APÉNDICES

APÉNDICE I

Archivo caso13barras.m

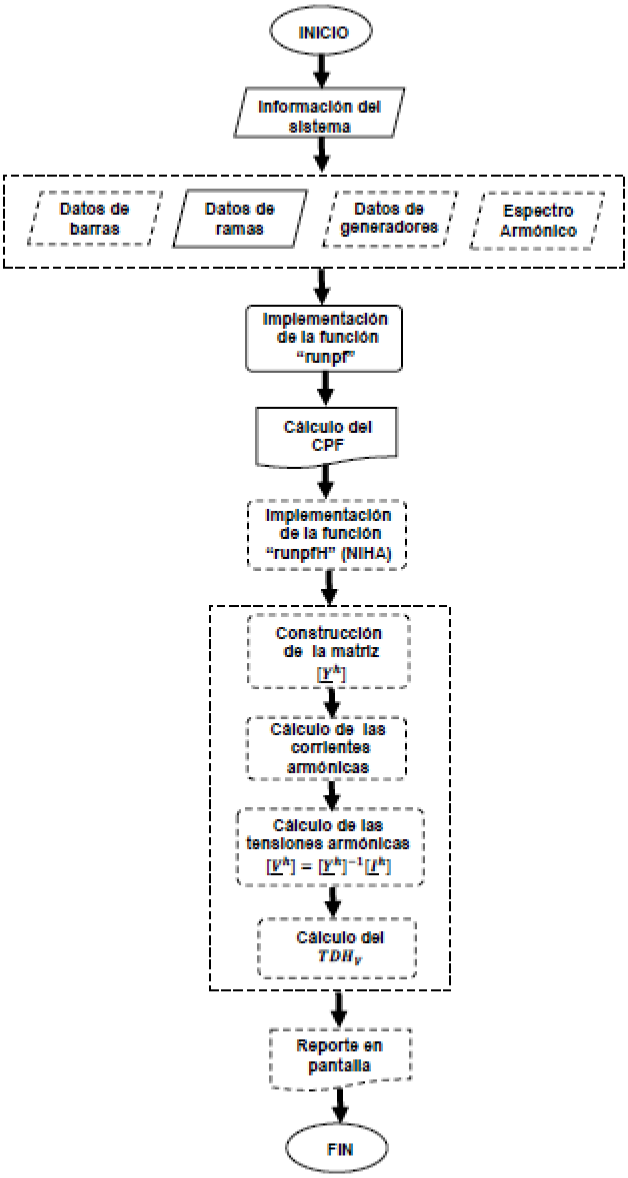
A continuación se presenta la estructura “mpc ”en un archivo .m que almacena los datos del Sistema de trece (13) barras para

```
function mpc = caso13barras
%   MATPOWER
%   $Id: caso13barras.m  2015-8-22 20:41:32 francis$
%   $Id: caso13barras.m  2017-10-30  Claris$
______________________________________________________________________________________________________________________________________
%% MATPOWER Case Format : Version 2
mpc.version = '2';
%%-----  Power Flow Data  -----%%
______________________________________________________________________________________________________________________________________
%% system MVA base
mpc.baseMVA = 10;
______________________________________________________________________________________________________________________________________
%% bus data

%        bus_i    type    Pd    Qd    Gs    Bs    area    Vm    Va
         baseKV zone    Vmax   Vmin  nonl_load  filter Ql Qc
mpc.bus = [

         1    3    0    0    0    0    1    1    0    69    1    1.05    0.95  0    0    0    0;
         2    2    0    0    0    0    1    1    0    13.8   2    1.05    0.95  0    0    0    0;
         3    1    0.6    0.53  0    0    1    1    0    0.48    4    1.05    0.95    1    0    0    0;%Carga no lineal
conectada
         4    1    0    0    0    0    1    1    0    69    1    1.05    0.95  0    0    0    0;
         5    1    2.24   2    0    0    1    1    0    13.8   3    1.05    0.95  0    1    0    0;
         6    1    0    0    0    0    1    1    0    13.8   3    1.05    0.95  0    0    0    0;
         7    1    1.15   0.29  0    0    1    1    0    0.48    5    1.05    0.95    1    0    0    0;%Carga no lineal
conectada
         8    1    1.31   1.13 0    0    1    1    0    4.16    6    1.05    0.95  0    0    0    0;
         9    1    0    0    0    0    1    1    0    13.8   3    1.05    0.95  0    0    0    0;
         10   1    0.81   0.8    0    0    1    1    0    0.48    7    1.05    0.95    1    0    0    0;%Carga no lineal
conectada
         11   1    0    0    0    0    1    1    0    13.8   3    1.05    0.95  0    0    0    0;
         12   1    0.37   0.33  0    0    1    1    0    0.48    8    1.05    0.95    1    0    0    0;%Carga no lineal
conectada
         13   1    2.8    2.5  0    0    1    1    0    2.4    9    1.05    0.95  0    0    0    0;
];

______________________________________________________________________________________________________________________________________
%% generator data
%        bus      Pg       Qg  Qmax Qmin    Vg       mBase    status    Pmax
         Pmin     Pc1      Pc2      Qc1min   Qc1max   Qc2min   Qc2max    ramp_agc
         ramp_10  ramp_30  ramp_q   apf  Rg  Xg
mpc.gen = [
         1  0  0   50   0   1      10   1   100   0  0  0  0    0    0    0  0    0    0    0    0  0.00045    0.00990;
         2  2  0   50   0   0.995  10   0   100   0  0  0  0    0    0    0  0    0    0    0    0  0.00192    0.07168;
   ];


______________________________________________________________________________________________________________________________________
%% branch data
%        fbus     tbus     r          x       b       rateA    rateB    rateC
         ratio    angle    status     angmin  angmax
mpc.branch = [
   1      4   0.00139   0.00296   0     150   0     0    1       0     1   -360   360; %transformador
   2      3   0.06391   0.37797   0     150   0     0   0.975   0     1   -360   360; %transformador
```

```
   2      5   0.00122    0.00243    0     150    0      0     1          0       1   -360    360;
   4      5   0.00313    0.05324    0     150    0      0     1          0       1   -360    360;
   5      6   0.00075    0.00063    0     150    0      0     1          0       1   -360    360;
   5      9   0.00157    0.00131    0     150    0      0     1          0       1…-360    360;
   5     11   0.00109    0.00091    0     150    0      0     1          0       1   -360    360;
   6      7   0.05918    0.35510    0     150    0      0     1          0       1   -360    360; %transformador tap nominal
   6      8   0.04314    0.34514    0     150    0      0     0.95       0       1   -360    360; %transformador tap nominal
   9     10   0.05829    0.37888    0     150    0      0     0.975      0       1   -360    360; %transformador
  11     12   0.05575    0.36240    0     150    0      0     1          0       1   -360    360; %transformador
  11     13   0.01218    0.14616    0     150    0      0     0.95       0       1   -360    360; %transformador

   ];

_____________________________________________________________________________________________________________
___________________________________________
%% Harmonic Source Data
% #H          percent   angle
mpc.HSData=[

   1     100.00    0.00
   3     0.00      0.00
   5     18.24    -55.68
   7     11.90    -84.11
   9     0.00      0.00
   11    5.73     -143.56
   13    4.01     -175.58
   15    0.00      0.00
   17    1.93      111.39
   19    1.39      68.30
   21    0.00      0.00
   23    0.94     -24.61
   25    0.86     -67.64
   27    0.00      0.00
   29    0.71     -145.46
   31    0.62      176.83
   33    0.00      0.00
   35    0.44      97.40
   37    0.38      54.36


];
```

APÉNDICE II

Metodología NIHA empleada para la obtención de HPF en MATPOWER



APÉNDICE III

Funcion Idx_bus.m

```
function [PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, VA, BASE_KV, ZONE, VMAX, VMIN,NONL_LOAD,FILTER,XL, XC,LAM_P,LAM_Q,MU_VMAX,MU_VMIN ] = idx_busH
```

```
%
%   columns 1-13 must be included in input matrix (in case file)
%    1  BUS_I       bus number (positive integer)
%    2  BUS_TYPE    bus type (1 = PQ, 2 = PV, 3 = ref, 4 = isolated)
%    3  PD          Pd, real power demand (MW)
%    4  QD          Qd, reactive power demand (MVAr)
%    5  GS          Gs, shunt conductance (MW demanded at V = 1.0 p.u.)
%    6  BS          Bs, shunt susceptance (MVAr injected at V = 1.0 p.u.)
%    7  BUS_AREA    area number, (positive integer)
%    8  VM          Vm, voltage magnitude (p.u.)
%    9  VA          Va, voltage angle (degrees)
%    10 BASE_KV     baseKV, base voltage (kV)
%    11 ZONE        zone, loss zone (positive integer)
%    12 VMAX        maxVm, maximum voltage magnitude (p.u.)
%    13 VMIN        minVm, minimum voltage magnitude (p.u.)
%    14 NONL_LOAD   nonl_load HARMONIC SOURCES, 1 -  harmonic source is connected to the bus, 0 - not harmonic source is connected to the bus
%    15 FILTER
%    16 XL
%    17 XC

%   columns 14-17 are added to matrix after OPF solution
%   they are typically not present in the input matrix
%               (assume OPF objective function has units, u)
%    15 LAM_P       Lagrange multiplier on real power mismatch (u/MW)
%    16 LAM_Q       Lagrange multiplier on reactive power mismatch (u/MVAr)
%    17 MU_VMAX     Kuhn-Tucker multiplier on upper voltage limit (u/p.u.)
%    18 MU_VMIN     Kuhn-Tucker multiplier on lower voltage limit (u/p.u.)
%
%
%   additional constants, used to assign/compare values in the BUS_TYPE column
%    1  PQ    PQ bus
%    2  PV    PV bus
%    3  REF   reference bus
```

```
%    4  NONE  isolated bus
%
%   See also DEFINE_CONSTANTS.

%   MATPOWER
%   Copyright (c) 1996-2015 by Power System Engineering Research Center (PSERC)
%   by Ray Zimmerman, PSERC Cornell
%
%   $Id: idx_bus.m 2644 2015-03-11 19:34:22Z ray $
%
%   This file is part of MATPOWER.
%   Covered by the 3-clause BSD License (see LICENSE file for details).
%   See http://www.pserc.cornell.edu/matpower/ for more info.
```

```
%% define bus types
PQ      = 1;
PV      = 2;
REF     = 3;
NONE    = 4;
```

```
%% define the indices
BUS_I       = 1;    %% bus number (1 to 29997)
BUS_TYPE    = 2;    %% bus type (1 - PQ bus, 2 - PV bus, 3 - reference bus, 4 - isolated bus)
PD          = 3;    %% Pd, real power demand (MW)
QD          = 4;    %% Qd, reactive power demand (MVAr)
GS          = 5;    %% Gs, shunt conductance (MW at V = 1.0 p.u.)
BS          = 6;    %% Bs, shunt susceptance (MVAr at V = 1.0 p.u.)
BUS_AREA    = 7;    %% area number, 1-100
VM          = 8;    %% Vm, voltage magnitude (p.u.)
VA          = 9;    %% Va, voltage angle (degrees)
BASE_KV     = 10;   %% baseKV, base voltage (kV)
ZONE        = 11;   %% zone, loss zone (1-999)
VMAX        = 12;   %% maxVm, maximum voltage magnitude (p.u.)      (not in PTI format)
VMIN        = 13;   %% minVm, minimum voltage magnitude (p.u.)      (not in PTI format)
NONL_LOAD   = 14;%%HARMONIC SOURCES, 1  harmonic source is connected to the bus, 0 - not
harmonic source is connected to the bus
FILTER      = 15;
XL          = 16;
XC          = 17;
```

```
%% included in opf solution, not necessarily in input
```

```
%% assume objective function has units, u
```

```
LAM_P     = 18;   %% Lagrange multiplier on real power mismatch (u/MW)
LAM_Q     = 19;   %% Lagrange multiplier on reactive power mismatch (u/MVAr)
MU_VMAX   = 20;   %% Kuhn-Tucker multiplier on upper voltage limit (u/p.u.)
MU_VMIN   = 21;   %% Kuhn-Tucker multiplier on lower voltage limit (u/p.u.)
```

