Lab 8: Estimating Velocity and Acceleration Using Numerical Differentiation

One application for numerical differentiation is in feedback control. Feedback control involves using sensors to measure the output(s) of a system or process and applying corrective control signals to keep the system output(s) at some desired set-point or following some desired path. Feedback control is applied to electromechanical systems, chemical processes, aircraft control, and a host of other applications. Almost every engineering discipline at CEAS includes at least one course in feedback control. We are going to look at a simple scenario in this lab.

Assume we have a two-link robot arm and have sensors to measure the angular positions, θ_1 and θ_2. Also, assume that we need to know angular velocities and angular accelerations but do not have sensors to measure these quantities. Instead, we will estimate angular velocity and angular acceleration from the angular position measurements using numerical differentiation.

A. Practice Exercise

The table below shows position measurements taken every 0.75 seconds.

Time (sec)	**0**	**0.75**	**1.5**	**2.25**	**3.0**
Position (deg)	**10**	**12**	**13**	**11**	**15**
Velocity (deg/sec)	**0**	2.6667	1.3333	-2.6667	5.3333
Acceleration (deg/sec^2)	**0**	-1.7778	-5.333	10.6667	

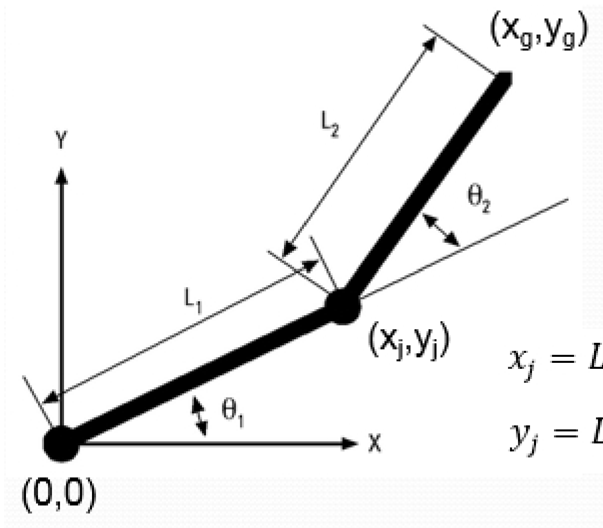
Complete the table using the following formulas to estimate velocity and acceleration from the position measurements then check values with your T.A.

$$\text{Velocity(t)} \approx \frac{\text{Position(t)} - \text{Position}(\text{t} - \Delta\text{t})}{\Delta\text{t}}$$

$$\text{Acceleration(t)} \approx \frac{\text{Position}(\text{t} + \Delta\text{t}) - 2 * \text{Position(t)} + \text{Position}(\text{t} - \Delta\text{t})}{\Delta\text{t}^2}$$

B. Estimating Velocity and Acceleration for Two-Link Robot Arm

Figure 1 shows a diagram for a two-link robot arm. Assume we have sensors to measure the angular positions, θ_1 and θ_2. Also, assume that we need to know angular velocity and angular acceleration but do not have sensors to measure these quantities. Instead, we will estimate angular velocity and angular acceleration from the angular position measurements using numerical differentiation.



$$x_g = L_1\cos(\theta_1) + L_2\cos(\theta_1 + \theta_2)$$

$$y_g = L_1\sin(\theta_1) + L_2\sin(\theta_1 + \theta_2)$$

$$x_j = L_1\cos(\theta_1)$$

$$y_j = L_1\sin(\theta_1)$$

Figure 1: Two-Link Robot Arm

Assume the initial base angle (Θ_1) is 30°, the initial joint angle (Θ_2) is 20°, Link 1 is 5ft long and Link 2 is 4ft long. The gripper coordinates at the end of Link 2 will then be xg = 6.9 ft. and yg = 5.56 ft. We wish to move the robot are along a smooth path to a new set of gripper coordinates (xg = 2 ft. and yg = 4 ft.).

The following polynomial equations for base angle and joint angle will move the gripper to the desired position.

