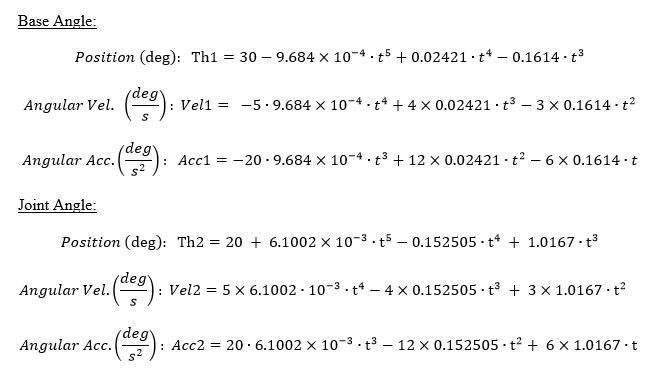
**ENED 1091 HW#4**

**Due Thursday, March 3, 2016 at noon**

The following formulas will be useful for all the problems in this assignment and are taken from the 2-link robot arm problem from HW#2.



Use your work from homework #2, problem 4, to calculate the a’s needed for this problem. You also have the position equations in the same problem. Make sure you don’t clear these values as you work through these problems.

**Problem 1:**  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 polynomial equations and save the results in vectors.
* Using a 2PT reverse estimate, calculate estimates for 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 velocities for the base and joint angles.
* 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.
* 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: \_\_\_\_\_0.346603978125033\_\_\_\_\_\_\_\_\_**

**Maximum Estimation Error for Joint Angular Velocity: \_\_\_\_\_2.183347364062485\_\_\_\_\_\_\_\_\_**

**PASTE MATLAB SCRIPT HERE:**

%Kyle O'Connor

%

%HW 4

%

%2/27/2016

home;

clear all;

clc;

Lbase = 5;

Ljoint = 4;

time = 0:.75:9.75;

dt = time(2) - time(1);

%Finding Base and Joint Angles/Velocities

for i = 1:length(time)

ThBase(i) = 30 + (-9.684\*(10^-4))\*time(i)^5 + (0.02421)\*time(i)^4 + (-.1614)\*time(i)^3;

ThJoint(i) = 20 + (0.0061002) \*time(i)^5 + (-0.152505)\*time(i)^4 + (1.0167) \* time(i)^3;

ActualBaseVel(i) = (-5\*(9.684\*(10^-4))\*time(i)^4) + (4\*0.02421\*time(i)^3) - (3\*0.1614\*time(i)^2);

ActualJointVel(i) = ((5\*(6.1002\*(10^-3)))\*time(i)^4) - (4\*0.152505\*time(i)^3) + (3\*1.0167\*time(i)^2);

end

%2 Point Reverse

for i = 2:length(time)

VelBaseEst(i) = (ThBase(i) - ThBase(i-1))/dt;

VelJointEst(i) = (ThJoint(i) - ThJoint(i-1))/dt;

end

%Plot Actual/Estimated

figure

plot(time,ActualBaseVel)

hold on;

plot(time,VelBaseEst)

plot(time,ActualJointVel)

plot(time,VelJointEst)

xlabel('Time (sec)')

ylabel('Velocity (deg/sec)')

legend('Actual Base Vel', 'Base Vel Est.','Actual Joint Vel','Joint Vel Est.')

title('Estimated Joint/Base Velocity vs. Actual Velocity')

hold off;

for i = 1:length(time)

VelBaseError(i) = abs(ActualBaseVel(i) - VelBaseEst(i));

VelJointError(i) = abs(ActualJointVel(i) - VelJointEst(i));

end

figure

plot(time, VelBaseError)

hold on;

plot(time, VelJointError)

title('Velocity Estimation Error')

legend('Base Error','Joint Error')

xlabel('Time (sec)')

ylabel('Velocity (deg/sec)')

MaxBaseVelError = max(VelBaseError);

MaxJointVelError = max(VelJointError);

%Finding Coordinates of Base/Joint

for i = 1:length(time)

YBase(i) = 5\*sin(ThBase(i));

XBase(i) = 5\*cos(ThBase(i));

XJoint(i) = 4\*cos(ThJoint(i));

YJoint(i) = 4\*sin(ThJoint(i));

end

% for i = 1:14

% plot(XBase(1),YBase(i),'ro')

% hold on;

% plot(XJoint(i),YJoint(i),'bo')

% pause();

% end

%Displacement of Gripper from Arm start

for i = 1:length(time)

Finalx(i) = XBase(i) + XJoint(i);

Finaly(i) = YBase(i) + YJoint(i);

end

**Problem 2:** Repeat Problem 1 to estimate the angular acceleration of the base and joint by applying the 2nd derivative estimate to the base and joint angles.