APÉNDICE IV

MakeYbusH_filter

```
function YbusH = makeYbusH_filter( baseMVA, bus, branch, gen, harm)
```

```
%MAKEYBUS   Builds the bus admittance matrix and branch admittance matrices.
%   [YBUS, YF, YT] = MAKEYBUS(MPC)
%   [YBUS, YF, YT] = MAKEYBUS(BASEMVA, BUS, BRANCH)
%
%   Returns the full bus admittance matrix (i.e. for all buses) and the
%   matrices YF and YT which, when multiplied by a complex voltage vector,
%   yield the vector currents injected into each line from the "from" and
%   "to" buses respectively of each line. Does appropriate conversions to p.u.
%   Inputs can be a MATPOWER case struct or individual BASEMVA, BUS and
%   BRANCH values. Bus numbers must be consecutive beginning at 1 (internal
%   ordering).
%
%   See also MAKEJAC, MAKESBUS, EXT2INT.

%   MATPOWER
%   Copyright (c) 1996-2015 by Power System Engineering Research Center (PSERC)
%   by Ray Zimmerman, PSERC Cornell
%
%   $Id: makeYbus.m 2644 2015-03-11 19:34:22Z ray $
%
%   This file is part of MATPOWER.
%   Covered by the 3-clause BSD License (see LICENSE file for details).
%   See http://www.pserc.cornell.edu/matpower/ for more info.
```

```
%%              Construci?n matriz YbusH
%%%% para construir la YBusH
if nargin < 3

    baseMVA = mpc.baseMVA;
    bus     = mpc.bus;
    branch  = mpc.branch;
    gen       =mpc.gen;    %%agregado para utilizar la matriz gen para las impedancias de los generadores y/o 
S/E
    hsdat     =mpc.HSData; %%agregado para utilizar la matriz HSData que contiene el espectro arm?nico de 
la carga no lineal
```

```
end
________________________________________________________________________________
%% constants
nb = size(bus, 1)    ;     %% number of buses
nl = size(branch, 1)  ;    %% number of lines
________________________________________________________________________________
%% define named indices into bus, branch matrices
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...
    VA, BASE_KV, ZONE, VMAX, VMIN,NONL_LOAD, FILTER, XL, XC, LAM_P, LAM_Q,
MU_VMAX, MU_VMIN] = idx_busH;
[F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, RATE_C, ...
    TAP, SHIFT, BR_STATUS, PF, QF, PT, QT, MU_SF, MU_ST, ...
    ANGMIN, ANGMAX, MU_ANGMIN, MU_ANGMAX] = idx_brch ;
[GEN_BUS, PG, QG,  QMAX, QMIN, VG, MBASE, GEN_STATUS, PMAX, PMIN, ...
    MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN, PC1, PC2, QC1MIN, QC1MAX, ...
    QC2MIN, QC2MAX, RAMP_AGC, RAMP_10, RAMP_30, RAMP_Q, APF,RG,XG] = idx_genH;

[HARM,PERCENT,ANGLE]=idx_HSData;

h = harm;
________________________________________________________________________________
%%
k=zeros(nb,1);
%Vfund=bus(:,VM).*bus(:,BASE_KV)/sqrt(3);  %% Tensiones de fase fundamentales en kV
Vfund=bus(:,VM);  %% Tensiones de fase fundamentales en pu


________________________________________________________________________________
%%  Construcci?n de matriz de admitancia Arm?nica


________________________________________________________________________________
%% modelado armonico de las ramas

stat = branch(:, BR_STATUS);                %% ramas en servicios
Yshh = stat ./ (branch(:, BR_R) + (1j* branch(:, BR_X)*h)); %% admitancia en serie de la matriz de las
ramas
Bch = stat .* branch(:, BR_B)*h;                      %% suceptancia de la rama
tap = ones(nl, 1)  ;                           %% por defecto tap ratio = 1
i = find(branch(:, TAP));                  %% posiciones de tap distintas de cero
tap(i) = branch(i, TAP);                   %% assign non-zero tap ratios
tap = tap .* exp(1j*pi/180 * branch(:, SHIFT)); %% tap con desfasaje
Ytth = Yshh + 1j*Bch/2;
Yffh = Ytth ./ (tap .* conj(tap));
```

```
Yfth = - Yshh ./ conj(tap);
Ytfh = - Yshh ./ tap;
```

```
%% construcci?n de la matriz de conexiones
f = branch(:, F_BUS) ;                    %% list of "from" buses
t = branch(:, T_BUS) ;                    %% list of "to" buses
Cf = sparse(1:nl, f, ones(nl, 1), nl, nb)  ;  %% connection matrix for line & from buses
Ct = sparse(1:nl, t, ones(nl, 1), nl, nb) ;   %% connection matrix for line & to buses

%% build Yf and Yt such that Yf * V is the vector of complex branch currents injected
%% at each branch's "from" bus, and Yt is the same for the "to" bus end
i = [1:nl; 1:nl]'    ;                    %% double set of row indices
Yfh = sparse(i, [f; t], [Yffh; Yfth], nl, nb);
Yth = sparse(i, [f; t], [Ytfh; Ytth], nl, nb);
Yramash = Cf' * Yfh + Ct' * Yth ;%%admitancias de las ramas
```

```
 %% modelado paralelo de las cargas pasivas en el dominio armonico

V_bus_fund= bus(:,VM);             % modulo  de tensiones a Frec. fundamental
p = find(bus(:, PD));              %% indice vector de MW carga
busp = bus(p, PD)/baseMVA ;        %% vector de potencia activa carga en p.u
q = find(bus(:, QD));              %% indice vector de MVAR carga
busq = bus(q, QD)/baseMVA  ;       %% vector de potencia reactiva carga en p.u
Rcarga=V_bus_fund(p).^2./busp;
Xcarga=V_bus_fund(q).^2./busq;

Yc= 1./Rcarga + 1./(1j*Xcarga*h);
Ych=sparse(p, q, Yc, nb, nb);
```

```
%%  Admitancias arm?nicas de las cargas
g1=find(bus(:,NONL_LOAD));  %%Se toma los n?meros de las barras en donde estan las cargas no lineal
Ych(g1,g1)=0;              %%Se asigna cero a esas cargas para luego ser modeladas como inyecciones de
corriente
Ycargah=Ych  ;             %%admitancias armonicas de las cargas
```

```
%% modelado arm?nico de los elementos filtro shunt
%Zc=(V^2/Potencia aparente)/(V^2/Sbase)=-jxc =Sbase/Selemento  --modeloarmonico---> XC=-jxc/h --
>Yshunt=jh/xc= (Selemento/Sbase)*h

V_bus=V_bus_fund.*bus(:, BASE_KV);
xL=bus(:, XL).*(baseMVA./(V_bus.^2)); % Valores en pu
xC=bus(:, XC).*(baseMVA./(V_bus.^2));
```

```
Yshu = -1j.*(1./((xL*h)-(1./(xC*h))));%% admitancia armonica de elementos shunt

Yshunth=sparse(1:nb, 1:nb, Yshu, nb, nb);

______________________________________________________________________________________
%% modelado arm?nico de los generadores

stat = gen(:, GEN_STATUS) ;
pk = find(gen(:,GEN_BUS));
Yg_h = stat ./ (gen(:, RG) + (1j* gen(:, XG)*h));
Yg_h=sparse(pk, pk, Yg_h, nb, nb);

YbusH = Yramash + Ycargah + Yshunth + Yg_h;%% Ybus para cada arm?nico

end
```