Base Angle Trajectory:

$$\theta_1\ (\mathrm{deg}):\ \mathrm{ThB} = 30 - 9.684 \times 10^{-4} \cdot t^5 + 0.02421 \cdot t^4 - 0.1614 \cdot t^3$$

Joint Angle Trajectory:

$$\theta_2\ (\mathrm{deg}):\ \mathrm{ThJ} = 20 + 6.1002 \times 10^{-3} \cdot t^5 - 0.152505 \cdot t^4 + 1.0167 \cdot t^3$$

Assuming the robot follows the desired trajectories, the corresponding angular velocities would simply be the derivative of the angle trajectories:

$$\text{Base Vel.} \left(\frac{\mathrm{deg}}{\mathrm{s}}\right):\ \mathrm{VelB} = -5 \cdot 9.684 \times 10^{-4} \cdot t^4 + 4 \times 0.02421 \cdot t^3 - 3 \times 0.1614 \cdot t^2$$

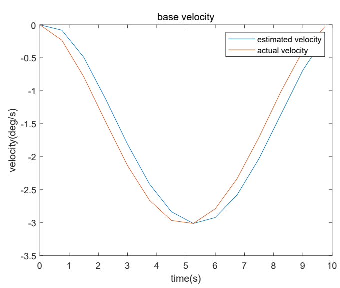
$$\text{Joint Vel.} \left(\frac{\mathrm{deg}}{\mathrm{s}}\right):\ \mathrm{VelJ} = 5 \times 6.1002 \cdot 10^{-3} \cdot t^4 - 4 \times 0.152505 \cdot t^3 + 3 \times 1.0167 \cdot t^2$$

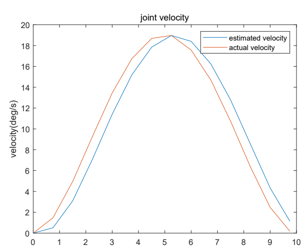
<u>Part 1</u>: In a script file do the following:

- Create a vector for time starting at 0, incrementing by 0.75, and ending at 9.75 seconds.
- Calculate the base angle position and joint angle position for each time using the given equations for base angle and joint angle trajectories and save the results in vectors.
- Using a 2PT reverse estimate, calculate estimates for the angular velocity of the base and angular velocity of the joint for each time and save the results in vectors.
- Calculate the actual angular velocities for each time using the given polynomial equations.
- Plot the estimated and actual velocity for the base angle on the same graph.
- Plot the estimated and actual velocity for the joint angle together on another graph.
- Calculate the absolute value of the estimation error for each velocity (i.e., abs(Vel_2PT – Vel)).
- Plot the absolute estimation error for the angular velocity of the base and joint angles together on another graph.
- Calculate the maximum estimation error for the base angular velocity and the maximum estimation error for the joint angular velocity. Use fprintf statements to display these values.

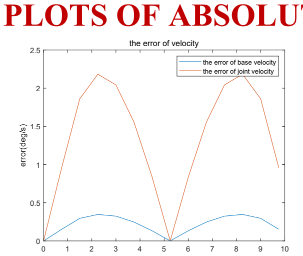
All plots should be titled, labeled (include units), and have a legend (if appropriate).

PLOTS OF ESTIMATED AND ACTUAL VELOCITIES (Joint and Base)





PLOTS OF ABSOLUTE ESTIMATION ERROR (Joint and Base)



Maximum Estimation Error for Base Angular Velocity: ___0.3466（deg/s） ___________

Maximum Estimation Error for Joint Angular Velocity: _2.1833(deg/s)_____________

<u>Part 2</u>:

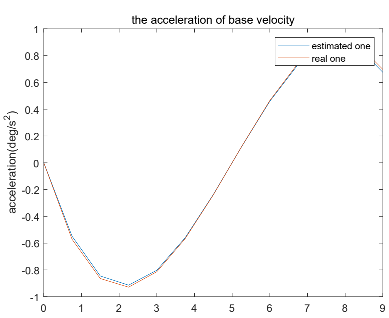
Assuming the robot follows the desired trajectories, the corresponding angular accelerations would simply be the 2^{nd} derivative of the angle trajectories:

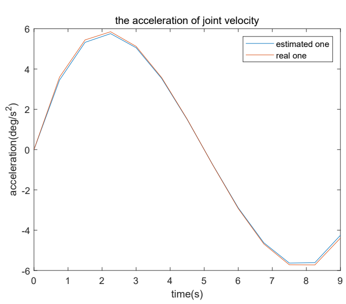
$$\text{Base Acc.}\left(\frac{\text{deg}}{\text{s}^2}\right):\ \ \text{AccB} = -20 \cdot 9.684 \times 10^{-4} \cdot t^3 + 12 \times 0.02421 \cdot t^2 - 6 \times 0.1614 \cdot t$$

$$\text{Joint Acc.}\left(\frac{\text{deg}}{\text{s}^2}\right):\ \text{AccJ} = 20 \cdot 6.1002 \times 10^{-3} \cdot t^3 - 12 \times 0.152505 \cdot t^2 + 6 \times 1.0167 \cdot t$$

