



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

Máster en Electrónica Industrial y Automática

MÁSTER EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE MÁSTER

ANEXOS

**NUEVO CONVERTIDOR BASADO EN MMC PARA APLICACIONES
FOTOVOLTAICAS**

Autor: D. José Arroyo Galán
Tutores: D. Luis Carlos Herrero de Lucas
Fernando Martínez Rodrigo
Departamento de Ingeniería Electrónica

Valladolid, 10 de julio de 2019

1 CÓDIGO MPPT PERTURBACIÓN & OBSERVACIÓN (P&O)

```
function D = P_O(P,V)

% *****
% ---- PARÁMETROS DE ENTRADA
%       P = POTENCIA DEL PANEL FOTOVOLTAICO
%       V = TENSIÓN DEL PANEL FOTOVOLTAICO
% ---- PARÁMETRO DE SALIDA
%       D = CICLO DE SERVICIO PARA EL CC/CC
% *****

% Se definen variables persistentes para mantener su valor en la siguiente
% iteración.
persistent D_ANT;
persistent P_ANT;
persistent V_ANT;

% En la primera iteración, se inicializan las variables persistentes a un
% valor determinado.
if isempty(D_ANT)
    D_ANT = 0.5;
end
if isempty(P_ANT)
    P_ANT = 20;
end
if isempty(V_ANT)
    V_ANT = 10;
end

% Se define el incremento del ciclo de servicio que se empleará para
% perturbar el funcionamiento del panel y movernos así por la curva de
% máxima potencia.
deltaD = 0.001;
% Si la potencia actual es mayor que la potencia leída en el instante
% anterior...
if(P > P_ANT)
    % Si la tensión actual es mayor que la tensión leída en el instante
    % anterior...
    if(V > V_ANT)
        % Nos encontramos a la izquierda del MPP. El siguiente desplazamiento
        % debe ser a la derecha (aumentar tensión), por lo que aumentamos el
        % ciclo de servicio del CC/CC.
        D = D_ANT - deltaD;
    else
        % Nos encontramos a la derecha del MPP. El siguiente desplazamiento
        % debe ser a la izquierda (reducir tensión), por lo que aumentamos el
        % ciclo de servicio del CC/CC.
        D = D_ANT + deltaD;
    end
end
% Si la potencia actual es menor que la potencia leída en el instante
% anterior...
else
    % Si la tensión actual es mayor que la tensión leída en el instante
    % anterior...
    if(V > V_ANT)
        % Nos encontramos a la derecha del MPP. El siguiente desplazamiento
        % debe ser a la izquierda (reducir tensión), por lo que aumentamos el
        % ciclo de servicio del CC/CC.
        D = D_ANT + deltaD;
    else
        % Nos encontramos a la izquierda del MPP. El siguiente desplazamiento
        % debe ser a la derecha (aumentar tensión), por lo que aumentamos el
        % ciclo de servicio del CC/CC.
        D = D_ANT - deltaD;
    end
end

% Se almacenan los valores actuales de ciclo de servicio, tensión del panel
% y potencia del panel, ya que en la siguiente iteración se corresponderán
% con los valores anteriores.
D_ANT = D;
V_ANT = V;
P_ANT = P;

End
```