APÉNDICE V

Función runpfHfilter

```
function [MVAbase, bus, gen, branch, success, et, Vh] = ...
        runpfHfilter(casedata, mpopt)
________________________________________________________________________________
%RUNPF  Runs a Harmonic Power Flow.
________________________________________________________________________________
%%-----  initialize  -----
%% Definicion de los nombres de los indices de las matrices bus, gen, branch
 [PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, VA, BASE_KV,...
    ZONE, VMAX, VMIN, NONL_LOAD,LAM_P, FILTER, XL, XC, LAM_Q, MU_VMAX, MU_VMIN]
= idx_busH;

[F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, RATE_C, ...
    TAP, SHIFT, BR_STATUS, PF, QF, PT, QT, MU_SF, MU_ST, ...
    ANGMIN, ANGMAX, MU_ANGMIN, MU_ANGMAX] = idx_brch ;

[GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, GEN_STATUS, PMAX, PMIN, ...
    MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN, PC1, PC2, QC1MIN, QC1MAX, ...
    QC2MIN, QC2MAX, RAMP_AGC, RAMP_10, RAMP_30, RAMP_Q, APF,RG, XG] = idx_genH;

[HARM,PERCENT,ANGLE]=idx_HSData;
________________________________________________________________________________
%% Definicion de argumentos
if nargin < 4
    solvedcase = '' ;           %% don't save solved case
    if nargin < 3
        fname = '' ;            %% don't print results to a file
        if nargin < 2
            mpopt = mpoption;       %% use default options
            if nargin < 1
                casedata = 'caso13barras_v2'; %% default data file is 'case13paper.m'
            end
        end
    end
end
________________________________________________________________________________
%% leer datos
mpc = loadcase(casedata);
________________________________________________________________________________
```

```
%% Conversion interna
mpc = ext2int(mpc);
[baseMVA, bus, gen, branch,hsdat] = deal(mpc.baseMVA, mpc.bus, mpc.gen, mpc.branch, mpc.HSData);
________________________________________________________________________________________________
%% Tama?o de cosas
nb = size(bus, 1);      %% Numero de barras
nl = size(branch, 1);   %% Numero de ramas
ng = size(gen, 1);      %% Numero de generadores
NroHarm=size(hsdat,1) ;%% tama?o de la fila de la matriz de la data de la fuente arm?nica
________________________________________________________________________________________________
%% -----  Correr flujo de carga convencional -----
t0 = clock;
if mpopt.verbose > 0
   v = mpver('all');
   fprintf('\nMATPOWER Version %s, %s', v.Version, v.Date);
end
________________________________________________________________________________________________
                        %%  Formulacion
fprintf(' -- AC Harmonic Power Flow Calculation \n');

[MVAbase, bus, gen, branch, success, et] = runpf(casedata) ;



Harm_max=max( hsdat(:,HARM));%% m?ximo arm?nico hasta el cual se calcular? el espectro
if mpopt.verbose
      fprintf('\nHarmonics Power Flow through no iterative harmonic analysis method (NIHA)\n');
end
if mpopt.verbose
      fprintf('\nHarmonics voltages spectrum #%d \n',Harm_max);
end

success = 1;
mpc.et = etime(clock, t0);
mpc.success = success;
et=mpc.et;
fd=1;
________________________________________________________________________________________________
%% calculo del factor de potencia del sistema
________________________________________________________________________________________________
%% Determinaci?n de la corriente fundamental en la barra de carga no lineal

S=(bus(:, PD)+1j*bus(:, QD))/(baseMVA);%%vector de potencias complejas en la barras en pu
V1= bus(:,VM).*exp(1j*bus(:,VA)*pi/180); % Vector de tensiones de barras en formato complejo real + j
```

```
imaginaria
Io=conj(S)./conj(V1);                    %%vector de corriente a frecuencia fundamental en todas las barras de carga

If=bus(:,NONL_LOAD).*Io; %%vector de inyecci?n de corriente a fre. fundamental p.u en la barra de carga no lineal
```

```
%% lazo iterativo para determinar tensiones armonicas de las barras

Vh=zeros(nb,NroHarm);
Vh(:,1)=V1;     %% asigna el vector de tensiones fundamentales a la 1er columna de la matriz de tensiones armonicas
```

```
%% SEleccion de analisis armonico o analisis integrando el filtro de sintonia
for jk=2:NroHarm % contador de cantidad de armonicos
```

```
%%  Matriz de admitancia armonica

    k=2*jk-1; % orden del armonico 3,5,7....
    YbusH = makeYbusH_filter( baseMVA, bus, branch,gen,k);
    if det(YbusH)==0
        fprintf(fd, '\nWarning: Matrix is singular to working precision', et);
    else
        fprintf(fd, '\nConverged in %.2f seconds', et);
    end

    Mod_Ihar= abs(If)*hsdat(jk,PERCENT)/100; %modulo de la corriente armonica escalada

    Fase_Ihar= angle(If)+hsdat(jk,ANGLE)*pi/180*bus(:,NONL_LOAD); %fase de la corriente armonica escalada

    Ihar=Mod_Ihar.*exp(1j*Fase_Ihar);

    Vh(:,jk)=YbusH\Ihar;
```

```
end
```

```
%% Determinacion de la taza de distorsion de armonico THD

idx_paper=[1,4,5,2,3,6,7,8,9,11,12,13,10];
 abs(Vh(idx_paper,1)).*bus(idx_paper,BASE_KV)/sqrt(3)*1000 ;% fundamental
 abs(Vh(idx_paper,3)).*bus(idx_paper,BASE_KV)/sqrt(3)*1000 ;% 5 armonico
 abs(Vh(idx_paper,4)).*bus(idx_paper,BASE_KV)/sqrt(3)*1000 ;% 7 armonico
```

```
 THDv = sqrt(sum(abs(Vh(:,2:end)).^2,2))./abs(Vh(:,1))*100; % Tasa distorsion
 THDv(idx_paper);
 %% factor de potencia del sistema

%% funiones en el dominio del tiempo
% fs=60;
% Ts=1/fs;
% ws=2*pi*fs;
% Ns=1000;
%
% MVh=abs(Vh);
% PHVh=angle(Vh);
% t=linspace(0,Ts,Ns);
%
% k=1:2:37;
%
% for i=1:Ns
%     vb7(i)=sum(MVh(7,:).*cos(k*ws*t(i)+PHVh(7,:)));
%     vb5(i)=sum(MVh(5,:).*cos(k*ws*t(i)+PHVh(5,:)));
% end
%
% figure(1)
% plot(t,vb7,t,vb5)
% grid on
%
```

```
%% creaci?n de un vector externo de los indices de las barras
i2e = bus(:, BUS_I) ;
%
% Parametros
    g1=find(bus(:,NONL_LOAD));
%
%
    OUT_ALL         = mpopt.out.all;
    OUT_FORCE       = mpopt.out.force;
    SUPPRESS        = mpopt.out.suppress_detail;
%
if SUPPRESS == -1
    if size(bus, 1) > 500
        SUPPRESS = 1;
    else
        SUPPRESS = 0;
    end
```

```
 end
     OUT_SYS_SUM     = OUT_ALL == 1 || (OUT_ALL == -1 && mpopt.out.sys_sum);
     OUT_BUS         = OUT_ALL == 1 || (OUT_ALL == -1 && ~SUPPRESS && mpopt.out.bus);
    % areas
    xfmr = find(branch(:, TAP));                    %% indices of transformers
    nzld = find((bus(:, PD) | bus(:, QD)) & bus(:, BUS_TYPE) ~= NONE); %barras donde hay cargas lineales
    sorted_areas = sort(bus(:, BUS_AREA)); %?reas del sistema
    s_areas = sorted_areas([1; find(diff(sorted_areas))+1]);    %% numeros de ?reas
    nzsh = find((bus(:, GS) | bus(:, BS)) & bus(:, BUS_TYPE) ~= NONE); %barras donde hay elementos
shunt
    allg = find( ~isload(gen) ); %barras donde hay generaci?n
    ong  = find( gen(:, GEN_STATUS) > 0 & ~isload(gen) );
    onld = find( gen(:, GEN_STATUS) > 0 &  isload(gen) );
```

```
%%  Resumen del sistema
if OUT_SYS_SUM && (success || OUT_FORCE)
    fprintf(fd,
'\n================================================================================');
    fprintf(fd, '\n|     System Summary                                                           |');
    fprintf(fd,
'\n================================================================================');
    fprintf(fd, '\n\n                     How many?                          ');
    fprintf(fd, '\n------------------------------------------------------------------     ');
    fprintf(fd, '\nBuses                 %6d            ', nb);
    fprintf(fd, '\nGenerators            %6d         ', length(allg));
    fprintf(fd, '\nCommitted Gens         %5d     ', length(ong));
    fprintf(fd, '\nLoads                  %5d               ', length(nzld)+length(onld));
    fprintf(fd, '\n   Fixed               %5d          ', length(nzld)-sum(bus(:,NONL_LOAD)));
    fprintf(fd, '\n   Dispatchable        %5d      ', length(onld));
    fprintf(fd, '\n   Total NON_LOAD      %5d      ',sum(bus(:,NONL_LOAD)));
    fprintf(fd, '\nShunts                 %5d     ', length(nzsh) );
    fprintf(fd, '\nBranches               %5d     ', nl );
    fprintf(fd, '\nTransformers           %5d    ', length(xfmr) );
    fprintf(fd, '\nAreas                  %5d', length(s_areas));
    fprintf(fd, '\nNumber bus NON_LOAD    %5d       ', g1);
    fprintf(fd, '\nharmonic (spectrum)    %5d ', max( hsdat(:,HARM)));
    %fprintf(fd, '\n');
    fprintf(fd, '\nMaximum');
    fprintf(fd, '\n--------------------------------');
%
% nb = numero de barras
Vharm=zeros(nb,4);
for j=1:nb
```

```
    Vharm(j,1:4)=abs(Vh(j,1:4))*bus(j,BASE_KV)/sqrt(3)*1000;
end
 end
```

```
    %% data de la barras- Tensiones armonicas
 if OUT_BUS && (success || OUT_FORCE)



fprintf(fd,'\n================================================================================');
  fprintf(fd, '\n|                              Bus Data                                          |');
 fprintf(fd,'\n================================================================================\n');
  fprintf('   bus    ')
   for k=1:2:7
    fprintf(fd, ' V%d(L-N)  \t',k);
   end
  fprintf(fd,'   THDv[%%]\n')

  for j=1:nb
  fprintf(fd, '      %g\t',bus(j, BUS_I));
  fprintf(fd, '   %g   \t',Vharm(j,1:4));
  fprintf(fd, ' %g \t',THDv(j));
  fprintf(fd, ' \n');

 end

 end

end
```

APÉNDICE VI

Factorpotencia

A continuación se muestra el código de la función para calcular factor de potencia del sistema en el punto de común acople.

```
function fp=FactorPotencia(Pg,Qg)
________________________________________________________________________________
%% define named indices into bus, gen, branch matrices
% [PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...
%     VA, BASE_KV, ZONE, VMAX, VMIN,NONL_LOAD,FILTER,XL,
XC,LAM_P,LAM_Q,MU_VMAX,MU_VMIN ] = idx_busH;
%
% [GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, GEN_STATUS, PMAX, PMIN, ...
%     MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN, PC1, PC2, QC1MIN, QC1MAX, ...
%     QC2MIN, QC2MAX, RAMP_AGC, RAMP_10, RAMP_30, RAMP_Q, APF,RG,XG] = idx_genH;
________________________________________________________________________________
% % %% Definicion de argumentos
% % if nargin < 2
% %      mpopt = mpoption;       %% use default options
% %      if nargin < 1
% %         casedata = 'caso13barras'; %% default data file is 'case13paper.m'
% %      end
% % end
% %% leer datos
% mpc = loadcase(casedata);
% %% Conversion interna
% mpc = ext2int(mpc);
% [baseMVA, bus, gen] = deal(mpc.baseMVA, mpc.bus, mpc.gen);
________________________________________________________________________________
%% Calculo del factor de potencia en el PCC = Barra 1

 St= sqrt(Pg^2+Qg^2);
 Ps=Pg; % Potencia trifasica en MW
 fp=Ps/St ;%factor de potencia en el PCC
```