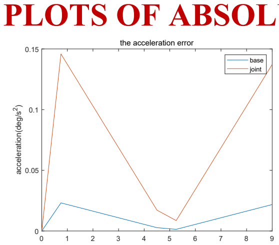
- Add code to the script developed in Part 1 to estimate the angular acceleration of the base and joint angles **by applying the 2^{nd} derivative estimate** to the base and joint angle measurements.
- Calculate the actual angular accelerations for each time using the given polynomial equations for base and joint acceleration.
- Plot the estimated and actual acceleration for the base angle on the same graph.
- Plot the estimated and actual acceleration for the joint angle together on another graph.
- Calculate the absolute value of the estimation error for both the base and joint accelerations.
- Plot the absolute estimation error for the angular acceleration of both the base and joint angles together on another graph.
- Calculate the maximum estimation error for the base angular acceleration and the maximum estimation error for the joint angular acceleration. Use fprintf statements to display these values.

PLOTS OF ESTIMATED AND ACTUAL ACCELERATIONS (Joint and Base)





PLOTS OF ABSOLUTE ESTIMATION ERROR (Joint and Base)



Maximum Estimation Error for Base Angular Acceleration: ______0.023_deg/s^2_______

Maximum Estimation Error for Joint Angular Acceleration: ____0.146___deg/s^2_______

PASTE MATLAB SCRIPT HERE:

```
%% lab 8
% name :Horace
clear ;clc;close all
%% calculate position and velocity
time=0:0.75:9.75;
thb=zeros(1,length(time));
thj=zeros(1,length(time));
for i=1:length(time)
thb(i)=30-9.684*(10^(-4))*(time(i))^(5)+0.02421*time(i)^4-0.1614*time(i)^3;
thj(i)=20+6.1002*10^(-3)*time(i)^5-0.152505*time(i)^4+1.0167*time(i)^3;
end
vb=zeros(1,length(time));
vj=zeros(1,length(time));
for i=2:length(time)
    vb(i)=(thb(i)-thb(i-1))/0.75;
    vj(i)=(thj(i)-thj(i-1))/0.75;
end
velb=zeros(1,length(time));
velj=zeros(1,length(time));
for i=1:length(time)
velb(i)=-5*9.684*10^(-4)*time(i)^4+4*0.02421*time(i)^3-3*0.1614*time(i)^2;
velj(i)=5*6.1002*10^(-3)*time(i)^4-4*0.152505*time(i)^3+3*1.0167*time(i)^2;
end
%% plot
figure(1)
plot(time,vb,time,velb)
xlabel('time(s)')
ylabel('velocity(deg/s)')
legend('estimated velocity','actual velocity')
title('base velocity')
figure(2)
plot(time,vj,time,velj)
xlabel('time(s)')
ylabel('velocity(deg/s)')
legend('estimated velocity','actual velocity')
title('joint velocity')
errorvb=abs(velb-vb);
errorvj=abs(velj-vj);
figure(3)
plot(time,errorvb,time,errorvj)
title('the error of velocity')
xlabel('time(s)')
ylabel('error(deg/s)')
legend('the error of base velocity','the error of joint velocity')

%% find maximum
maxerrb=max(errorvb);
```

```
maxerrj=max(errorvj);

fprintf('the maxium error base angular velocity is %0.4f deg/s and the joint
angular velocity is %0.4f deg/s \n',maxerrb,maxerrj)
%% part 2
% calculate acceleration
baseacc=zeros(1,length(time)-1);
jointacc=zeros(1,length(time)-1);
for i=2:length(time)-1
    baseacc(i)=(thb(i+1)+thb(i-1)-2*thb(i))/0.75^2;
    jointacc(i)=(thj(i+1)+thj(i-1)-2*thj(i))/0.75^2;
end
for i=1:length(time)-1
   realba(i)=-20*9.684*10^(-4)*time(i)^3+12*0.02421*time(i)^2-
6*0.1614*time(i);
   realja(i)=20*6.1002*10^(-3)*time(i)^3-
12*0.152505*time(i)^2+6*1.0167*time(i);
end
%% plot
figure(4)
plot((time(1:end-1)),baseacc,(time(1:end-1)),realba)
title('the acceleration of base velocity')
legend('estimated one ','real one')
xlabel('time(s)')
ylabel('acceleration(deg/s^2)')
figure(5)
plot(time(1:end-1),jointacc,time(1:end-1),realja)
title('the acceleration of joint velocity')
legend('estimated one ','real one')
xlabel('time(s)')
ylabel('acceleration(deg/s^2)')
%%
erroraccb=abs(realba-baseacc);
erroraccj=abs(realja-jointacc);
figure(6)
plot(time(1:end-1),erroraccb,time(1:end-1),erroraccj)
title('the acceleration error')
legend('base  ','joint')
xlabel('time(s)')
ylabel('acceleration(deg/s^2)')
basemax=max(erroraccb);
jointmax=max(erroraccj);
fprintf('the maximum value of base error is %0.3f deg/s^2 and joint is %0.3f
deg/s^2 \n',basemax,jointmax);
```

Part 3: The results in Parts 1 and 2 indicate that the largest estimation error occurs with joint angle. Find the largest acceptable DeltaT ***(within two places behind the decimal point)*** that ensures that the maximum angular velocity estimation error will be less than 1 deg/sec.

