Table of Contents


```
clear all
close all
```

SECTION 1 - STATIC CALIBRATION

```
%%%%%%%%%%%   %%%%%%%%%%%  %%%%%%%%%%%  %%%%%%%%%%%%%  %%%%%%%%%%%%%
 %%%%%%%%%%%  %        %       %%%%%%%%%%%%
%             %            %                  %              %
 %         %  % %      %            %
%             %            %                  %              %
 %         %  %  %     %            %
%             %            %                  %              %
 %         %  %   %    %            %
%%%%%%%%%%%   %%%%%%%%%%%  %                  %              %
 %         %  %    %   %            %
          %   %            %                  %              %
 %         %  %     %  %            %
          %   %            %                  %              %
 %         %  %      % %            %
          %   %            %                  %              %
 %         %  %       %%            %
%%%%%%%%%%%   %%%%%%%%%%%  %%%%%%%%%%%        %        %%%%%%%%%%%%%
 %%%%%%%%%%%  %        %       %%%%%%%%%%%%
```

Data Input - Hand Recorded

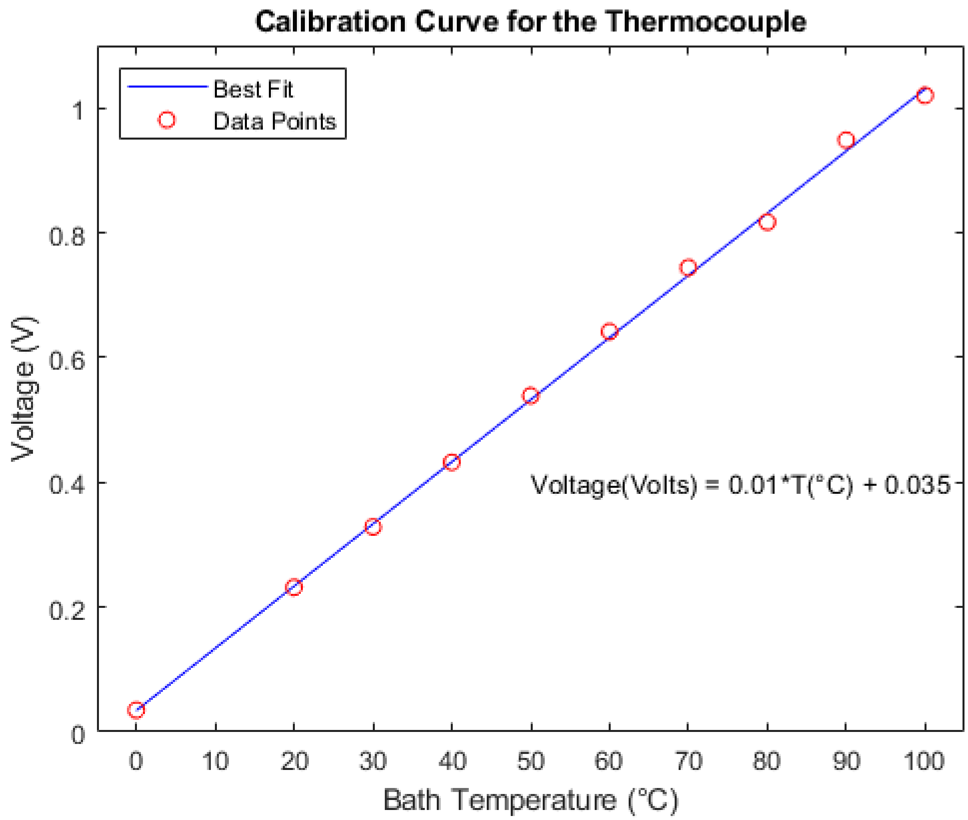
```
% Recorded Temperatures from Lab 2 - 2/5/19
Zero_Temp      = [0    ,0    ,0    ,0    ,0    ,0    ,0    ,0    ,0
  ,0    ]; % Degrees Celcius
```

```
Zero_Volt     =
 [.0347,.0383,.0362,.0314,.0366,.0372,.0337,.0318,.0339,.0355]; % Volt
Zero_Volt_Avg = mean(Zero_Volt); % Volt

Bath_Temp = [0            ,20   ,30   ,40   ,50   ,60   ,70   ,80
  ,90   ,100  ]; % Degrees Celcius
Ther_Resi = [27200
  ,12190,8160 ,5650 ,3940 ,2890 ,2160 ,1720 ,1340 ,1020 ]; % Ohms
Amp_Volt  =
 [Zero_Volt_Avg,.2317,.3285,.4321,.5387,.6416,.7441,.8171,.9484,1.020]; %
 Volt
```

Part 1 - Done within Word

Part 3 - Calibration Linear Curve - Using Resistance to Calculate Temperature of the Thermistor

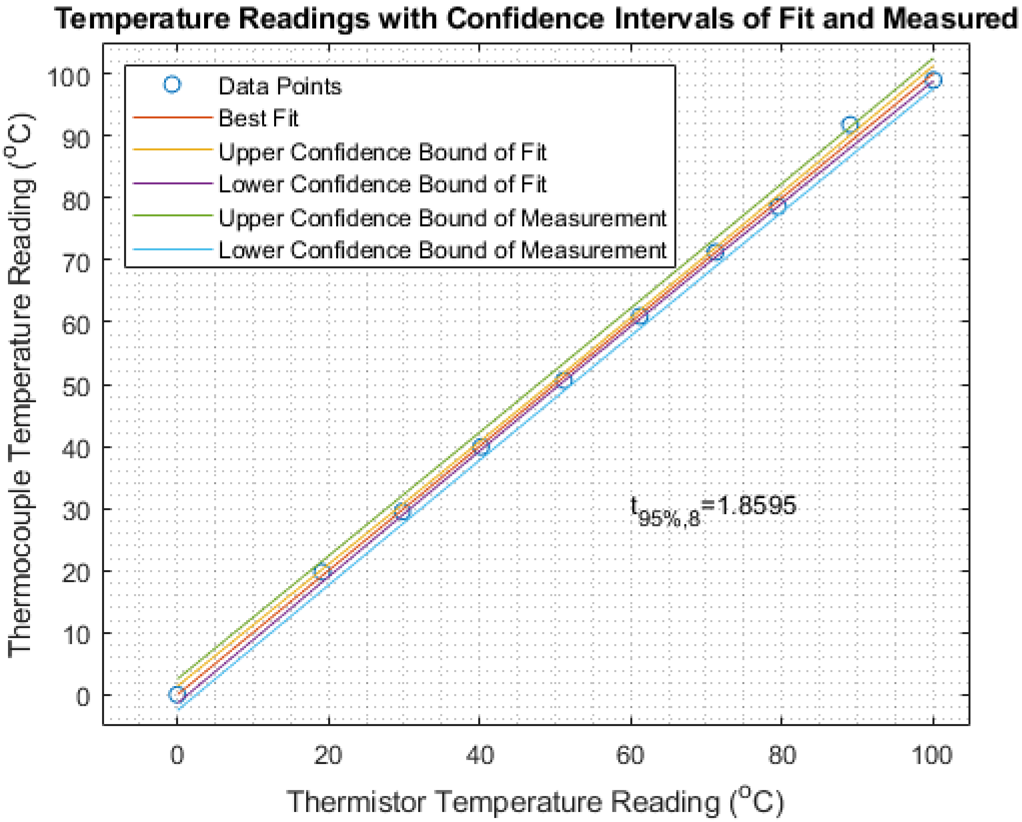
```
Beta = 3343;    % (K)   - Calculated on Document
R_0  = 9736;    % (Ohm) - Calculated on Document
T_0  = 298.15 ; % (K)   - Given as Room Temperature Water

Temp_Thermistor_Actual = zeros(10,1); % Creating array of zeros

for i = 1:length(Bath_Temp) % Calculating the Temperature of the
 Thermistor from the Resistance Data
    Temp_Thermistor_Actual(i) =  ( ( (1/Beta) * log((Ther_Resi(i))/
R_0) + (1/T_0) )^-1 ) -273.15;
end

Fit_Voltage = polyfit(Temp_Thermistor_Actual,Amp_Volt',1); %
 Calculates the slope and y-intercept to calculate fitted data of
 Voltage from actual temperatures
Fit_1_Voltage = Fit_Voltage(1).*Temp_Thermistor_Actual +
 Fit_Voltage(2); % Calculates the 10 voltages that use the paramaters
 calculated above

% Plotting of the raw Bath temperature and voltage measurement with
 the best fit line
figure(1)
plot(Temp_Thermistor_Actual,Fit_1_Voltage,'Color','b')
hold on
plot(Bath_Temp,Amp_Volt,'o','Color','r')
xlabel('Bath Temperature (°C)')
ylabel('Voltage (V)')
xlim([-5,105])
ylim([0,1.1])
text(50,0.4,'Voltage(Volts) = 0.01*T(°C) + 0.035')
title('Calibration Curve for the Thermocouple')
legend('Best Fit', 'Data Points', 'Location', 'Northwest')
```



Part 4 - Converting Thermocouple Voltage Measurements to Temperature. Compare with Thermistor Measurement

```
% Calculating the Thermocouple Temperatures in the Water baths from
 the calibration curve found in Part 3
N = length(Amp_Volt);
Temp_Thermocouple_Actual = zeros(N,1);
for i = 1:N
    Temp_Thermocouple_Actual(i) = (Amp_Volt(i)-Fit_Voltage(2))/
Fit_Voltage(1); % Calculates Temperature measurements from the
 voltages recorded
end

% Calculates the parameters for a fit line comparing the Thermocouple
 and Thermistor Temperatures
Best_Fit_2 =
 polyfit(Temp_Thermocouple_Actual,Temp_Thermistor_Actual,1);

% Calculating the Values of the fit line comparing the Thermocouple to
 the Thermistor
Fit_2 = Best_Fit_2(1).*Temp_Thermistor_Actual + Best_Fit_2(2); %2
```

```
% Stat Calculations
nu = length(Temp_Thermistor_Actual)-(2); % Calculating nu for
 statistic calculation
t_nup =tinv(.95, nu); % Calculating the t value for measuring
 confidence for fit data and measured
Temp_Zero_Mean = mean(Temp_Thermistor_Actual); % Calculating the mean
 of Thermistor
sumbot = sum((Temp_Thermistor_Actual-Temp_Zero_Mean).^2); %
 Denominator of the confidence calculation
syx = (sum((Temp_Thermocouple_Actual-Fit_2).^2)/nu).^.5; % Numerator
 for conidence calculation

% Calculating the confidence interval for the fit
confit = t_nup*syx*(1/
length(Temp_Thermistor_Actual)+(Temp_Thermistor_Actual-
Temp_Zero_Mean).^2/ sumbot).^.5;
% Calculating the confidence interval for the measured
conmeas = t_nup*syx*(1+1/
length(Temp_Thermistor_Actual)+(Temp_Thermistor_Actual-
Temp_Zero_Mean).^2/sumbot).^.5;

% Plotting
figure(2)
plot(Temp_Thermistor_Actual,Temp_Thermocouple_Actual,'o')
hold on
plot(Temp_Thermistor_Actual,Fit_2)
plot(Temp_Thermistor_Actual, Fit_2+confit, Temp_Thermistor_Actual,
 Fit_2-confit, Temp_Thermistor_Actual, Fit_2+conmeas,
 Temp_Thermistor_Actual, Fit_2-conmeas)
xlabel('Thermistor Temperature Reading (^oC)')
ylabel('Thermocouple Temperature Reading (^oC)')
text(60,30,['t_9_5_%_,_8=' num2str(t_nup)])
title('Temperature Readings with Confidence Intervals of Fit and
 Measured')
legend('Data Points', 'Best Fit', 'Upper Confidence Bound of
 Fit', 'Lower Confidence Bound of Fit', 'Upper Confidence Bound
 of Measurement', 'Lower Confidence Bound of Measurement',
  'Location', 'northwest')
xlim([-10,105])
ylim([-5,105])
grid minor
```



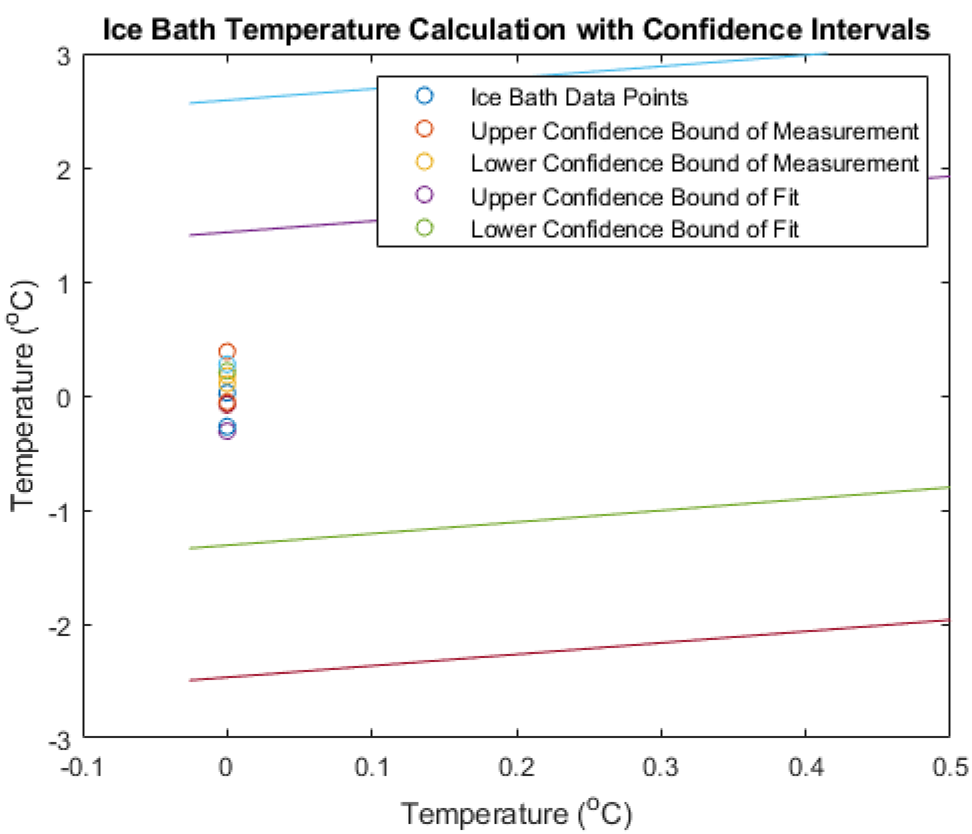
Part 2 - Repetitive Ice Water Measurements - Used for Word Document Submission

```
N = length(Zero_Temp);
Zero_Temp_Actual = zeros(N,1);
for i = 1:N
    Zero_Temp_Actual(i) = (Zero_Volt(i)-Fit_Voltage(2))/
Fit_Voltage(1); % Calculates Temperature measurements from the
 voltages recorded
end

% Mean Calculation of Temperatures
Zero_Temp_Mean = mean(Zero_Temp_Actual);

for i = 1:length(Zero_Temp)
    Std_Dev = sqrt(((Zero_Temp_Actual(i)-Zero_Temp_Mean)^2)/(N-1));
end

% From Tables
t_vP = 2.262; % 95% Confidence, 10 Data Points

% Calculations
Std_Dev_Mean = Std_Dev / N;
Population_Mean_Upper = Zero_Temp_Mean + t_vP*Std_Dev_Mean;
```

```
Population_Mean_Lower = Zero_Temp_Mean - t_vP*Std_Dev_Mean;
```

Part 5 - Ice Bath Measurement Comparison to the fit and confidence interval of fit and measurements

```
% Plotting
figure(3)
plot(0,Zero_Temp_Actual,'o') % Data point of Calculated temperature
 and 0 degrees C
hold on
% Plotting confidence intervals from Part 4 on graph to compare
plot(Temp_Thermistor_Actual, Fit_2+confit, Temp_Thermistor_Actual,
 Fit_2-confit, Temp_Thermistor_Actual, Fit_2+conmeas,
 Temp_Thermistor_Actual, Fit_2-conmeas)
ylim([-3,3])
xlim([-.1,.5])
xlabel('Temperature (^oC)')
ylabel('Temperature (^oC)')
legend('Ice Bath Data Points', 'Upper Confidence Bound of
 Measurement', 'Lower Confidence Bound of Measurement', 'Upper
 Confidence Bound of Fit', 'Lower Confidence Bound of Fit',
  'Location', 'northeast')
title('Ice Bath Temperature Calculation with Confidence Intervals')
```



SECTION 2 - DYNAMIC CALIBRATION

```
%%%%%%%%%%%   %%%%%%%%%%%  %%%%%%%%%%%  %%%%%%%%%%%%%  %%%%%%%%%%%%%
 %%%%%%%%%%%  %        %       %%%%%%%%%%%%%%%%%%%%%
%             %            %                  %              %
 %         %  % %      %            %        %
%             %            %                  %              %
 %         %  %  %     %            %        %
%             %            %                  %              %
 %         %  %   %    %            %        %
%%%%%%%%%%%   %%%%%%%%%%%  %                  %              %
 %         %  %    %   %            %        %
          %   %            %                  %              %
 %         %  %     %  %            %        %
          %   %            %                  %              %
 %         %  %      % %            %        %
          %   %            %                  %              %
 %         %  %       %%            %        %
%%%%%%%%%%%   %%%%%%%%%%%  %%%%%%%%%%%        %        %%%%%%%%%%%%%
 %%%%%%%%%%%  %        %       %%%%%%%%%%%%%%%%%%%%%
```

Part 1

```
% BIB = Bare Ice Boil
% BBI = Bare Boil Ice
% BIA = Bare Ice Air
% BIW = Bare Ice Water
% AIB = Aluminum Ice Boil
% ABI = Aluminum Boil Ice
% SIB = Steel Ice Boil
% SBI = Steel Boil Ice

% Bare Thermocoupler, N = 50,000
BIB_Time = xlsread('Part2Data.xlsx','bareiceboil' ,'A9:A50008'); %
 Second
BIB_Volt = xlsread('Part2Data.xlsx','bareiceboil' ,'B9:B50008'); %
 Volt
BBI_Time = xlsread('Part2Data.xlsx','bareboilice' ,'A9:A50008');
BBI_Volt = xlsread('Part2Data.xlsx','bareboilice' ,'B9:B50008');

% Bare Thermocoupler, N = 12,000
BIA_Time = xlsread('Part2Data.xlsx','bareiceair'  ,'A9:A50008'); %
 Second
BIA_Volt = xlsread('Part2Data.xlsx','bareiceair'  ,'B9:B50008'); %
 Volt
BIW_Time = xlsread('Part2Data.xlsx','bareicewater','A9:A50008');
BIW_Volt = xlsread('Part2Data.xlsx','bareicewater','B9:B50008');


% Aluminum and Steel Insulated Thermocoupler, N = 5,000
AIB_Time = xlsread('Part2Data.xlsx','aliceboil'   ,'A9:A5008'); %
 Second
```