APÉNDICE VII

CorreccionFP

Acontinuacion se muetran los códigos para corregir FP en sus valores iniciales.

```
Function [Qc,Xc,Cf]=CorreccionFP (fp ,V ,Ps ,fpnew)
________________________________________________________________________________
    Qc= Ps *(tan(acos(fp))-tan(acos(fpnew))); % calculo del potencia reactiva compensada MVAr
    %Qc=roundn(Qc,-1);                  % aproximando el valor de potencia reactiva MVAr
    %[Qc]=comercial(Qc);
    Xc=V^2/Qc ;        %reactancia del capacitor ohm
    Cf=1/(2*pi*60*Xc);  %capacitancia en faradios
```

APÉNDICE VIII

ReacInductiva

A continuación función encargada de calcular le inductancia, reactancia inductiva y frecuencias del filtro

```
function [Xl,Lf,Lsys,fsys,fc]= ReacInductiva(indFilter,baseMVA, bus, branch,gen,fc_defaul)
________________________________________________________________________________
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...
   VA, BASE_KV, ZONE, VMAX, VMIN,NONL_LOAD,FILTER,XL,
XC,LAM_P,LAM_Q,MU_VMAX,MU_VMIN ] = idx_busH;
________________________________________________________________________________
f=60;
Xc=bus(indFilter,XC); % reactancia capacitica en ohm

Vbase=bus(indFilter,BASE_KV); % Voltaje base trifasico kV
Sbase=baseMVA; % potencia trifafica en MVA
Zbase= Vbase^2/Sbase; % Z base en ohm
________________________________________________________________________________
YbusH1 = makeYbusH_v3( baseMVA, bus, branch,gen,1); % matriz admintancia a frecuencia fundamental
en pu
        Zbus=inv(YbusH1);
        Zsys= Zbus(indFilter,indFilter);
        XLsys=imag(Zsys)*Zbase;
        Lsys=XLsys/(2*pi*f);

 C=1/(2*pi*60*Xc);
 %Lp=(1/((2*pi)^2*(4.7*f)^2*C))-Lsys;
 Lf=(1/((2*pi)^2*(fc_defaul*f)^2*C));
 Xl=Lf*(2*pi*f); % reactancia inductiva en ohm
fsys=1/(2*pi*sqrt((Lf+Lsys)*C));
fc=1/(2*pi*sqrt(Lf*C));
```

APÉNDICE IX

Valoresbase

A continuacon código de la función la cual realiza el calculo de valores máximos del sistema y valores rms y pico del filtro.

```
function [Inor, Vnor, Vnorp, Qnor, vbusfilter_pico,
Vcap,irms_filter,kVar_ent]=valoresbase(casedata,PhVhpu)
________________________________________________________________________________
%% Definicion de los nombres de los indices de las matrices bus, gen, branch

 [PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, VA, BASE_KV,...
    ZONE, VMAX, VMIN, NONL_LOAD, FILTER, XL, XC] = idx_busH;

[F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, RATE_C, ...
    TAP, SHIFT, BR_STATUS] = idx_brch ;

[GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, GEN_STATUS, PMAX, PMIN, ...
    MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN, PC1, PC2, QC1MIN, QC1MAX, ...
    QC2MIN, QC2MAX, RAMP_AGC, RAMP_10, RAMP_30, RAMP_Q, APF,RG, XG] = idx_genH;

[HARM,PERCENT,ANGLE]=idx_HSData;
________________________________________________________________________________
%% Definicion de argumentos
if nargin < 2
     mpopt = mpoption;       %% use default options
     if nargin < 1
        casedata = 'caso13barras'; %% default data file is 'case13paper.m'
     end
end
%% leer datos
mpc = loadcase(casedata);
%% Conversion interna
mpc = ext2int(mpc);
[baseMVA, bus, gen, branch,hsdat] = deal(mpc.baseMVA, mpc.bus, mpc.gen, mpc.branch, mpc.HSData);


________________________________________________________________________________
%% Tama?o de cosas
nb = size(bus, 1);      %% Numero de barras
nl = size(branch, 1);   %% Numero de ramas
ng = size(gen, 1);      %% Numero de generadores
NroHarm=size(hsdat,1) ;%% tama?o de la fila de la matriz de la data de la fuente arm?nica
________________________________________________________________________________
```

```
%%
indFilter = find(bus(:,FILTER)); %identifica la fila donde esta el filtro
VnomCap=bus(indFilter,BASE_KV);   % Voltaje nominal banco capacitivo = tension base de la barra

fs=60; %frecuencia electrica
ws=2*pi*fs;
Ts=1/fs; % periodo de la onda
Ns=1000; % numero de muestras
ts=linspace(0,Ts,Ns);
vbusfilter=zeros(Ns,1);
Hmax=2* NroHarm-1; % maximo armonico
k=1:2:Hmax;
```

```
% CALCULO DE TENSION DEL BANCO CAPACITIVO
MVh=abs(PhVhpu(indFilter,:))'; PHVh=angle(PhVhpu(indFilter,:))';

for i=1:Ns
    vbusfilter(i)=sqrt(2)*sum(MVh.*cos(k'.*ws*ts(i)+PHVh));
end

vbusfilter_pico=abs(max(vbusfilter))*bus(indFilter,BASE_KV)/sqrt(3);

figure(1)
plot(ts,vbusfilter)
grid on
grid on
 xlabel('t')
 ylabel('V (Kv)')

Vcap=sqrt(sum((abs(PhVhpu(indFilter,:)).*bus(indFilter,BASE_KV)).^2))/sqrt(3);% Tension rms en el
banco capacitivo
```

```
% CALCULO DE CORREINTE DEL BANCO CAPACITIVO

Ibus_filter= (bus(indFilter,BASE_KV)/sqrt(3)*(PhVhpu(indFilter,:))./(-1i*(bus(indFilter,XC))./k +
1i*(bus(indFilter,XL)*k))).';

MIh=abs(Ibus_filter); PHIh=angle(Ibus_filter);
ibusfilter=zeros(Ns,1);
for i=1:Ns
    ibusfilter(i)=sqrt(2)*sum(MIh.*cos(k'.*ws*ts(i)+PHIh));
end
```

```
figure(2)
plot(ts,ibusfilter)
grid on
 xlabel('t')
 ylabel('I (KA)')

irms_filter=sqrt(sum(MIh.^2));
```

```
%CALCULO DE LA POTENCIA REACTIVA QUE ENTREGA EL BANCO CAPACITIVO
kVar_ent=sqrt(3)*irms_filter*Vcap;
```

```
% LIMITES ACEPTABLES SEGUN LA NORMA

Qc=bus(indFilter,BS);
M=dlmread('MvarNominal.txt');
mayor=find(M>=Qc );
A=M(mayor);
Snorma = A(1);

 Inor=1.35*(Snorma/(VnomCap*sqrt(3)));% la corriente nominal en kA
 Vnor=1.1*(VnomCap/sqrt(3));% la tension en kV
 Vnorp=1.2*(VnomCap/sqrt(3))*sqrt(2);% la tension en kV
 Qnor=1.35*Snorma;%Potencia en Mvar
```

```
 %% creaci?n de un vector externo de los indices de las barras
e=0;b=0;d=0;a=0;
if irms_filter>Inor
    e=((irms_filter)/(Inor/1.35))*100;
    E='YES';
else
    e=((irms_filter)/(Inor/1.35))*100;
    E='NO';
end
if Vcap>Vnor
    b=((Vcap)/(Vnor/1.1))*100;
    B='YES';
else
    b=((Vcap)/(Vnor/1.1))*100;
    B='NO';
end
if vbusfilter_pico>Vnorp
    d=((vbusfilter_pico)/(Vnorp/1.2))*100;
```

```
   D='YES';
else
   d=((vbusfilter_pico)/(Vnorp/1.2))*100;
   D='NO';
end
if kVar_ent>Qnor
   a=((kVar_ent)/(Qnor/1.35))*100;
   A='YES';
else
    a=((kVar_ent)/(Qnor/1.35))*100;
   A='NO';
end
```

```
%%  Resumen de la verificacion de la norma IEEE 1036
 fprintf(
'\n============================================================================');
 fprintf( '\n|     Verification of the standard IEEE 1036                                          |');
 fprintf(
'\n============================================================================');
 fprintf('\n');
 fprintf(' Irms_max[kA]     Irms_filtro[kA]     Limit[%%]       Calculated[%%]    exceeds the limits \n');
 fprintf('\n');
 fprintf('   %8.4f        %8.4f           %4.0f           %8.2f          %s \n', Inor, irms_filter,135,e,E );
 fprintf('\n');
 fprintf(' Vrms_max[kV]     Vrms_filtro[kV]     Limit[%%]       Calculated[%%]    exceeds the limits\n');
 fprintf('\n');
 fprintf('   %8.4f        %8.4f           %4.0f           %8.2f          %s \n', Vnor,Vcap ,110, b,B );
 fprintf('\n');
 fprintf('  Vp_max[kV]       Vp_filtro[kV]      Limit[%%]       Calculated[%%]     exceeds the limits\n');
 fprintf('\n');
 fprintf('   %8.4f        %8.4f           %4.0f           %8.2f          %s \n', Vnorp,vbusfilter_pico ,120,d,D  );
 fprintf('\n');
 fprintf('  kVar_max[Mvar]    kVar_filtro[Mvar]   Limit[%%]       Calculated[%%]     exceeds the limits\n');
 fprintf('\n');
 fprintf('   %8.4f        %8.4f           %4.0f           %8.2f          %s \n', Qnor,kVar_ent ,135,a,A  );
 fprintf('\n');

end
```

