```
%% j211filtfilt.m
%% Description
% j211filtfilt -- Zero-phase filtering method that conforms to SAE J211
%
% SEE: SAE J211-1 for more information
%
% ALSO SEE <j211_response.html |j211_response|> for frequency and phase plots of the
filters
%
% Note that for the sampling rate you are using, you will need to make sure
% that the filter meets the corridors specified in J211. We have found that
% for sampling rates below ~15,000 Hz, the original scale factor given
% (2.0775) is insufficient to meet the corridors. use j211_response.m to
% check filter response before use. This code is provided without warranty.
% Use at your own risk.
%
%% Usage
%  y = j211filtfilt(cfc,samplerate,x)
%% Input Arguments
% * *cfc* | |double| | _cfc_ is the filter class (can be 60, 108, 180, 600, or 1000
Hz)
% * *samplerate* | |double| | _samplerate_ is the sampling rate of the
% input data array (_x_)
% * *x* | |array| | _x_ is the input data array to be filtered
%% Output Arguments
% * *y* | |array| | _y_ is the filtered data
%%
%  ------------------------------------------------------------------
%   © Copyright 2009-2017                Jeffrey T. Somers
%    All Rights Reserved                    KBRwyle
%  ------------------------------------------------------------------
%  Revision Log:
%  Version  Date        Modified By     Reason
%  ------------------------------------------------------------------
%  2.0      21-MAY-09   J. Somers       Wrote to work with my scripts and to
%                                           make sure the filter falls within
%                                           the SAE corridor.  Added padding to
%                                           the beginning and end to minimize
%                                           startup effects
%  2.1      22-SEP-14   J. Somers       Added b and a outputs
%  2.1.1        03-JUN-17   J. Somers       Formatted for commercial crew
%
%% Code
function [y,b,a] = j211filtfilt(cfc,samplerate,x)
%%
% *Check Input Arguments*
if size(x,1) > size(x,2)
        x = x' ;
        flag = 1 ;
else
        flag = 0 ;
end
len = length(x) ;
%%
% *Setup filter specs*
T=1/samplerate; %sample period in seconds
% sf = 2.0775 ; % actual scale factor from J211. Smpf dependent

switch cfc
        case 60,
                sf = 2.2 ;
                if samplerate < 2700
                        error(['Filter Performance does not meet SAE J211
requirements ' ...
                                'for a ' num2str(samplerate) ' Hz'])
                end
        case 108,
                sf = 2.2 ;
                if samplerate < 5000
                        error(['Filter Performance does not meet SAE J211
requirements ' ...
                                'for a ' num2str(samplerate) ' Hz'])
                end
        case 180,
                sf = 2.2 ;
                if samplerate < 6200
                        error(['Filter Performance does not meet SAE J211
requirements ' ...
                                'for a ' num2str(samplerate) ' Hz'])
                end
        case 600,
                sf = 2.0 ;
                if samplerate < 7000
                        error(['Filter Performance does not meet SAE J211
requirements ' ...
                                'for a ' num2str(samplerate) ' Hz'])
                end
        case 1000,
                if samplerate < 9600
                        error(['Filter Performance does not meet SAE J211
requirements ' ...
                                'for a ' num2str(samplerate) ' Hz'])
                end
                if samplerate < 6442*2
                        sf = 2.3 ;
                else
```

```
                        sf = 2 ;
                end
        otherwise,
                sf = 2.2 ;
end
wd=2*pi*cfc*sf ; %cfc = channel filter class
wa=sin(wd*T/2)/cos(wd*T/2);
a0=wa^2/(1.0+sqrt(2)*wa+wa^2);
a1=2*a0;
a2=a0;
b0=1;
b1=-2*(wa^2-1)/(1+sqrt(2)*wa+wa^2);
b2=(-1+sqrt(2)*wa-wa^2)/(1+sqrt(2)*wa+wa^2);
a=[b0 -b1 -b2]; %coefficients for filter from SAE J211
b=[a0  a1  a2]; %coefficients for filter from SAE J211

%%
% *Pad data at beginning and end*
numPts = floor(10 * samplerate / 1000) ; %% 10 ms of data
x1 = zeros(1,numPts);
x = [x1 x x1] ;

%%
% *Filter data*
y=filtfilt(b,a,x);

%%
% *Remove padding*
if length(numPts:length(y)-numPts) > len
        y = y(numPts+1:end-numPts) ;
else
        y = y(numPts:end-numPts) ;
end
if flag
        y = y' ;
end
```