DeltaT = _______0.34_____

Include your code and/or explanation of how you arrived at this value for DeltaT:

```
%% lab 8
% name :Horace
clear ;clc;close all
%% calculate position and velocity
t=input('the number of deltat ');
time=0:t:9.75;
thb=zeros(1,length(time));
thj=zeros(1,length(time));
for i=1:length(time)
thb(i)=30-9.684*(10^(-4))*(time(i))^(5)+0.02421*time(i)^4-0.1614*time(i)^3;
thj(i)=20+6.1002*10^(-3)*time(i)^5-0.152505*time(i)^4+1.0167*time(i)^3;
end
vb=zeros(1,length(time));
vj=zeros(1,length(time));
for i=2:length(time)
    vb(i)=(thb(i)-thb(i-1))/t;
    vj(i)=(thj(i)-thj(i-1))/t;
end
velb=zeros(1,length(time));
velj=zeros(1,length(time));
for i=1:length(time)
velb(i)=-5*9.684*10^(-4)*time(i)^4+4*0.02421*time(i)^3-3*0.1614*time(i)^2;
velj(i)=5*6.1002*10^(-3)*time(i)^4-4*0.152505*time(i)^3+3*1.0167*time(i)^2;
end
%% plot
figure(1)
plot(time,vb,time,velb)
xlabel('time(s)')
ylabel('velocity(deg/s)')
legend('estimated velocity','actual velocity')
title('base velocity')
figure(2)
plot(time,vj,time,velj)
xlabel('time(s)')
ylabel('velocity(deg/s)')
legend('estimated velocity','actual velocity')
title('joint velocity')
errorvb=abs(velb-vb);
errorvj=abs(velj-vj);
figure(3)
plot(time,errorvb,time,errorvj)
title('the error of velocity')
xlabel('time(s)')
ylabel('error(deg/s)')
legend('the error of base velocity','the error of joint velocity')

%% find maximum
maxerrb=max(errorvb);
maxerrj=max(errorvj);

fprintf('the maxium error base angular velocity is %0.4f deg/s and the joint
angular velocity is %0.4f deg/s \n',maxerrb,maxerrj)
%% part 2
% calculate acceleration
```

```
baseacc=zeros(1,length(time)-1);
jointacc=zeros(1,length(time)-1);
for i=2:length(time)-1
    baseacc(i)=(thb(i+1)+thb(i-1)-2*thb(i))/t^2;
    jointacc(i)=(thj(i+1)+thj(i-1)-2*thj(i))/t^2;
end
for i=1:length(time)-1
   realba(i)=-20*9.684*10^(-4)*time(i)^3+12*0.02421*time(i)^2-
6*0.1614*time(i);
   realja(i)=20*6.1002*10^(-3)*time(i)^3-
12*0.152505*time(i)^2+6*1.0167*time(i);
end
%% plot
figure(4)
plot((time(1:end-1)),baseacc,(time(1:end-1)),realba)
title('the acceleration of base velocity')
legend('estimated one ','real one')
xlabel('time(s)')
ylabel('acceleration(deg/s^2)')
figure(5)
plot(time(1:end-1),jointacc,time(1:end-1),realja)
title('the acceleration of joint velocity')
legend('estimated one ','real one')
xlabel('time(s)')
ylabel('acceleration(deg/s^2)')
%%
erroraccb=abs(realba-baseacc);
erroraccj=abs(realja-jointacc);
figure(6)
plot(time(1:end-1),erroraccb,time(1:end-1),erroraccj)
title('the acceleration error')
legend('base  ','joint')
xlabel('time(s)')
ylabel('acceleration(deg/s^2)')
basemax=max(erroraccb);
jointmax=max(erroraccj);
fprintf('the maximum value of base error is %0.3f deg/s^2 and joint is %0.3f
deg/s^2 \n',basemax,jointmax);
```

Explanation:try several times of value t

What to submit: This document with all required information included.