APÉNDICE X

Función runfreqscan

Se presenta la función encargada de realizar barrido en frecuencia.

```
Function [MVAbase, bus, gen, branch, success, et] = ...
        runfreqscan(casedata, mpopt)
      clc; close all
________________________________________________________________________
%%-----  initialize  -----
%% Definicion de los nombres de los indices de las matrices bus, gen, branch
% 1   2   3   4     5      6         7   8   9   10  11        12  13  14
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, VA, BASE_KV,...
  ZONE, VMAX, VMIN, NONL_LOAD, FILTER, XL, XC] = idx_busH;

[F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, RATE_C, ...
  TAP, SHIFT, BR_STATUS] = idx_brch ;

[GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, GEN_STATUS, PMAX, PMIN, ...
  MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN, PC1, PC2, QC1MIN, QC1MAX, ...
  QC2MIN, QC2MAX, RAMP_AGC, RAMP_10, RAMP_30, RAMP_Q, APF,RG, XG] = idx_genH;

[HARM,PERCENT,ANGLE]=idx_HSData;
________________________________________________________________________
%% Definicion de argumentos
if nargin < 4
  solvedcase = '' ;             %% don't save solved case
  if nargin < 3
    fname = '' ;             %% don't print results to a file
    if nargin < 2
      mpopt = mpoption;      %% use default options
      if nargin < 1
        casedata = 'caso13barras_v9'; %% default data file is 'case13paper.m'
      end
    end
  end
end
________________________________________________________________________
%% leer datos
mpc = loadcase(casedata);
________________________________________________________________________
%% Conversion interna
mpc = ext2int(mpc);
[baseMVA, bus, gen, branch,hsdat] = deal(mpc.baseMVA, mpc.bus, mpc.gen, mpc.branch, mpc.HSData);
```

```
%% Tama?o de cosas
nb = size(bus, 1);      %% Numero de barras
nl = size(branch, 1);   %% Numero de ramas
ng = size(gen, 1);      %% Numero de generadores
NroHarm=size(hsdat,1);
```

```
%% Inicializacion bus de cargas no lineal

bus(:,NONL_LOAD)=zeros(nb,1);
```

```
%% Bus donde esta carga no lineal o fuente de estimulo
%bus_rh = input('\n Indique la barra de estimulo (bus referencia arm?nica) = ');
bus_rh = find(bus(:,FILTER));
bus(bus_rh,NONL_LOAD)=1;
```

```
%% Fuente de estimulo normalizada
%HSnorm=[hsdat(:,1),ones(NroHarm,1),zeros(NroHarm,1)]; % fuente armonica normalizada
%% Fuente de corriente normalizada (usualmente donde esta carga no lineal)

Ih_norm=bus(:,NONL_LOAD); %vector de inyecci?n de corriente normalizada en 1 pu
```

```
%% lazo iterativo para determinar tensiones armonicas (Impedancias) de las barras

Vh=zeros(nb,NroHarm);
Harm_max=2*NroHarm-1;
Npto=1000; % cantidad de puntos en el dominio xHarm
xHarm=linspace(1,Harm_max,Npto); % rango del dominio xHarm
```

```
for jk=1:Npto % contador de cantidad de puntos
   t0 = clock;
%%  Matriz de admitancia armonica
   YbusH = makeYbusH_filter( baseMVA, bus, branch,gen,xHarm(jk));

   Vh(:,jk)=YbusH\Ih_norm;
End
```

```
%% Determinar impedancia transferencia

%bus_tr = input('\n Indique la barra transferencia (bus respuesta arm?nica) = ');
bus_tr =find(bus(:,FILTER));
Vbnorm=abs(Vh(bus_tr,:)); % bus 1
```

```
figure
plot(xHarm,Vbnorm);
grid on
xlabel('harm')
ylabel('Z (pu)')

end
```

APÉNDICE XI

Función RunpfFilter

Acontinuacon función encargada de realizar el deseño de un filtro LC serie.

```
function [MVAbase, bus, gen, branch, success, et] = ...
            runfilter (casedata, mpopt)
________________________________________________________________________________
        clc; close all; clear all;
________________________________________________________________________________


%%-----  initialize  -----
%% define named indices into bus, gen, branch matrices
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...
    VA, BASE_KV, ZONE, VMAX, VMIN,NONL_LOAD,FILTER,XL,
XC,LAM_P,LAM_Q,MU_VMAX,MU_VMIN ] = idx_busH;

[GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, GEN_STATUS, PMAX, PMIN, ...
    MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN, PC1, PC2, QC1MIN, QC1MAX, ...
    QC2MIN, QC2MAX, RAMP_AGC, RAMP_10, RAMP_30, RAMP_Q, APF,RG,XG] = idx_genH;

[F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, RATE_C, ...
    TAP, SHIFT, BR_STATUS, PF, QF, PT, QT, MU_SF, MU_ST, ...
    ANGMIN, ANGMAX, MU_ANGMIN, MU_ANGMAX] = idx_brch;

[HARM,PERCENT,ANGLE]=idx_HSData;


________________________________________________________________________________
%% Definicion de argumentos
if nargin < 2
     mpopt = mpoption;       %% use default options
     if nargin < 1
        casedata = 'caso13barras'; %% default data file is 'case13paper.m'
     end
end
________________________________________________________________________________
%% leer datos
mpc = loadcase(casedata);
________________________________________________________________________________
%% Conversion interna
mpc = ext2int(mpc);
```

```
[baseMVA, bus, gen, branch,hsdat] = deal(mpc.baseMVA, mpc.bus, mpc.gen, mpc.branch, mpc.HSData);
```
____

```
%% Ubicacion del filtro en el sistema
indFilter = find(bus(:,FILTER)); %identifica la fila donde esta el filtro
```
____

```
%% flujo de carga convencional para obtener el fp en el PCC
[MVAbase, bus, gen, branch, success, et] = runpf(casedata);   %% calculo del factor de potencia inicial sin compensacion
fp_inicial=FactorPotencia(gen(1,PG),gen(1,QG)); % Factor de potencia y potencia inicial en el PCC

num=1;[num,fp_inicial,0,0]
```
____

```
%% Valores por definicion del factor de potencia y THD
fp_defaul=0.95;% Factor de potencial al cual se decea corregir el sistema
THD_defaul=5;% total de distorsion armonica de voltaje
Vbus_filter = bus(indFilter,VM)*bus(indFilter,BASE_KV);  % voltaje trifasico en kV donde se conectara el filtro
Ps=gen(1,PG);
fs=60;
fc_defaul=4.7;
if fp_inicial < fp_defaul % Si se requiere compensacion reactiva entra el lazo
```
____

```
%% calculo del banco capacitivo inicial
num=2;fname=['caso13barras_v',int2str(num)];
[Qc,Xc,Cf]=CorreccionFP(fp_inicial ,Vbus_filter ,Ps ,fp_defaul); % Calculo del Banco pacitivo...
% Qc potencia MVAr trifasicos, Xc = ohm
bus(indFilter,BS)= Qc; % Almacena la potencia reactiva MVAr
bus(indFilter,XC)= Xc; % Almacena la reactancia capacitiva del banco
savecasefilter(fname, baseMVA, bus, gen, branch,hsdat);
[MVAbase, bus, gen, branch] = runpf(fname) ;%calculo del Flujo de carga incluye condensador
fp_correjido = FactorPotencia(gen(1,PG),gen(1,QG));   % Nuevo factor de potencia solo C filtro

[num,fp_inicial,fp_correjido,0]
[MVAbase, bus, gen, branch, success, et, PhVhpu] = runpfH_v3(fname);
THDvo = sqrt(sum(abs(PhVhpu(indFilter,2:end)).^2,2))./abs(PhVhpu(indFilter,1))*100; % THD barra filtro

aux5=1;% Variable de control para determinar si se cumple limtes de potencia reactiva
while aux5==1;
```
____

```
aux4=1;% Variable de control para determinar si se cumple limtes de tension
while aux4==1;
```
____

```
aux3=1;% Variable de control para determinar si se cumple limtes de corrientes
```

```
while aux3==1;
________________________________________________________________________________
aux2=1;  % Variable de control para determinar si se cumpple thd voltaje
while aux2 ==1;
________________________________________________________________________________
var=0; %variable auxiliar para romper los ciclos en caso de problemas de resonancia
aux1=1; % Variable de control para determinar el reactor del filtro
Qcf=bus(indFilter,BS);
num=3;fname=['caso13barras_v',int2str(num)];
________________________________________________________________________________
while aux1==1 % Determinacion de la L filtro y verificacion de factor de potencia
________________________________________________________________________________
%% Calculo de Reactancia inductiva
   [XLf,Lf,Lsys,fsys,fc]= ReacInductiva(indFilter,baseMVA, bus, branch,gen,fc_defaul);%modulo para el
calculo de la reactanica inductiva del filtro
%-------------------------------------------------------------------------------------
   Vbus_filter=bus(indFilter,VM)*bus(indFilter,BASE_KV); % voltaje trifasico barra del filtro
   VLf=XLf/(XLf-Xc)*Vbus_filter; % voltaje trifasico sobre el inductor filtro
   QLf= VLf^2/XLf; %Potencia que consume la Lf del filtro
%-------------------------------------------------------------------------------------
   bus(indFilter,BS)=Qcf -QLf;
   bus(indFilter,XL)= XLf ; % Almacena inductancia filtro
   %bus(:,VM)= 1.0 ; bus(:,VA)= 0 ; gen(1,QG)=0;
   savecasefilter(fname, baseMVA, bus, gen, branch,hsdat);
   [MVAbase, bus, gen, branch] = runpf(fname);
   fp_filtro = FactorPotencia(gen(1,PG),gen(1,QG));  % Nuevo factor de potencia incluyendo L filtro
%-------------------------------------------------------------------------------------------


  if fp_filtro <= fp_defaul
     num=num+1;fname=['caso13barras_v',int2str(num)];
     Qcf=Qcf*(1+0.01);
     bus(indFilter,BS)=Qcf-QLf;
     Vnomf=bus(indFilter,BASE_KV); % voltaje trifasico barra del filtro
     Xcnew=Vnomf^2/Qcf;
     bus(indFilter,XC)= Xcnew;
     %bus(:,VM)= 1.0 ; bus(:,VA)= 0 ; gen(1,QG)=0;
     savecasefilter(fname, baseMVA, bus, gen, branch,hsdat);
     [MVAbase, bus, gen, branch] = runpf(fname);
     fp_filtro = FactorPotencia(gen(1,PG),gen(1,QG));  % Nuevo factor de potencia incluyendo L filtro
     [num,fp_inicial,fp_correjido,fp_filtro];
     num=num+1;fname=['caso13barras_v',int2str(num)];
  else
       aux1=0;
```

```
  end
end % fin while aux 1
```

```
%% Distorsion armonica

[MVAbase, bus, gen, branch, success, et, PhVhpu] = runpfHfilter(fname);
%THDv0 = sqrt(sum(abs(Vh(:,2:end)).^2,2))./abs(Vh(:,1))*100; % Tasa distorsion
THDv = sqrt(sum(abs(PhVhpu(indFilter,2:end)).^2,2))./abs(PhVhpu(indFilter,1))*100; % THD barra filtro



if THDv>THD_defaul  % limite IEEE 519_2014 = 8%
    runfreqscan(fname) %barrido en frecuencia del sistema
    decicion = input('\n El problema es por la frecuencia de dise~o [Y/N]) = ','s');
  if decicion == 'N'
      num=num+1;fname=['caso13barras_v',int2str(num)];
      Qcf=Qcf*(1+0.01)  ;
      bus(indFilter,BS)=Qcf-QLf;
      Vnomf=bus(indFilter,BASE_KV); % voltaje trifasico barra del filtro
      Xcnew=Vnomf^2/Qcf;
      bus(indFilter,XC)= Xcnew;
      %bus(:,VM)= 1.0 ; bus(:,VA)= 0 ; gen(1,QG)=0;
      savecasefilter(fname, baseMVA, bus, gen, branch,hsdat);
      [MVAbase, bus, gen, branch] = runpf(fname);
      fp_filtro = FactorPotencia(gen(1,PG),gen(1,QG));  % Nuevo factor de potencia incluyendo L filtro
      [num,fp_inicial,fp_correjido,fp_filtro]
      num=num+1;fname=['caso13barras_v',int2str(num)];
  elseif decicion == 'Y'
      fprintf ('\n Se debe realizar una reconsideracion del dise~o para el FILTRO DE SINTONIA SIMPLE,
ajustando a una nueva frecuencia ')
      var=1;

    if var==1;
    break
    end

  end
else
aux2=0;

end
end % fin while aux2

if var==1
```

```
    break
end
_____________________________________________________________________________________________
_
[Inor, Vnor, Vnorp, Qnor, vbusfilter_pico,
Vcap,irms_filter,kVar_ent,vbusfilter]=valoresbase(fname,PhVhpu);
if irms_filter>Inor
     num=num+1;fname=['caso13barras_v',int2str(num)];
     Qcf=Qcf*(1+0.01);
     bus(indFilter,BS)=Qcf-QLf;
     Vnomf=bus(indFilter,BASE_KV); % voltaje trifasico barra del filtro
     Xcnew=Vnomf^2/Qcf;
     bus(indFilter,XC)= Xcnew;
     %bus(:,VM)= 1.0 ; bus(:,VA)= 0 ; gen(1,QG)=0;
     savecasefilter(fname, baseMVA, bus, gen, branch,hsdat);
     [MVAbase, bus, gen, branch] = runpf(fname);
     fp_filtro = FactorPotencia(gen(1,PG),gen(1,QG));  % Nuevo factor de potencia incluyendo L filtro
     [num,fp_inicial,fp_correjido,fp_filtro]
     num=num+1;fname=['caso13barras_v',int2str(num)];
else
aux3=0;
end
end% fin while aux3

if var==1
    break
end
if Vcap>Vnor
    fprintf ('\n El ajuste a realiza en el Banco capaitivo es en tension, ')
    aux4=0;
else
   aux4=0;
end

end% fin while aux4
if var==1
    break
end
if kVar_ent>Qnor
    num=num+1;fname=['caso13barras_v',int2str(num)];
     Qcf=Qcf*(1+0.01)  ;
     bus(indFilter,BS)=Qcf-QLf;
     Vnomf=bus(indFilter,BASE_KV); % voltaje trifasico barra del filtro
```

```
    Xcnew=Vnomf^2/Qcf;
    bus(indFilter,XC)= Xcnew;
    %bus(:,VM)= 1.0 ; bus(:,VA)= 0 ; gen(1,QG)=0;
    savecasefilter(fname, baseMVA, bus, gen, branch,hsdat);
    [MVAbase, bus, gen, branch] = runpf(fname);
    fp_filtro = FactorPotencia(gen(1,PG),gen(1,QG));  % Nuevo factor de potencia incluyendo L filtro
    [num,fp_inicial,fp_correjido,fp_filtro]
    num=num+1;fname=['caso13barras_v',int2str(num)];

else
  aux5=0;
end

end% fin while aux5
if var==1
   quit cancel
else
```

```
%% Resumen  del Calcula del filtro

 fprintf('\n=====================================================================');
 fprintf( '\n|      Filter Summary                                          |');
fprintf('\n=====================================================================');
 fprintf('\n');
 fprintf(' Location_bus   FP_corrected      THDv[%%]     Qc[Mvar]       XLf[ohm]       XCf[ohm]\n');
 fprintf('\n');
 fprintf(' %4.0f           %8.4f        %8.4f     %8.4f       %8.4f      %8.4f
\n',indFilter,fp_filtro,THDv,bus(indFilter,BS),bus(indFilter,XL),bus(indFilter,XC));
 fprintf('\n');
 fprintf('\n');
end
else
   fprintf ('No se contempla el estudio del Dise?o para el FILTRO MINIMO de SINTONIA ')
 end


end
```