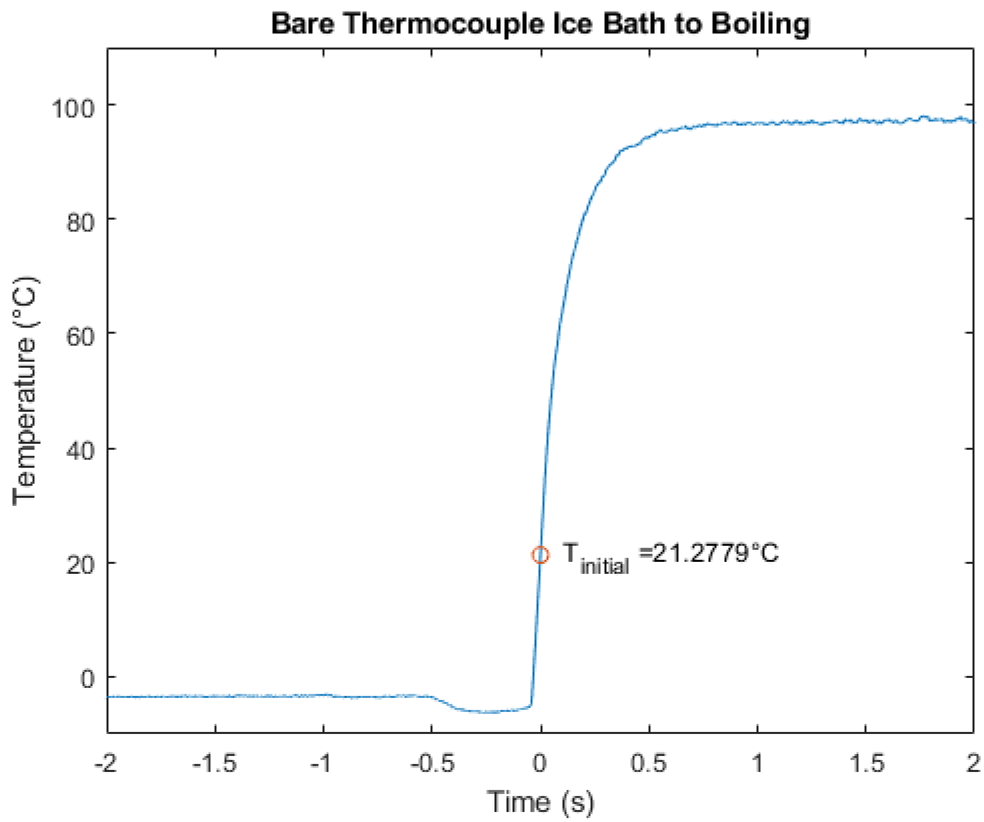
```
AIB_Volt = xlsread('Part2Data.xlsx','aliceboil'   ,'B9:B5008'); % Volt
ABI_Time = xlsread('Part2Data.xlsx','alboilice'   ,'A9:A5008');
ABI_Volt = xlsread('Part2Data.xlsx','alboilice'   ,'B9:B5008');
SIB_Time = xlsread('Part2Data.xlsx','steeliceboil','A9:A5008');
SIB_Volt = xlsread('Part2Data.xlsx','steeliceboil','B9:B5008');
SBI_Time = xlsread('Part2Data.xlsx','steelboilice','A9:A5008');
SBI_Volt = xlsread('Part2Data.xlsx','steelboilice','B9:B5008');

% Using the previous calibration curve to convert all voltage
 measurements to temperature
BIB_Temp = (BIB_Volt - Fit_Voltage(2))./Fit_Voltage(1);
BBI_Temp = (BBI_Volt - Fit_Voltage(2))./Fit_Voltage(1);
BIA_Temp = (BIA_Volt - Fit_Voltage(2))./Fit_Voltage(1);
BIW_Temp = (BIW_Volt - Fit_Voltage(2))./Fit_Voltage(1);
AIB_Temp = (AIB_Volt - Fit_Voltage(2))./Fit_Voltage(1);
ABI_Temp = (ABI_Volt - Fit_Voltage(2))./Fit_Voltage(1);
SIB_Temp = (SIB_Volt - Fit_Voltage(2))./Fit_Voltage(1);
SBI_Temp = (SBI_Volt - Fit_Voltage(2))./Fit_Voltage(1);

% Smoothing data
Span   = 50;
Window = ones(Span,1)/Span;
BIB_Temp_Smooth = conv(BIB_Temp,Window,'same');
BIB_Temp_Smooth = BIB_Temp_Smooth(1000:end-1000);
BIB_Time = BIB_Time(1000:end-1000);

Span   = 50;
Window = ones(Span,1)/Span;
BBI_Temp_Smooth = conv(BBI_Temp,Window,'same');
BBI_Temp_Smooth = BBI_Temp_Smooth(1000:end-1000);
BBI_Time = BBI_Time(1000:end-1000);

Span   = 500;
Window = ones(Span,1)/Span;
BIA_Temp_Smooth = conv(BIA_Temp,Window,'same');
BIA_Temp_Smooth = BIA_Temp_Smooth(250:end-250);
BIA_Time = BIA_Time(250:end-250);

Span   = 500;
Window = ones(Span,1)/Span;
BIW_Temp_Smooth = conv(BIW_Temp,Window,'same');
BIW_Temp_Smooth = BIW_Temp_Smooth(250:end-250);
BIW_Time = BIW_Time(250:end-250);

Span   = 30;
Window = ones(Span,1)/Span;
AIB_Temp_Smooth = conv(AIB_Temp,Window,'same');
AIB_Temp_Smooth = AIB_Temp_Smooth(100:end-100);
AIB_Time = AIB_Time(100:end-100);

Span   = 30;
Window = ones(Span,1)/Span;
ABI_Temp_Smooth = conv(ABI_Temp,Window,'same');
ABI_Temp_Smooth = ABI_Temp_Smooth(100:end-100);
```

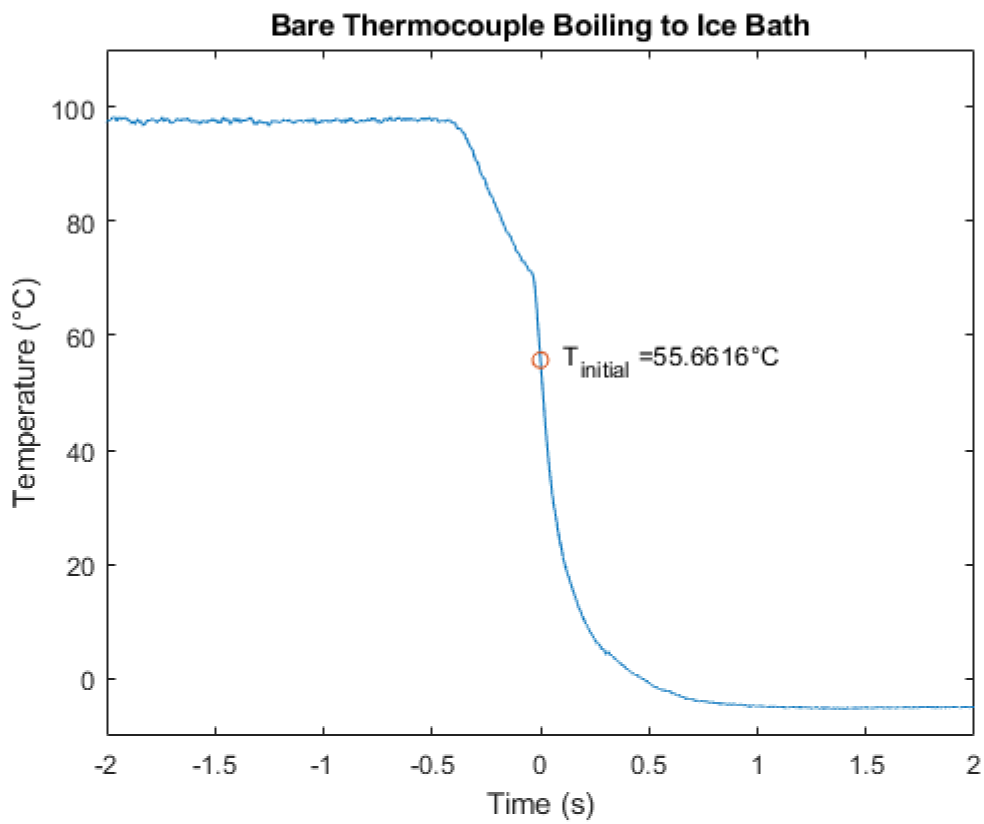
```
ABI_Time = ABI_Time(100:end-100);

Span   = 30;
Window = ones(Span,1)/Span;
SIB_Temp_Smooth = conv(SIB_Temp,Window,'same');
SIB_Temp_Smooth = SIB_Temp_Smooth(100:end-100);
SIB_Time = SIB_Time(100:end-100);

Span   = 30;
Window = ones(Span,1)/Span;
SBI_Temp_Smooth = conv(SBI_Temp,Window,'same');
SBI_Temp_Smooth = SBI_Temp_Smooth(100:end-100);
SBI_Time = SBI_Time(100:end-100);

% Using a sliding, first order polynomial to determine time of maximum
 slope

%BIB
Window = 15;
MaxSlope = 0;
for i = Window+1:length(BIB_Time)-Window-1
    p = polyfit(BIB_Time(i-Window:i+Window),BIB_Temp_Smooth(i-Window:i
+Window),1);
    if (abs(p(1))>MaxSlope)
        MaxSlope = abs(p(1));
        startfitBIB = i;
    end
end
BIB_Time = BIB_Time - BIB_Time(startfitBIB);

%BBI
Window = 15;
MaxSlope = 0;
for i = Window+1:length(BBI_Time)-Window-1
    p = polyfit(BBI_Time(i-Window:i+Window),BBI_Temp_Smooth(i-Window:i
+Window),1);
    if (abs(p(1))>MaxSlope)
        MaxSlope = abs(p(1));
        startfitBBI = i;
    end
end
BBI_Time = BBI_Time - BBI_Time(startfitBBI);

%BIA
Window = 10;
MaxSlope = 0;
for i = Window+1:length(BIA_Time)-Window-1
    p = polyfit(BIA_Time(i-Window:i+Window),BIA_Temp_Smooth(i-Window:i
+Window),1);
    if (abs(p(1))>MaxSlope)
        MaxSlope = abs(p(1));
        startfitBIA = i;
    end
end
```

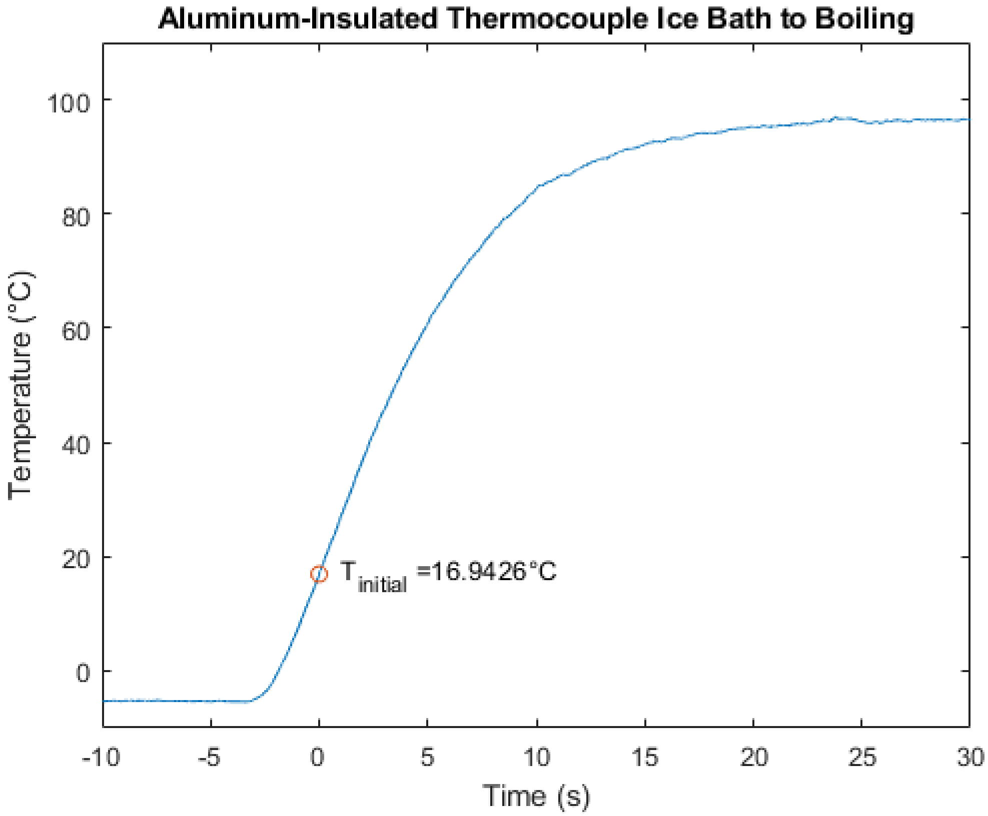
```
BIA_Time = BIA_Time - BIA_Time(startfitBIA);

%BIW
Window = 10;
MaxSlope = 0;
for i = Window+1:length(BIW_Time)-Window-1
    p = polyfit(BIW_Time(i-Window:i+Window),BIW_Temp_Smooth(i-Window:i
+Window),1);
    if (abs(p(1))>MaxSlope)
        MaxSlope = abs(p(1));
        startfitBIW = i;
    end
end
BIW_Time = BIW_Time - BIW_Time(startfitBIW);

%AIB
Window = 5;
MaxSlope = 0;
for i = Window+1:length(AIB_Time)-Window-1
    p = polyfit(AIB_Time(i-Window:i+Window),AIB_Temp_Smooth(i-Window:i
+Window),1);
    if (abs(p(1))>MaxSlope)
        MaxSlope = abs(p(1));
        startfitAIB = i;
    end
end
AIB_Time = AIB_Time - AIB_Time(startfitAIB);

%ABI
Window = 5;
MaxSlope = 0;
for i = Window+1:length(ABI_Time)-Window-1
    p = polyfit(ABI_Time(i-Window:i+Window),ABI_Temp_Smooth(i-Window:i
+Window),1);
    if (abs(p(1))>MaxSlope)
        MaxSlope = abs(p(1));
        startfitABI = i;
    end
end
ABI_Time = ABI_Time - ABI_Time(startfitABI);

%SIB
Window = 5;
MaxSlope = 0;
for i = Window+1:length(SIB_Time)-Window-1
    p = polyfit(SIB_Time(i-Window:i+Window),SIB_Temp_Smooth(i-Window:i
+Window),1);
    if (abs(p(1))>MaxSlope)
        MaxSlope = abs(p(1));
        startfitSIB = i;
    end
end
SIB_Time = SIB_Time - SIB_Time(startfitSIB);
```

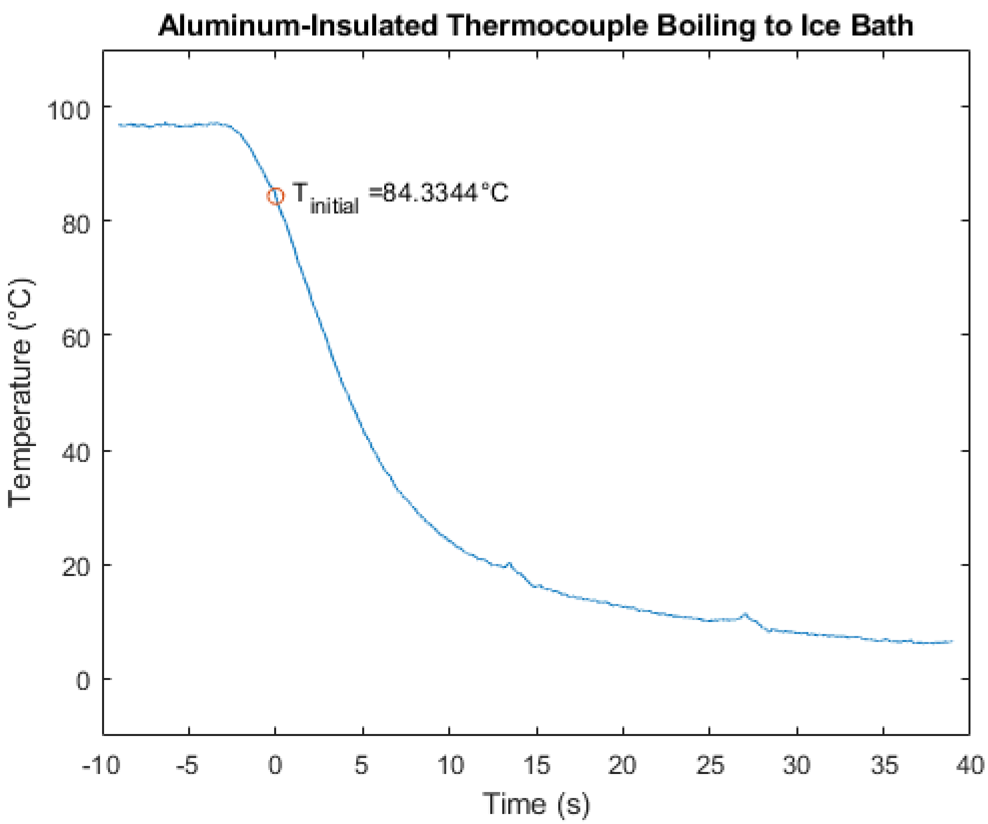
```
%SBI
Window = 5;
MaxSlope = 0;
for i = Window+1:length(SBI_Time)-Window-1
    p = polyfit(SBI_Time(i-Window:i+Window),SBI_Temp_Smooth(i-Window:i
+Window),1);
    if (abs(p(1))>MaxSlope)
        MaxSlope = abs(p(1));
        startfitSBI = i;
    end
end
SBI_Time = SBI_Time - SBI_Time(startfitSBI);

% Plotting

figure(4)
plot(BIB_Time,BIB_Temp_Smooth)
hold on
plot(BIB_Time(startfitBIB),BIB_Temp_Smooth(startfitBIB),'o')
title('Bare Thermocouple Ice Bath to Boiling')
xlabel('Time (s)')
ylabel('Temperature (°C)')
ylim([-10,110])
xlim([-2,2])
text(BIB_Time(startfitBIB)+.1,BIB_Temp_Smooth(startfitBIB),strcat('T_{initial}
 = ' ,num2str(BIB_Temp_Smooth(startfitBIB)),'°C'))

figure(5)
plot(BBI_Time,BBI_Temp_Smooth)
hold on
plot(BBI_Time(startfitBBI),BBI_Temp_Smooth(startfitBBI),'o')
title('Bare Thermocouple Boiling to Ice Bath')
xlabel('Time (s)')
ylabel('Temperature (°C)')
ylim([-10,110])
xlim([-2,2])
text(BBI_Time(startfitBBI)+.1,BBI_Temp_Smooth(startfitBBI),strcat('T_{initial}
 = ' ,num2str(BBI_Temp_Smooth(startfitBBI)),'°C'))

figure(6)
plot(AIB_Time,AIB_Temp_Smooth)
hold on
plot(AIB_Time(startfitAIB),AIB_Temp_Smooth(startfitAIB),'o')
title('Aluminum-Insulated Thermocouple Ice Bath to Boiling')
xlabel('Time (s)')
ylabel('Temperature (°C)')
ylim([-10,110])
xlim([-10,30])
text(AIB_Time(startfitAIB)+1,AIB_Temp_Smooth(startfitAIB),strcat('T_{initial}
 = ' ,num2str(AIB_Temp_Smooth(startfitAIB)),'°C'))

figure(7)
plot(ABI_Time,ABI_Temp_Smooth)
hold on
```

