APPENDIX F: MATLAB CODE USED TO DETERMINE EXTRAPOLATION FACTOR

This appendix provides the MATLAB code that was used to evaluate the linearity of the "close" pressure as well as calculate the errors for the different extrapolation factors. The selection of the extrapolation factor based on these errors is described in Chapter 5.

Main Program

```
%This is a program that determines the voltage values that
%should be used to perform linear extrapolation. This program does
%the following on the data corresponding to a single transducer

%1.  Evaluates linearity of pressure close to the transducer
%2.  Evaluates linearity of voltage waveform
%3.  Evaluates linearity of different possible extrapolation
%    voltages

close all;
clear all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Determine the voltage data files and the pressure data    %
%files that will be used.                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%*.mat file of RLC information
file='/dunn/bigelow/Transducer_impedance/00066.mat';
load(file); %Load RLC values and resonance frequency

%Location of data files.
path='/dunn/bigelow/nonlinear_data/00066/11-10-01b/';

%The voltage will be the voltage waveform that has been
%applied to the real transducer

v_file='v00066p3_';

%The pressure will be the pressure waveform measured close
%to the surface of the transducer

c_file='c00066p3_';
```

```
%Settings for RA-30/Attenuators
RA30={'30_' '24_' '18_' '12_' '06_' '00_'};

ritec={'02' '06' '10' '14' '18' '22' '30' '38' '46' '54' '58' ...
         '62' '66' '70' '78' '86'};

%Output data desired
mv_p=0; %1 = record "close" waveforms as a movie
mv_v=0; %1 = record v waveforms as a movie

%Calibration factor
cal_factor=0.0425; %V/MPa for the Hydrophone.
f_m=freq_res*1e6; %Assumed resonance frequency
Q_error=6.25e-5; %Quantization error voltage of O-scope at
                 %2mV setting

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Read in the data
di=0;
V=[];
Vr=[];
P=[];
for ii=1:length(RA30)
   for ji=1:length(ritec)
      dat_c=[path c_file char(RA30(ii)) char(ritec(ji)) '.txt'];
      dat_v=[path v_file char(RA30(ii)) char(ritec(ji)) '.txt'];

      fid_c=fopen(dat_c,'r');

      if fid_c~=-1
         di=di+1;
         %Get close pressure data
         A=fscanf(fid_c,'%f',[2,inf]);

         p=(A(2,:)-mean(A(2,:)))/cal_factor;
         tc=A(1,:);

         %Get voltage data
         fid_v=fopen(dat_v,'r');
         B=fscanf(fid_v,'%f',[2,inf]);

         v=100*(B(2,:)-mean(B(2,:)));
         tv=B(1,:);

         %Save desired close pressure values
         if mv_p==1
```

```
   figure(1)
   clf
   plot(tc*1e6,p/max(abs(p)))
   axis([min(tc*1e6) max(tc*1e6) -1 1])
   grid

   Mov_p(di)=getframe;
end

p_max(di)=max(p);
p_min(di)=abs(min(p));
p_avg(di)=(max(p)-min(p))/2;

P=[P p'];

%Save desired voltage values
if mv_v==1
   figure(2)
   clf
   plot(tv*1e6,v/max(abs(v)))
   axis([min(tv*1e6) max(tv*1e6) -1 1])
   grid

   Mov_v(di)=getframe;
end


v_max(di)=max(v);
v_min(di)=abs(min(v));
v_pp(di)=(max(v)-min(v));
v_op(di)=max(abs(v));

V=[V v'];

%Find the magnitude of v at the fundamental frequency
Vsn_cf(di)=find_Vfo(v,tv,f_m);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Find the signal values assuming an RLC circuit
%Determine value for each frequency
Vs=fft(v);

M=length(Vs);

%Find corresponding freq. values
d_f1=[0:(M/2-1)]*2*pi/M;
```

```
        d_f2=[(M/2):(M-1)]*2*pi/M - 2*pi;

        d_f=[d_f1 d_f2];

