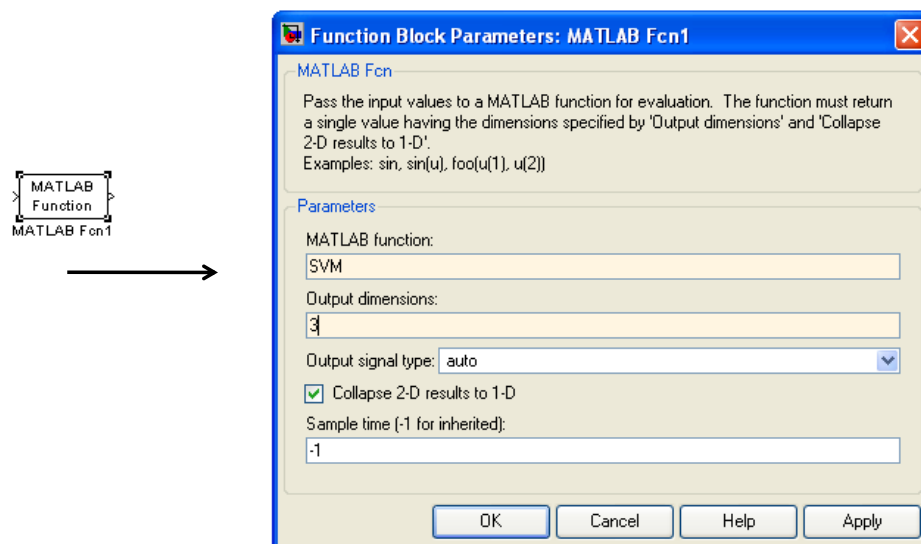


Fig. 11. Función SVM que considera tres salidas.

La función SVM efectúa la modulación vectorial. Parte del código que puede utilizar esta función es:

SVM.m

```
function [Vout] = SVM(Vin);
%
% Se debe repetir la definicion de todas las variables publicas comunes entre "initial" y "SVM"
%
global cuenta;          % Cuenta el tiempo de cada vector.
global vectors;         % Almacena información de todos los vectores
global T1 T2 T3 T4 T5 T6 T7; % Tiempos utilizados.
global V1 V2 V3 V4 V5 V6 V7; % Vectores utilizados.
global sectors;         % orden de los vectores.
global Ts;              % 1/(switching frequency)
global Tr ;             % resolución del contador (mínimo 500uS)
global CountF;          % Cuenta Final;

Valfa=Vin(1);           % Voltaje alfa de entrada (normalmente salida de controladores)
Vbeta=Vin(2);           % Voltaje beta de entrada (normalmente salida de controladores)
E=Vin(3);               % Voltaje del enlace DC.

if (cuenta==1)          % Primer instante del ciclo de muestreo, se deben calcular los tiempos y vectores
                        % en este punto.

    Th_o=atan2(Vbeta,Valfa); % Encuentra el angulo del vector de salida
    if (Th_o<0)
        Th_o=Th_o+2*pi;
    end
    SECTOR=floor((Th_o+pi/3)*3/pi); % Sencilla formula que encuentra el sector a partir de Tho
    if (sector==7)                % captura y corrige posibles errores de redondeo
        SECTOR=1;
    end;
    Theta= Th_o-(pi/3)*(SECTOR-1) % Encuentra el angulo con respecto al primer vector del
    % sector a considerer.
    %
    % Encuentra los vectores pre-arreglados del sector, que se encuentran almacenados en el
    % arreglo publico sectors.
    %
    L1=sectors(SECTOR,1);
    L2=sectors(SECTOR,2);
    L3=sectors(SECTOR,3);
    L4=sectors(SECTOR,4);
    L5=sectors(SECTOR,5);
    L6=sectors(SECTOR,6);
    L7=sectors(SECTOR,7);
    %
    % Los vectores L1....L7 se utilizan en este sector.
    %
    V1=vectors(L1,1:3)*E/2; % V1 tiene la fila complete del vector L1
    V2=vectors(L2,1:3) *E/2; % V2 tiene la fila complete del vector L2
    V3=vectors(L3,1:3) *E/2; % V3 tiene la fila complete del vector L3
    V4=vectors(L4,1:3) *E/2; % V4 tiene la fila complete del vector L4
    V5=vectors(L5,1:3) *E/2; % V5 tiene la fila complete del vector L5
    V6=vectors(L6,1:3) *E/2; % V6 tiene la fila complete del vector L6
```

```

V7=vectors(L7,1:3) *E/2; % V7 tiene la fila complete del vector L7
%
% En este punto se debe insertar la rutina que calcula los ciclos de trabajo D1 y D2 utilizando
% Theta, Valfa, Vbeta y E.
% D1 y D2 Corresponden a los vectores uno y dos del sector en que se esta trabajando.
% CUIDADO QUE NO ES TAN FACIL
%
D0=1-(D1+D2); % ciclo de trabajo vectores cero.
T1=round(0.25*D0*CountF); % Tiempo del primer vector cero T0/4
T2=round(0.5*D1*CountF); % Tiempo del primer vector no nulo
T3=round(0.5*D2*CountF); % Tiempo del segundo vector no nulo
T4=2*T1; % Tiempo total del Segundo vector cero T0/2
T5=T3 % Aplica propiedad espejo del PWM simetrico
T6=T2;
T7=T1;
End; (En este punto finaliza el if (cuenta= =1) )
%
% Despues de calcular los tiempos envia la salida a Vout
%
if (cuenta<=T1)
Vao=V1(1);
Vbo=V1(2);
Vco=V1(3);
Vout=[VaoVboVco];
elseif ((cuenta>T1)&(cuenta<=(T1+T2)))
Vao=V2(1);
Vbo=V2(2);
Vco=V2(3);
Vout=[VaoVboVco];
elseif ((cuenta>(T1+T2))&(cuenta<=(T1+T2+T3)))
Vao=V3(1);
Vbo=V3(2);
Vco=V3(3);
Vout=[VaoVboVco];
:
:
: Continua de la misma forma hasta llegar a:
elseif ((cuenta>(T1+T2+T3+T4+T5+T6))&(cuenta<=CountF))
Vao=V7(1);
Vbo=V7(2);
Vco=V7(3);
Vout=[VaoVboVco];
End % En este punto finaliza el if (cuenta<=T1)
%
% Ahora Finaliza la rutina
%
cuenta=cuenta+1;
if (cuenta>CountF) % Termino el periodo de switching
cuenta=1; % Comienza de nuevo
end;

return % Vuelve a simulink