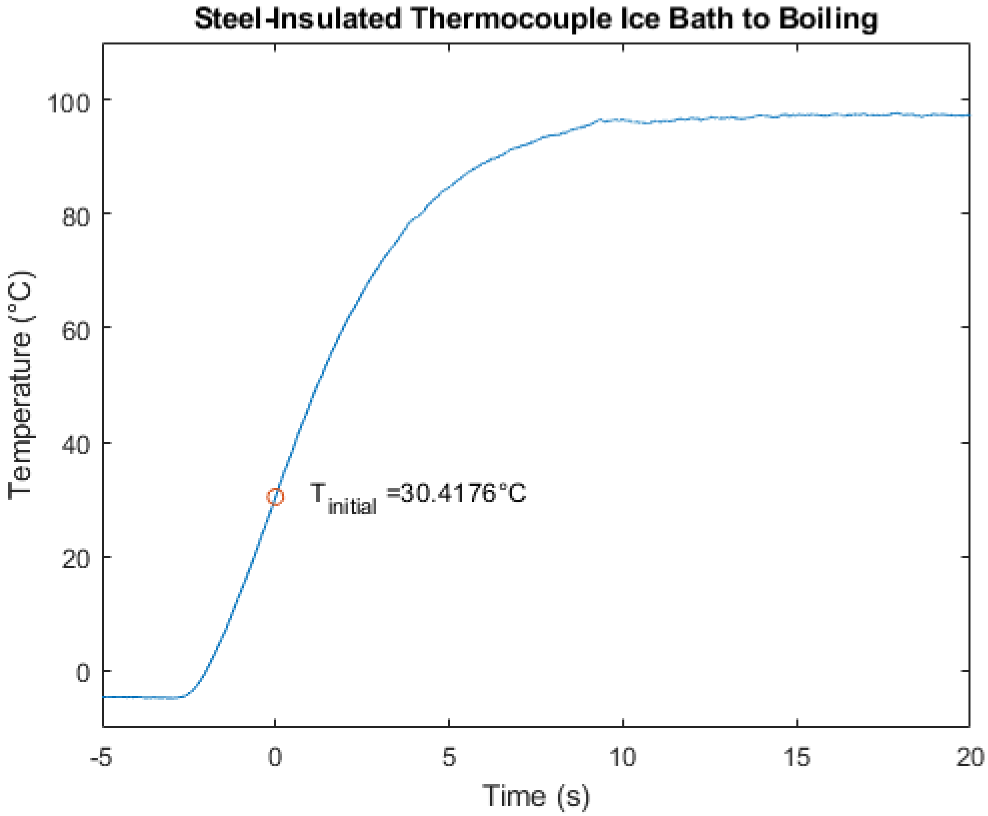
```
plot(ABI_Time(startfitABI),ABI_Temp_Smooth(startfitABI),'o')
title('Aluminum-Insulated Thermocouple Boiling to Ice Bath')
xlabel('Time (s)')
ylabel('Temperature (°C)')
ylim([-10,110])
xlim([-10,40])
text(ABI_Time(startfitABI)+1,ABI_Temp_Smooth(startfitABI),strcat('T_{initial}
 = ' ,num2str(ABI_Temp_Smooth(startfitABI)),'°C'))

figure(8)
plot(SIB_Time,SIB_Temp_Smooth)
hold on
plot(SIB_Time(startfitSIB),SIB_Temp_Smooth(startfitSIB),'o')
title('Steel-Insulated Thermocouple Ice Bath to Boiling')
xlabel('Time (s)')
ylabel('Temperature (°C)')
ylim([-10,110])
xlim([-5,20])
text(SIB_Time(startfitSIB)+1,SIB_Temp_Smooth(startfitSIB),strcat('T_{initial}
 = ' ,num2str(SIB_Temp_Smooth(startfitSIB)),'°C'))

figure(9)
plot(SBI_Time,SBI_Temp_Smooth)
hold on
plot(SBI_Time(startfitSBI),SBI_Temp_Smooth(startfitSBI),'o')
title('Steel-Insulated Thermocouple Boiling to Ice Bath')
xlabel('Time (s)')
ylabel('Temperature (°C)')
ylim([-10,110])
xlim([-5,40])
text(SBI_Time(startfitSBI)+1,SBI_Temp_Smooth(startfitSBI),strcat('T_{initial}
 = ' ,num2str(SBI_Temp_Smooth(startfitSBI)),'°C'))
```


Bare Thermocouple Ice Bath to Boiling
Temperature (°C)
Time (s)
T initial =21.2779°C
100
80
60
40
20
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

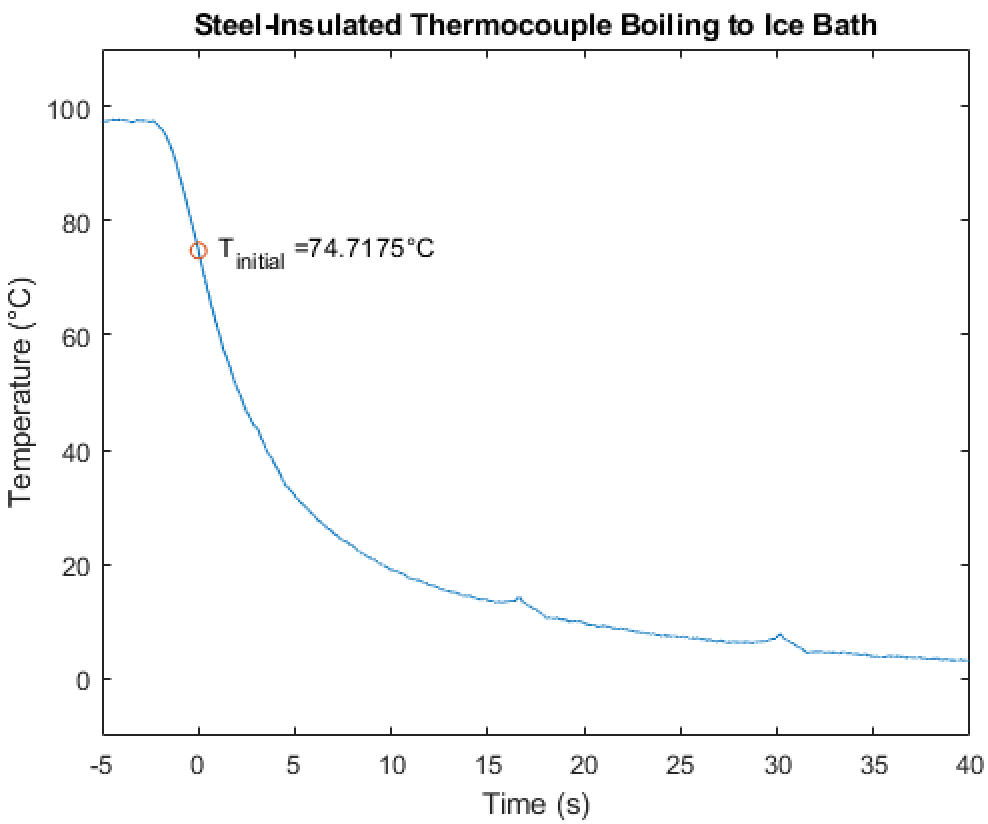

Bare Thermocouple Boiling to Ice Bath
Temperature (°C)
Time (s)
T initial =55.6616°C
100
80
60
40
20
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Aluminum-Insulated Thermocouple Ice Bath to Boiling
Temperature (°C)
Time (s)
$T_{initial}$ =16.9426°C
100
80
60
40
20
0
-10
-5
0
5
10
15
20
25
30


Aluminum-Insulated Thermocouple Boiling to Ice Bath
Temperature (°C)
Time (s)
$T_{initial}$ =84.3344°C
100
80
60
40
20
0
-10
-5
0
5
10
15
20
25
30
35
40



Steel-Insulated Thermocouple Ice Bath to Boiling
Temperature (°C)
Time (s)
T initial =30.4176°C
100
80
60
40
20
0
-5
0
5
10
15
20



Steel-Insulated Thermocouple Boiling to Ice Bath
Temperature (°C)
Time (s)
T initial =74.7175°C
100
80
60
40
20
0
-5
0
5
10
15
20
25
30
35
40

Part 2

```
% BIB = Bare Ice Boil
% BBI = Bare Boil Ice
% BIA = Bare Ice Air
% BIW = Bare Ice Water
% AIB = Aluminum Ice Boil
% ABI = Aluminum Boil Ice
% SIB = Steel Ice Boil
% SBI = Steel Boil Ice

% Calculate Final Temperature Average for each Curve
BIB_Temp_Final = mean(BIB_Temp_Smooth(end-1000:end));
BBI_Temp_Final = mean(BBI_Temp_Smooth(end-1000:end));
AIB_Temp_Final = mean(AIB_Temp_Smooth(end-1000:end));
ABI_Temp_Final = mean(ABI_Temp_Smooth(end-1000:end));
SIB_Temp_Final = mean(SIB_Temp_Smooth(end-1000:end));
SBI_Temp_Final = mean(SBI_Temp_Smooth(end-1000:end));

% Calculating Gamma Values for BIB
GammaBIB = zeros(length(BIB_Time),1);
for i = 1:length(BIB_Time)
    GammaBIB(i) = (BIB_Temp_Final-BIB_Temp_Smooth(i))/(BIB_Temp_Final-
BIB_Temp_Smooth(startfitBIB));
end

% Obtaining the indicies of the Gamma vector corresponding to specific
 values
for i = 1:length(BIB_Time)
    if GammaBIB(i) < 1
        GammaBIB_1 = i;
        break
    end
end

for i = 1:length(BIB_Time)
    if GammaBIB(i) < 0.7
        GammaBIB_7 = i;
        break
    end
end

for i = 1:length(BIB_Time)
    if GammaBIB(i) < 0.2
        GammaBIB_2 = i;
        break
    end
end

for i = 1:length(BIB_Time)
    if GammaBIB(i) < 0
        GammaBIB_0 = i;
        break
```

```
    end
end

% Calculating Gamma Values for BBI
GammaBBI = zeros(length(BBI_Time),1);
for i = 1:length(BBI_Time)
    GammaBBI(i) = (BBI_Temp_Final-BBI_Temp_Smooth(i))/(BBI_Temp_Final-
BBI_Temp_Smooth(startfitBBI));
end

% Obtaining the indicies of the Gamma vector corresponding to specific
 values
for i = 1:length(BBI_Time)
    if GammaBBI(i) < 1
        GammaBBI_1 = i;
        break
    end
end

for i = 1:length(BBI_Time)
    if GammaBBI(i) < 0.7
        GammaBBI_7 = i;
        break
    end
end

for i = 1:length(BBI_Time)
    if GammaBBI(i) < 0.2
        GammaBBI_2 = i;
        break
    end
end

for i = 1:length(BBI_Time)
    if GammaBBI(i) < 0
        GammaBBI_0 = i;
        break
    end
end

% Calculating Gamma Values for BIB
GammaAIB = zeros(length(AIB_Time),1);
for i = 1:length(AIB_Time)
    GammaAIB(i) = (AIB_Temp_Final-AIB_Temp_Smooth(i))/(AIB_Temp_Final-
AIB_Temp_Smooth(startfitAIB));
end

% Obtaining the indicies of the Gamma vector corresponding to specific
 values
for i = 1:length(AIB_Time)
    if GammaAIB(i) < 1
        GammaAIB_1 = i;
        break
    end
```

```
end

for i = 1:length(AIB_Time)
    if GammaAIB(i) < 0.7
        GammaAIB_7 = i;
        break
    end
end

for i = 1:length(AIB_Time)
    if GammaAIB(i) < 0.2
        GammaAIB_2 = i;
        break
    end
end

for i = 1:length(AIB_Time)
    if GammaAIB(i) < 0
        GammaAIB_0 = i;
        break
    end
end

% Calculating Gamma Values for BIB
GammaABI = zeros(length(ABI_Time),1);
for i = 1:length(ABI_Time)
    GammaABI(i) = (ABI_Temp_Final-ABI_Temp_Smooth(i))/(ABI_Temp_Final-
ABI_Temp_Smooth(startfitABI));
end

% Obtaining the indicies of the Gamma vector corresponding to specific
 values
for i = 1:length(ABI_Time)
    if GammaABI(i) < 1
        GammaABI_1 = i;
        break
    end
end

for i = 1:length(ABI_Time)
    if GammaABI(i) < 0.7
        GammaABI_7 = i;
        break
    end
end

for i = 1:length(ABI_Time)
    if GammaABI(i) < 0.2
        GammaABI_2 = i;
        break
    end
end

for i = 1:length(ABI_Time)
```

```
    if GammaABI(i) < 0
        GammaABI_0 = i;
        break
    end
end

% Calculating Gamma Values for BIB
GammaSIB = zeros(length(SIB_Time),1);
for i = 1:length(SIB_Time)
    GammaSIB(i) = (SIB_Temp_Final-SIB_Temp_Smooth(i))/(SIB_Temp_Final-
SIB_Temp_Smooth(startfitSIB));
end

% Obtaining the indicies of the Gamma vector corresponding to specific
 values
for i = 1:length(SIB_Time)
    if GammaSIB(i) < 1
        GammaSIB_1 = i;
        break
    end
end

for i = 1:length(SIB_Time)
    if GammaSIB(i) < 0.7
        GammaSIB_7 = i;
        break
    end
end

for i = 1:length(SIB_Time)
    if GammaSIB(i) < 0.2
        GammaSIB_2 = i;
        break
    end
end

for i = 1:length(SIB_Time)
    if GammaSIB(i) < 0
        GammaSIB_0 = i;
        break
    end
end

% Calculating Gamma Values for BIB
GammaSBI = zeros(length(SBI_Time),1);
for i = 1:length(SBI_Time)
    GammaSBI(i) = (SBI_Temp_Final-SBI_Temp_Smooth(i))/(SBI_Temp_Final-
SBI_Temp_Smooth(startfitSBI));
end

% Obtaining the indicies of the Gamma vector corresponding to specific
 values
for i = 1:length(SBI_Time)
    if GammaSBI(i) < 1
```

```
            GammaSBI_1 = i;
            break
        end
    end

    for i = 1:length(SBI_Time)
        if GammaSBI(i) < 0.7
            GammaSBI_7 = i;
            break
        end
    end

    for i = 1:length(SBI_Time)
        if GammaSBI(i) < 0.2
            GammaSBI_2 = i;
            break
        end
    end

    for i = 1:length(SBI_Time)
        if GammaSBI(i) < 0
            GammaSBI_0 = i;
            break
        end
    end

    % Creating Specific Gamma Log vectors using the indicies from above

    GammaBIB_01 = GammaBIB(GammaBIB_1:GammaBIB_0);
    BIB_Time_01 = BIB_Time(GammaBIB_1:GammaBIB_0);
    GammaBBI_01 = GammaBBI(GammaBBI_1:GammaBBI_0);
    BBI_Time_01 = BBI_Time(GammaBBI_1:GammaBBI_0);
    GammaBIB_27 = GammaBIB(GammaBIB_7:GammaBIB_2);
    BIB_Time_27 = BIB_Time(GammaBIB_7:GammaBIB_2);
    GammaBBI_27 = GammaBBI(GammaBBI_7:GammaBBI_2);
    BBI_Time_27 = BBI_Time(GammaBBI_7:GammaBBI_2);

    GammaAIB_01 = GammaAIB(GammaAIB_1:GammaAIB_0);
    AIB_Time_01 = AIB_Time(GammaAIB_1:GammaAIB_0);
    GammaABI_01 = GammaABI(GammaABI_1:GammaABI_0);
    ABI_Time_01 = ABI_Time(GammaABI_1:GammaABI_0);
    GammaAIB_27 = GammaAIB(GammaAIB_7:GammaAIB_2);
    AIB_Time_27 = AIB_Time(GammaAIB_7:GammaAIB_2);
    GammaABI_27 = GammaABI(GammaABI_7:GammaABI_2);
    ABI_Time_27 = ABI_Time(GammaABI_7:GammaABI_2);

    GammaSIB_01 = GammaSIB(GammaSIB_1:GammaSIB_0);
    SIB_Time_01 = SIB_Time(GammaSIB_1:GammaSIB_0);
    GammaSBI_01 = GammaSBI(GammaSBI_1:GammaSBI_0);
    SBI_Time_01 = SBI_Time(GammaSBI_1:GammaSBI_0);
    GammaSIB_27 = GammaSIB(GammaSIB_7:GammaSIB_2);
    SIB_Time_27 = SIB_Time(GammaSIB_7:GammaSIB_2);
    GammaSBI_27 = GammaSBI(GammaSBI_7:GammaSBI_2);
    SBI_Time_27 = SBI_Time(GammaSBI_7:GammaSBI_2);
```

