
```

%control en modo corriente por picos
% Declaracion de requerimientos
Vin_min = 130;      % tensiones de entrada Vrms
Vin_max = 265;
Vin_nom = 220;

fline_nom = 50;     %frecuencia de la linea Hz
fline_min = 50;

Vout_nom = 65;
Iout_nom = 1.5;     %corriente de salida nominal A
Vripple = 2;        % tension de ripple en V
fsw = 100000;       %frecuencia de switch HZ
rend = 0.85;        %rendimiento
Rout = Vout_nom/Iout_nom;
%fin requerimientos

%parametros dispositivos utilizados
Vds_max = 600;      %del MOSFET
Vf = 0.7;           %caida en el diodo en directa
Vbulk_min = 120; % minima tension a la que se escarga el Cbulk
Resr = 0.5;         %Resistencia serie equivalente Cout
Vzener = 10;        %tension nominal del zener que alimenta al TL431
CTR = 1;
%fin parametros dispositivos utilizados

%Bulk capacitor and Minimum Bulk voltage
Pout = Vout_nom * Iout_nom
Pin = Pout / rend

Num = 2 * Pin * (asin(Vbulk_min / (sqrt(2)*Vin_min))/pi + 0.25);
Den = (2*Vin_min*Vin_min - Vbulk_min * Vbulk_min) * fline_min;
fprintf(1,'Valor minimo de capacitor de entrada\n');
Cin = Num/Den
Cin = 180e-6;
fprintf(1,'Se elije %1.2f\n', Cin)
%fin calculo Cbulk

% relacion de vueltas transformador
Vbulk_max = sqrt(2) * Vin_max;
Vreflected = 0.8*(Vds_max - 1.3*Vbulk_max);
Nps = Vreflected / Vout_nom % se elije el proximo valor menor
%falta calculo de devanado auxiliar

%tension en el diodo
Vdiode_max = Vbulk_max / Nps + Vout_nom
%maximo Duty cicle
Num = Nps *(Vout_nom + Vf);
Den = Vbulk_min + Nps*(Vout_nom + Vf);
Dmax = Num / Den

```

```

%9.2.2.3
%inductancia del primario
Num = Vbulk_min^2*((Nps*Vout_nom)/(Vbulk_min+Nps*Vout_nom))^2;
Den = 0.2*Pin*fsw;
Lp = Num / Den
%calculo Ipk mosfet
Sum1 = (Pin*(Vbulk_min + Nps*Vout_nom)) / (Vbulk_min*Nps*Vout_nom);
N = (Nps*Vout_nom)/(Vbulk_min+(Nps*Vout_nom));
Sum2 = (0.5*Vbulk_min/Lp)*(N/fsw);
clear N;
Ipk_mosfet = Sum1 + Sum2;
%calculo Irms MOSFET
Irms_mosfet = sqrt(((Dmax^3/3)*(Vbulk_min/
(Lp*fsw))^2)-((Dmax*Dmax*Ipk_mosfet*Vbulk_min)/
(Lp*fsw)))+(Dmax*Ipk_mosfet*Ipk_mosfet))

Ipk_diode = Nps * Ipk_mosfet
Iavg_diode = Iout_nom

%9.2.2.4 Output capacitor
Num = Iout_nom*Nps*Vout_nom;
Den = (0.001)*Vout_nom*fsw*(Vbulk_min+Nps*Vout_nom); %el ripple se
    divide en 100 para pasarlo a por ciento
Cout = Num / Den %se toma el mayor estandar cercano
Cout = 100e-6

%9.2.2.5 red sensora de corriente
Rcs = 1 / Ipk_mosfet %ver compensacion para disminuir la perdida en
    esta R
fprintf(1,'Ver si se consigue este valor de Rcs si no hay que
    cambiarlo por uno cercano\n');
%9.2.2.6
Rg = 10
%9.2.2.7

%9.2.2.8
Cct=1000e-12
Rrt = 8200
fsw=1.72*.5/(Cct*Rrt)

%9.2.2.9
%9.2.2.10.1 POWER STAGE POLOS AND ZEROS
%determinarlos polos y ceros de la etapa de potencia
%primero se determina si esta en CCM comparando Lpcritica y Lp para el
%rango de tension de entrada
Lpcrit1 = ((Rout*Nps*Nps)/(2*fsw))*((Vin_min/
(Vin_min*Vout_nom*Nps))^2)
Lpcrit2 = ((Rout*Nps*Nps)/(2*fsw))*((Vin_max/
(Vin_max*Vout_nom*Nps))^2);
if(Lpcrit1 < Lp && Lpcrit2 < Lp)
    fprintf(1,'Funciona en CCM, de lo contrario los calculos
        siguientes estan mal\n');