APÉNDICE XII

Fichero fuente el cual describe el circuito en PSpice

```
EstudioPentracionArmonica
*
.LIB PQ.LIB
.LIB CFL.LIB


.TRAN   0.002m  3004.167m    2987.5m   0.02m  SKIPBP
.OPTIONS ABSTOL=0.001A CHGTOL=0.001C ITL5=0 RELTOL=0.001 VNTOL=0.05
.PARAM   FREQ=60  TPER={1/FREQ} ws={2*PI*FREQ}
* Valores Bases
.PARAM   Vb={13800} Sb={10e6} Zb={Vb*Vb/Sb} Ibase={Sb/(SQRT(3)*Vb)}
* fuente de tension barra referencia
.PARAM     UEFA={Vb/SQRT(3)} FASE_A={-0.5*PI/180}


* Modelo de la carga no lineal
* Barra 7
.PARAM   P7= 1150K Q7= 290K V7pu=0.980 faseV7deg={-4.72+0} faseV7={faseV7deg*PI/180}
.PARAM   P7pu={P7/Sb} Q7pu={Q7/Sb} S7pu={SQRT(P7pu*P7pu+Q7pu*Q7pu)}
.PARAM   I7pu={S7pu/V7pu} I7={I7pu*Ibase} faseI7rad={-ATAN(Q7pu/P7pu)+faseV7}
faseI7={faseI7rad*180/PI}
* Barra 3
.PARAM   P3= 600K Q3= 530K V3pu=0.995 faseV3deg=-3.53 faseV3={faseV3deg*PI/180}
.PARAM   P3pu={P3/Sb} Q3pu={Q3/Sb} S3pu={SQRT(P3pu*P3pu+Q3pu*Q3pu)}
.PARAM   I3pu={S3pu/V3pu} I3={I3pu*Ibase} faseI3rad={-ATAN(Q3pu/P3pu)+faseV3}
faseI3={faseI3rad*180/PI}
* Barra 10
.PARAM   P10= 810K Q10= 800K V10pu=0.981 faseV10deg=-4.16 faseV10={faseV10deg*PI/180}
.PARAM   P10pu={P10/Sb} Q10pu={Q10/Sb} S10pu={SQRT(P10pu*P10pu+Q10pu*Q10pu)}
.PARAM   I10pu={S10pu/V10pu} I10={I10pu*Ibase} faseI10rad={-ATAN(Q10pu/P10pu)+faseV10}
faseI10={faseI10rad*180/PI}
* Barra 12
.PARAM   P12= 370K Q12= 330K V12pu=0.979 faseV12deg=-3.08 faseV12={faseV12deg*PI/180}
```

```
.PARAM   P12pu={P12/Sb} Q12pu={Q12/Sb} S12pu={SQRT(P12pu*P12pu+Q12pu*Q12pu)}

.PARAM   I12pu={S12pu/V12pu} I12={I12pu*Ibase} faseI12rad={-ATAN(Q12pu/P12pu)+faseV12}
faseI12={faseI12rad*180/PI}

*

* **************** Sistema externo *********************

* nodos 100-1

.PARAM   R101pu=4.5e-4   X101pu=0.009989  R101={R101pu*Zb} X101={X101pu*Zb}
L101={X101/ws}

*

*  *************** Impedancias de ramas ******************

* nodos 1-5

.PARAM          R15pu=0.0045199     X15pu=0.0562       R15={R15pu*Zb}
         X15={X15pu*Zb}             L15={X15/ws}

* nodos 5-6

.PARAM          R56pu=0.00075   X56pu=0.00063  R56={R56pu*Zb}
         X56={X56pu*Zb}             L56={X56/ws}

* nodos 6-7

.PARAM   R67pu=0.059184      X67pu=0.35510      R67={R67pu*Zb}
         X67={X67pu*Zb}             L67={X67/ws}

* nodos 6-8

.PARAM   R68pu=0.0432        X68pu=0.3451       R68={R68pu*Zb}
         X68={X68pu*Zb}             L68={X68/ws}

* nodos 5-10

.PARAM   R510pu=0.0599       X510pu=0.3801      R510={R510pu*Zb}    X510={X510pu*Zb}
         L510={X510/ws}

* nodos 5-11

.PARAM   R511pu=0.00109      X511pu=0.00091     R511={R511pu*Zb}    X511={X511pu*Zb}
         L511={X511/ws}

* nodos 11-12

.PARAM   R1112pu=0.0558      X1112pu=0.3624     R1112={R1112pu*Zb}
         X1112={X1112pu*Zb}         L1112={X1112/ws}

* nodos 11-13

.PARAM   R1113pu=0.01218 X1113pu=0.1462         R1113={R1113pu*Zb}
         X1113={X1113pu*Zb}         L1113={X1113/ws}

* nodos 3-5

.PARAM   R35pu=0.0651        X35pu=0.3803       R35={R35pu*Zb}
         X35={X35pu*Zb}             L35={X35/ws}

