% CÁLCULO DE RESTRICCIONES DE ENERGÍA Y POTENCIA INDIVIDUALES

%Cálculo de Emin y Emáx para cada PEV

Eni\_min=zeros(nPEV,tiempo);

Eni\_max=zeros(nPEV,tiempo);

SOC\_ARR=zeros(nPEV,tiempo);

Emin\_evol=[];

Emax\_evol=[];

ESOC\_evol=[];

for k=1:nPEV;

Eni\_min(k,Hc(k)-hinicial)=SOC\_a(k); % Condicion inicial

Eni\_max(k,Hc(k)-hinicial)=SOC\_a(k); % Condicion inicial

for n=Hp(k)+hinicial-1:tiempo;

Eni\_min(k,n)=SOC\_Da(k); % Condicion final

Eni\_max(k,n)=SOC\_Da(k); % Condicion final

end

for n=1:Hni(k)-1;

Eni\_min(k,(Hp(k)+hinicial-n))=max((Eni\_min(k,(Hp(k)+hinicial-n+1))-(Pcarga/(1/delta\_t))),SOC\_a(k));

end

Evol\_min=Eni\_min(k,:); %Esta sentencia pone el valor del SOC\_a de cada PEV donde Eni\_min tiene el valor 0

Evol\_min(Eni\_min(k,:)<=0)=SOC\_a(1,k+1-1);

Emin\_evol=[Emin\_evol;Evol\_min];

for n=1:Hni(k)-1

Eni\_max(k,(Hc(k)-hinicial+n))=min((Eni\_max(k,(Hc(k)-hinicial+n-1))+(Pcarga/(1/delta\_t))),SOC\_Da(k));

end

Evol\_max=Eni\_max(k,:); %Esta sentencia pone el valor del SOC\_a de cada PEV donde Eni\_max tiene el valor 0

Evol\_max(Eni\_max(k,:)<=0)=SOC\_a(1,k+1-1);

Emax\_evol=[Emax\_evol;Evol\_max];

SOC\_ARR(k,Hc(k)-hinicial:end)=ones(1,tiempo+hinicial+1-Hc(k)).\*SOC\_a(k);

Evol\_SOC=SOC\_ARR(k,:); %Esta sentencia pone el valor del SOC\_a de cada PEV donde SOC\_ARR tiene el valor 0

Evol\_SOC(SOC\_ARR(k,:)<=0)=SOC\_a(1,k+1-1);

ESOC\_evol=[ESOC\_evol;Evol\_SOC];

end

Emax\_evol1=[Emax\_evol(1:nPEV,2:tiempo),60\*ones(nPEV,1)]; % El 60 es la capacidad total del PEV

Emin\_evol1=[Emin\_evol(1:nPEV,2:tiempo),60\*ones(nPEV,1)]; % El 60 es la capacidad total del PEV

%Cálculo de Pmáx para cada PEV

Pmax=zeros(nPEV,tiempo);

for k=1:nPEV;

for n=Hc(k)-hinicial:Hp(k)+hinicial

Pmax(k,n)=Pcarga;

end

end

%------------------------------------------------------------------------------

% CÁLCULO DE RESTRICCIONES DE ENERGÍA y POTENCIA POR AGREGADOR

for a=1:nag

SumEmin([a],[1:tiempo])=sum(Eni\_min([cPEV(a)+1:cPEV(a+1)],[1:tiempo]));

end

for a=1:nag

SumEmax([a],[1:tiempo])=sum(Eni\_max([cPEV(a)+1:cPEV(a+1)],[1:tiempo]));

end

for a=1:nag

SumSOC\_A([a],[1:tiempo])=sum(SOC\_ARR([cPEV(a)+1:cPEV(a+1)],[1:tiempo]));%no tiene en cuenta la disponibilidad de potencia

end

SumSOC\_total=sum(SumSOC\_A);

SumEmin\_total=sum(SumEmin)-SumSOC\_total;%restriciones de energia para el PDT

SumEmax\_total=sum(SumEmax)-SumSOC\_total;%restriciones de energia para el PDT

SumPmax1=zeros(nag,tiempo);

for a=1:nag

SumPmax1([a],[1:tiempo])=sum(Pmax([cPEV(a)+1:cPEV(a+1)],[1:tiempo]));%no tiene en cuenta la disponibilidad de potencia

end

SumPmax=min(SumPmax1,Pdisp);