MAE 219 Assignment 4, Part 2

```
% Original Author: Allison Okamura, September 20, 2015
% Last Modified By: David Lim, October 10, 2024

% Clear the workspace.
clear;
```

Constants

```
% device and human parameters
m = 0.03;   % effective mass at the handle, kg
b = 1;   % viscous damping, Ns/m
kh = 500;   % human hand stiffness, N/m
bh = 10;   % human hand damping, Ns/m

% times for dynamic simulation
tstart = 0;   % s
tend = 10;   % s
T = 0.1*10^-3;   % time increment, s
t = (tstart:T:tend)';   % time vector

% human input
omega = 0.4*2*pi;   % frequency of user's desired motion, rad/s
A = 0.04;   % amplitude of user's desired motion, m
xd = A*sin(omega*t) - 0.01;   % user's desired hand position, in m
vd = A*omega*cos(omega*t);   % user's desired hand velocity, in m/s

% default virtual wall parameters
kwall = 500;   % wall stiffness, N/m
xwall = 0.025;   % wall position, m

% ************************************
% effect-specific parameters
effect = "Sensor Quantization";   % change this to change the effect
switch effect
    case "Sensor Quantization"
        label = "Nominal System Parameters";
        deltax = 0.0005;   % position sensor resolution, m
    case "Nonlinear Friction"
        label = "User Moves in Free Space";
        xwall = 0.05;   % user moving in free space
        vt = 0.01;   % speed threshold, m/s
        bl = 10*b;   % large damping, Ns/m
    case "Actuator Saturation"
        label = "User Moves Inside Wall";
        xwall = -0.05;   % user moving inside wall
        ft = 15;   % force threshold, N
    case "Sample and Hold"
        label = "Nominal System Parameters";
        deltat = 0.05;   % sampling period
    case "Zero-Order Hold"
```

```
        label = "Nominal System Parameters";
        deltat = 0.05;   % hold period
    otherwise
        error("Effect not valid.")
end
% ************************************
```

State Tracking

```
xh = zeros(length(t),1);    % handle position
vh = zeros(length(t),1);    % handle velocity
ah = zeros(length(t),1);    % handle acceleration
fa = zeros(length(t),1);    % force applied by the actuator
ffelt = zeros(length(t),1); % force felt by the human
xq = zeros(length(t),1);   % measured position
```

Dynamic Simulation

```
for sim = 1:2

    for i = 1:length(t)

        % integrate the main state derivatives
        if (i == 1)
            % first time step has no difference between desired and actual
handle position
            vh(i) = vd(i);
            xh(i) = xd(i);
        else
            % simple Euler integration (you could use something more
accurate!)
            vh(i) = vh(i-1) + ah(i-1) * T;
            xh(i) = xh(i-1) + vh(i-1) * T;
        end

        xq(i) = xh(i);

        % ************************************
        if effect == "Sample and Hold" & sim == 2   % if sample and hold in
effect
            if rem(t(i),deltat) ~= 0   % if not time to sample position
                xq(i) = xq(i-1);   % hold previous position measurement
            end
        end
        % ************************************

        % ************************************
        if effect == "Sensor Quantization" & sim == 2   % if sensor
quantization in effect
            xq(i) = deltax * floor(xh(i)/deltax);   % quantization of xh with
resolution deltax
        end
        % ************************************
```

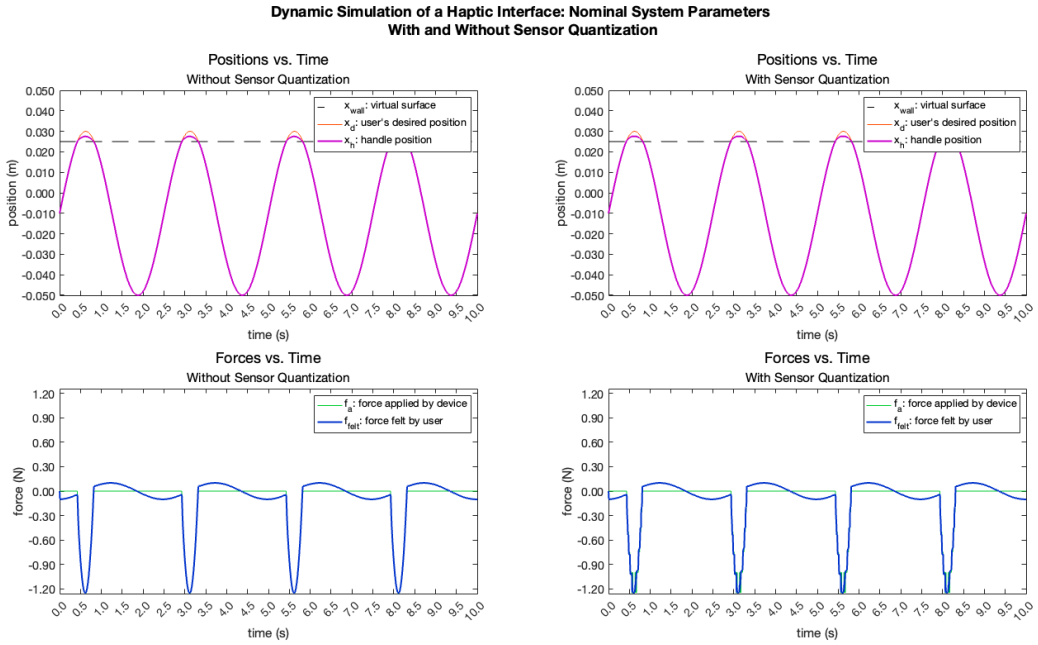
```
        % force applied by the virtual environment
        if (xq(i) > xwall)   % if the user is inside the wall
            fa(i) = kwall * (xwall - xq(i));
        else   % if the user is outside the wall
            fa(i) = 0;
        end

        % ************************************
        if effect == "Zero-Order Hold" & sim == 2   % if zero-order hold in
effect
            if rem(t(i),deltat) ~= 0   % if not time to update the force
                fa(i) = fa(i-1);   % hold previous force output
            end
        end
        % ************************************

        % ************************************
        if effect == "Actuator Saturation" & sim == 2   % if actuator
saturation in effect
            if abs(fa(i)) > ft   % if actuator is saturated
                fa(i) = sign(fa(i)) * ft;   % limit force applied
            end
        end
        % ************************************

        % force between the hand and the handle
        fh = kh * (xd(i) - xh(i)) + bh * (vd(i) - vh(i));

        % force felt by the user
        ffelt(i) = -fh;

        % default friction force
        ff = -b * vh(i);

        % ************************************
        if effect == "Nonlinear Friction" & sim == 2   % if nonlinear
friction if effect
            if abs(vh(i)) < vt   % if speed within threshold
                ff = -bl * vh(i);   % large damping
            end
        end
        % ************************************

        % Compute the sum of forces on the handle: applied force, human
force, and friction force.
        ftotal = fa(i) + fh + ff;

        % Compute the handle's new acceleration for the next iteration.
        ah(i) = ftotal / m;

    end

    if sim == 1   % if ran simulation without effect
```

```
        % save data in matrices
        xdata1 = [xwall*ones(1,length(t))' xd xh];
        fdata1 = [fa, ffelt];
    else
        % save data in matrices
        xdata2 = [xwall*ones(1,length(t))' xd xh];
        fdata2 = [fa, ffelt];
    end

end
```