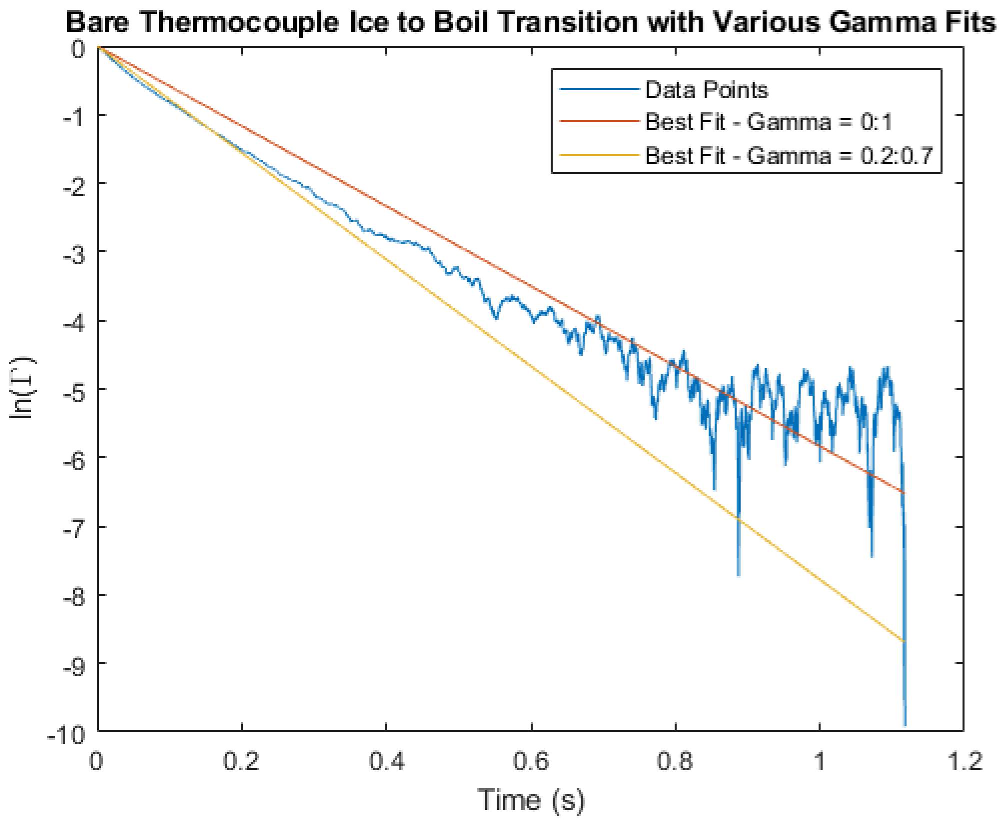
```
% Creating Log values of Gamma
GammaBIB_Log_01 = log(GammaBIB_01);
GammaBBI_Log_01 = log(GammaBBI_01);
GammaBIB_Log_27 = log(GammaBIB_27);
GammaBBI_Log_27 = log(GammaBBI_27);

GammaAIB_Log_01 = log(GammaAIB_01);
GammaABI_Log_01 = log(GammaABI_01);
GammaAIB_Log_27 = log(GammaAIB_27);
GammaABI_Log_27 = log(GammaABI_27);

GammaSIB_Log_01 = log(GammaSIB_01);
GammaSBI_Log_01 = log(GammaSBI_01);
GammaSIB_Log_27 = log(GammaSIB_27);
GammaSBI_Log_27 = log(GammaSBI_27);

% Finding the Fit line forced to the origin

SumX = 0;
SumY = 0;
for i = 1:length(GammaBIB_Log_01)
    SumX = GammaBIB_Log_01(i) * BIB_Time_01(i) + SumX;
    SumY = BIB_Time_01(i)^2 + SumY;
end
CoBIB_01 = SumX/SumY;
FitBIB_01 = CoBIB_01.*BIB_Time_01;


SumX = 0;
SumY = 0;
for i = 1:length(GammaBBI_Log_01)
    SumX = GammaBBI_Log_01(i) * BBI_Time_01(i) + SumX;
    SumY = BBI_Time_01(i)^2 + SumY;
end
CoBBI_01 = SumX/SumY;
FitBBI_01 = CoBBI_01.*BBI_Time_01;


SumX = 0;
SumY = 0;
for i = 1:length(GammaBIB_Log_27)
    SumX = GammaBIB_Log_27(i) * BIB_Time_27(i) + SumX;
    SumY = BIB_Time_27(i)^2 + SumY;
end
CoBIB_27 = SumX/SumY;
FitBIB_27 = CoBIB_27.*BIB_Time_01;

SumX = 0;
SumY = 0;
for i = 1:length(GammaBBI_Log_27)
    SumX = GammaBBI_Log_27(i) * BBI_Time_27(i) + SumX;
    SumY = BBI_Time_27(i)^2 + SumY;
end
```

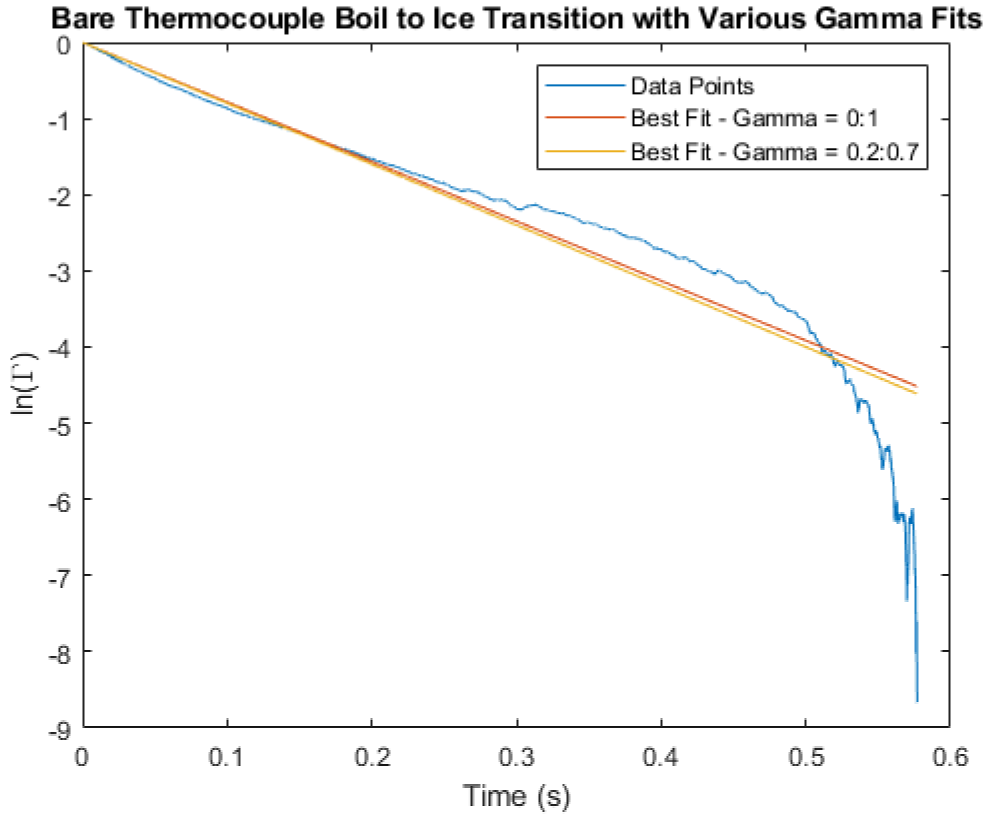
```
CoBBI_27 = SumX/SumY;
FitBBI_27 = CoBBI_27.*BBI_Time_01;


SumX = 0;
SumY = 0;
for i = 1:length(GammaAIB_Log_01)
    SumX = GammaAIB_Log_01(i) * AIB_Time_01(i) + SumX;
    SumY = AIB_Time_01(i)^2 + SumY;
end
CoAIB_01 = SumX/SumY;
FitAIB_01 = CoAIB_01.*AIB_Time_01;


SumX = 0;
SumY = 0;
for i = 1:length(GammaABI_Log_01)
    SumX = GammaABI_Log_01(i) * ABI_Time_01(i) + SumX;
    SumY = ABI_Time_01(i)^2 + SumY;
end
CoABI_01 = SumX/SumY;
FitABI_01 = CoABI_01.*ABI_Time_01;


SumX = 0;
SumY = 0;
for i = 1:length(GammaAIB_Log_27)
    SumX = GammaAIB_Log_27(i) * AIB_Time_27(i) + SumX;
    SumY = AIB_Time_27(i)^2 + SumY;
end
CoAIB_27 = SumX/SumY;
FitAIB_27 = CoAIB_27.*AIB_Time_01;

SumX = 0;
SumY = 0;
for i = 1:length(GammaABI_Log_27)
    SumX = GammaABI_Log_27(i) * ABI_Time_27(i) + SumX;
    SumY = ABI_Time_27(i)^2 + SumY;
end
CoABI_27 = SumX/SumY;
FitABI_27 = CoABI_27.*ABI_Time_01;



SumX = 0;
SumY = 0;
for i = 1:length(GammaSIB_Log_01)
    SumX = GammaSIB_Log_01(i) * SIB_Time_01(i) + SumX;
    SumY = SIB_Time_01(i)^2 + SumY;
end
CoSIB_01 = SumX/SumY;
FitSIB_01 = CoSIB_01.*SIB_Time_01;
```

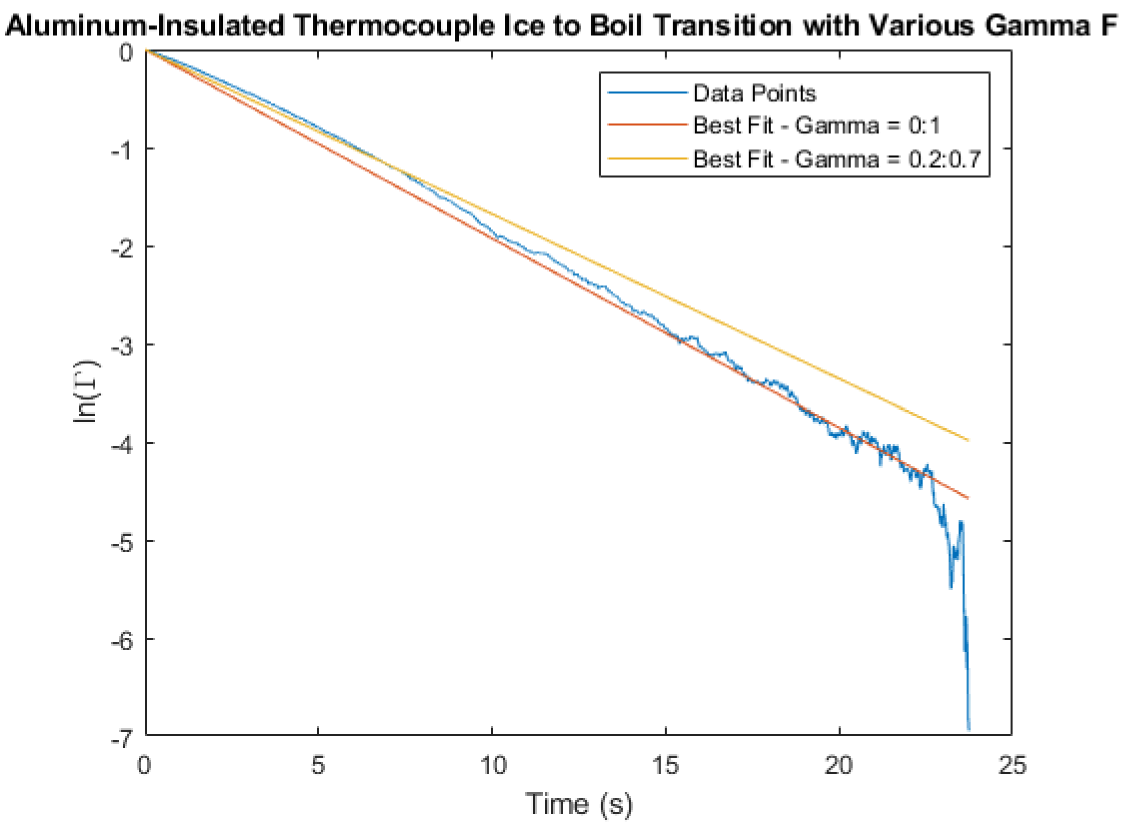
```
SumX = 0;
SumY = 0;
for i = 1:length(GammaSBI_Log_01)
    SumX = GammaSBI_Log_01(i) * SBI_Time_01(i) + SumX;
    SumY = SBI_Time_01(i)^2 + SumY;
end
CoSBI_01 = SumX/SumY;
FitSBI_01 = CoSBI_01.*SBI_Time_01;


SumX = 0;
SumY = 0;
for i = 1:length(GammaSIB_Log_27)
    SumX = GammaSIB_Log_27(i) * SIB_Time_27(i) + SumX;
    SumY = SIB_Time_27(i)^2 + SumY;
end
CoSIB_27 = SumX/SumY;
FitSIB_27 = CoSIB_27.*SIB_Time_01;

SumX = 0;
SumY = 0;
for i = 1:length(GammaSBI_Log_27)
    SumX = GammaSBI_Log_27(i) * SBI_Time_27(i) + SumX;
    SumY = SBI_Time_27(i)^2 + SumY;
end
CoSBI_27 = SumX/SumY;
FitSBI_27 = CoSBI_27.*SBI_Time_01;

% Plotting the Figure

figure(10)
plot(BIB_Time_01,GammaBIB_Log_01)
hold on
plot(BIB_Time_01,FitBIB_01)
plot(BIB_Time_01,FitBIB_27)
title('Bare Thermocouple Ice to Boil Transition with Various Gamma
 Fits')
legend('Data Points', 'Best Fit - Gamma = 0:1' , 'Best Fit - Gamma =
 0.2:0.7', 'Location', 'northeast')
xlabel('Time (s)')
ylabel('ln(\Gamma)')

figure(11)
plot(BBI_Time_01,GammaBBI_Log_01)
hold on
plot(BBI_Time_01,FitBBI_01)
plot(BBI_Time_01,FitBBI_27)
title('Bare Thermocouple Boil to Ice Transition with Various Gamma
 Fits')
legend('Data Points', 'Best Fit - Gamma = 0:1' , 'Best Fit - Gamma =
 0.2:0.7', 'Location', 'northeast')
xlabel('Time (s)')
ylabel('ln(\Gamma)')
```

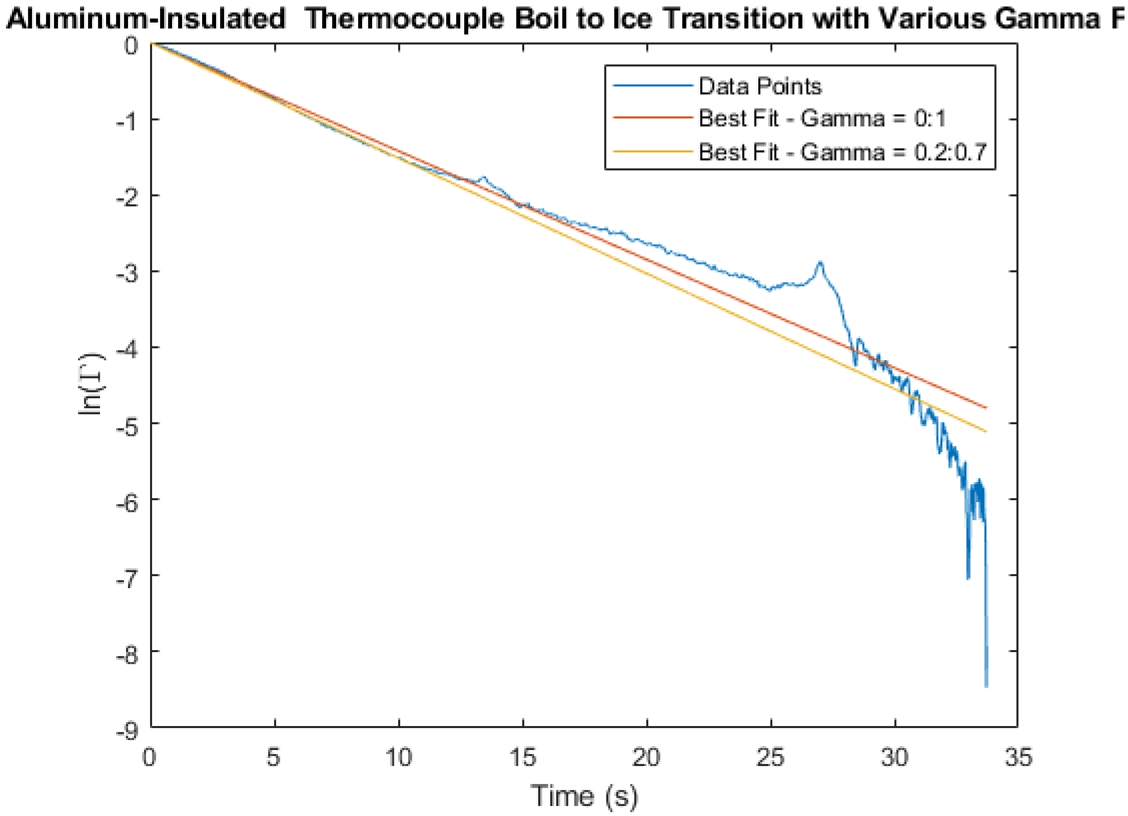
```
figure(12)
plot(AIB_Time_01,GammaAIB_Log_01)
hold on
plot(AIB_Time_01,FitAIB_01)
plot(AIB_Time_01,FitAIB_27)
title('Aluminum-Insulated Thermocouple Ice to Boil Transition with
 Various Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0:1' , 'Best Fit - Gamma =
 0.2:0.7', 'Location', 'northeast')
xlabel('Time (s)')
ylabel('ln(\Gamma)')

figure(13)
plot(ABI_Time_01,GammaABI_Log_01)
hold on
plot(ABI_Time_01,FitABI_01)
plot(ABI_Time_01,FitABI_27)
title('Aluminum-Insulated  Thermocouple Boil to Ice Transition with
 Various Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0:1' , 'Best Fit - Gamma =
 0.2:0.7', 'Location', 'northeast')
xlabel('Time (s)')
ylabel('ln(\Gamma)')

figure(14)
plot(SIB_Time_01,GammaSIB_Log_01)
hold on
plot(SIB_Time_01,FitSIB_01)
plot(SIB_Time_01,FitSIB_27)
title('Steel-Insulated Thermocouple Ice to Boil Transition with
 Various Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0:1' , 'Best Fit - Gamma =
 0.2:0.7', 'Location', 'northeast')
xlabel('Time (s)')
ylabel('ln(\Gamma)')

figure(15)
plot(SBI_Time_01,GammaSBI_Log_01)
hold on
plot(SBI_Time_01,FitSBI_01)
plot(SBI_Time_01,FitSBI_27)
title('Steel-Insulated Thermocouple Boil to Ice Transition with
 Various Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0:1' , 'Best Fit - Gamma =
 0.2:0.7', 'Location', 'northeast')
xlabel('Time (s)')
ylabel('ln(\Gamma)')

Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
```

