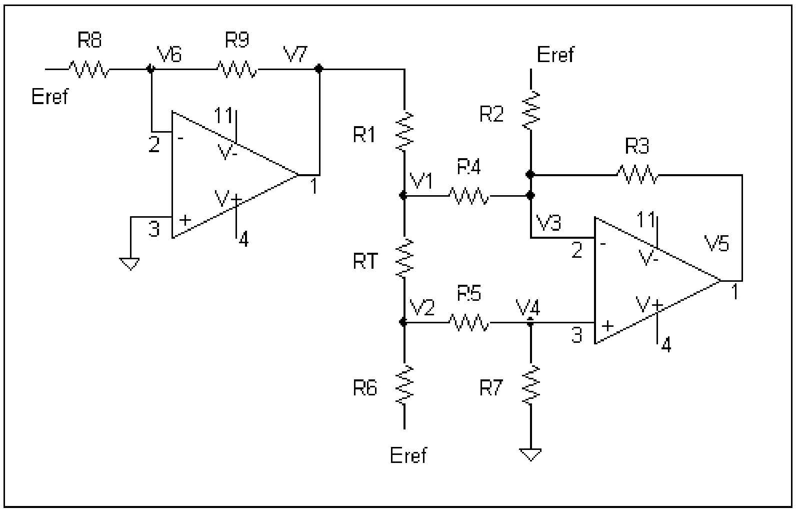
11/14/2006

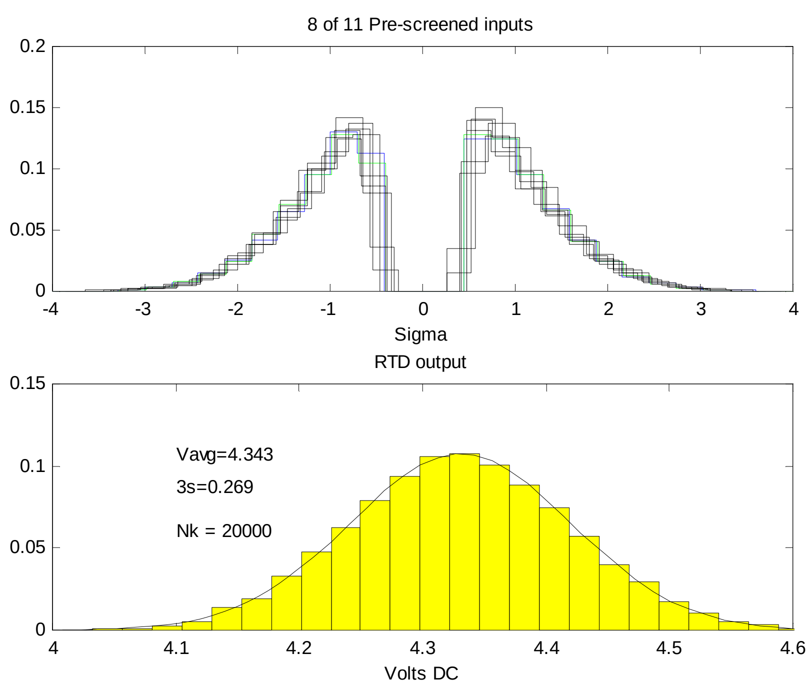
Bimodal (Gapped) input.

Quite often vendors will pre-screen components and remove the tighter tolerances. When this happens it is of some value to know what affect these bimodal input distributions will have on the ouput. The M-File rtdbimod.m will tell us that for the DC case.

The following schematic supersedes that shown on page 20 of the book.