**PLOTS OF ESTIMATED AND ACTUAL ACCELERATIONS (Joint and Base)**

****

**PLOTS OF ABSOLUTE ESTIMATION ERROR (Joint and Base)**



**Maximum Estimation Error for Base Angular Acceleration: \_\_\_\_0.023150812500002\_\_\_\_\_\_\_\_\_\_**

**Maximum Estimation Error for Joint Angular Acceleration: \_\_\_\_\_0.145832906250004\_\_\_\_\_\_\_\_\_**

**PASTE MATLAB SCRIPT HERE:**

%Kyle O'Connor

%

%HW 4

%

%2/27/2016

home;

clear all;

close all;

clc;

Lbase = 5;

Ljoint = 4;

time = 0:.75:9.75;

dt = time(2) - time(1);

%Finding Base and Joint Angles/Velocities

for i = 1:length(time)

ThBase(i) = 30 + (-9.684\*(10^-4))\*time(i)^5 + (0.02421)\*time(i)^4 + (-.1614)\*time(i)^3;

ThJoint(i) = 20 + (0.0061002) \*time(i)^5 + (-0.152505)\*time(i)^4 + (1.0167) \* time(i)^3;

ActualBaseVel(i) = (-5\*(9.684\*(10^-4))\*time(i)^4) + (4\*0.02421\*time(i)^3) - (3\*0.1614\*time(i)^2);

ActualJointVel(i) = ((5\*(6.1002\*(10^-3)))\*time(i)^4) - (4\*0.152505\*time(i)^3) + (3\*1.0167\*time(i)^2);

ActualBaseAcc(i) = (-20\*(9.684\*(10^-4))\*time(i)^3) + (12\*0.02421\*time(i)^2) - (6\*0.1614\*time(i));

ActualJointAcc(i) = (20\*(6.1002\*(10^-3))\*time(i)^3) - (12\*0.152505\*time(i)^2) + (6\*1.0167\*time(i));

end

%2 Point Reverse

for i = 2:length(time)

VelBaseEst(i) = (ThBase(i) - ThBase(i-1))/dt;

VelJointEst(i) = (ThJoint(i) - ThJoint(i-1))/dt;

end

%Second Derivative

for i = 2:length(time)-1

AccBaseEst(i) = (ThBase(i+1) - 2\*ThBase(i) + ThBase(i -1))/(dt^2);

AccJointEst(i) = (ThJoint(i+1) - 2\*ThJoint(i) + ThJoint(i-1))/(dt^2);

end

%Velocity Plot/Error

figure

plot(time,ActualBaseVel)

hold on;

plot(time,VelBaseEst)

plot(time,ActualJointVel)

plot(time,VelJointEst)

xlabel('Time (sec)')

ylabel('Velocity (deg/sec)')

legend('Actual Base Vel', 'Base Vel Est.','Actual Joint Vel','Joint Vel Est.')

title('Estimated Joint/Base Velocity vs. Actual Velocity')

hold off;

for i = 1:length(time)

VelBaseError(i) = abs(ActualBaseVel(i) - VelBaseEst(i));

VelJointError(i) = abs(ActualJointVel(i) - VelJointEst(i));

end

figure

plot(time, VelBaseError)

hold on;

plot(time, VelJointError)

title('Velocity Estimation Error')

legend('Base Error','Joint Error')

xlabel('Time (sec)')

ylabel('Velocity (deg/sec)')

MaxBaseVelError = max(VelBaseError);

MaxJointVelError = max(VelJointError);

%Acc Plotting/Error

figure

plot((1:length(time)-1),AccBaseEst,'r')

hold on;

plot((1:length(time)-1),AccJointEst,'g')

plot((1:length(time)),ActualBaseAcc,'b')

plot((1:length(time)),ActualJointAcc,'k')

title('Accleration Estimate vs Actual')

xlabel('Time (sec)')

ylabel('Accleration (deg/sec^2)')

legend('Base Est','Joint Est','Actual Base','Actual Joint')

AccBaseError(14) = 0;