2 CÓDIGO MPPT CONDUCTANCIA INCREMENTAL

```
function D = COND_INC(I,V)

% *****
% ---- PARÁMETROS DE ENTRADA
%       I = CORRIENTE DEL PANEL FOTOVOLTAICO
%       V = TENSIÓN DEL PANEL FOTOVOLTAICO
% ---- PARÁMETRO DE SALIDA
%       D = CICLO DE SERVICIO PARA EL CC/CC
% *****

% Se definen variables persistentes para mantener su valor en la siguiente
% iteración.
persistent D_ANT;
persistent I_ANT;
persistent V_ANT;

% En la primera iteración, se inicializan las variables persistentes a un
% valor determinado.
if isempty(D_ANT)
    D_ANT = 0.5;
end
if isempty(I_ANT)
    I_ANT = 2;
end
if isempty(V_ANT)
    V_ANT = 10;
end

% Se define el incremento del ciclo de servicio que se empleará para
% perturbar el funcionamiento del panel y movernos así por la curva de
% máxima potencia.
deltaD = 0.001;

% Se calcula los incrementos experimentados en la corriente y en la tensión
% aportada por el panel fotovoltaico.
INC_I = I - I_ANT;
INC_V = V - V_ANT;

% Si el incremento en la tensión es nulo...
if(INC_V == 0)
    % Y el incremento en la corriente también es nulo...
    if(INC_I == 0)
        % Nos encontramos en el MPP, por lo que el ciclo de servicio se
        % mantiene constante.
        D = D_ANT;
    else
        % Si el incremento en la tensión es nulo pero el incremento de la
        % corriente es positivo...
        if(INC_I > 0)
            % Se debe incrementar la tensión de trabajo del panel
            D = D_ANT - deltaD;
        else
            % Se debe decrementar la tensión de trabajo del panel
            D = D_ANT + deltaD;
        end
    end
end
% Si el incremento de la tensión no es nulo...
else
    if((INC_I/INC_V) == -(I/V))
        % En el MPP. Se mantiene el ciclo de servicio.
        D = D_ANT;
    else
        if((INC_I/INC_V) > -(I/V))
            % Nos encontramos a la izquierda del MPP. Se debe aumentar la
            % tensión de trabajo del panel.
            D = D_ANT - deltaD;
        else
            % Nos encontramos a la derecha del MPP. Se debe disminuir la
            % tensión de trabajo del panel.
            D = D_ANT + deltaD;
        end
    end
end

% Se almacenan los valores actuales de ciclo de servicio, tensión del panel
```

```
% y potencia del panel, ya que en la siguiente iteración se corresponderán
% con los valores anteriores.
D_ANT = D;
V_ANT = V;
I_ANT = I;

end
```

3 CÓDIGO DEL MODULADOR

```
function [nzsuf,nzinf] = fcn(Vref, Triang, Vdc)

    if Vref < Triang
        nzsuf = 5;
        nzinf = 0;
    elseif Vref > Triang && Vref < (Triang + Vdc/5)
        nzsuf = 4;
        nzinf = 1;
    elseif Vref > Triang && Vref < (Triang + 2*Vdc/5)
        nzsuf = 3;
        nzinf = 2;
    elseif Vref > Triang && Vref < (Triang + 3*Vdc/5)
        nzsuf = 2;
        nzinf = 3;
    elseif Vref > Triang && Vref < (Triang + 4*Vdc/5)
        nzsuf = 1;
        nzinf = 4;
    else
        nzsuf = 0;
        nzinf = 5;
    end
end
```


