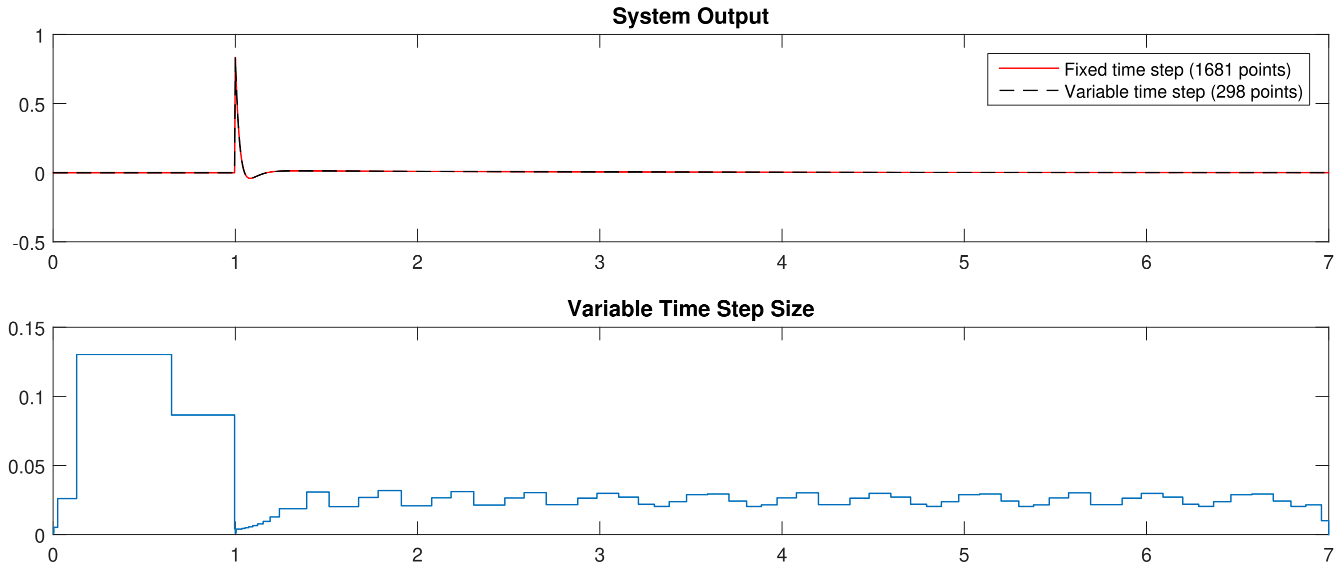
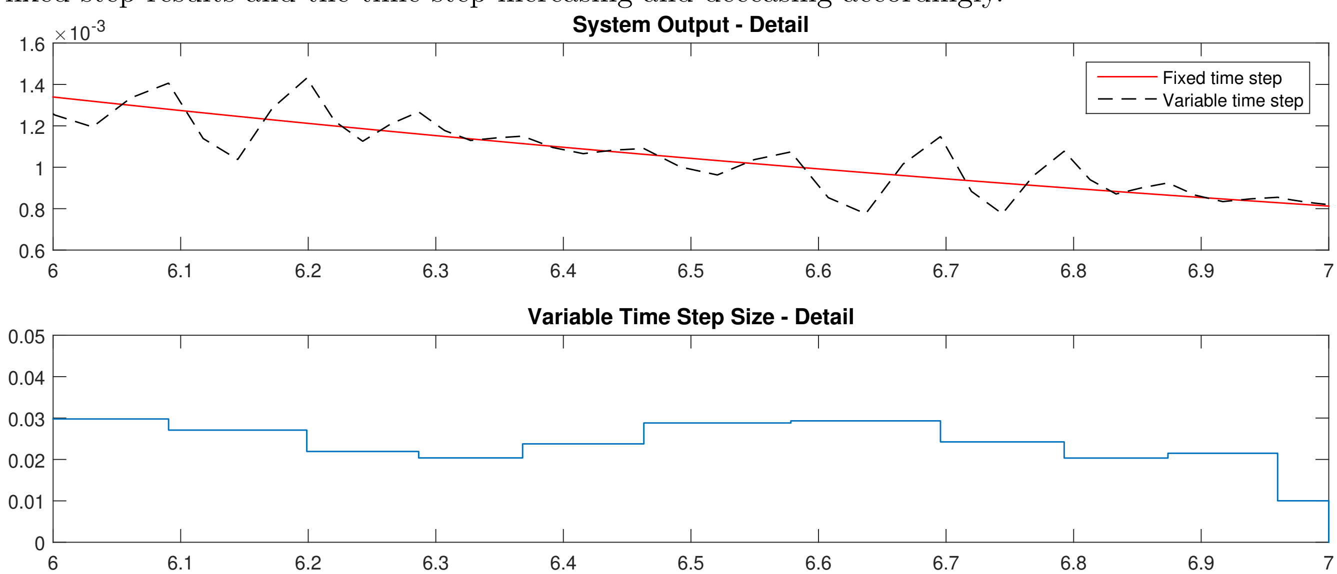
Step Results of a 3rd Order State Space System

A fixed time step of 1/4 cycle (≈4.2 seconds) using `lsim` was compared to a variable time step `ode45` solution. The default ODE tolerance settings were altered to produce an acceptable match in output (code line 23).