```

E. Otras Consideraciones

Como se explicó al principio de este apunte, la modulación se efectúa con respecto a un punto “O” ficticio. Por lo cual las tensiones V_{ao} , V_{bo} y V_{co} están referidas a ese

punto. Sin perder generalidad se puede asumir que el estator de la máquina está conectado en estrella con un neutro (que también puede ser ficticio). Desde el punto de vista del neutro del motor de inducción, las tensiones son $V_{ao}=V_{an}-V_{on}$, $V_{bo}=V_{bn}-V_{on}$ y $V_{co}=V_{cn}-V_{on}$.

La máquina es alimentada por las tensiones de línea. Por lo tanto la tensión entre las fases a y b es:

$$V_{ab} = V_{an} - V_{on} - V_{bn} + V_{on} = V_{an} - V_{bn} \quad (18)$$

Es decir las tensiones de secuencia cero desaparecen de las tensiones de línea. En este caso las tensiones de secuencia cero se definen como:

$$V_0 = \frac{1}{3} [V_{ao} + V_{bo} + V_{co}] \quad (19)$$

e incluyen la tensión V_{on} así como los terceros armónicos que pudieran inyectarse a la salida del modulador para incrementar el índice de modulación.

Para fines de simulación se pueden utilizar tensiones medidas con respecto al neutro de la máquina, calculadas desde las tensiones de línea aplicadas por el conversor. Esto se calcula como:

$$\begin{bmatrix} V_{an} \\ V_{bn} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} V_{ab} \\ V_{bc} \end{bmatrix} \quad V_{cn} = -\frac{1}{3} (V_{an} + V_{bn}) \quad (20)$$

recuerde que si aplica directamente las tensiones de salida del modulador a la máquina, (como tensiones de fase de salida) estaría aplicando tensiones de secuencia cero al motor de inducción. Eso no es posible debido a que el motor está conectado a las tensiones de línea las cuales no contienen estos componentes armónicos.