4 CÓDIGO MPPT – MÓDULO MMC AISLADO

```
function [g1,g2] = fcn(V_PANEL,I_PANEL,IPM)

% *****
% ---- PARÁMETROS DE ENTRADA
%       I_PANEL = CORRIENTE DEL PANEL FOTOVOLTAICO
%       V_PANEL = TENSIÓN DEL PANEL FOTOVOLTAICO
%       IPM = CORRIENTE POR LA RAMA DEL MMC
% ---- PARÁMETROS DE SALIDA
%       g1 = ESTADO DEL POLO SUPERIOR DEL MÓDULO MMC
%       g2 = ESTADO DEL POLO INFERIOR DEL MÓDULO MMC
% *****

% Se calcula la potencia a la que está trabajando el panel
P_PANEL=V_PANEL*I_PANEL;

% Se definen variables persistentes para mantener su valor en la siguiente
% iteración.
persistent V_PANEL_ANT P_PANEL_ANT;

% En la primera iteración, se inicializan las variables persistentes a un
% valor determinado.
if isempty(V_PANEL_ANT)
    V_PANEL_ANT = 10;
end
if isempty(P_PANEL_ANT)
    P_PANEL_ANT = 20;
end

% Si la potencia actual es mayor que la potencia leída en el instante
% anterior...
if(P_PANEL > P_PANEL_ANT)
    % Si la tensión actual es mayor que la tensión leída en el instante
    % anterior...
    if(V_PANEL > V_PANEL_ANT)
        % Nos encontramos a la izquierda del MPP. El siguiente desplazamiento
        % debe ser a la derecha (aumentar tensión).
        if(IPM > 0)
            % Si la corriente por la rama del módulo es positiva, se permite
            % su entrada al módulo MMC para cargar el condensador.
            g1=1;
            g2=0;
        else
            % Si la corriente por la rama del módulo es negativa, se escoge
            % el estado de los polos propicio para que no se evacúe energía.
            g1=0;
            g2=1;
        end
    end
    % Si la tensión actual es menor que la potencia leída en el instante
    % anterior...
    else % La siguiente perturbación debe ser negativa
        % Si la corriente por la rama del módulo es positiva, se escoge
        % el estado de los polos más propicio para evacuar energía con
        % las condiciones dadas (o para que al menos no aumente
        % demasiado).
        if(IPM > 0)
            g1=0;
            g2=1;
        else
            % Si la corriente por la rama del módulo es negativa, se permite
            % comunicación con el condensador para que ayude a sacar energía
            g1=1;
            g2=0;
        end
    end
end
else % La potencia es menor que la anterior
    % Se repite el proceso pero a la inversa.
    if(V_PANEL > V_PANEL_ANT) % A la drch de la curva P-V
        %Siguiente perturbación negativa
        if(IPM > 0)
            g1=0;
            g2=1;
        else
            g1=1;
            g2=0;
        end
    end
end
end
```

```
end
else % A la izq de la curva P-V
    %Siguiente perturbación positiva
    if(IPM > 0)
        g1=1;
        g2=0;
    else
        g1=0;
        g2=1;
    end
end
end

% Se almacenan los valores de potencia y tensión para emplearlos en la
% siguiente iteración.
P_PANEL_ANT=P_PANEL;
V_PANEL_ANT=V_PANEL;

end
```

5 ALGORITMO MPPT + ORDENACIÓN REDISEÑADO

```
function Ctrl =
fcn(Vc1,Vc2,Vc3,Vc4,Vc5,Ipv1,Ipv2,Ipv3,Ipv4,Ipv5,Vc1_ant,Vc2_ant,Vc3_ant,Vc
4_ant,Vc5_ant,Ipv1_ant,Ipv2_ant,Ipv3_ant,Ipv4_ant,Ipv5_ant,nz,IPM)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Potencias actuales y anteriores de los paneles
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P_PANEL_1=Ipv1*Vc1;
P_PANEL_2=Ipv2*Vc2;
P_PANEL_3=Ipv3*Vc3;
P_PANEL_4=Ipv4*Vc4;
P_PANEL_5=Ipv5*Vc5;

P_PANEL_1_ANT = Vc1_ant*Ipv1_ant;
P_PANEL_2_ANT = Vc2_ant*Ipv2_ant;
P_PANEL_3_ANT = Vc3_ant*Ipv3_ant;
P_PANEL_4_ANT = Vc4_ant*Ipv4_ant;
P_PANEL_5_ANT = Vc5_ant*Ipv5_ant;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MPPT Panel 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(P_PANEL_1 > P_PANEL_1_ANT)
    if(Vc1 > Vc1_ant) %A la izq de la curva P-V
        %Siguiente perturbación positiva
        if(IPM > 0)
            g11=1;
        else
            g11=0;
        end
    else % A la drch de la curva P-V
        %Siguiente perturbación negativa
        if(IPM > 0)
            g11=0;
        else
            g11=1;
        end
    end
else
    if(Vc1 > Vc1_ant) % A la drch de la curva P-V
        %Siguiente perturbación negativa
        if(IPM > 0)
            g11=0;
        else
            g11=1;
        end
    else % A la izq de la curva P-V
        %Siguiente perturbación positiva
        if(IPM > 0)
            g11=1;
        else
            g11=0;
        end
    end
end
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MPPT Panel 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(P_PANEL_2 > P_PANEL_2_ANT)
    if(Vc2 > Vc2_ant) %A la izq de la curva P-V
        %Siguiete perturbación positiva
        if(IPM > 0)
            g12=1;
        else
            g12=0;
        end
    else % A la drch de la curva P-V
        %Siguiete perturbación negativa
        if(IPM > 0)
            g12=0;
        else
            g12=1;
        end
    end
end
else
    if(Vc2 > Vc2_ant) % A la drch de la curva P-V
        %Siguiete perturbación negativa
        if(IPM > 0)
            g12=0;
        else
            g12=1;
        end
    else % A la izq de la curva P-V
        %Siguiete perturbación positiva
        if(IPM > 0)
            g12=1;
        else
            g12=0;
        end
    end
end
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MPPT Panel 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(P_PANEL_3 > P_PANEL_3_ANT)
    if(Vc3 > Vc3_ant) %A la izq de la curva P-V
        %Siguiete perturbación positiva
        if(IPM > 0)
            g13=1;
        else
            g13=0;
        end
    else % A la drch de la curva P-V
        %Siguiete perturbación negativa
        if(IPM > 0)
            g13=0;
        else
            g13=1;
        end
    end
end
else
    if(Vc3 > Vc3_ant) % A la drch de la curva P-V
        %Siguiete perturbación negativa
        if(IPM > 0)
            g13=0;
        else