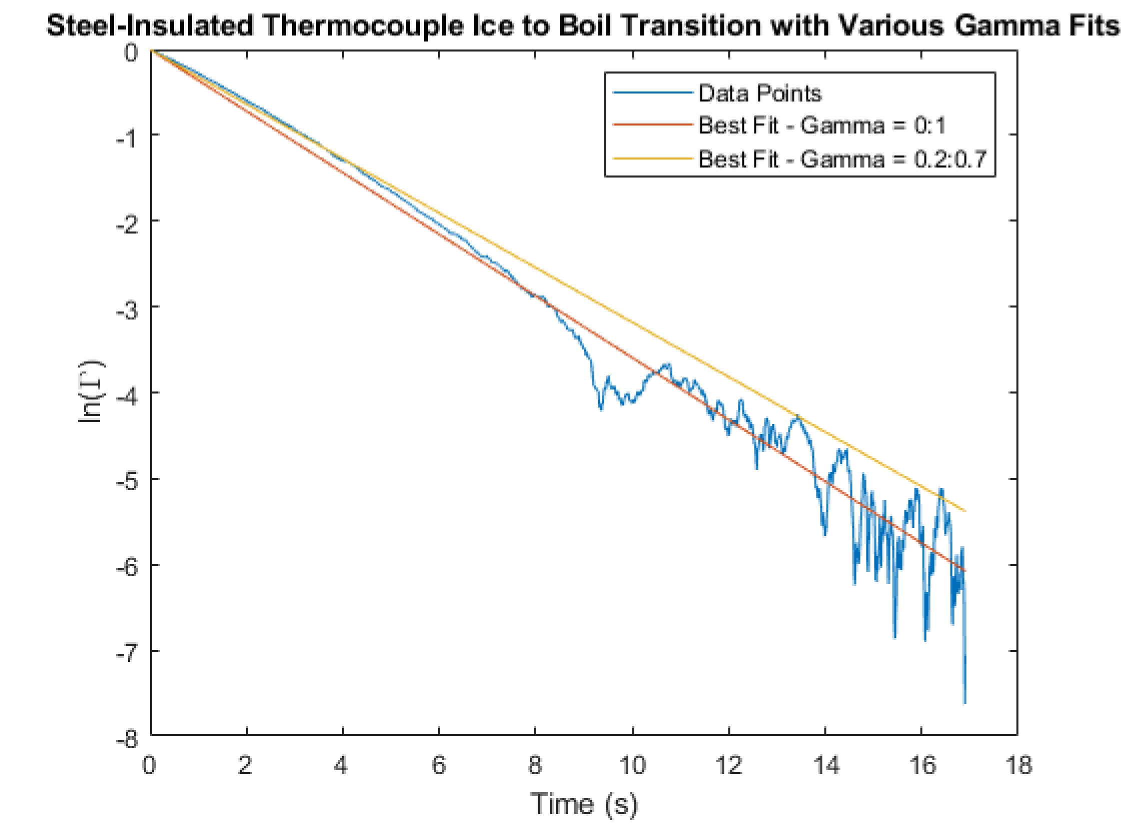



Bare Thermocouple Boil to Ice Transition with Various Gamma Fits
Data Points
Best Fit - Gamma = 0:1
Best Fit - Gamma = 0.2:0.7
ln(Γ)
Time (s)
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
0
0.1
0.2
0.3
0.4
0.5
0.6

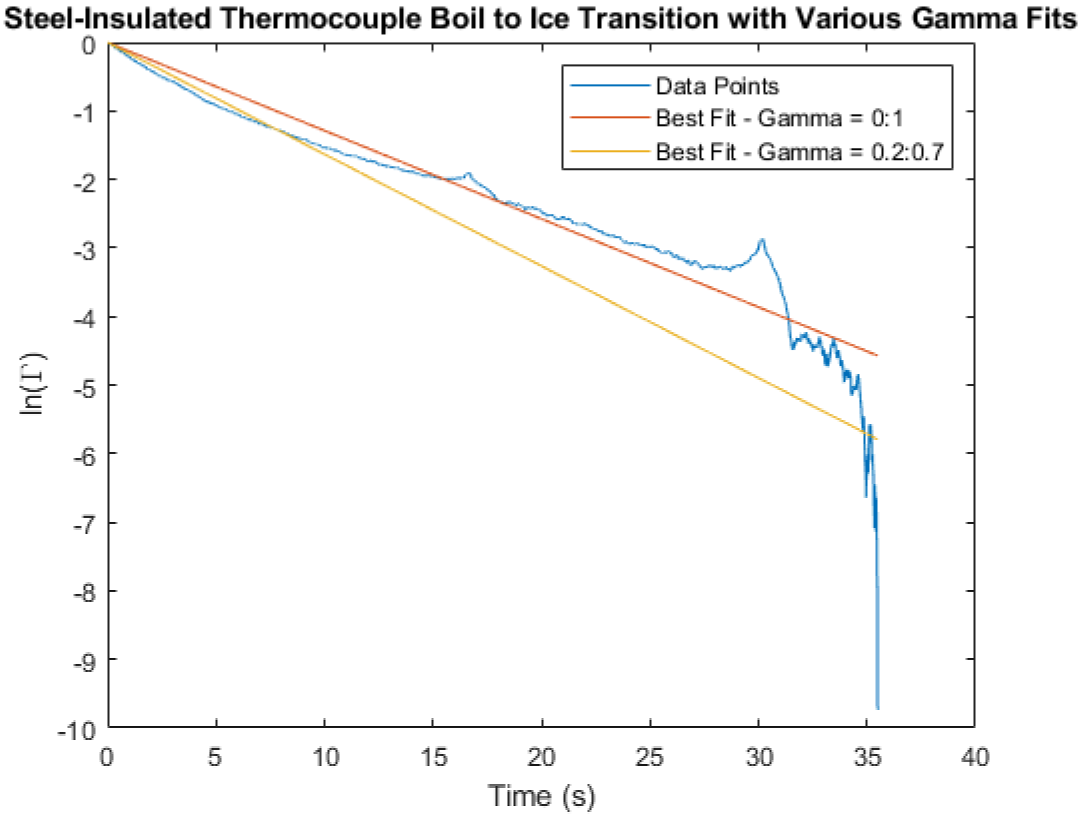

Aluminum-Insulated Thermocouple Ice to Boil Transition with Various Gamma F
Data Points
Best Fit - Gamma = 0:1
Best Fit - Gamma = 0.2:0.7
ln(Γ)
Time (s)
0
-1
-2
-3
-4
-5
-6
-7
0
5
10
15
20
25



Aluminum-Insulated Thermocouple Boil to Ice Transition with Various Gamma F
Data Points
Best Fit - Gamma = 0:1
Best Fit - Gamma = 0.2:0.7
ln(Γ)
Time (s)
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
0
5
10
15
20
25
30
35



Steel-Insulated Thermocouple Ice to Boil Transition with Various Gamma Fits
Data Points
Best Fit - Gamma = 0:1
Best Fit - Gamma = 0.2:0.7
ln(Γ)
Time (s)
0
-1
-2
-3
-4
-5
-6
-7
-8
0
2
4
6
8
10
12
14
16
18



Part 3

```
% Calculation for Bare Ice Boil and Boil Ice

BIB_Temp_Smooth_01 = BIB_Temp_Smooth(GammaBIB_1:GammaBIB_0);
BBI_Temp_Smooth_01 = BBI_Temp_Smooth(GammaBBI_1:GammaBBI_0);
BIB_Temp_Smooth_27 = BIB_Temp_Smooth(GammaBIB_7:GammaBIB_2);
BBI_Temp_Smooth_27 = BBI_Temp_Smooth(GammaBBI_7:GammaBBI_2);

BIB_01_Cons = real(1/CoBIB_01);
BBI_01_Cons = real(1/CoBBI_01);
BIB_27_Cons = real(1/CoBIB_27);
BBI_27_Cons = real(1/CoBBI_27);

for i = 1:length(BIB_Time_01)
    if BIB_Time_01(i) > 5*BIB_01_Cons
        BIB_Time_01_5T = i;
        break
    end
end

for i = 1:length(BBI_Time_01)
    if BBI_Time_01(i) > 5*BBI_01_Cons
        BBI_Time_01_5T = i;
        break
```

```
    end
end


for i = 1:length(BIB_Time_27)
    if i > 5*BIB_27_Cons
        BIB_Time_27_5T = i;
        break
    end
end

for i = 1:length(BBI_Time_27)
    if i > 5*BBI_27_Cons
        BBI_Time_27_5T = i;
        break
    end
end

BIB_Fit_01 = BIB_Temp_Smooth(startfitBIB) + (BIB_Temp_Final-
BIB_Temp_Smooth(startfitBIB))*(1-exp(BIB_Time/BIB_01_Cons));
BBI_Fit_01 = BBI_Temp_Smooth(startfitBBI) + (BBI_Temp_Final-
BBI_Temp_Smooth(startfitBBI))*(1-exp(BBI_Time/BBI_01_Cons));
BIB_Fit_27 = BIB_Temp_Smooth(startfitBIB) + (BIB_Temp_Final-
BIB_Temp_Smooth(startfitBIB))*(1-exp(BIB_Time/BIB_27_Cons));
BBI_Fit_27 = BBI_Temp_Smooth(startfitBBI) + (BBI_Temp_Final-
BBI_Temp_Smooth(startfitBBI))*(1-exp(BBI_Time/BBI_27_Cons));

% Finding when time = 0
BIB_Fit_01 = BIB_Fit_01(startfitBIB:end);
BBI_Fit_01 = BBI_Fit_01(startfitBBI:end);
BIB_Fit_27 = BIB_Fit_27(startfitBIB:end);
BBI_Fit_27 = BBI_Fit_27(startfitBBI:end);

BIB_Time_0 = BIB_Time(startfitBIB:end);
BBI_Time_0 = BBI_Time(startfitBBI:end);

BIB_01_Cons = -real(1/CoBIB_01);
BBI_01_Cons = -real(1/CoBBI_01);
BIB_27_Cons = -real(1/CoBIB_27);
BBI_27_Cons = -real(1/CoBBI_27);

for i = 1:length(BIB_Time_01)
    if BIB_Time_01(i) > 5*BIB_01_Cons
        BIB_Time_01_5T = BIB_Time_01(i);
        break
    end
end

for i = 1:length(BBI_Time_01)
    if BBI_Time_01(i) > 5*BBI_01_Cons
        BBI_Time_01_5T = BBI_Time_01(i);
        break
    end
end
```

```
for i = 1:length(BIB_Time_27)
    if BIB_Time_27(i) > 5*BIB_27_Cons
        BIB_Time_27_5T = BIB_Time_27(i);
        break
    end
end

for i = 1:length(BBI_Time_27)
    if BBI_Time_27(i) > 5*BBI_27_Cons
        BBI_Time_27_5T = BBI_Time_27(i);
        break
    end
end

BIB_Temp_Smooth_s = BIB_Temp_Smooth(startfitBIB:end);

S_yx_BIB_01 = 0;
for i = 1:length(BIB_Temp_Smooth_s)
    S_yx_BIB_01 = (BIB_Temp_Smooth_s(i)-BIB_Fit_01(i))^2 +
 S_yx_BIB_01;
end
S_yx_BIB_01 = sqrt(S_yx_BIB_01/length(BIB_Temp_Smooth_s));


S_yx_BIB_27 = 0;
for i = 1:length(BIB_Temp_Smooth_s)
    S_yx_BIB_27 = (BIB_Temp_Smooth_s(i)-BIB_Fit_27(i))^2 +
 S_yx_BIB_27;
end
S_yx_BIB_27 = sqrt(S_yx_BIB_27/length(BIB_Temp_Smooth_s));


BBI_Temp_Smooth_s = BBI_Temp_Smooth(startfitBBI:end);

S_yx_BBI_01 = 0;
for i = 1:length(BBI_Temp_Smooth_s)
    S_yx_BBI_01 = (BBI_Temp_Smooth_s(i)-BBI_Fit_01(i))^2 +
 S_yx_BBI_01;
end
S_yx_BBI_01 = sqrt(S_yx_BBI_01/length(BBI_Temp_Smooth_s));

S_yx_BBI_27 = 0;
for i = 1:length(BBI_Temp_Smooth_s)
    S_yx_BBI_27 = (BBI_Temp_Smooth_s(i)-BBI_Fit_27(i))^2 +
 S_yx_BBI_27;
end
S_yx_BBI_27 = sqrt(S_yx_BBI_27/length(BBI_Temp_Smooth_s));

% Plotting

figure(16)
plot(BIB_Time,BIB_Temp_Smooth)
hold on
```

```
plot(BIB_Time_0,BIB_Fit_27)
plot(BIB_Time_0,BIB_Fit_01)
xlim([-0.1,BIB_Time_01_5T])
ylim([-10,110])
xlabel('Time (s)')
ylabel('Temperature (^oC)')
text(0.4,60,strcat('s_{yx} = ', num2str(S_yx_BIB_01), 'K',' For Gamma
 = (0,1)'))
text(0.4,50,strcat('s_{yx} = ', num2str(S_yx_BIB_27), 'K',' For Gamma
 = (0.2,0.7)'))
ax = gca;
ax.YAxisLocation = 'origin';
title('Bare Thermocouple with Different Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0.2:0.7', 'Best Fit - Gamma
 = 0:1', 'Location', 'southeast')
grid minor

figure(17)
plot(BBI_Time,BBI_Temp_Smooth)
hold on
plot(BBI_Time_0,BBI_Fit_27)
plot(BBI_Time_0,BBI_Fit_01)
xlim([-0.1,BBI_Time_01_5T])
ylim([-10,110])
xlabel('Time (s)')
ylabel('Temperature (^oC)')
text(0.4,60,strcat('s_{yx} = ', num2str(S_yx_BBI_01), 'K',' For Gamma
 = (0,1)'))
text(0.4,50,strcat('s_{yx} = ', num2str(S_yx_BBI_27), 'K',' For Gamma
 = (0.2,0.7)'))
ax = gca;
ax.YAxisLocation = 'origin';
title('Bare Thermocouple with Different Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0.2:0.7', 'Best Fit - Gamma
 = 0:1', 'Location', 'northeast')
grid minor

% Calculation for Aluminum
AIB_Temp_Smooth_01 = AIB_Temp_Smooth(GammaAIB_1:GammaAIB_0);
ABI_Temp_Smooth_01 = ABI_Temp_Smooth(GammaABI_1:GammaABI_0);
AIB_Temp_Smooth_27 = AIB_Temp_Smooth(GammaAIB_7:GammaAIB_2);
ABI_Temp_Smooth_27 = ABI_Temp_Smooth(GammaABI_7:GammaABI_2);

AIB_01_Cons = real(1/CoAIB_01);
ABI_01_Cons = real(1/CoABI_01);
AIB_27_Cons = real(1/CoAIB_27);
ABI_27_Cons = real(1/CoABI_27);

for i = 1:length(AIB_Time_01)
    if AIB_Time_01(i) > 5*AIB_01_Cons
        AIB_Time_01_5T = i;
        break
    end
end
```

```
for i = 1:length(ABI_Time_01)
    if ABI_Time_01(i) > 5*ABI_01_Cons
        ABI_Time_01_5T = i;
        break
    end
end


for i = 1:length(AIB_Time_27)
    if i > 5*AIB_27_Cons
        AIB_Time_27_5T = i;
        break
    end
end

for i = 1:length(ABI_Time_27)
    if i > 5*ABI_27_Cons
        ABI_Time_27_5T = i;
        break
    end
end

AIB_Fit_01 = AIB_Temp_Smooth(startfitAIB) + (AIB_Temp_Final-
AIB_Temp_Smooth(startfitAIB))*(1-exp(AIB_Time/AIB_01_Cons));
ABI_Fit_01 = ABI_Temp_Smooth(startfitABI) + (ABI_Temp_Final-
ABI_Temp_Smooth(startfitABI))*(1-exp(ABI_Time/ABI_01_Cons));
AIB_Fit_27 = AIB_Temp_Smooth(startfitAIB) + (AIB_Temp_Final-
AIB_Temp_Smooth(startfitAIB))*(1-exp(AIB_Time/AIB_27_Cons));
ABI_Fit_27 = ABI_Temp_Smooth(startfitABI) + (ABI_Temp_Final-
ABI_Temp_Smooth(startfitABI))*(1-exp(ABI_Time/ABI_27_Cons));

% Finding when time = 0
AIB_Fit_01 = AIB_Fit_01(startfitAIB:end);
ABI_Fit_01 = ABI_Fit_01(startfitABI:end);
AIB_Fit_27 = AIB_Fit_27(startfitAIB:end);
ABI_Fit_27 = ABI_Fit_27(startfitABI:end);

AIB_Time_0 = AIB_Time(startfitAIB:end);
ABI_Time_0 = ABI_Time(startfitABI:end);

AIB_01_Cons = -real(1/CoAIB_01);
ABI_01_Cons = -real(1/CoABI_01);
AIB_27_Cons = -real(1/CoAIB_27);
ABI_27_Cons = -real(1/CoABI_27);

for i = 1:length(AIB_Time_01)
    if AIB_Time_01(i) > 5*AIB_01_Cons
        AIB_Time_01_5T = AIB_Time_01(i);
        break
    end
end

for i = 1:length(ABI_Time_01)
```

```
    if ABI_Time_01(i) > 5*ABI_01_Cons
        ABI_Time_01_5T = ABI_Time_01(i);
        break
    end
end


for i = 1:length(AIB_Time_27)
    if AIB_Time_27(i) > 5*AIB_27_Cons
        AIB_Time_27_5T = AIB_Time_27(i);
        break
    end
end

for i = 1:length(ABI_Time_27)
    if ABI_Time_27(i) > 5*ABI_27_Cons
        ABI_Time_27_5T = ABI_Time_27(i);
        break
    end
end

AIB_Temp_Smooth_s = AIB_Temp_Smooth(startfitAIB:end);
S_yx_AIB_01 = 0;
for i = 1:length(AIB_Temp_Smooth_s)
    S_yx_AIB_01 = (AIB_Temp_Smooth_s(i)-AIB_Fit_01(i))^2 +
 S_yx_AIB_01;
end
S_yx_AIB_01 = sqrt(S_yx_AIB_01/length(AIB_Temp_Smooth_s));

S_yx_AIB_27 = 0;
for i = 1:length(AIB_Temp_Smooth_s)
    S_yx_AIB_27 = (AIB_Temp_Smooth_s(i)-AIB_Fit_27(i))^2 +
 S_yx_AIB_27;
end
S_yx_AIB_27 = sqrt(S_yx_AIB_27/length(AIB_Temp_Smooth_s));


ABI_Temp_Smooth_s = ABI_Temp_Smooth(startfitABI:end);

S_yx_ABI_01 = 0;
for i = 1:length(ABI_Temp_Smooth_s)
    S_yx_ABI_01 = (ABI_Temp_Smooth_s(i)-ABI_Fit_01(i))^2 +
 S_yx_ABI_01;
end
S_yx_ABI_01 = sqrt(S_yx_ABI_01/length(ABI_Temp_Smooth_s));

S_yx_ABI_27 = 0;
for i = 1:length(ABI_Temp_Smooth_s)
    S_yx_ABI_27 = (ABI_Temp_Smooth_s(i)-ABI_Fit_27(i))^2 +
 S_yx_ABI_27;
end
S_yx_ABI_27 = sqrt(S_yx_ABI_27/length(ABI_Temp_Smooth_s));

% Plotting
```

```
figure(18)
plot(AIB_Time,AIB_Temp_Smooth)
hold on
plot(AIB_Time_0,AIB_Fit_27)
plot(AIB_Time_0,AIB_Fit_01)
ylim([-10,110])
xlabel('Time (s)')
ylabel('Temperature (^oC)')
text(10,60,strcat('s_{yx} = ', num2str(S_yx_AIB_01), 'K',' For Gamma =
 (0,1)'))
text(10,50,strcat('s_{yx} = ', num2str(S_yx_AIB_27), 'K',' For Gamma =
 (0.2,0.7)'))
ax = gca;
ax.YAxisLocation = 'origin';
title('Aluminum-Insulated Thermocouple with Different Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0.2:0.7', 'Best Fit - Gamma
 = 0:1', 'Location', 'southeast')
grid minor

figure(19)
plot(ABI_Time,ABI_Temp_Smooth)
hold on
plot(ABI_Time_0,ABI_Fit_27)
plot(ABI_Time_0,ABI_Fit_01)
ylim([-10,110])
xlabel('Time (s)')
ylabel('Temperature (^oC)')
text(10,60,strcat('s_{yx} = ', num2str(S_yx_ABI_01), 'K',' For Gamma =
 (0,1)'))
text(10,50,strcat('s_{yx} = ', num2str(S_yx_ABI_27), 'K',' For Gamma =
 (0.2,0.7)'))
ax = gca;
ax.YAxisLocation = 'origin';
title('Aluminum-Insulated Thermocouple with Different Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0.2:0.7', 'Best Fit - Gamma
 = 0:1', 'Location', 'northeast')
grid minor

% Calculation for Steel
SIB_Temp_Smooth_01 = SIB_Temp_Smooth(GammaSIB_1:GammaSIB_0);
SBI_Temp_Smooth_01 = SBI_Temp_Smooth(GammaSBI_1:GammaSBI_0);
SIB_Temp_Smooth_27 = SIB_Temp_Smooth(GammaSIB_7:GammaSIB_2);
SBI_Temp_Smooth_27 = SBI_Temp_Smooth(GammaSBI_7:GammaSBI_2);

SIB_01_Cons = real(1/CoSIB_01);
SBI_01_Cons = real(1/CoSBI_01);
SIB_27_Cons = real(1/CoSIB_27);
SBI_27_Cons = real(1/CoSBI_27);

for i = 1:length(SIB_Time_01)
    if SIB_Time_01(i) > 5*SIB_01_Cons
        SIB_Time_01_5T = i;
        break
```