*
```

```
* ********* Impedancias de cargas ************************

* nodo 8-0

.PARAM    R80pu=6.836   X80pu=7.93   R80={R80pu*Zb}   X80={X80pu*Zb}   L80={X80/ws}

* nodo 3-0

.PARAM    R30pu=15.688  X30pu=17.76  R30={R30pu*Zb}   X30={X30pu*Zb}   L30={X30/ws}

* nodo 13-0

.PARAM    R130pu=3.23   X130pu=3.62  R130={R130pu*Zb} X130={X130pu*Zb} L130={X130/ws}

* nodo 12-0

.PARAM    R120pu=25.934 X120pu=29.08 R120={R120pu*Zb} X120={X120pu*Zb} L120={X120/ws}

* nodo 10-0

.PARAM    R100pu=11.316 X100pu=11.46 R100={R100pu*Zb} X100={X100pu*Zb} L100={X100/ws}

* nodo 7-0

.PARAM    R70pu=8.3513  X70pu=33.12  R70={R70pu*Zb}   X70={X70pu*Zb}   L70={X70/ws}

* nodo 5-0

.PARAM    R50pu=4.4162  X50pu=4.95  R50={R50pu*Zb}   X50={X50pu*Zb}   L50={X50/ws}



* Valor incial del condensador

.PARAM   Cfi=9.2385e-5


* Valor filtro serie LC

.PARAM   Cf=85.224e-6  Lf=3.737e-3


* sistema externo

EA                100       0         VALUE
          {1.0094*SQRT(2)*UEFA*SIN(ws*TIME+FASE_A)}

R1                100       101       {R101}

L1                101       1         {L101}

* equivalente bus1 a bus 5

Vdc1      1   102 DC                  0

R2                102       103       {R15}

L2                103       5         {L15}

* equivalente bus 5- bus 6
```

```
R3          5      51     {R56}

L3          51     6      {L56}

* equivalente bus 6- bus 7

R4          6      61     {R67}

L4          61     7      {L67}

* equivalente bus 6- bus 8

R5          6      62     {R68}

L5          62     8      {L68}

* equivalente bus 5- 10

R7          5      52     {R510}

L7          52     10     {L510}

* equivalente bus 5- bus 11

R11         5      53     {R511}

L11         53     11     {L511}

* equivalente bus 11- bus 12

R12         11     111    {R1112}

L12         111    12     {L1112}

* equivalente bus 11- bus 13

R14         11     112    {R1113}

L14         112    13     {L1113}

* equivalente bus 3 - 5

R9          5      54     {R35}

L9          54     3      {L35}


* Carga lineal bus 3

Vdc3    3   31     DC            0

*Rc3        31     0      {R30}

*Lc3        31     0      {L30}

* Carga no linael bus 3

I3_1        31     0      SIN(0 {SQRT(2)*I3} {FREQ} 0 0 {faseI3})

I3_5        31     0      SIN(0 {0.1824*SQRT(2)*I3} {5*FREQ}  0 0  {faseI3-
55.68})

I3_7        31     0      SIN(0 {0.1190*SQRT(2)*I3} {7*FREQ}  0 0  {faseI3-
```

```
84.11})

I3_11            31      0      SIN(0 {0.0573*SQRT(2)*I3} {11*FREQ} 0 0 {faseI3-
143.56})

I3_13            31      0      SIN(0 {0.0401*SQRT(2)*I3} {13*FREQ} 0 0 {faseI3-
175.58})

I3_17            31      0      SIN(0 {0.0193*SQRT(2)*I3} {17*FREQ} 0 0
{faseI3+111.39})

I3_19            31      0      SIN(0 {0.0139*SQRT(2)*I3} {19*FREQ} 0 0
{faseI3+68.30})

I3_23            31      0      SIN(0 {0.0094*SQRT(2)*I3} {23*FREQ} 0 0 {faseI3-
24.61})

I3_25            31      0      SIN(0 {0.0086*SQRT(2)*I3} {25*FREQ} 0 0 {faseI3-
67.64})

I3_29            31      0      SIN(0 {0.0071*SQRT(2)*I3} {29*FREQ} 0 0 {faseI3-
145.46})

I3_31            31      0      SIN(0 {0.0062*SQRT(2)*I3} {31*FREQ} 0 0
{faseI3+176.83})

I3_35            31      0      SIN(0 {0.0044*SQRT(2)*I3} {35*FREQ} 0 0
{faseI3+97.40})

I3_37            31      0      SIN(0 {0.0038*SQRT(2)*I3} {37*FREQ} 0 0
{faseI3+54.36})


* equivalente bus 5- 0

Vdc5      5   55      DC             0

Rc5               55      0      {R50}

Lc5               55      0      {L50}


* carga lineal Bus 7

Vdc7      7   71      DC             0

*Rc7              71      0      {R70}

*Lc7              71      0      {L70}

* carga no lineal Bus 7

I7_1              71      0      SIN(0 {SQRT(2)*I7} {FREQ} 0 0 {faseI7})

I7_5              71      0      SIN(0 {0.1824*SQRT(2)*I7} {5*FREQ}  0 0  {faseI7-
55.68})

I7_7              71      0      SIN(0 {0.1190*SQRT(2)*I7} {7*FREQ}  0 0  {faseI7-
84.11})

I7_11             71      0      SIN(0 {0.0573*SQRT(2)*I7} {11*FREQ} 0 0 {faseI7-
143.56})
```

```
I7_13     71    0     SIN(0 {0.0401*SQRT(2)*I7} {13*FREQ} 0 0 {faseI7-175.58})

I7_17     71    0     SIN(0 {0.0193*SQRT(2)*I7} {17*FREQ} 0 0 {faseI7+111.39})

I7_19     71    0     SIN(0 {0.0139*SQRT(2)*I7} {19*FREQ} 0 0 {faseI7+68.30})

I7_23     71    0     SIN(0 {0.0094*SQRT(2)*I7} {23*FREQ} 0 0 {faseI7-24.61})

I7_25     71    0     SIN(0 {0.0086*SQRT(2)*I7} {25*FREQ} 0 0 {faseI7-67.64})

I7_29     71    0     SIN(0 {0.0071*SQRT(2)*I7} {29*FREQ} 0 0 {faseI7-145.46})

I7_31     71    0     SIN(0 {0.0062*SQRT(2)*I7} {31*FREQ} 0 0 {faseI7+176.83})

I7_35     71    0     SIN(0 {0.0044*SQRT(2)*I7} {35*FREQ} 0 0 {faseI7+97.40})

I7_37     71    0     SIN(0 {0.0038*SQRT(2)*I7} {37*FREQ} 0 0 {faseI7+54.36})



* carga lineal bus 8

Vdc8   8  81   DC         0

Rc8         81    0       {R80}

Lc8         81    0       {L80}



* Carga lineal bus 10

Vdc10  10  131  DC        0

*Rc10   131   0    {R100}

*Lc10   131   0    {L100}

* Carga no linael bus 10

I10_1     131   0     SIN(0 {SQRT(2)*I10} {FREQ} 0 0 {faseI10})

I10_5     131   0     SIN(0 {0.1824*SQRT(2)*I10} {5*FREQ}  0 0  {faseI10-55.68})

I10_7     131   0     SIN(0 {0.1190*SQRT(2)*I10} {7*FREQ}  0 0  {faseI10-84.11})

I10_11    131   0     SIN(0 {0.0573*SQRT(2)*I10} {11*FREQ} 0 0 {faseI10-143.56})

I10_13    131   0     SIN(0 {0.0401*SQRT(2)*I10} {13*FREQ} 0 0 {faseI10-175.58})

I10_17    131   0     SIN(0 {0.0193*SQRT(2)*I10} {17*FREQ} 0 0
```

```
{faseI10+111.39})

I10_19            131     0       SIN(0 {0.0139*SQRT(2)*I10} {19*FREQ} 0 0
{faseI10+68.30})

I10_23            131     0       SIN(0 {0.0094*SQRT(2)*I10} {23*FREQ} 0 0 {faseI10-
24.61})

I10_25            131     0       SIN(0 {0.0086*SQRT(2)*I10} {25*FREQ} 0 0 {faseI10-
67.64})

I10_29            131     0       SIN(0 {0.0071*SQRT(2)*I10} {29*FREQ} 0 0 {faseI10-
145.46})

I10_31            131     0       SIN(0 {0.0062*SQRT(2)*I10} {31*FREQ} 0 0
{faseI10+176.83})

I10_35            131     0       SIN(0 {0.0044*SQRT(2)*I10} {35*FREQ} 0 0
{faseI10+97.40})

I10_37            131     0       SIN(0 {0.0038*SQRT(2)*I10} {37*FREQ} 0 0
{faseI10+54.36})



* Carga lineal bus 12

Vdc12   12   121      DC              0

*Rc12      121        0       {R120}

*Lc12      121        0       {L120}

* Carga no linael bus 12

I12_1             121     0       SIN(0 {SQRT(2)*I12} {FREQ} 0 0 {faseI12})

I12_5             121     0       SIN(0 {0.1824*SQRT(2)*I12} {5*FREQ}  0 0  {faseI12-
55.68})

I12_7             121     0       SIN(0 {0.1190*SQRT(2)*I12} {7*FREQ}  0 0  {faseI12-
84.11})

I12_11            121     0       SIN(0 {0.0573*SQRT(2)*I12} {11*FREQ} 0 0 {faseI12-
143.56})

I12_13            121     0       SIN(0 {0.0401*SQRT(2)*I12} {13*FREQ} 0 0 {faseI12-
175.58})

I12_17            121     0       SIN(0 {0.0193*SQRT(2)*I12} {17*FREQ} 0 0
{faseI12+111.39})

I12_19            121     0       SIN(0 {0.0139*SQRT(2)*I12} {19*FREQ} 0 0
{faseI12+68.30})

I12_23            121     0       SIN(0 {0.0094*SQRT(2)*I12} {23*FREQ} 0 0 {faseI12-
24.61})

I12_25            121     0       SIN(0 {0.0086*SQRT(2)*I12} {25*FREQ} 0 0 {faseI12-
67.64})

I12_29            121     0       SIN(0 {0.0071*SQRT(2)*I12} {29*FREQ} 0 0 {faseI12-
145.46})

I12_31            121     0       SIN(0 {0.0062*SQRT(2)*I12} {31*FREQ} 0 0
```

```
{faseI12+176.83})

I12_35          121     0       SIN(0 {0.0044*SQRT(2)*I12} {35*FREQ} 0 0
{faseI12+97.40})

I12_37          121     0       SIN(0 {0.0038*SQRT(2)*I12} {37*FREQ} 0 0
{faseI12+54.36})


* carga lineal bus 13

Vdc13   13  135     DC              0

Rc13    135     0       {R130}

Lc13    135     0       {L130}



* Circuito compensador inicial

*Vdcf   5  58       DC              0

*Cfi                58 0    {Cfi}


* Circuito filtro

Vdcf    5  58       DC              0

Lf                  58      59      {Lf}

Cf      59 0        {Cf}

*.STEP PARAM xcc 0.0005 10 2

* Fourier de las corrientes (para diagramas de contorno). Las dos ultimas dan

* el valor eficaz (1/sqrt(2)) de las intensidades normalizadas que consumen los

* rectificadores. A la fase hay que restarle pi/2.

* Fourier de las tensiones para comprobar caidas de tension ok (mirando componente fund.)

*.FOUR 60 15 V(5,0)

.PROBE

.PRINT TRAN V(5)

*.PROBE RMS(I(Vdc))

*.PROBE I(LCCB)  I(LtB) I(LmB)

*.PROBE  I(Vsh)

.END
```