```

```

        g13=1;
    end
else % A la izq de la curva P-V
    %Siguiete perturbación positiva
    if(IPM > 0)
        g13=1;
    else
        g13=0;
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MPPT Panel 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(P_PANEL_4 > P_PANEL_4_ANT)
    if(Vc4 > Vc4_ant) %A la izq de la curva P-V
        %Siguiete perturbación positiva
        if(IPM > 0)
            g14=1;
        else
            g14=0;
        end
    else % A la drch de la curva P-V
        %Siguiete perturbación negativa
        if(IPM > 0)
            g14=0;
        else
            g14=1;
        end
    end
end
else
    if(Vc4 > Vc4_ant) % A la drch de la curva P-V
        %Siguiete perturbación negativa
        if(IPM > 0)
            g14=0;
        else
            g14=1;
        end
    else % A la izq de la curva P-V
        %Siguiete perturbación positiva
        if(IPM > 0)
            g14=1;
        else
            g14=0;
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MPPT Panel 5
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(P_PANEL_5 > P_PANEL_5_ANT)
    if(Vc5 > Vc5_ant) %A la izq de la curva P-V
        %Siguiete perturbación positiva
        if(IPM > 0)
            g15=1;
        else
            g15=0;
        end
    else % A la drch de la curva P-V
        %Siguiete perturbación negativa

```

```

        if(IPM > 0)
            g15=0;
        else
            g15=1;
        end
    end
else
    if(Vc5 > Vc5_ant) % A la drch de la curva P-V
        %Siguiete perturbación negativa
        if(IPM > 0)
            g15=0;
        else
            g15=1;
        end
    else % A la izq de la curva P-V
        %Siguiete perturbación positiva
        if(IPM > 0)
            g15=1;
        else
            g15=0;
        end
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Almaceno el valor de los polos según el MPPT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
POLOS_PM = [g11;g12;g13;g14;g15];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ver qué panel está más alejado del MPPT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Se calcula los incrementos experimentados en la corriente y en la tensión
% aportada por el panel fotovoltaico.
INC_I_1 = Ipv1 - Ipv1_ant;
INC_V_1 = Vc1 - Vc1_ant;

INC_I_2 = Ipv2 - Ipv2_ant;
INC_V_2 = Vc2 - Vc2_ant;

INC_I_3 = Ipv3 - Ipv3_ant;
INC_V_3 = Vc3 - Vc3_ant;

INC_I_4 = Ipv4 - Ipv4_ant;
INC_V_4 = Vc4 - Vc4_ant;

INC_I_5 = Ipv5 - Ipv5_ant;
INC_V_5 = Vc5 - Vc5_ant;

%Cálculo del valor de la derivada de la potencia de cada panel
dPdV1 = Ipv1 + Vc1*(INC_I_1/INC_V_1);
dPdV2 = Ipv2 + Vc2*(INC_I_2/INC_V_2);
dPdV3 = Ipv3 + Vc3*(INC_I_3/INC_V_3);
dPdV4 = Ipv4 + Vc4*(INC_I_4/INC_V_4);
dPdV5 = Ipv5 + Vc5*(INC_I_5/INC_V_5);

dPdV = [dPdV1;dPdV2;dPdV3;dPdV4;dPdV5];

