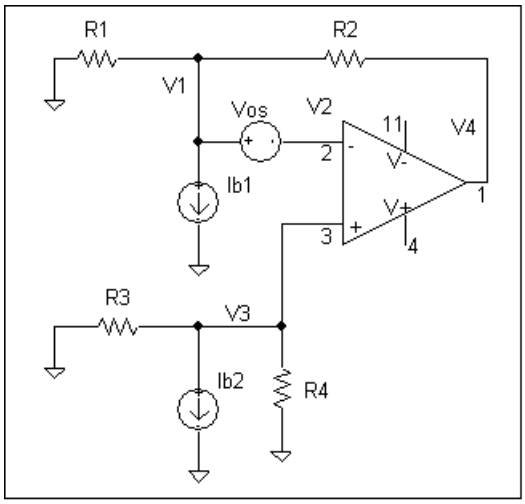
11/9/2006

MCA of Opamp DC Offsets)

Opamp offsets can be analyzed from the following schematic (also shown on p.70 of the book)



In the following analysis, the values of Ib, Ios, and Vos are changed to Ib = 100nA, Ios = 20nA, & Vos = 5mV.

This histograms for these values are: (See M-file offsets.m below)

Fig 24. Opamp offsets MCA, Normal dist

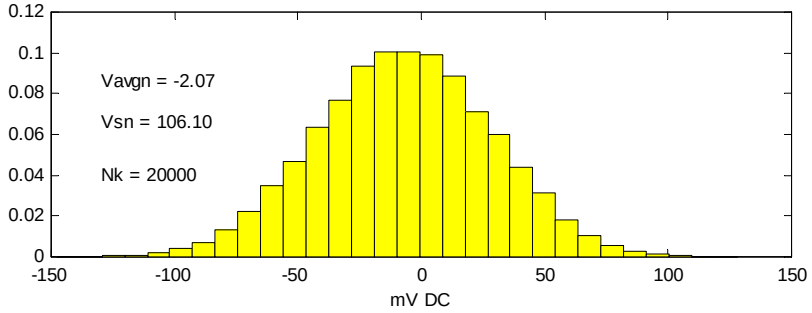
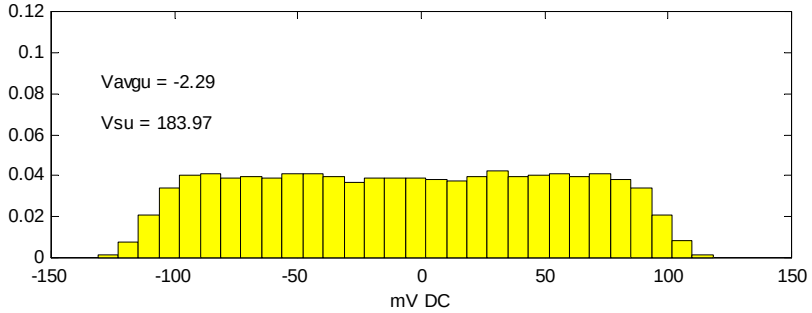


Fig 25. Opamp offsets MCA, Uniform dist



```
% dc opamp offsets
% File:  c:\M_files\bookupdate\offsets.m
% uses MATLAB function G6.m
% updated 11/09/06
clear;clc;tic;
mV=1e-3;nA=1e-9;K=1e3;
R1=10*K;R2=200*K;R3=10*K;R4=200*K;
% From LM156 data sheet:
Vos=5*mV;Ib=100*nA;Ios=20*nA;
Ib1=Ib-Ios/2;Ib2=Ib+Ios/2;
%
% Under the broad assumption that the average value of Vos = 0
% and the average bias currents are half of maximum:
%
Nav=[R1 R2 R3 R4 Vos Ib1 Ib2];
Nom=[R1 R2 R3 R4 0 Ib1/2 Ib2/2];
%
Vo=G6(Nom);
%
Nk=20000; %<<<<<<<<<<<<<<<<<<< Nk
%
Nc=size(Nom,2);randn('state',sum(100*clock));
Tn=zeros(Nc,Nk);Tu=Tn;
Tr=0.02;
% Assuming Vos varies from -5 to +5 mV, and Ib1 & Ib2 are not negative
T=[ -Tr -Tr -Tr -Tr -Vos 0 0;Tr Tr Tr Tr Vos Ib1 Ib2];
%
nb=30; % number of histogram bins
for k=1:Nk
   for w=1:4 % Multiplying resistor tolerances
      Tn(w,k)=Nav(w)*(((T(2,w)-T(1,w))/6)*(randn+3)+T(1,w)+1);
      Tu(w,k)=Nav(w)*((T(2,w)-T(1,w))*rand+T(1,w)+1);
   end % end w loop
   for w=5:7 % Adding tolerance units to Vos, Ib1, & Ib2
      Tn(w,k)=((T(2,w)-T(1,w))/6)*(randn+3)+T(1,w);
      Tu(w,k)=(T(2,w)-T(1,w))*rand+T(1,w);
   end % end w loop
      Vn(k)=G6(Tn(:,k));Vu(k)=G6(Tu(:,k));
end % end k loop
%
% Get Normal histogram
%
Vs1=3*std(Vn);Vavg1=mean(Vn);
h1=hist(Vn,nb)/Nk;VL=min(Vn);VH=max(Vn);
intv=(VH-VL)/nb;
q=1:nb;bin1(q)=VL+intv*(q-1); % Vectorize
Vhi1=Vavg1+Vs1;Vlo1=Vavg1-Vs1;
Vsn=sprintf('%2.2f\n',Vs1);
Vavgn=sprintf('%2.2f\n',Vavg1);
%
% Get Uniform histogram
%
Vs2=3*std(Vu);Vavg2=mean(Vu);
h2=hist(Vu,nb)/Nk;VL=min(Vu);VH=max(Vu);
intv=(VH-VL)/nb;
q=1:nb;bin2(q)=VL+intv*(q-1);
Vhi2=Vavg2+Vs2;Vlo2=Vavg2-Vs2;
Vsu=sprintf('%2.2f\n',Vs2);
Vavgu=sprintf('%2.2f\n',Vavg2);
```

```
%
subplot(2,1,1)
bar(bin1,h1,1,'y');
set(gca,'FontSize',8);
s1='Fig 24. Opamp offsets MCA, Normal dist';
title(s1);
xlabel('mV DC');
grid off;
axis([-150 150 0 0.12]);
text(-130,0.08,['Vavgn = ',Vavgn],'FontSize',8);
text(-130,0.06,['Vsn = ',Vsn],'FontSize',8);
text(-130,0.04,['Nk = ',num2str(Nk)],'FontSize',8);
%
subplot(2,1,2)
bar(bin2,h2,1,'y');
set(gca,'FontSize',8);
s2='Fig 25. Opamp offsets MCA, Uniform dist';
title(s2);
xlabel('mV DC')
grid off;
axis([-150 150 0 0.12]);
text(-130,0.08,['Vavgu = ',Vavgu],'FontSize',8);
text(-130,0.06,['Vsu = ',Vsu],'FontSize',8);
figure(1)
ET=toc



function y = G6(X)
% fcn for offsets.m offset analysis
%G6(R1,R2,R3,R4,Vos,Ib1,Ib2)
R1=X(1);R2=X(2);R3=X(3);R4=X(4);Vos=X(5);Ib1=X(6);Ib2=X(7);
Vo=(Vos+Ib1*R1*R2/(R1+R2)-Ib2*R3*R4/(R3+R4))*(1+R2/R1);
y=Vo*1e3; % output in mV
%y=Vo;
```