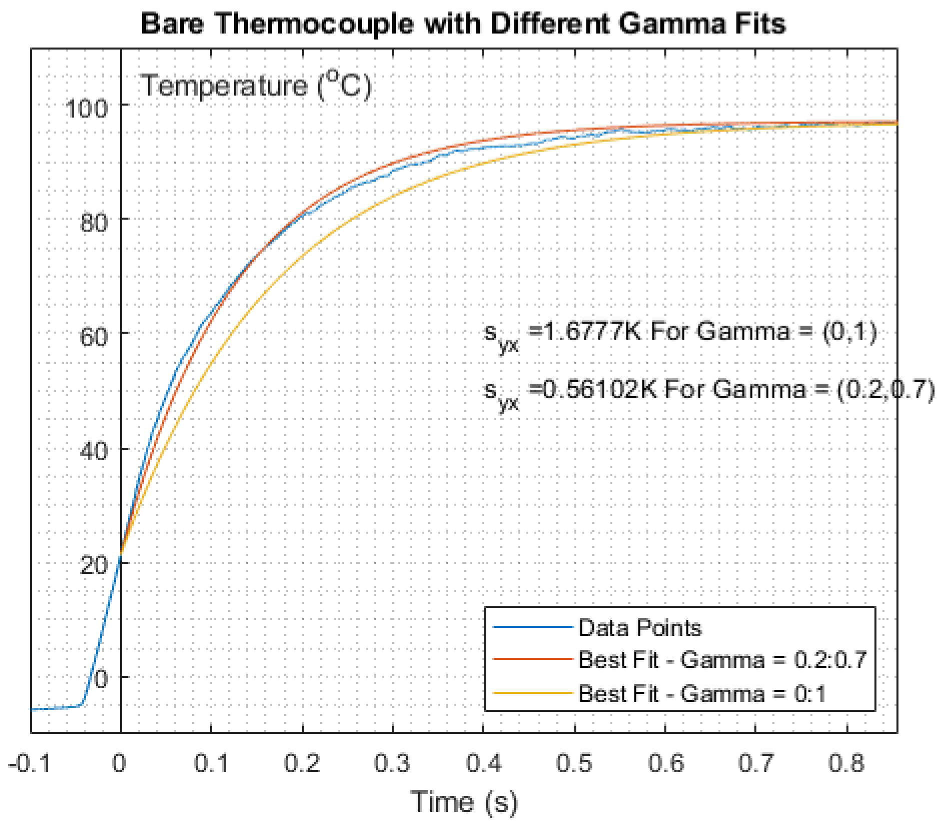
```
    end
end

for i = 1:length(SBI_Time_01)
    if SBI_Time_01(i) > 5*SBI_01_Cons
        SBI_Time_01_5T = i;
        break
    end
end

for i = 1:length(SIB_Time_27)
    if i > 5*SIB_27_Cons
        SIB_Time_27_5T = i;
        break
    end
end

for i = 1:length(SBI_Time_27)
    if i > 5*SBI_27_Cons
        SBI_Time_27_5T = i;
        break
    end
end

SIB_Fit_01 = SIB_Temp_Smooth(startfitSIB) + (SIB_Temp_Final-
SIB_Temp_Smooth(startfitSIB))*(1-exp(SIB_Time/SIB_01_Cons));
SBI_Fit_01 = SBI_Temp_Smooth(startfitSBI) + (SBI_Temp_Final-
SBI_Temp_Smooth(startfitSBI))*(1-exp(SBI_Time/SBI_01_Cons));
SIB_Fit_27 = SIB_Temp_Smooth(startfitSIB) + (SIB_Temp_Final-
SIB_Temp_Smooth(startfitSIB))*(1-exp(SIB_Time/SIB_27_Cons));
SBI_Fit_27 = SBI_Temp_Smooth(startfitSBI) + (SBI_Temp_Final-
SBI_Temp_Smooth(startfitSBI))*(1-exp(SBI_Time/SBI_27_Cons));

% Finding when time = 0
SIB_Fit_01 = SIB_Fit_01(startfitSIB:end);
SBI_Fit_01 = SBI_Fit_01(startfitSBI:end);
SIB_Fit_27 = SIB_Fit_27(startfitSIB:end);
SBI_Fit_27 = SBI_Fit_27(startfitSBI:end);

SIB_Time_0 = SIB_Time(startfitSIB:end);
SBI_Time_0 = SBI_Time(startfitSBI:end);

SIB_01_Cons = -real(1/CoSIB_01);
SBI_01_Cons = -real(1/CoSBI_01);
SIB_27_Cons = -real(1/CoSIB_27);
SBI_27_Cons = -real(1/CoSBI_27);

for i = 1:length(SIB_Time_01)
    if SIB_Time_01(i) > 5*SIB_01_Cons
        SIB_Time_01_5T = SIB_Time_01(i);
        break
    end
end
```

```
for i = 1:length(SBI_Time_01)
    if SBI_Time_01(i) > 5*SBI_01_Cons
        SBI_Time_01_5T = SBI_Time_01(i);
        break
    end
end

for i = 1:length(SIB_Time_27)
    if SIB_Time_27(i) > 5*SIB_27_Cons
        SIB_Time_27_5T = SIB_Time_27(i);
        break
    end
end

for i = 1:length(SBI_Time_27)
    if SBI_Time_27(i) > 5*SBI_27_Cons
        SBI_Time_27_5T = SBI_Time_27(i);
        break
    end
end

SIB_Temp_Smooth_s = SIB_Temp_Smooth(startfitSIB:end);

S_yx_SIB_01 = 0;
for i = 1:length(SIB_Temp_Smooth_s)
    S_yx_SIB_01 = (SIB_Temp_Smooth_s(i)-SIB_Fit_01(i))^2 +
 S_yx_SIB_01;
end
S_yx_SIB_01 = sqrt(S_yx_SIB_01/length(SIB_Temp_Smooth_s));

S_yx_SIB_27 = 0;
for i = 1:length(SIB_Temp_Smooth_s)
    S_yx_SIB_27 = (SIB_Temp_Smooth_s(i)-SIB_Fit_27(i))^2 +
 S_yx_SIB_27;
end
S_yx_SIB_27 = sqrt(S_yx_SIB_27/length(SIB_Temp_Smooth_s));


SBI_Temp_Smooth_s = SBI_Temp_Smooth(startfitSBI:end);

S_yx_SBI_01 = 0;
for i = 1:length(SBI_Temp_Smooth_s)
    S_yx_SBI_01 = (SBI_Temp_Smooth_s(i)-SBI_Fit_01(i))^2 +
 S_yx_SBI_01;
end
S_yx_SBI_01 = sqrt(S_yx_SBI_01/length(SBI_Temp_Smooth_s));

S_yx_SBI_27 = 0;
for i = 1:length(SBI_Temp_Smooth_s)
    S_yx_SBI_27 = (SBI_Temp_Smooth_s(i)-SBI_Fit_27(i))^2 +
 S_yx_SBI_27;
end
S_yx_SBI_27 = sqrt(S_yx_SBI_27/length(SBI_Temp_Smooth_s));
```

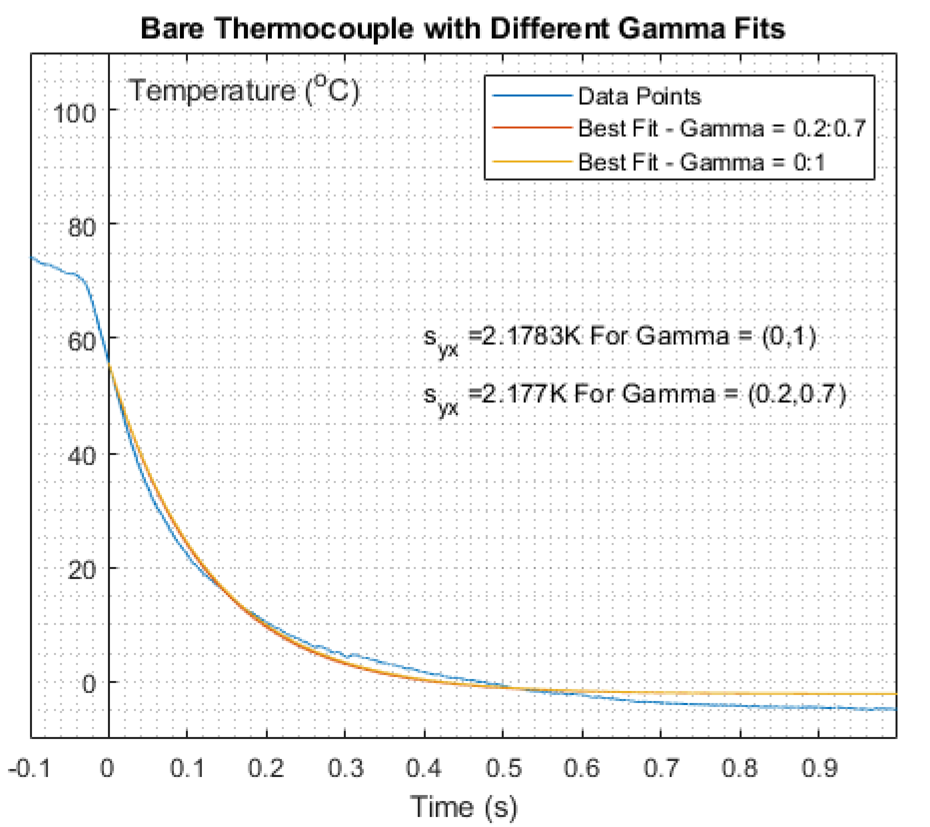
```
% Plotting

figure(20)
plot(SIB_Time,SIB_Temp_Smooth)
hold on
plot(SIB_Time_0,SIB_Fit_27)
plot(SIB_Time_0,SIB_Fit_01)
ylim([-10,110])
xlabel('Time (s)')
ylabel('Temperature (^oC)')
text(10,60,strcat('s_{yx} = ', num2str(S_yx_SIB_01), 'K',' For Gamma =
 (0,1)'))
text(10,50,strcat('s_{yx} = ', num2str(S_yx_SIB_27), 'K',' For Gamma =
 (0.2,0.7)'))
ax = gca;
ax.YAxisLocation = 'origin';
title('Steel-Insulated Thermocouple with Different Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0.2:0.7', 'Best Fit - Gamma
 = 0:1', 'Location', 'southeast')
grid minor

figure(21)
plot(SBI_Time,SBI_Temp_Smooth)
hold on
plot(SBI_Time_0,SBI_Fit_27)
plot(SBI_Time_0,SBI_Fit_01)
ylim([-10,110])
xlabel('Time (s)')
ylabel('Temperature (^oC)')
text(10,60,strcat('s_{yx} = ', num2str(S_yx_SBI_01), 'K',' For Gamma =
 (0,1)'))
text(10,50,strcat('s_{yx} = ', num2str(S_yx_SBI_27), 'K',' For Gamma =
 (0.2,0.7)'))
ax = gca;
ax.YAxisLocation = 'origin';
title('Steel-Insulated Thermocouple with Different Gamma Fits')
legend('Data Points', 'Best Fit - Gamma = 0.2:0.7', 'Best Fit - Gamma
 = 0:1', 'Location', 'northeast')
grid minor

% Residual Calculations
Residual_BIB_01 = BIB_Temp_Smooth(startfitBIB:end) - BIB_Fit_01;
Residual_BIB_27 = BIB_Temp_Smooth(startfitBIB:end) - BIB_Fit_27;
Residual_BBI_01 = BBI_Temp_Smooth(startfitBBI:end) - BBI_Fit_01;
Residual_BBI_27 = BBI_Temp_Smooth(startfitBBI:end) - BBI_Fit_27;

Residual_AIB_01 = AIB_Temp_Smooth(startfitAIB:end) - AIB_Fit_01;
Residual_AIB_27 = AIB_Temp_Smooth(startfitAIB:end) - AIB_Fit_27;
Residual_ABI_01 = ABI_Temp_Smooth(startfitABI:end) - ABI_Fit_01;
Residual_ABI_27 = ABI_Temp_Smooth(startfitABI:end) - ABI_Fit_27;

Residual_SIB_01 = SIB_Temp_Smooth(startfitSIB:end) - SIB_Fit_01;
Residual_SIB_27 = SIB_Temp_Smooth(startfitSIB:end) - SIB_Fit_27;
Residual_SBI_01 = SBI_Temp_Smooth(startfitSBI:end) - SBI_Fit_01;
```

```
Residual_SBI_27 = SBI_Temp_Smooth(startfitSBI:end) - SBI_Fit_27;

% Plotting the figures for Residuals

figure(22)
plot(BIB_Time_0,Residual_BIB_01)
hold on
plot(BIB_Time_0,Residual_BIB_27)
xlabel('Time (s)')
ylabel('Residuals (^oC)')
text(3,5,strcat('s_{yx} = ', num2str(S_yx_BIB_01), 'K',' For Gamma =
 (0,1)'))
text(3,4,strcat('s_{yx} = ', num2str(S_yx_BIB_27), 'K',' For Gamma =
 (0.2,0.7)'))
title('Residuals for Bare Thermocouple with Different Gamma Fits')
legend('Residuals to Actual Data - Gamma = 0:1', 'Residuals to Actual
 Data - Gamma = 0.2:0.7', 'Location', 'northeast')
grid minor

figure(23)
plot(BBI_Time_0,Residual_BBI_01)
hold on
plot(BBI_Time_0,Residual_BBI_27)
xlabel('Time (s)')
ylabel('Residuals (^oC)')
text(1.5,1,strcat('s_{yx} = ', num2str(S_yx_BBI_01), 'K',' For Gamma =
 (0,1)'))
text(1.5,0,strcat('s_{yx} = ', num2str(S_yx_BBI_27), 'K',' For Gamma =
 (0.2,0.7)'))
title('Residuals for Bare Thermocouple with Different Gamma Fits')
legend('Residuals to Actual Data - Gamma = 0:1', 'Residuals to Actual
 Data - Gamma = 0.2:0.7', 'Location', 'northeast')
grid minor

figure(24)
plot(AIB_Time_0,Residual_AIB_01)
hold on
plot(AIB_Time_0,Residual_AIB_27)
xlabel('Time (s)')
ylabel('Residuals (^oC)')
text(15,-2,strcat('s_{yx} = ', num2str(S_yx_AIB_01), 'K',' For Gamma =
 (0,1)'))
text(15,-3,strcat('s_{yx} = ', num2str(S_yx_AIB_27), 'K',' For Gamma =
 (0.2,0.7)'))
title('Residuals for Aluminum-Insulated Thermocouple with Different
 Gamma Fits')
legend('Residuals to Actual Data - Gamma = 0:1', 'Residuals to Actual
 Data - Gamma = 0.2:0.7', 'Location', 'southeast')
grid minor

figure(25)
plot(ABI_Time_0,Residual_ABI_01)
hold on
plot(ABI_Time_0,Residual_ABI_27)
```