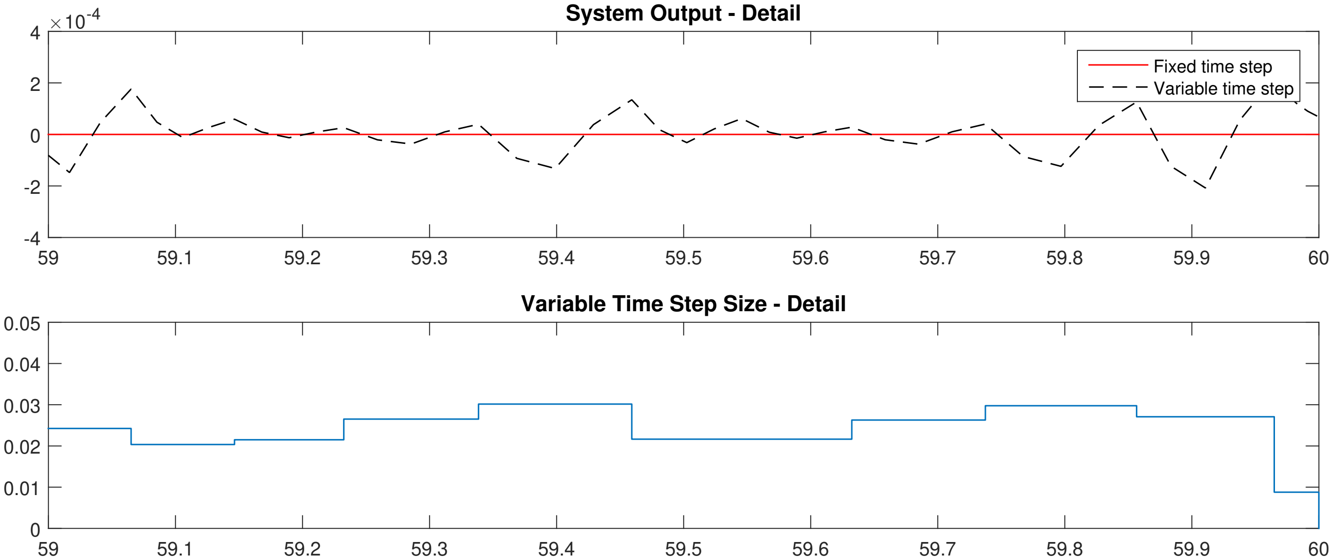
Data output plots appear the same upon inspection. The variable time step method requires fewer steps (≈ 5.5x less in this example) for very similar output. Time step size seems to oscillate around 0.026 seconds post-perturbance but is as large as 0.13 seconds before the step is executed.



A detail view of the output shows minor oscillations of the variable time step results around the fixed step results and the time step increasing and deceasing accordingly.



Running the simulation out to 60 seconds shows that this *time step oscillation* occurs continuously after an event. The large decease in time step size near the end of a calculated time interval appears to be a common `ode45` trait.



MATLAB Code

The `ode45` function requires a passed in function that uses `t` and returns derivatives. A simple `getXdot` function was written that performs such an action. Note that the time variable `t` is not used and most variables are global. This was done done to mimic PST methods.

```
function [ xdot ] = getXdot( t, x)
%getXdot return xdot from statespace for ODE45 use
%   t = filler variable
%   A = A matrix from system
%   x = initial state vector
%   B = B matrix from system
%   U = Input to system
global A B U
    xdot = A*x + B*U;
end
```

A PSS model from the miniWECC case was used as the system to test. Manipulation of `ode45` output is required for correct state operation and model output handling (lines 36 and 41).
`ode45` has a 'OutputFcn' option that may be useful in indexing, time advancement, required state/output handling, and/or network solution calls. Other `ode45` options exist that may also be useful in future development.

```
%% test to use ode solver to step PST-esq model
close all;clear;format compact;clc

%% pss model definition (miniWECC)
%           1   2   3   4       5   6   7       8       9       10
pss_con = [ 1          1   20  2     0.25 0.04 0.2   0.03      1.0     -1.0];

%% MATLAB model
tend = 60;
block1 = tf([pss_con(3)*pss_con(4), 0],[pss_con(4), 1]);
block2 = tf([pss_con(5), 1],[pss_con(6), 1]);
block3 = tf([pss_con(7), 1],[pss_con(8), 1]);

G= block1*block2*block3;
tL = 0:1/60/4:tend; % quarter cycle steps
modSig = zeros(size(tL,1),1);
modSig(tL>=1) = .001; % very small input to avoid limiter
yL = lsim(G,modSig,tL);

%% ODE45 attempt with statespace
% Configure ODE settings
%options = odeset('RelTol',1e-3,'AbsTol',1e-6); % default settings
options = odeset('RelTol',1e-5,'AbsTol',1e-8,'InitialStep', 1/60/4, 'MaxStep',20);

% manipulate test sytem to statespace
[num,den] = tfdata(G);
global A B U
[A,B,C,D] = tf2ss(num{1},den{1});
% initial conditions
x = zeros(size(A,1),1);
y0 = x;
U = 0;

% Pre-perturbance
[t1,y1] = ode45(@getXdot, [0,1-1/60/4],y0, options);
yOut1 = C*y1'+D*U;

% Step input
U = modSig(end);
[t2,y2] = ode45(@getXdot, [1,tend],y1(end,:)', options);
yOut2 = C*y2'+D*U;

% combining output
tCombined = [t1;t2];
yCombined = [yOut1, yOut2];
```