APÉNDICE XIII

Ficheros de salida en PSpice para el caso 1

Se presentan a continuación el fichero de salida de la simulación en PSpice caso de prueba sin compensación**.**Componentes de Fourier para construir el diagrama de barras:

DC COMPONENT = -1.7421E+00

HARMONIC NO	FREQUENCY (HZ)	FOURIER COMPONENT	NORMALIZED COMPONENT	PHASE (DEG)	NORMALIZED PHASE (DEG)
1	6.0000E+01	1.0727E+04	1.0000E+00	-3.9274E+00	0.0000E+00
2	1.2000E+02	1.9337E+00	1.8026E-04	5.1895E+00	1.3044E+01
3	1.8000E+02	8.0756E-01	7.5281E-05	3.9460E+00	1.5728E+01
4	2.4000E+02	8.2125E-01	7.6558E-05	-5.0584E+00	1.0651E+01
5	3.0000E+02	2.2313E+02	2.0801E-02	1.6379E+02	1.8343E+02
6	3.6000E+02	3.6436E-01	3.3965E-05	-2.0071E+01	3.4932E+00
7	4.2000E+02	1.9647E+02	1.8315E-02	1.3166E+02	1.5915E+02
8	4.8000E+02	1.2197E-01	1.1370E-05	9.0155E+01	1.2157E+02
9	5.4000E+02	1.1781E-01	1.0982E-05	-1.2540E+01	2.2806E+01
10	6.0000E+02	1.2526E-01	1.1677E-05	-1.0660E+02	-6.7330E+01
11	6.6000E+02	1.3682E+02	1.2754E-02	6.6593E+01	1.0979E+02
12	7.2000E+02	2.2312E-01	2.0799E-05	9.7181E+01	1.4431E+02
13	7.8000E+02	1.0875E+02	1.0137E-02	3.2538E+01	8.3594E+01
14	8.4000E+02	4.0597E-01	3.7845E-05	5.0529E+01	1.0551E+02
15	9.0000E+02	3.0294E-01	2.8240E-05	5.1979E+01	1.1089E+02

TOTAL HARMONIC DISTORTION = 3.2149E+00 PERCENT

APÉNDICE XIV

Ficheros de salida en PSpice para el caso 2

Se presentan a continuación el fichero de salida de la simulación en PSpice caso de prueba incluyendo el banco capacitivo.Componentes de Fourier para construir el diagrama de barras:

FOURIER COMPONENTS OF TRANSIENT RESPONSE V(5,0)

DC COMPONENT = -2.4950E+00

HARMONIC NO	FREQUENCY (HZ)	FOURIER COMPONENT	NORMALIZED COMPONENT	PHASE (DEG)	NORMALIZED PHASE (DEG)
1	6.0000E+01	1.1195E+04	1.0000E+00	8.5811E+01	0.0000E+00
2	1.2000E+02	5.8302E-01	5.2077E-05	1.3931E+01	-1.5769E+02
3	1.8000E+02	2.1915E-01	1.9575E-05	4.0299E+01	-2.1713E+02
4	2.4000E+02	3.4203E-01	3.0551E-05	3.9038E+01	-3.0420E+02
5	3.0000E+02	1.0895E+03	9.7319E-02	1.6805E+02	-2.6101E+02
6	3.6000E+02	4.2637E-01	3.8084E-05	1.1783E+02	-3.9703E+02
7	4.2000E+02	2.0291E+02	1.8125E-02	-1.0778E+02	-7.0845E+02
8	4.8000E+02	1.1480E-01	1.0255E-05	1.7652E+02	-5.0997E+02
9	5.4000E+02	9.4176E-02	8.4121E-06	1.6903E+02	-6.0326E+02
10	6.0000E+02	1.0351E-01	9.2460E-06	1.7566E+02	-6.8245E+02
11	6.6000E+02	4.0404E+01	3.6090E-03	-1.7760E+02	-1.1215E+03
12	7.2000E+02	5.2148E-02	4.6580E-06	-1.3432E+02	-1.1640E+03
13	7.8000E+02	2.2253E+01	1.9877E-03	-3.1512E+01	-1.1471E+03
14	8.4000E+02	7.5128E-02	6.7107E-06	-1.6783E+02	-1.3692E+03
15	9.0000E+02	4.5852E-02	4.0956E-06	-1.3569E+02	-1.4229E+03

TOTAL HARMONIC DISTORTION = 9.9078E+00 PERCENT

APÉNDICE XV

Ficheros de salida en PSpice para el caso 3

Se presentan a continuación el fichero de salida de la simulación en PSpice caso de prueba incluyendo el filtro LC serie. Componentes de Fourier para construir el diagrama de barras:

FOURIER COMPONENTS OF TRANSIENT RESPONSE V(5,0)

DC COMPONENT = -1.6974E+00

HARMONIC NO	FREQUENCY (HZ)	FOURIER COMPONENT	NORMALIZED COMPONENT	PHASE (DEG)	NORMALIZED PHASE (DEG)
1	6.0000E+01	1.1180E+04	1.0000E+00	-4.2015E+00	0.0000E+00
2	1.2000E+02	1.9915E+00	1.7814E-04	6.4617E+00	1.4865E+01
3	1.8000E+02	8.0930E-01	7.2389E-05	7.9975E+00	2.0602E+01
4	2.4000E+02	7.2305E-01	6.4674E-05	2.9826E+00	1.9789E+01
5	3.0000E+02	2.7491E+01	2.4590E-03	1.7385E+02	1.9486E+02
6	3.6000E+02	4.6813E-01	4.1873E-05	-1.1448E+00	2.4064E+01
7	4.2000E+02	8.1056E+01	7.2502E-03	1.4035E+02	1.6976E+02
8	4.8000E+02	1.7689E-01	1.5823E-05	2.2072E+01	5.5684E+01
9	5.4000E+02	2.2645E-01	2.0256E-05	-7.1839E+00	3.0630E+01
10	6.0000E+02	1.1941E-01	1.0681E-05	-2.8345E+01	1.3670E+01
11	6.6000E+02	7.3596E+01	6.5829E-03	7.5811E+01	1.2203E+02
12	7.2000E+02	1.4720E-01	1.3167E-05	6.3472E+01	1.1389E+02
13	7.8000E+02	6.1531E+01	5.5038E-03	4.2020E+01	9.6640E+01
14	8.4000E+02	2.9818E-01	2.6672E-05	4.8839E+01	1.0766E+02
15	9.0000E+02	2.4607E-01	2.2010E-05	4.4933E+01	1.0796E+02

TOTAL HARMONIC DISTORTION = 1.1501E+00 PERCENT

APÉNDICE XVI

Rutina integrada a runfilter para realizar las graficas de comparación con PSpice

```
%% importacion del Fourier de tension caso 1 del ?Pspice

filename = 'FourierVoltBus5Caso1.txt';
delimiterIn = ' ';
headerlinesIn = 5;
A = importdata(filename,delimiterIn,headerlinesIn);
MIpsp=A.data(1:2:end,3)/sqrt(3); % Valor rms de las amplitudes de corriente, armonicos impares

Vrms1=norm(MIpsp);
MVh=abs(PhVhpu1(indFilter,:));
MVh_kV=((abs(PhVhpu1(indFilter,:)).*bus(indFilter,BASE_KV))./sqrt(3))';% Tension rms en el banco
capacitivo

r1=(1:2:15)';% rango armonico
  figure(60)
 bar(r1,[MVh_kV(1:8,1),MIpsp*1e-3])
  legend('VFilter_-_m_a_t_l_a_b','V_f_i_l_t_e_r_-_P_s_p_i_c_e',1);
 xlabel('armonico (h)')
 ylabel('Amplitud (KV)')
xlim([0 16])
ylim([0 10])
grid on
______________________________________________________________________________
%% importacion del Fourier de tension caso 2 del ?Pspice

filename = 'FourierVoltBus5Caso2.txt';
delimiterIn = ' ';
headerlinesIn = 8;
A = importdata(filename,delimiterIn,headerlinesIn);
MIpsp=A.data(1:2:end,3)/sqrt(3); % Valor rms de las amplitudes de corriente, armonicos impares

Vrms2=norm(MIpsp);
MVh=abs(PhVhpu2(indFilter,:));
MVh_kV=((abs(PhVhpu2(indFilter,:)).*bus(indFilter,BASE_KV))./sqrt(3))';% Tension rms en el banco
capacitivo

r1=(1:2:15)';% rango armonico
```

```
 figure(62)
 bar(r1,[MVh_kV(1:8,1),MIpsp*1e-3])
  legend('VFilter_-_m_a_t_l_a_b','V_f_i_l_t_e_r_-_P_s_p_i_c_e',1);
 xlabel('armonico (h)')
 ylabel('Amplitud (KV)')
xlim([2 16])
ylim([0 0.8])
grid on
```

```
%% importacion del Fourier de tension caso 3 del ?Pspice

filename = 'FourierVoltBus5Caso3.txt';
delimiterIn = ' ';
headerlinesIn = 8;
A = importdata(filename,delimiterIn,headerlinesIn);
MIpsp=A.data(1:2:end,3)/sqrt(3); % Valor rms de las amplitudes de corriente, armonicos impares

Vrms3=norm(MIpsp);
MVh=abs(PhVhpu3(indFilter,:));
MVh_kV=((abs(PhVhpu3(indFilter,:)).*bus(indFilter,BASE_KV))./sqrt(3))';% Tension rms en el banco capacitivo

r1=(1:2:15)';% rango armonico
  figure(63)
 bar(r1,[MVh_kV(1:8,1),MIpsp*1e-3])
 %title(['Vrms_p_f_H = ',num2str(Vcap), 'kV','   Vfilter_P_s_p_i_c_e = ',num2str(Vrms3*1e-3), 'kV'])
 legend('VFilter_-_m_a_t_l_a_b','V_f_i_l_t_e_r_-_P_s_p_i_c_e',1);
 xlabel('armonico (h)')
 ylabel('Amplitud (kV)')
xlim([0 16])
ylim([0 10])
grid on
```