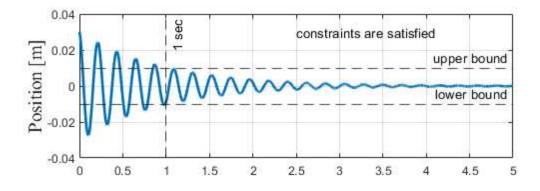
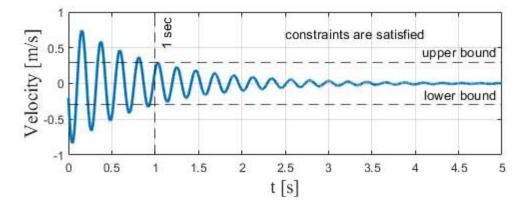
```
J=.016;
m=12;
k=10.^4;
r=.34;
% find the damping coefficient using the plot below:
c num = 31
\% define the time interval and the number of integration steps
   = 10000;
    = zeros(n,1);
tf = 5;
t(1) = 0;
delt = (tf-t(1))/n;
% initializing variables
   = zeros(n,1);
    = zeros(n,1);
% assign initial conditions (be careful of the units)
x(1) = .03;
v(1) = -.2;
% start the integration directly from the second step (with i=2)
for i=2:1:n
    % update the velocity and acceleration for x at step i
    a=((-c_num./((J./r.^2)+m)).*v(i-1) + (-k./((J./r.^2)+m)).*x(i-1));
    \% integrate x and v
    x(i) = x(i-1)+v(i-1).*delt;
    v(i) = v(i-1)+a.*delt;
    % integrate time
    t(i) = t(i-1)+delt;
end % end of each integration step
% plot results
figure(10)
clf;
% plot position
subplot(2,1,1)
box on; grid on; hold on;
plot(t,x,'linewidth',2);
ylabel('Position [m]', 'fontsize', 14, 'fontname', 'times');
% find the indices of all elements that are smaller or equal to 1
time flag = (t <= 1);
% find the index of the last element
time_ind = find(time_flag,1,'last');
% Check the constraint violation for position:
if all(abs(x(time_ind:end))<0.01)</pre>
    yline(0.01,'k--','upper bound');
    yline(-0.01, 'k--', 'lower bound');
    text(2.5,0.03,'constraints are satisfied');
else
```

```
yline(0.01,'r--','upper bound');
    yline(-0.01,'r--','lower bound');
    text(2.5,0.03, 'constraints are violated');
end
xline(1,'k--','1 sec');
% plot velocity
subplot(2,1,2)
box on; grid on; hold on;
plot(t,v,'linewidth',2);
xlabel('t [s]', 'fontsize', 14, 'fontname', 'times');
ylabel('Velocity [m/s]', 'fontsize', 14, 'fontname', 'times');
% check constraint violation for speeds
if all(abs(v(time_ind:end))<0.3)</pre>
    yline(0.3,'k--','upper bound');
    yline(-0.3, 'k--', 'lower bound');
    text(2.5,0.7,'constraints are satisfied');
else
    yline(0.3,'r--','upper bound');
    yline(-0.3,'r--','lower bound');
    text(2.5,0.7,'constraints are violated');
end
xline(1,'k--','1 sec');
```

c_num =

31





Published with MATLAB® R2020a