        dt=tv(2)-tv(1);

        freq=d_f*(1/dt)/(2*pi);

        %Now for each frequency, find the voltage accross the
        %resistor
        for fi=1:length(freq)
            w=freq(fi)*2*pi;

            if abs(w)<1e-12
               Vout(fi)=0;
            else
               Zc1=R1 + j*w*L1 + 1/(j*w*C1);
               Vout(fi)=(R1/Zc1)*Vs(fi);
            end
        end

        v2=real(ifft(Vout));

        %Find some normalization factors
        v2op(di)=max(abs(v2)); %The maximum amplitude out
        v2pp(di)=max(v2)-min(v2); %The peak-peak value out
        V2sn(di)=max(abs(Vout)); %The voltage value at maximum
                                 %frequency

        Vr=[Vr v2'];

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        fclose(fid_c);
        fclose(fid_v);

     end
  end
end

%Set the reference value to be the lowest voltage that the error
%due to Quantization is less than 2 percent
Qer=10;
ref=0;
while Qer>=2
   ref=ref+1;
   Qer=100*Q_error/(cal_factor*p_max(ref));
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Linearity of system
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Determine the errors in Linear Extrapolation of p values based on
%p values
MINp_maxp=100*abs((p_min - p_min(ref)*(p_max/p_max(ref)))./p_min);
AVGp_maxp=100*abs((p_avg - p_avg(ref)*(p_max/p_max(ref)))./p_avg);

MAXp_minp=100*abs((p_max - p_max(ref)*(p_min/p_min(ref)))./p_max);
AVGp_minp=100*abs((p_avg - p_avg(ref)*(p_min/p_min(ref)))./p_avg);

MAXp_avgp=100*abs((p_max - p_max(ref)*(p_avg/p_avg(ref)))./p_max);
MINp_avgp=100*abs((p_min - p_min(ref)*(p_avg/p_avg(ref)))./p_min);

figure(3)
clf
subplot(311)
plot(v_pp,MINp_maxp,'b--')
hold
plot(v_pp,AVGp_maxp,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
title('Errors w/ p_c_l_o_s_e Extrapolation')
axis([min(v_pp) max(v_pp) 0 10])
legend('= Error in p_r','= Error in p_a_v_g')

subplot(312)
plot(v_pp,MAXp_minp,'k')
hold
plot(v_pp,AVGp_minp,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
axis([min(v_pp) max(v_pp) 0 10])
legend('= Error in p_c','= Error in p_a_v_g')

subplot(313)
plot(v_pp,MAXp_avgp,'k')
hold
plot(v_pp,MINp_maxp,'b--')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
```

```
axis([min(v_pp) max(v_pp) 0 10])
legend('= Error in p_c','= Error in p_r')

%Determine the errors in Linear Extrapolation of v values based on
%v values
MINv_maxv=100*abs((v_min - v_min(ref)*(v_max/v_max(ref)))./v_min);
VPPv_maxv=100*abs((v_pp - v_pp(ref)*(v_max/v_max(ref)))./v_pp);

MAXv_minv=100*abs((v_max - v_max(ref)*(v_min/v_min(ref)))./v_max);
VPPv_minv=100*abs((v_pp - v_pp(ref)*(v_min/v_min(ref)))./v_pp);

MAXv_vpp=100*abs((v_max - v_max(ref)*(v_pp/v_pp(ref)))./v_max);
MINv_vpp=100*abs((v_min - v_min(ref)*(v_pp/v_pp(ref)))./v_min);

figure(4)
clf
subplot(311)
plot(v_pp,MINv_maxv,'b--')
hold
plot(v_pp,VPPv_maxv,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in v')
title('Errors w/ v Extrapolation')
axis([min(v_pp) max(v_pp) 0 10])
legend('= Error in V_m_i_n','= Error in V_p_p')

subplot(312)
plot(v_pp,MAXv_minv,'k')
hold
plot(v_pp,VPPv_minv,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in v')
axis([min(v_pp) max(v_pp) 0 10])
legend('= Error in V_m_a_x','= Error in V_p_p')

subplot(313)
plot(v_pp,MAXv_vpp,'k')
hold
plot(v_pp,MINv_vpp,'b--')
grid
xlabel('v_p_p (V)')
ylabel('% Error in v')
axis([min(v_pp) max(v_pp) 0 10])
legend('= Error in V_m_a_x','= Error in V_m_i_n')
```

```
%Quantify the linearity (only consider p_min extrap).
