Table of Contents


```
function [SSE] =
 M4Calibration_014_05(yL1,yH1,ts1,tau1,yData,time,heating)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132 Program Description
% This function produces an SSE value given a set of data and
 important parameters.
% A "set" contains time data and one column of temperature data.
%
% Function Call
%  [SSE] = M4Calibration_014_05(yL1,yH1,ts1,tau1,yData,time,heating)
%
% Input Arguments
% 1. yL1: minimum temperature for data
%   2. yH1: maximum temperature for data
%   3. ts1: time step for data
%   4. tau1: tau for data
%   5. yData: the temperature portion of the raw data
% 6. time: the time portion of the raw data
% 7. heating: a boolean variable that represents if the data is
 heating or cooling, 1 - heating, 0 - cooling
%
% Output Arguments
% 1. SSE: sum of squares of of error for data
%
% Assignment Information
% Assignment:         Final Project
%   Team ID:            014-05
%   Team Members:       Alex Pieprzycki, apieprzy@purdue.edu
%                       Colin Jamison, cjamison@purdue.edu
%                       Peter Swales, pswales@purdue.edu
%      Micah Huffman, mhuffma11@purdue.edu
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION ---

```
%Code modified from M2 Calibration by Colin Jamison and Micah

index = find(time >= ts1);  % Parsing data at time step
index = index(1);
```

CALCULATIONS ---

```
half2_t = time(index + 1:length(time));  % Time values proceeding the
 time step
half1_y = ones(1,index)';  % Initalizing vector of ones


if(heating)
 half1_y = yL1 * half1_y;  % Each time value up to the time step has
 the same temp. value
 half2_y = yL1 + (yH1 - yL1) * (1 - exp(-(half2_t - ts1) ./ tau1));
  % Second part of the piecewise function to calculate respective
 temperature values for time values after the time step
 y_new = [half1_y;half2_y];  % Recombining both parts of the piecewise
 function produces one 'smooth' string of data
 SSE = sum((yData - y_new).^2) / length(yData); %SSE for heating
else
 half1_y = yH1 * half1_y;  % Each time value up to the time step has
 the same temp. value
 half2_y = yL1 + (yH1 - yL1) * (exp(-(half2_t - ts1) ./ tau1));
  % Second part of the piecewise function to calculate respective
 temperature values for time values after the time step
 y_new = [half1_y;half2_y];  % Recombining both parts of the piecewise
 function produces one 'smooth' string of data
 SSE = sum((yData - y_new).^2) / length(yData); %SSE for cooling
end


ymin =

    0.5077


ymax =

  100.8127


tstep =

    1.3633
```

```
tau =

    0.2342


isHeating =

     0


SSE =

    1.7475
```

FORMATTED TEXT & FIGURE DISPLAYS ---

COMMAND WINDOW OUTPUTS ---

ACADEMIC INTEGRITY STATEMENT ---

I/We have not used source code obtained from any other unauthorized source, either modified or unmodified. Neither have I/we provided access to my/our code to another. The project I/we am/are submitting is my/our own original work.

Published with MATLAB® R2016a