```

```

% Ordenar los vectores de derivadas según convengan
if(IPM >= 0)
    % Si la corriente es positiva, nos centramos en dPdV positivas primero
    [~,dPdV_orden]=sort(dPdV,1,'descend');
else
    % Si la corriente es negativa, nos centramos en dPdV negativas primero
    [~,dPdV_orden]=sort(dPdV,1,'ascend');
end

% Determinar cuantos módulos van a conducir
if (nz==0)
    Ctrl_sin_ord=[0;0;0;0;0];
elseif (nz==1)
    Ctrl_sin_ord=[1;0;0;0;0];
elseif (nz==2)
    Ctrl_sin_ord=[1;1;0;0;0];
elseif (nz==3)
    Ctrl_sin_ord=[1;1;1;0;0];
elseif (nz==4)
    Ctrl_sin_ord=[1;1;1;1;0];
else
    Ctrl_sin_ord=[1;1;1;1;1];
end

% Contabilizo los polos que deberían conducir según el algoritmo MPPT
n_POLOS_PM = POLOS_PM(1)+POLOS_PM(2)+POLOS_PM(3)+POLOS_PM(4)+POLOS_PM(5);

if(nz==n_POLOS_PM)
    [~,POLOS_PM_sort]=sort(POLOS_PM,1,'descend');
    % crear una matriz de dos columnas: vector de control desordenado y
orden
    matriz_ctrl=cat(2,Ctrl_sin_ord,POLOS_PM_sort);
    % ordenar el vector de control y extraerlo de la matriz de dos columnas
    Ctrl=(sortrows(matriz_ctrl,2))*[1;0];
elseif(nz<n_POLOS_PM)
    % Tengo que priorizar los polos a poner en conducción según dPdV
    POLOS_PM_selec=[0;0;0;0;0];
    i=0;
    for j=1:5
        if(POLOS_PM(dPdV_orden(j))==1)
            POLOS_PM_selec(dPdV_orden(j))=1;
            i=i+1;
            if(i==nz)
                break;
            end
        end
    end
    end
    [~,POLOS_PM_sort]=sort(POLOS_PM_selec,1,'descend');
    % crear una matriz de dos columnas: vector de control desordenado y
orden
    matriz_ctrl=cat(2,Ctrl_sin_ord,POLOS_PM_sort);
    % ordenar el vector de control y extraerlo de la matriz de dos columnas
    Ctrl=(sortrows(matriz_ctrl,2))*[1;0];
else
    % Caso en el que nz>n_POLOS_PM
    % En primer lugar, adquirir los polos que deben conducir si o si
    POLOS_PM_selec=[0;0;0;0;0];
    POLOS_PM_selec=POLOS_PM

```

```

%Calculo los polos que faltan por poner en conducción
dif_polos=nz-n_POLOS_PM

% Poner a 1 los polos más cercanos al MPPT según dPdV
i=0;
for j=1:5
    if (POLOS_PM(dPdV_orden(j))~=1)
        POLOS_PM_selec(dPdV_orden(j))=1;
        i=i+1;
        if (i==dif_polos)
            break;
        end
    end
end

[~,POLOS_PM_sort]=sort(POLOS_PM_selec,1,'descend');
% crear una matriz de dos columnas: vector de control desordenado y
orden
matriz_ctrl=cat(2,Ctrl_sin_ord,POLOS_PM_sort);
% ordenar el vector de control y extraerlo de la matriz de dos columnas
Ctrl=(sortrows(matriz_ctrl,2))*[1;0];
end

end

```