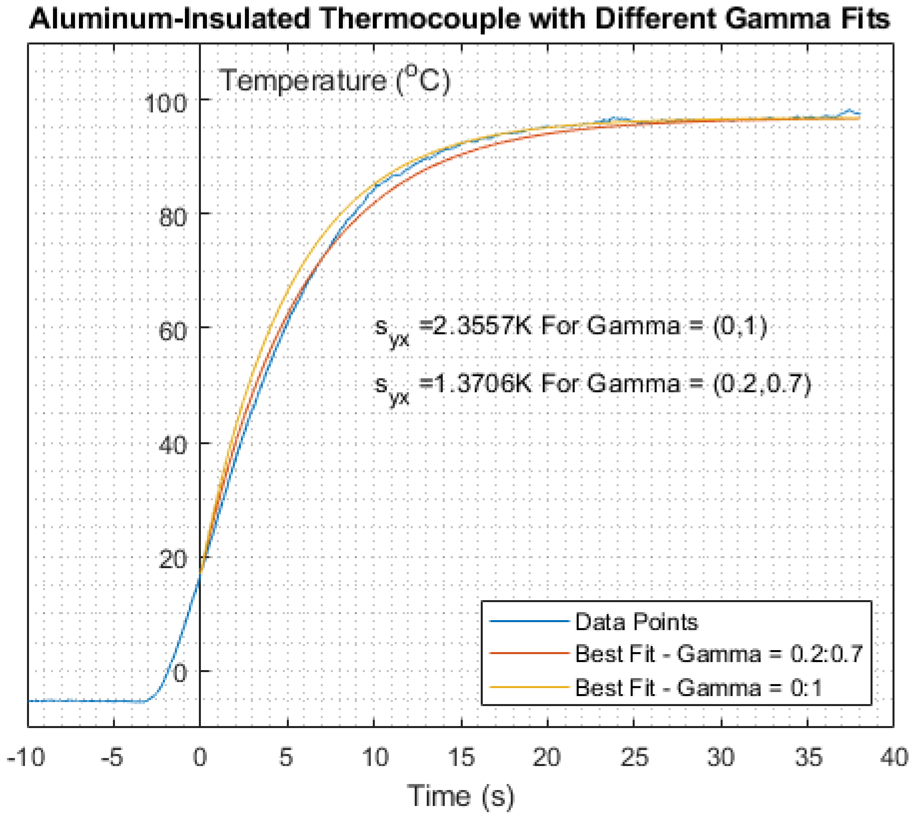
```
xlabel('Time (s)')
ylabel('Residuals (^oC)')
text(15,-1,strcat('s_{yx} = ', num2str(S_yx_ABI_01), 'K',' For Gamma =
 (0,1)'))
text(15,-2,strcat('s_{yx} = ', num2str(S_yx_ABI_27), 'K',' For Gamma =
 (0.2,0.7)'))
title('Residuals for Aluminum-Insulated Thermocouple with Different
 Gamma Fits')
legend('Residuals to Actual Data - Gamma = 0:1', 'Residuals to Actual
 Data - Gamma = 0.2:0.7', 'Location', 'northeast')
grid minor

figure(26)
plot(SIB_Time_0,Residual_SIB_01)
hold on
plot(SIB_Time_0,Residual_SIB_27)
xlabel('Time (s)')
ylabel('Residuals (^oC)')
text(15,-2,strcat('s_{yx} = ', num2str(S_yx_SIB_01), 'K',' For Gamma =
 (0,1)'))
text(15,-3,strcat('s_{yx} = ', num2str(S_yx_SIB_27), 'K',' For Gamma =
 (0.2,0.7)'))
title('Residuals for Steel-Insulated Thermocouple with Different Gamma
 Fits')
legend('Residuals to Actual Data - Gamma = 0:1', 'Residuals to Actual
 Data - Gamma = 0.2:0.7' , 'Location', 'northeast')
grid minor

figure(27)
plot(SBI_Time_0,Residual_SBI_01)
hold on
plot(SBI_Time_0,Residual_SBI_27)
xlabel('Time (s)')
ylabel('Residuals (^oC)')
text(15,-4,strcat('s_{yx} = ', num2str(S_yx_SBI_01), 'K',' For Gamma =
 (0,1)'))
text(15,-6,strcat('s_{yx} = ', num2str(S_yx_SBI_27), 'K',' For Gamma =
 (0.2,0.7)'))
title('Residuals for Steel-Insulated Thermocouple with Different Gamma
 Fits')
legend('Residuals to Actual Data - Gamma = 0:1', 'Residuals to Actual
 Data - Gamma = 0.2:0.7', 'Location', 'southeast')
grid minor
```



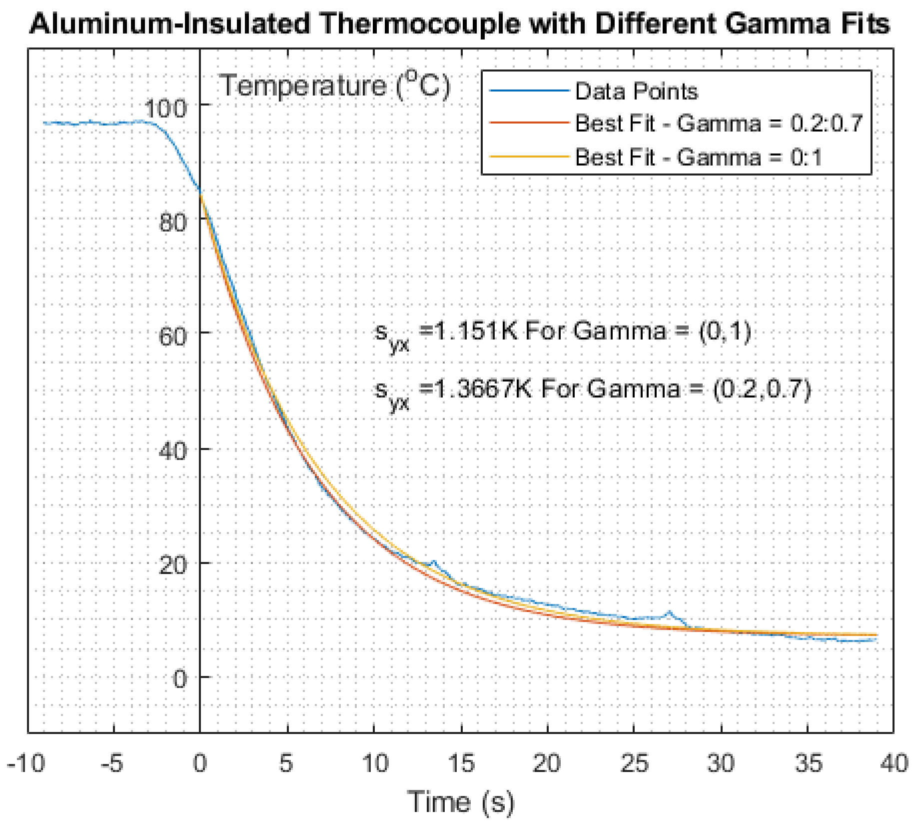
Bare Thermocouple with Different Gamma Fits
Temperature (^oC)
s_{yx} =1.6777K For Gamma = (0,1)
s_{yx} =0.56102K For Gamma = (0.2,0.7)
Data Points
Best Fit - Gamma = 0.2:0.7
Best Fit - Gamma = 0:1
Time (s)



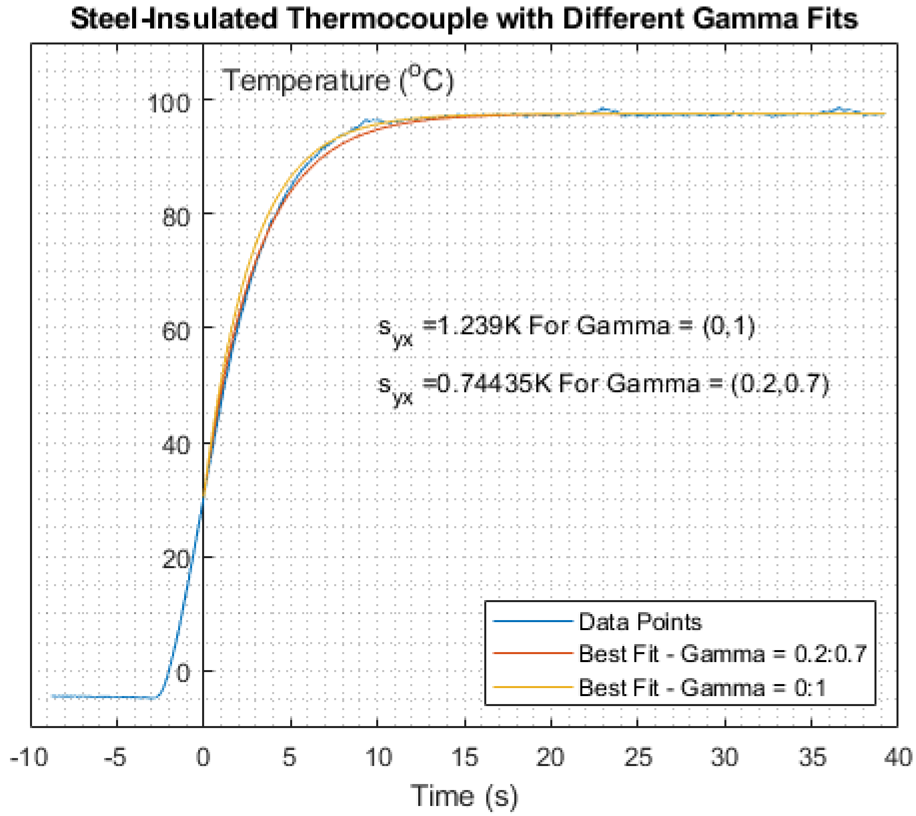
Bare Thermocouple with Different Gamma Fits
Temperature (^oC)
Data Points
Best Fit - Gamma = 0.2:0.7
Best Fit - Gamma = 0:1
s_{yx} =2.1783K For Gamma = (0,1)
s_{yx} =2.177K For Gamma = (0.2,0.7)
Time (s)



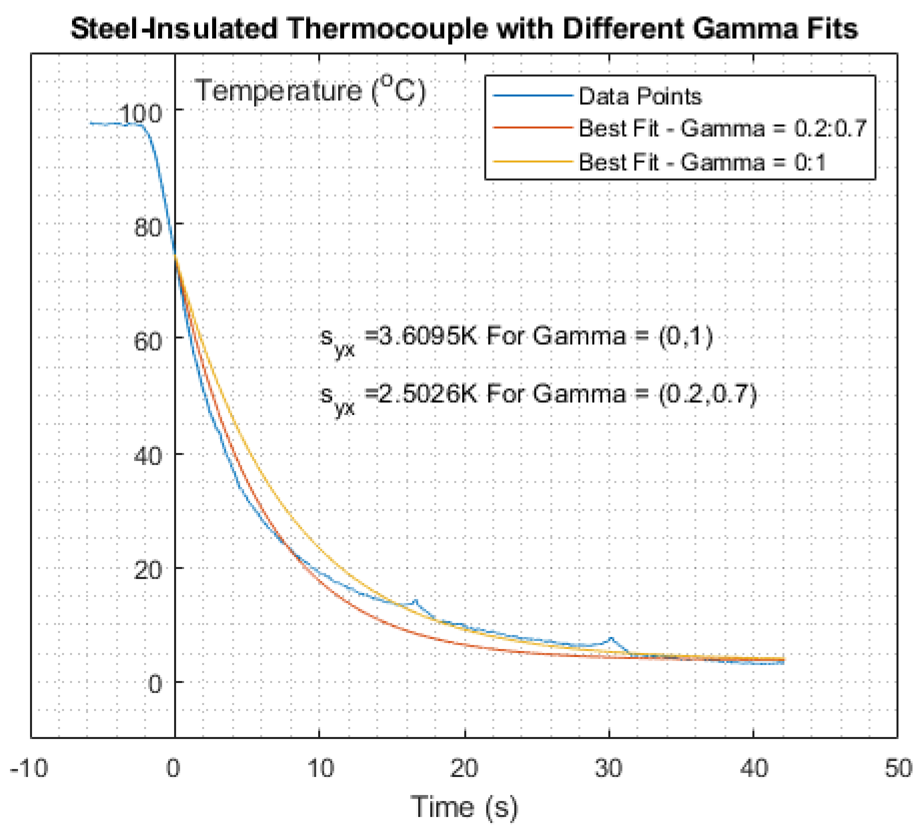
Aluminum-Insulated Thermocouple with Different Gamma Fits
Temperature (°C)
100
80
60
40
20
0
-10
-5
0
5
10
15
20
25
30
35
40
Time (s)
s_yx =2.3557K For Gamma = (0,1)
s_yx =1.3706K For Gamma = (0.2,0.7)
Data Points
Best Fit - Gamma = 0.2:0.7
Best Fit - Gamma = 0:1



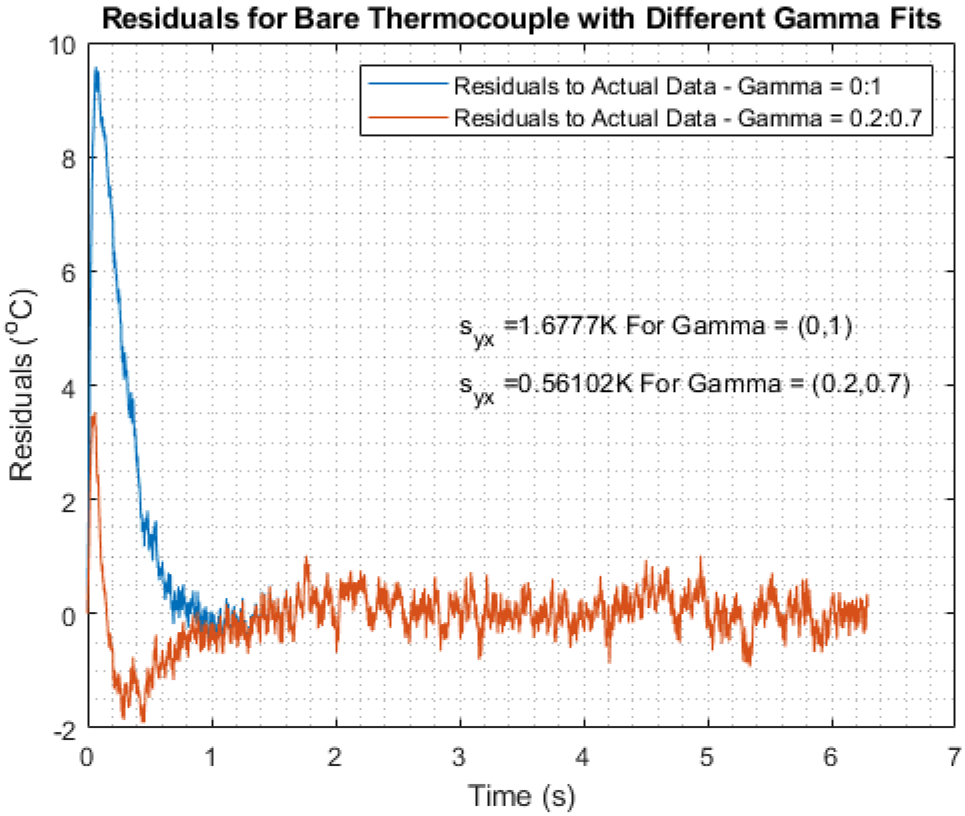
Aluminum-Insulated Thermocouple with Different Gamma Fits
Temperature (°C)
100
80
60
40
20
0
-10
-5
0
5
10
15
20
25
30
35
40
Time (s)
Data Points
Best Fit - Gamma = 0.2:0.7
Best Fit - Gamma = 0:1
s_yx =1.151K For Gamma = (0,1)
s_yx =1.3667K For Gamma = (0.2,0.7)

