

```
function y = low_pass(t,x,cutoff)
%LOW_PASS Low Pass Filter
% Inputs are:
% t      :a numeric array of M time in seconds
% x      :a numeric array of M tracked variable
% cutoff :a scalar cutoff frequency in rad/s
%
% Output is:
% y      :a numeric array of M filtered tracked variable

arguments
    t (1,:) {mustBeNumeric, mustBeReal}
    x (1,:) {mustBeNumeric, mustBeReal}
    cutoff {mustBeNumeric, mustBeReal}
end

L = length(x);
sample_freq = L./t(end);

% FFT to determine acceptable cutoff frequency
y = fft(x);
P2 = abs(y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);

% Plot Amplitude spectrum and cutoff frequency
figure()
f = sample_freq*(0:(L/2))/L;
plot(2*pi*f,P1)
title('Single-Sided Amplitude Spectrum of X(t)')
xlabel('f (rad/s)')
ylabel('|P1(f)|')
xline(cutoff,'r','Cutoff Frequency')

% Low pass filter
G=tf(cutoff,[1 cutoff]);
y=lsim(G,x,t);
end
```