```
% Author:   Bradley Anderson
% Date:     Oct-21 2017
% Name:     transferFunctionPlot
% Purpose:  Takes a numerator array, demoninator array, and settling
%           time. Creates a transfer function from parameters and
%           plots a step response. Overlays plot with rise time,
%           settling time of given ST, peak time, and percent
%           overshoot. Returns stepinfo and axes handle.

function [S, AxH] = transferFunctionPlot(numerator, demoninator, ST)

Ts = tf(numerator, demoninator);
[Y, T] = step(Ts);

% Format Title
% Takes coefficeint arrays and converts them into symbols that can be
% interpretted by latex
syms s
numSymPoly = sym(numerator);
denSymPoly = sym(demoninator);
numSym = poly2sym(numSymPoly, s);
denSyn = poly2sym(denSymPoly, s);
tfSym = numSym/denSyn;
tfTitle = latex(tfSym);

% Get info about the transfer function
S = stepinfo(Ts, 'SettlingTimeThreshold', ST);

% Plot formatting
AxH = axes;
func = plot(AxH,T, Y, 'LineWidth',2);
hold on
xlabel(AxH, 'Time (s)');
ylabel(AxH, 'Step Response');
title(AxH, sprintf('T(s) = $$ %s $$', tfTitle), 'Interpreter','latex')

% Strady state of function
steadyState = plot(AxH, [min(T), max(T)],[Y(end) Y(end)],'--k');

% Calculate ten percent time and index of that value in the time array
tenPercent = (Y(end)-Y(1))*0.1;
[~, tpIndex] = min( abs(Y-tenPercent) );

% Calculate ninety percent time and index of that value in the time
 array
ninetyPercentTime = S.RiseTime + T(tpIndex);
[~, npIndex] = min( abs(T-ninetyPercentTime) );

% Find index of settling time in time array
[~, stIndex] = min( abs(T - S.SettlingTime) );

% Plot into variables in order to set legend later
```

```
riseTime0 = plot(AxH, [T(tpIndex) T(tpIndex)], [Y(1) Y(tpIndex)], '--
b');
riseTime1 = plot(AxH, [ninetyPercentTime ninetyPercentTime], [Y(1)
 Y(npIndex)], '--r');
settTime = plot(AxH, [S.SettlingTime S.SettlingTime], [Y(1)
 Y(stIndex)],'--g');
peakTime = plot(AxH, [S.PeakTime S.PeakTime], [Y(1) S.Peak],'--c');
overshoot = plot(AxH, [T(1) S.PeakTime], [S.Peak S.Peak],'--m');

% Legend requires proper order, not tuples
legend(AxH, [func steadyState riseTime0 riseTime1 settTime peakTime
 overshoot], ...
    {'Step response', 'Steady state', ...
    ['10% = ', num2str(T(tpIndex)), ' s'], ...
    ['Rise time = ', num2str(S.RiseTime), ' s'], ...
    [num2str(ST*100),'% Settling time = ' num2str(S.SettlingTime) '
 s'] ...
    ['Peak time = ', num2str(S.PeakTime), ' s'], ...
    [num2str(S.Overshoot), '% Overshoot']}, ...
    'Location', 'south')

end
```

```
Not enough input arguments.

Error in transferFunctionPlot (line 12)
Ts = tf(numerator, demoninator);
```

*Published with MATLAB® R2016a*