Prlin_err=mean(MAXp_minp(ref:length(v_pp)));

figure(5)
clf
semilogx(v_pp,MAXp_minp)
grid
hold
semilogx(v_pp,Prlin_err*ones(1,length(v_pp)),'-.')
semilogx(v_pp(ref),MAXp_minp(ref),'o')
xlabel('v_p_p (V)')
ylabel('% Error in p_c')
legend('= Error in p_c (close)','= Mean Error','= Reference
Waveform')
title('Errors w/ p_c_l_o_s_e Extrapolation')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Linearity of Extrapolation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Determine the errors in Linear Extrapolation of p values based on
%v values
%v_op Based
MAXp_vop=100*(p_max - p_max(ref)*(v_op/v_op(ref)))./p_max;
MINp_vop=100*(p_min - p_min(ref)*(v_op/v_op(ref)))./p_min;
AVGp_vop=100*(p_avg - p_avg(ref)*(v_op/v_op(ref)))./p_avg;

%v_pp Based
MAXp_vpp=100*(p_max - p_max(ref)*(v_pp/v_pp(ref)))./p_max;
MINp_vpp=100*(p_min - p_min(ref)*(v_pp/v_pp(ref)))./p_min;
AVGp_vpp=100*(p_avg - p_avg(ref)*(v_pp/v_pp(ref)))./p_avg;

figure(6)
clf
subplot(211)
plot(v_pp,MAXp_vop,'k')
hold
plot(v_pp,MINp_vop,'b--')
plot(v_pp,AVGp_vop,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
title('Errors w/ V_o_p Extrapolation')
axis([min(v_pp) max(v_pp) -8 8])
legend('= Error in p_c','= Error in p_r','= Error in p_a_v_g');

subplot(212)
```

```
plot(v_pp,MAXp_vpp,'k')
hold
plot(v_pp,MINp_vpp,'b--')
plot(v_pp,AVGp_vpp,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
title('Errors w/ V_p_p Extrapolation')
axis([min(v_pp) max(v_pp) -8 8])

%Determine std error terms
PcEB(1)=mean(abs(MAXp_vop(ref:length(v_pp))));
PrEB(1)=mean(abs(MINp_vop(ref:length(v_pp))));
PaEB(1)=mean(abs(AVGp_vop(ref:length(v_pp))));

PcEB(2)=mean(abs(MAXp_vpp(ref:length(v_pp))));
PrEB(2)=mean(abs(MINp_vpp(ref:length(v_pp))));
PaEB(2)=mean(abs(AVGp_vpp(ref:length(v_pp))));

%Determine the errors in Linear Extrapolation of p values based on
%v2 values

%v2op Based
MAXp_v2op=100*(p_max - p_max(ref)*(v2op/v2op(ref)))./p_max;
MINp_v2op=100*(p_min - p_min(ref)*(v2op/v2op(ref)))./p_min;
AVGp_v2op=100*(p_avg - p_avg(ref)*(v2op/v2op(ref)))./p_avg;

%v2pp Based
MAXp_v2pp=100*(p_max - p_max(ref)*(v2pp/v2pp(ref)))./p_max;
MINp_v2pp=100*(p_min - p_min(ref)*(v2pp/v2pp(ref)))./p_min;
AVGp_v2pp=100*(p_avg - p_avg(ref)*(v2pp/v2pp(ref)))./p_avg;

figure(7)
clf
subplot(211)
plot(v_pp,MAXp_v2op,'k')
hold
plot(v_pp,MINp_v2op,'b--')
plot(v_pp,AVGp_v2op,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
title('Errors w/ V_2_o_p Extrapolation')
axis([min(v_pp) max(v_pp) -8 8])
legend('= Error in p_c','= Error in p_r','= Error in p_a_v_g');

subplot(212)
```

```
plot(v_pp,MAXp_v2pp,'k')
hold
plot(v_pp,MINp_v2pp,'b--')
plot(v_pp,AVGp_v2pp,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
title('Errors w/ V_2_p_p Extrapolation')
axis([min(v_pp) max(v_pp) -8 8])

%Determine std error terms
PcEB(3)=mean(abs(MAXp_v2op(ref:length(v_pp))));
PrEB(3)=mean(abs(MINp_v2op(ref:length(v_pp))));
PaEB(3)=mean(abs(AVGp_v2op(ref:length(v_pp))));

PcEB(4)=mean(abs(MAXp_v2pp(ref:length(v_pp))));
PrEB(4)=mean(abs(MINp_v2pp(ref:length(v_pp))));
PaEB(4)=mean(abs(AVGp_v2pp(ref:length(v_pp))));