```

```

end;
%Ganancia a lazo abierto
Acs = 3; %del datasheet (ganancia sensor corriente)
t1 = (2*Lp*fsw)/(Rout*Nps*Nps);
M = Vout_nom * Nps / Vbulk_min;
Go = (Rout*Nps)/(Rcs*Acs)*(1/(((1-Dmax)^2/t1)+(2*M)+1));
Go = 20*log10(Go);
%calculo cero ESR y Cout
wesr_z = 1/(Resr * Cout);
fesr_z = wesr_z / (2*pi)
%Calculo cero en semiplano derecho, se calcula para el peor caso, es
decir
%cuando a f mas baja, pasa para maxima carga y minima Vbulk
wrhp_z = (Rout*(1-Dmax)^2*Nps*Nps)/(Lp*Dmax);
frhp_z = wrhp_z / (2*pi)

%POLO DOMINANTE Wp1
wp1 = (((1 - Dmax)^3/t1) + 1 + Dmax) / (Rout * Cout);
fp1 = wp1 / (2*pi)
%POLO DOBLE Wp2
fp2 = fsw / 2
wp2 = fsw * pi;

%9.2.2.10.2 COMPENZACION DE PENDIENTE
Mc = (1/pi+0.5)/(1 - Dmax);
Sn = (Vbulk_min*Rcs)/Lp;
Se = (Mc - 1)* Sn;

ton_min = Dmax / fsw;
Sosc = 1.7 / ton_min;
Ccsf = 100e-12
Cramp = 10e-9
Rramp = 22000
Rcsf = Rramp / (Sosc/Se - 1)
Rcsf = 4200

%Ganancia a lazo abierto
%en hertz
Gof = (Go * fp1)/( frhp_z * fesr_z);
Num_ol = Gof * conv([1 fesr_z],[-1 frhp_z]);
Den_ol = conv([1 fp1],[1/(fp2*fp2) 1/fp2 1]);
bode(Num_ol,Den_ol)

%Calculos realimentacion
fbw = frhp_z / 4;
%del bode vemos frecuencia y fase al lazo abierto para el ancho de
banda fbw
Ifb_ref = 869.3e-6;
%calculos de Rfbu y Rfbb
Rfbu = (65 - 2.495) / Ifb_ref
Rfbb = (2.495 * Rfbu) / (90 - 2.495)

```

```

fprintf(1,'se pone como Rfbb = 1.8k y Rfbu=62k por tolerancia, para el
    diseño final hay que usar los otros\n');
%para dar buen margen de fase se compensa el TL con un cero ubicado a
    1/10
%del BW
fcomp_z = fbw / 10;
wcomp_z = 2 * pi * fcomp_z;
Ccomp_z = 0.018e-6 %se elije por defecto
Rcomp_z = 1 / (wcomp_z * Ccomp_z)
Rcomp_z = 47000

%polarizacion del TL431 necesita 10mA, los que se proveen con el zener
Rtlbias = (Vout_nom - Vzener) / 10e-3

%se agrega un polo a frecuencia del fesr_z o frhp_z, el que sea menor
if (fesr_z < frhp_z)
    fcmp_p = fesr_z
else
    fcmp_p = frhp_z
end

Ccmp_p = 15e-9
Rcmp_p = 1 / (2 * pi * fcmp_p * Ccmp_p)
Rcmp_p = 3300
%con Rfbg se añade ganancia de DC p/ obtener el BW deseado
Rfbg = Rcmp_p / 3.3 % establece ganancia = 12

%funcion de transferencia del TL431
Num_tl = [(Rcomp_z*Ccomp_z*2*pi) 1];
Den_tl = [(Ccomp_z*Rfbu) 0];

%funcion de transferencia del compensador
Num_cmp = [1];
Den_cmp = [(1/fcmp_p) 1];

%funcion transferencia opto
Rled = 8200
Ropto = 8200
Num_opto = CTR * Ropto / Rled;

Num_total = conv(Num_ol,Num_tl);
Num_total = Rcmp_p * Num_opto * conv(Num_total,Num_cmp);
Den_total = conv(Den_ol,Den_tl);
Den_total = Rfbg * conv(Den_total,Den_cmp)*2*pi;

figure
bode(Num_total,Den_total)

Pout =