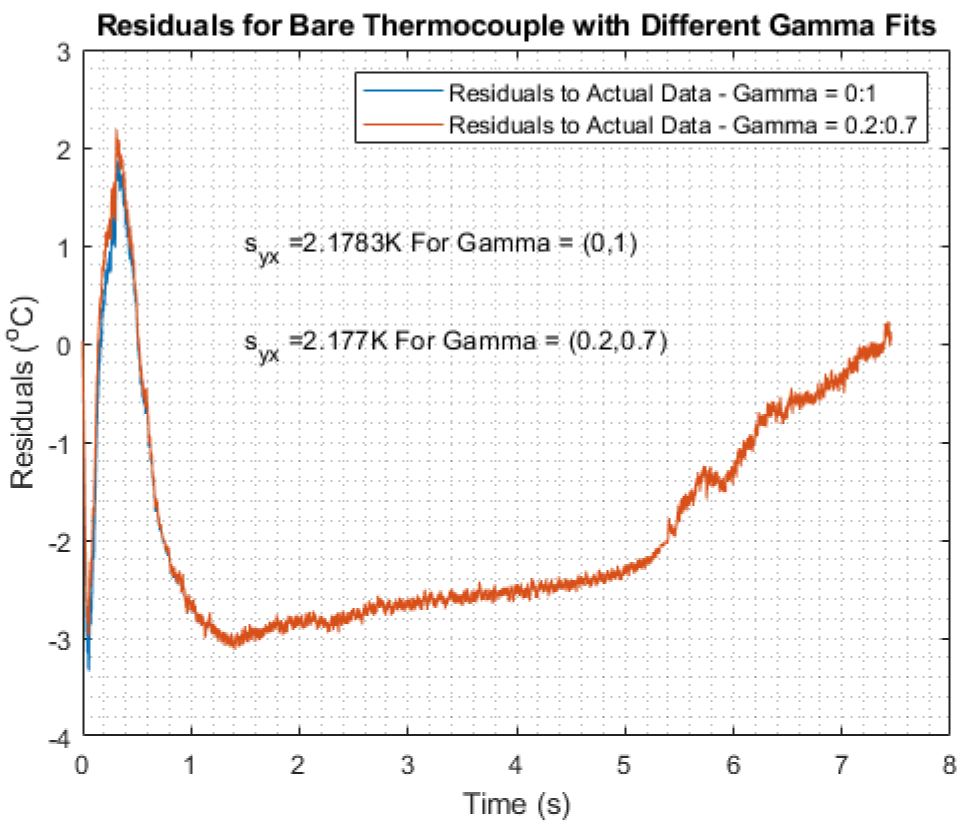


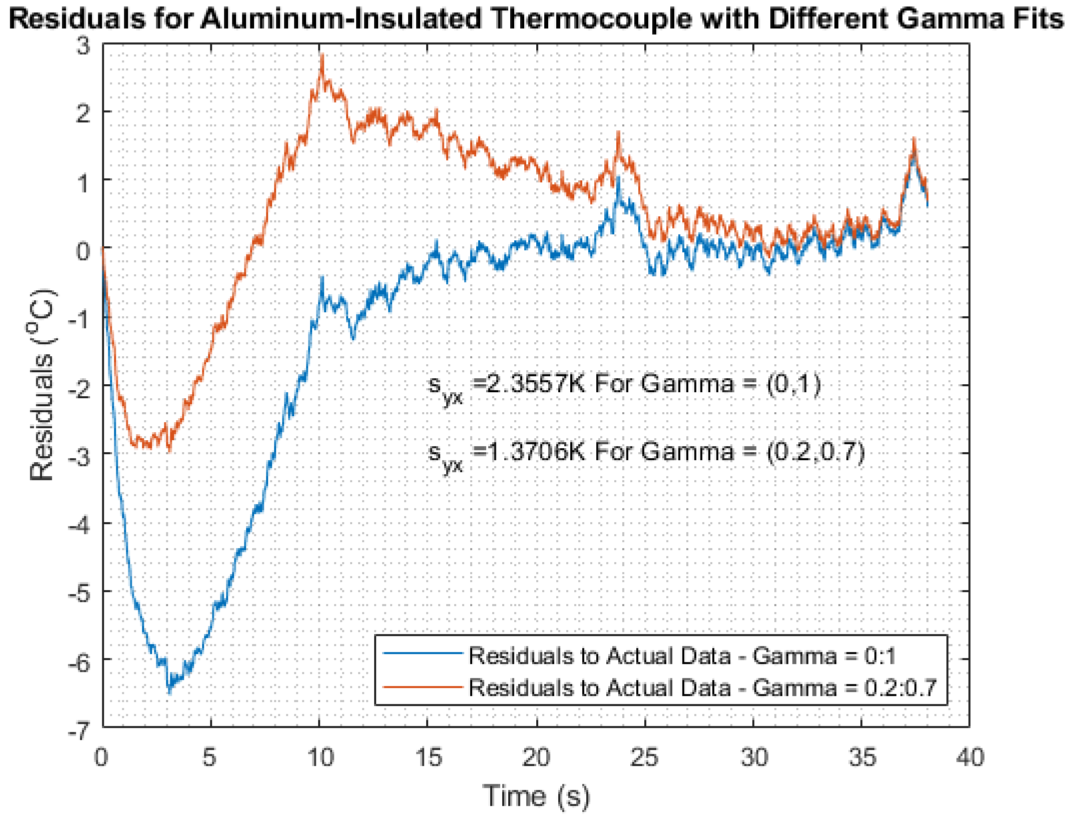
Steel-Insulated Thermocouple with Different Gamma Fits
Temperature (°C)
100
80
60
40
20
0
s_yx =1.239K For Gamma = (0,1)
s_yx =0.74435K For Gamma = (0.2,0.7)
Data Points
Best Fit - Gamma = 0.2:0.7
Best Fit - Gamma = 0:1
-10
-5
0
5
10
15
20
25
30
35
40
Time (s)

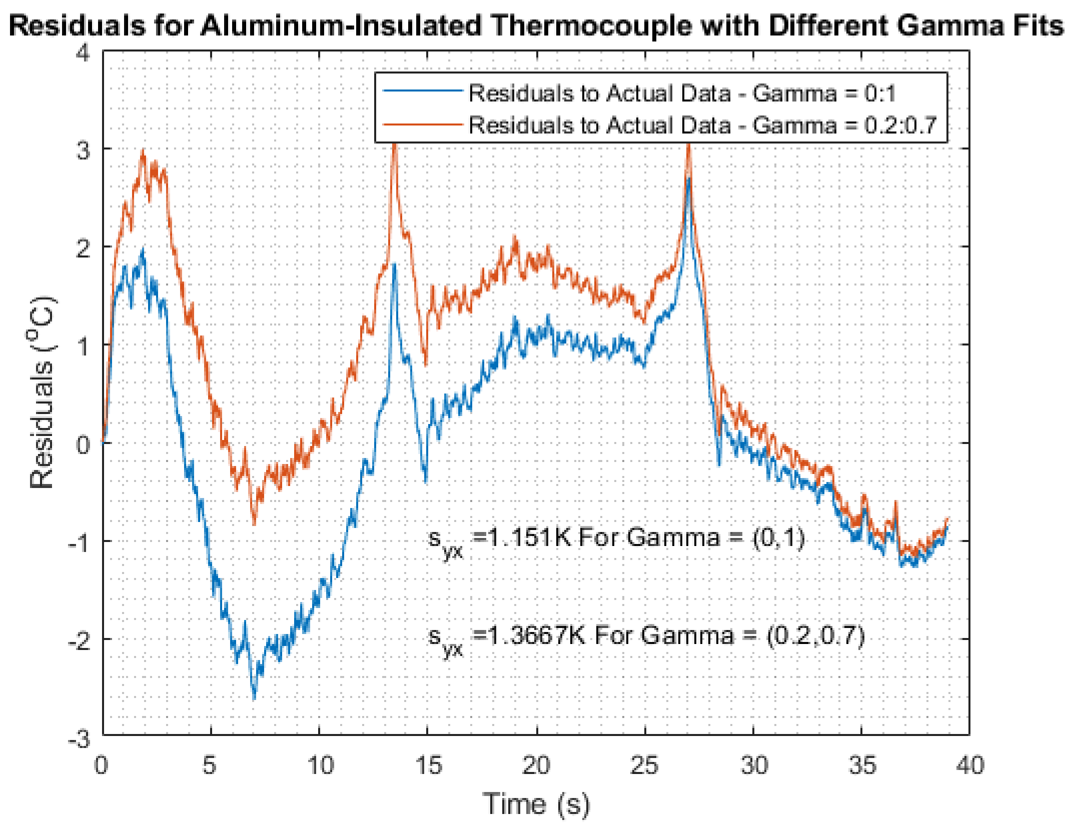


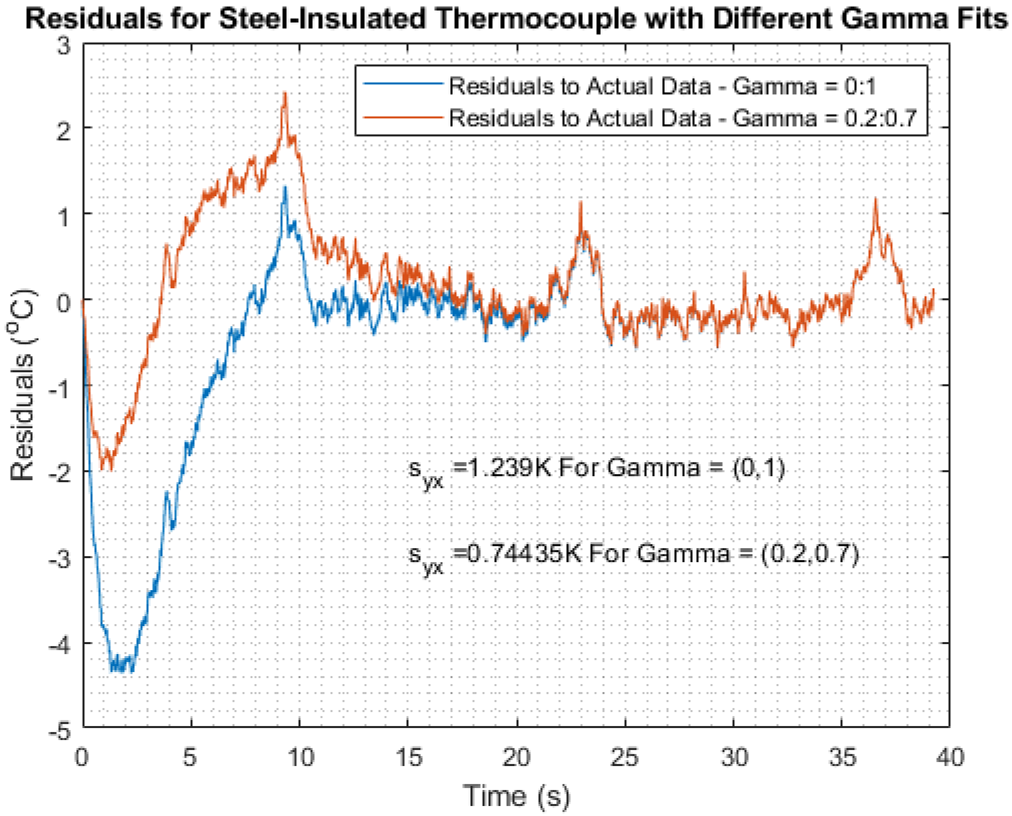
Steel-Insulated Thermocouple with Different Gamma Fits
Temperature (°C)
100
80
60
40
20
0
Data Points
Best Fit - Gamma = 0.2:0.7
Best Fit - Gamma = 0:1
s_yx =3.6095K For Gamma = (0,1)
s_yx =2.5026K For Gamma = (0.2,0.7)
-10
0
10
20
30
40
50
Time (s)


Residuals for Bare Thermocouple with Different Gamma Fits
Residuals to Actual Data - Gamma = 0:1
Residuals to Actual Data - Gamma = 0.2:0.7
s_{yx} =1.6777K For Gamma = (0,1)
s_{yx} =0.56102K For Gamma = (0.2,0.7)
Residuals (°C)
Time (s)

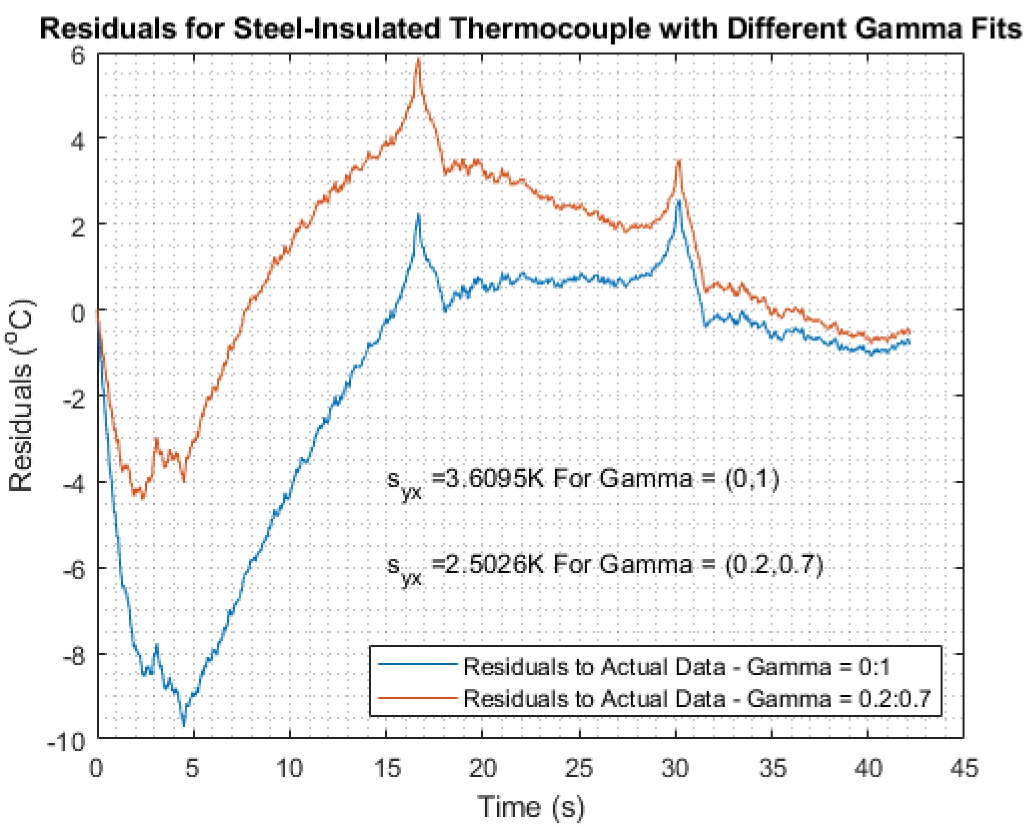

Residuals for Bare Thermocouple with Different Gamma Fits
Residuals to Actual Data - Gamma = 0:1
Residuals to Actual Data - Gamma = 0.2:0.7
s_{yx} =2.1783K For Gamma = (0,1)
s_{yx} =2.177K For Gamma = (0.2,0.7)
Residuals (°C)
Time (s)


Residuals for Aluminum-Insulated Thermocouple with Different Gamma Fits
Residuals (°C)
Time (s)
s_yx =2.3557K For Gamma = (0,1)
s_yx =1.3706K For Gamma = (0.2,0.7)
Residuals to Actual Data - Gamma = 0:1
Residuals to Actual Data - Gamma = 0.2:0.7


Residuals for Aluminum-Insulated Thermocouple with Different Gamma Fits
Residuals (°C)
Time (s)
Residuals to Actual Data - Gamma = 0:1
Residuals to Actual Data - Gamma = 0.2:0.7
s_yx =1.151K For Gamma = (0,1)
s_yx =1.3667K For Gamma = (0.2,0.7)



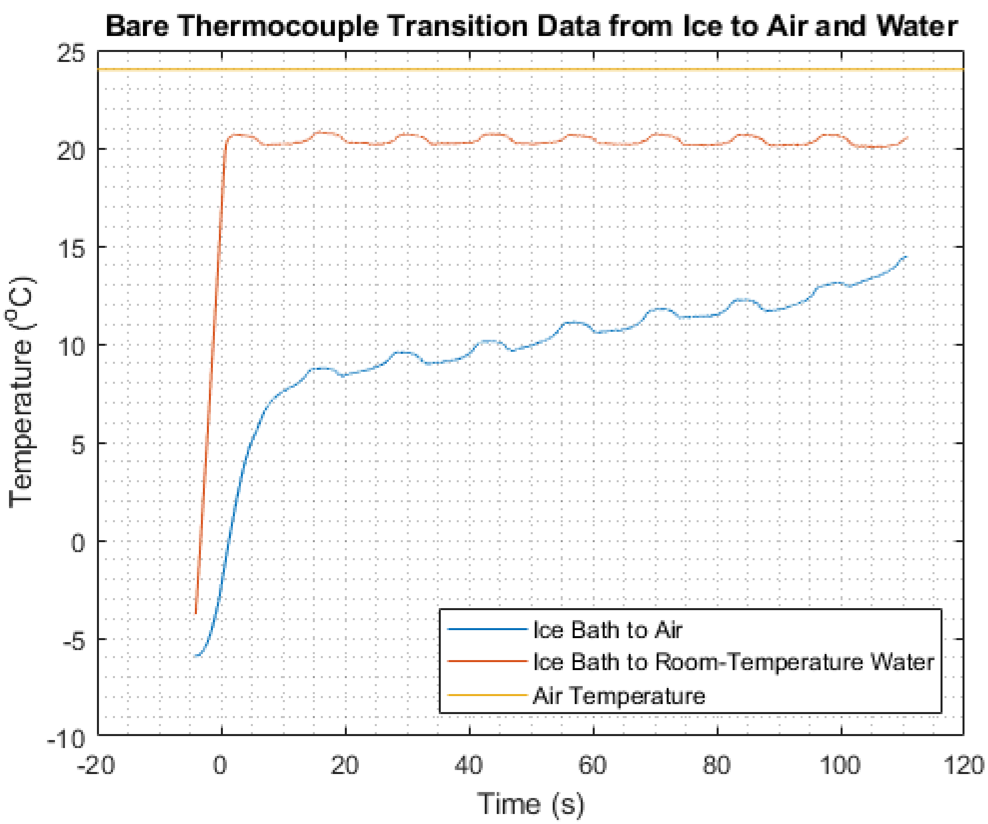
Residuals for Steel-Insulated Thermocouple with Different Gamma Fits
Residuals to Actual Data - Gamma = 0:1
Residuals to Actual Data - Gamma = 0.2:0.7
s_{yx} =1.239K For Gamma = (0,1)
s_{yx} =0.74435K For Gamma = (0.2,0.7)
Residuals (°C)
Time (s)



Residuals for Steel-Insulated Thermocouple with Different Gamma Fits
s_{yx} =3.6095K For Gamma = (0,1)
s_{yx} =2.5026K For Gamma = (0.2,0.7)
Residuals to Actual Data - Gamma = 0:1
Residuals to Actual Data - Gamma = 0.2:0.7
Residuals (°C)
Time (s)

Part 4

```
% Plotting for Water and Air - Temperature vs. Time
figure(28)
plot(BIA_Time ,BIA_Temp_Smooth)
hold on
plot(BIW_Time,BIW_Temp_Smooth)
plot(linspace(-20,120,100),24.*ones(100,1))
xlabel('Time (s)')
ylabel('Temperature (^oC)')
title('Bare Thermocouple Transition Data from Ice to Air and Water')
legend('Ice Bath to Air', 'Ice Bath to Room-Temperature Water', 'Air
 Temperature' ,'Location', 'southeast')
grid minor
```



Part 5 - Creating Array of Syx and Tau

```
Table_Tau_01 =
 [BIB_01_Cons,BBI_01_Cons,AIB_01_Cons,ABI_01_Cons,SIB_01_Cons,SBI_01_Cons];
Table_Tau_27 =
 [BIB_27_Cons,BBI_27_Cons,AIB_27_Cons,ABI_27_Cons,SIB_27_Cons,SBI_27_Cons];
Table_Syx_01 =
 [S_yx_BIB_01,S_yx_BBI_01,S_yx_AIB_01,S_yx_ABI_01,S_yx_SIB_01,S_yx_SBI_01];
Table_Syx_27 =
 [S_yx_BIB_27,S_yx_BBI_27,S_yx_AIB_27,S_yx_ABI_27,S_yx_SIB_27,S_yx_SBI_27];
```

```
Data_Table = [Table_Tau_01;Table_Tau_27;Table_Syx_01;Table_Syx_27];

xlswrite('Final_Data.xlsx',Data_Table)
```

Published with MATLAB® R2018a