Plotting

```
figure(1); clf;

% tick mark calculations
N = 5;
% find max
xmax = max( [abs(xdata1) abs(xdata2)] ,[],"all");
% create N tick marks above and below zero
xtic = round( xmax/N, 1,'significant');
xtics = [-fliplr(xtic:xtic:xmax) 0 xtic:xtic:xmax];
% find max
fmax = max( [abs(fdata1) abs(fdata2)] ,[],"all");
% create N tick marks above and below zero
ftic = round( fmax/N, 1,'significant');
ftics = [-fliplr(ftic:ftic:fmax) 0 ftic:ftic:fmax];

% positions without effect
subplot(2,2,1)
h = plot(t, xdata1);
set(h(1),'Color',[0 0 0],'LineWidth',1.0,'LineStyle','--')
set(h(2),'Color',[1 .3 0],'LineWidth',1)
set(h(3),'Color',[.8 0 .8],'LineWidth',2)
xlabel('time (s)')
ylabel('position (m)')
legend('x_{wall}: virtual surface','x_d: user''s desired position','x_h:
handle position')
subtitle(append('Without ',effect))
axis([tstart tend -xmax xmax])
set(gca,'FontSize',14)
title('Positions vs. Time','FontWeight','Normal','FontSize',18)
xticks(0:0.5:tend)
xtickformat('%.1f')
yticks(xtics)
ytickformat('%.3f')

% positions with effect
subplot(2,2,2)
h = plot(t, xdata2);
set(h(1),'Color',[0 0 0],'LineWidth',1.0,'LineStyle','--')
set(h(2),'Color',[1 .3 0],'LineWidth',1)
set(h(3),'Color',[.8 0 .8],'LineWidth',2)
```

```
xlabel('time (s)')
ylabel('position (m)')
legend('x_{wall}: virtual surface','x_d: user''s desired position','x_h:
handle position')
subtitle(append('With ',effect))
axis([tstart tend -xmax xmax])
set(gca,'FontSize',14)
title('Positions vs. Time','FontWeight','Normal','FontSize',18)
xticks(0:0.5:tend)
xtickformat('%.1f')
yticks(xtics)
ytickformat('%.3f')

% forces without effect
subplot(2,2,3)
h = plot(t, fdata1);
set(h(1),'Color',[0 .8 .2],'LineWidth',1)
set(h(2),'Color',[0 .2 .8],'LineWidth',2)
xlabel('time (s)')
ylabel('force (N)')
legend('f_a: force applied by device','f_{felt}: force felt by user')
subtitle(append('Without ',effect))
axis([tstart tend -fmax fmax])
set(gca,'FontSize',14)
title('Forces vs. Time','FontWeight','Normal','FontSize',18)
xticks(0:0.5:tend)
xtickformat('%.1f')
yticks(ftics)
ytickformat('%.2f')

% forces with effect
subplot(2,2,4)
h = plot(t, fdata2);
set(h(1),'Color',[0 .8 .2],'LineWidth',1)
set(h(2),'Color',[0 .2 .8],'LineWidth',2)
xlabel('time (s)')
ylabel('force (N)')
legend('f_a: force applied by device','f_{felt}: force felt by user')
subtitle(append('With ',effect))
axis([tstart tend -fmax fmax])
set(gca,'FontSize',14)
title('Forces vs. Time','FontWeight','Normal','FontSize',18)
xticks(0:0.5:tend)
xtickformat('%.1f')
yticks(ftics)
ytickformat('%.2f')

sgtitle([append('Dynamic Simulation of a Haptic Interface: ',label); append('
With and Without ',effect)],'FontSize',18,'FontWeight','Bold')
```



Published with MATLAB® R2024a