```

97.5000

$P_{in} =$

114.7059

Valor minimo de capacitor de entrada

$C_{in} =$

1.1266e-04

Se elije 0.00

$N_{ps} =$

1.3883

$V_{diode_max} =$

334.9367

$D_{max} =$

0.4319

$L_p =$

0.0012

$I_{rms_mosfet} =$

1.4650

$I_{pk_diode} =$

3.4010

$I_{avg_diode} =$

1.5000

$C_{out} =$

9.9053e-05

Cout =

1.0000e-04

Rcs =

0.4082

Ver si se consigue este valor de Rcs si no hay que cambiarlo por uno cercano

Rg =

10

Cct =

1.0000e-09

Rrt =

8200

fsw =

1.0488e+05

Lpcrit1 =

4.8897e-08

Funciona en CCM, de lo contrario los calculos siguientes estan mal

fesr_z =

3.1831e+03

frhp_z =

8.5918e+03

fp1 =

54.9086

$f_{p2} =$

$5.2439e+04$

$C_{csf} =$

$1.0000e-10$

$C_{ramp} =$

$1.0000e-08$

$R_{ramp} =$

22000

$R_{csf} =$

$1.0409e+03$

$R_{csf} =$

4200

$R_{fbu} =$

$7.1903e+04$

$R_{fbb} =$

$2.0501e+03$

se pone como $R_{fbb} = 1.8k$ y $R_{fbu}=62k$ por tolerancia, para el diseño final hay que usar los otros