The rather suprising output with all 11 inputs gapped is shown to be approximately Normal with no sign of a gap. However, the user is encouraged to change the tolerance of resistor R4 to 10%, and note the drastic difference in the output. Hence it appears that the Central Limit Theorem from statistics holds true only when all the random variables are approximately equally weighted.



```
% File c:\M_files\bookupdate\rtdbimod.m
% MCA of rtd circuit using pre-screened (gapped) inputs;
% uses MATLAB function G2a.m
% Revised and updated 11/14/06
clc;clear;tic;
% Component values (KOHms)
R1=4.53;R2=34.8;R3=132;R4=9.09;R5=9.09;E1=5;
R6=4.53;R7=27.4;R8=20;R9=20;RT=1.915;
Nom=[R1 R2 R3 R4 R5 R6 R7 R8 R9 RT E1];
Vo=G2a(Nom);
%
% "Real world" tolerances
%
Tinit=0.001;Tlife=0.002;ppm=1e-6;
TC1=50*ppm;TC2=25*ppm;
Thi=Tinit+Tlife+35*TC1;Tlo=-Tinit-Tlife-80*TC1;
Trhi=8.1*1e-4;Trlo=-Trhi;Trefhi=0.02+35*TC2;
Treflo=-0.02-80*TC2;
%
T=[ Tlo Tlo Tlo Tlo Tlo Tlo Tlo Tlo Tlo Trlo Treflo;
   Thi Thi Thi Thi Thi Thi Thi Thi Thi Trhi Trefhi];
%
%T(1,4)=-0.1;T(2,4)=0.1; % Set R4 to +/- 10% tolerance and note output changes.
%
Nk=20000; % <<<<<<<<<<<<<<<<<< Nk
%
Nc=length(Nom); % Number of components
nb=30; % Number of bins in histograms
Ng=50; % Number of point in ideal Gaussian curve
randn('state',sum(100*clock)); % randomize normal RNG seed
Tn=zeros(Nk,Nc);Yn=zeros(Nk,Nc);sp=0.5; % gap width is 2*sp (sigmas)
for w=1:Nc
   k=0;
   while k < Nk
      z=randn;
      if (z<-sp)|(z>sp) % accept only rv's outside of -sp to +sp gap
         k=k+1; % next rv
         Yn(k,w)=z; % store in Yn array
      end
   end
end
for k=1:Nk
   for p=1:Nc;
      Rn(p,k)=Nom(p)*(((T(2,p)-T(1,p))/6)*(Yn(k,p)+3)+T(1,p)+1);
   end
   Vm(k)=G2a(Rn(:,k));
end
%
% get Nc input histograms
%
for p=1:Nc
   Vav(p)=mean(Yn(:,p));
   Vsd(p)=3*std(Yn(:,p));
   hin(p,:)=hist(Yn(:,p),nb)/Nk;
   VL(p)=min(Yn(:,p));VH(p)=max(Yn(:,p));
   intv(p)=(VH(p)-VL(p))/nb;
   q=1:nb;bin1(p,q)=VL(p)+intv(p)*(q-1);
end
%
% get output histogram
%
Vs=std(Vm);Vavg=mean(Vm);
hout=hist(Vm,nb)/Nk;VL=min(Vm);VH=max(Vm);
intv2=(VH-VL)/nb;
q=1:nb;bin2(q)=VL+intv2*(q-1);
% Ideal Gaussian curve
intvn=(VH-VL)/Ng;
c1=intv2/(Vs*sqrt(2*pi));
for q=1:Ng
   x1(q)=intvn*(q-1)+VL;
```

```
   y1(q)=c1*exp((-(x1(q)-Vavg)^2/(2*Vs^2)));
end
%
Vhi2=Vavg+Vs;Vlo2=Vavg-Vs;
Vsr=sprintf('%2.3f\n',3*Vs);Vavgr=sprintf('%2.3f\n',Vavg);
%
subplot(2,1,2)
bar(bin2,hout,1,'y');
set(gca,'FontSize',[8]);
hold on
h=plot(x1-intv2/2,y1,'k');
hold off
title('RTD output');xlabel('Volts dc')
xlabel('Volts DC');
axis([4.0 4.6 0 0.15]);
%axis auto; % Use when R4 tolerance is set to 10%
text(4.1,0.1,['Vavg=',Vavgr],'FontSize',8);
text(4.1,0.08,['3s=',Vsr],'FontSize',8);
text(4.1,0.06,['Nk = ',num2str(Nk)],'FontSize',8);
%
subplot(2,1,1)
set(gca,'FontSize',8);
stairs(bin1(1,:),hin(1,:),'k');
hold on
stairs(bin1(2,:),hin(2,:),'b');stairs(bin1(3,:),hin(3,:),'g');
stairs(bin1(4,:),hin(4,:),'k');stairs(bin1(5,:),hin(5,:),'k');
stairs(bin1(6,:),hin(6,:),'k');stairs(bin1(7,:),hin(7,:),'k');
stairs(bin1(8,:),hin(8,:),'k');hold off;
title('8 of 11 Pre-screened inputs');
grid off
xlabel('Sigma');
axis([-4 4 0 0.2]);
figure(1)
ET=toc
```

```
function y = G2a(X)
% RTD function
% reduced order A matrix - no opamps
% X = [R1 R2 R3 R4 R5 R6 R7 R8 R9 RT E1]
R1=X(1);R2=X(2);R3=X(3);R4=X(4);R5=X(5);R6=X(6);
R7=X(7);R8=X(8);R9=X(9);RT=X(10);E1=X(11);
% A matrix
A=[1/R1+1/R4+1/RT -1/RT -1/R4 0 -1/R1;
-1/RT 1/R5+1/R6+1/RT -1/R5 0 0;
-1/R4 0 1/R2+1/R3+1/R4 -1/R3 0;
0 -1/R5 1/R5+1/R7 0 0;
0 0 0 0 -1/R9];
% B matrix
B=[0;E1/R6;E1/R2;0;E1/R8];
C=A\B;y=C(4);
```