%Determine the errors in Linear Extrapolation of pc values based
%on freq values

%Vsn_cf Based
MAXp_Vsn=100*(p_max - p_max(ref)*(Vsn_cf/Vsn_cf(ref)))./p_max;
MINp_Vsn=100*(p_min - p_min(ref)*(Vsn_cf/Vsn_cf(ref)))./p_min;
AVGp_Vsn=100*(p_avg - p_avg(ref)*(Vsn_cf/Vsn_cf(ref)))./p_avg;

MAXp_V2sn=100*(p_max - p_max(ref)*(V2sn/V2sn(ref)))./p_max;
MINp_V2sn=100*(p_min - p_min(ref)*(V2sn/V2sn(ref)))./p_min;
AVGp_V2sn=100*(p_avg - p_avg(ref)*(V2sn/V2sn(ref)))./p_avg;

figure(8)
clf
subplot(211)
plot(v_pp,MAXp_Vsn,'k')
hold
plot(v_pp,MINp_Vsn,'b--')
plot(v_pp,AVGp_Vsn,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
title('Errors w/ V(f_r_e_s) Extrapolation')
axis([min(v_pp) max(v_pp) -8 8])
legend('= Error in p_c','= Error in p_r','= Error in p_a_v_g');

subplot(212)
plot(v_pp,MAXp_V2sn,'k')
```

```
hold
plot(v_pp,MINp_V2sn,'b--')
plot(v_pp,AVGp_V2sn,'r-.')
grid
xlabel('v_p_p (V)')
ylabel('% Error in p')
title('Errors w/ V(f_m_a_x) Extrapolation')
axis([min(v_pp) max(v_pp) -8 8])

%Determine std error terms
PcEB(5)=mean(abs(MAXp_Vsn(ref:length(v_pp))));
PrEB(5)=mean(abs(MINp_Vsn(ref:length(v_pp))));
PaEB(5)=mean(abs(AVGp_Vsn(ref:length(v_pp))));

PcEB(6)=mean(abs(MAXp_V2sn(ref:length(v_pp))));
PrEB(6)=mean(abs(MINp_V2sn(ref:length(v_pp))));
PaEB(6)=mean(abs(AVGp_V2sn(ref:length(v_pp))));

figure(9)
clf
subplot(311)
bar(PcEB,'c')
grid

subplot(312)
bar(PrEB,'c')
grid

subplot(313)
bar(PaEB,'c')
grid

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Display error values.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('The %Error in p_close is ');
disp(Prlin_err);

disp('%ERROR IN p_c');
disp('The %Error w/ vop ');
disp(PcEB(1))
disp('The %Error w/ vpp ');
disp(PcEB(2))
disp('The %Error w/ v2op ');
disp(PcEB(3))
disp('The %Error w/ v2pp ');
```

```
disp(PcEB(4))
disp('The %Error w/ Vsn ');
disp(PcEB(5))
disp('The %Error w/ V2sn ');
disp(PcEB(6))

disp('%ERROR IN p_r');
disp('The %Error w/ vop ');
disp(PrEB(1))
disp('The %Error w/ vpp ');
disp(PrEB(2))
disp('The %Error w/ v2op ');
disp(PrEB(3))
disp('The %Error w/ v2pp ');
disp(PrEB(4))
disp('The %Error w/ Vsn ');
disp(PrEB(5))
disp('The %Error w/ V2sn ');
disp(PrEB(6))

disp('%ERROR IN p_avg');
disp('The %Error w/ vop ');
disp(PaEB(1))
disp('The %Error w/ vpp ');
disp(PaEB(2))
disp('The %Error w/ v2op ');
disp(PaEB(3))
disp('The %Error w/ v2pp ');
disp(PaEB(4))
disp('The %Error w/ Vsn ');
disp(PaEB(5))
disp('The %Error w/ V2sn ');
disp(PaEB(6))
```

Find Voltage Magnitude at Resonance Frequency

```
function [Vfo]=find_Vfo(v,time,fo)

%This is a MATLAB function that finds the value of voltage across
%the transducer at a specific frequency (transducer resonant
%frequency).

%Inputs
%    v = voltage waveform.
%    time = corresponding time values
%    fo = frequency value desired.
%Output
```

```
%    Vfo = value of voltage at desired frequency.
%****************************************************************

j=sqrt(-1);

%Determine multiplication factor.
Vff=exp(j*2*pi*fo*time);

%Multiply and integrate.
Vfo=2*abs(mean(Vff(:).*v(:)));
```