$C_{comp_z} =$

$1.8000e-08$

$R_{comp_z} =$

$4.1164e+04$

$R_{comp_z} =$

47000

Rtlbias =

5500

fcmp_p =

3.1831e+03

Ccmp_p =

1.5000e-08

Rcmp_p =

3.3333e+03

Rcmp_p =

3300

Rfbg =

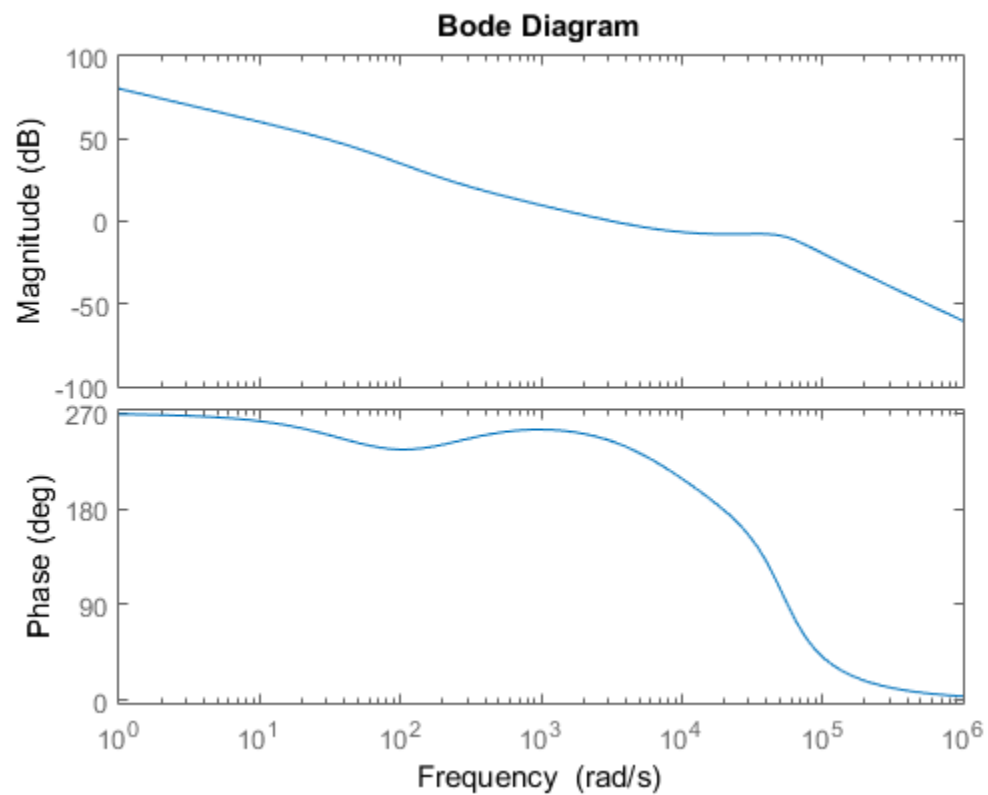
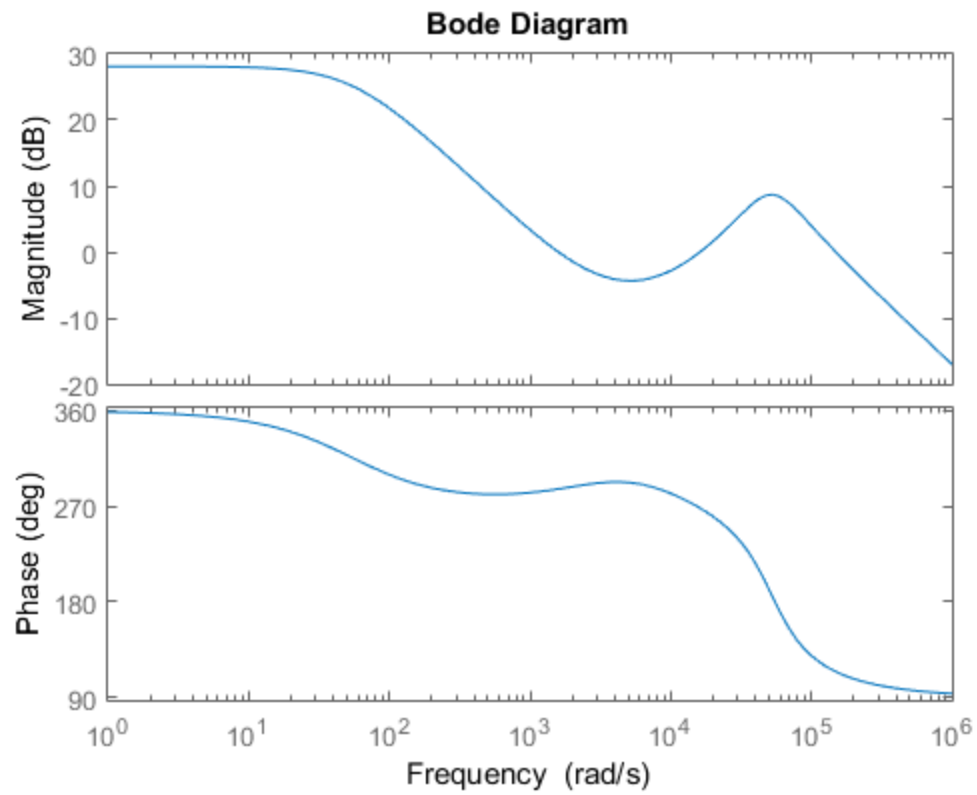
1000

Rled =

8200

Rocto =

8200



Published with MATLAB® R2015a