AccJointError(14) = 0;

for i = 1:length(time)-1

AccBaseError(i) = abs(ActualBaseAcc(i) - AccBaseEst(i));

AccJointError(i) = abs(ActualJointAcc(i) - AccJointEst(i));

end

figure

plot(time,AccBaseError)

hold on;

plot(time,AccJointError)

title('Accleration Estimation Error')

xlabel('Time (sec)')

ylabel('Accleration (deg/sec^2)')

legend('Base Error', 'Joint Error')

MaxBaseAcc = max(AccBaseError);

MaxJointAcc = max(AccJointError);

**Problem 3:** The results in Problems 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:**

dt = 0.01;

copydt = dt;

MaxJointVelError = 0;

while(MaxJointVelError < 1)

dt = dt + 0.01;

time = 0:dt:9.75;

for i = 1:length(time)

ThJoint(i) = 20 + (0.0061002) \*time(i)^5 + (-0.152505)\*time(i)^4 + (1.0167) \* time(i)^3;

ActualJointVel(i) = ((5\*(6.1002\*(10^-3)))\*time(i)^4) - (4\*0.152505\*time(i)^3) + (3\*1.0167\*time(i)^2);

end

for i = 2:length(time)

VelJointEst(i) = (ThJoint(i) - ThJoint(i-1))/dt;

end

VelJointError = abs(ActualJointVel - VelJointEst);

MaxJointVelError = max(VelJointError);

end

dt = dt- copydt;

**Checked for every value of Dt from 0 up until the MaxJointVelError was greater than one**

**And than took one incrementation away to get the Dt where MaxJointVelError was below one.**

**Problem 4:** This problem is designed to illustrate the effect of noise on derivative estimates. In some cases, a seemingly small amount a noise can be greatly amplified when taking derivatives of a noisy signal. As an example, consider the following signal: 0.01sin(1000t). The signal has a very small amplitude of 0.01. What happen when we take a derivative of this signal? Suddenly the amplitude grows by a factor of 1000 to an amplitude of 10! A 2nd derivative causes the amplitude to grow by a factor of 1 million to an amplitude 10,000!

For this problem, you need the data file, HW4.mat, posted on the Blackboard metasite with Homework #4. The MATLAB command: load HW4 will load the data into the MATLAB workspace. The data file has 2 vectors:

* t is a vector of times (seconds) starting at 0, incrementing by 0.34, and ending at 9.86 s.
* Th2\_n is a vector of noisy angular position measurements of the joint angle (deg) taken at the times in vector t.

1. Calculate the actual joint angular position at the times in vector t using the equation given at the beginning of this assignment. On the same plot, plot the actual angular position and the noisy angular position measurements.

**PLOT of Actual Joint Angle and the Sensor Measurements:**



**Do the measurements look very noisy to you?**

**The measurements do not look noisy at all.**

1. Using the noisy angular joint position measurements, calculate a 2 PT reverse estimate of the angular velocity. Plot the estimates on the same graph as the actual joint angular velocity.

**PLOT of Angular Velocity Estimate vs. Actual Angular Velocity:**



**Comment on the Accuracy of the Estimates:**

**The estimate is fairly close to the actual it seems as though the actual starts sooner than the estimate.**

1. Using the noisy angular joint position measurements, calculate an estimate of the angular acceleration of the joint. Plot the estimates on the same graph as the actual joint angular acceleration.

**PLOT of Angular Acceleration Estimate vs. Actual Angular Acceleration:**



**Comment on the Accuracy of the Estimates:**

**The accuracy of the estimate is far from the actual.**

**PASTE MATLAB CODE FOR PARTS (a) – (c):**

1. What have you learned from this problem?

%Kyle O'Connor

%

%HW 4

%

%2/27/2016

home;

clear all;

close all;

clc;

load HW4

dt = t(2)-t(1);

for i = 1: length(t)

ThJoint(i) = 20 + (0.0061002) \*t(i)^5 + (-0.152505)\*t(i)^4 + (1.0167) \* t(i)^3;

ActualJointVel(i) = ((5\*(6.1002\*(10^-3)))\*t(i)^4) - (4\*0.152505\*t(i)^3) + (3\*1.0167\*t(i)^2);

ActualJointAcc(i) = (20\*(6.1002\*(10^-3))\*t(i)^3) - (12\*0.152505\*t(i)^2) + (6\*1.0167\*t(i));

end

plot(t,ThJoint)

hold on;

plot(t,Th2\_n)

title('Actual vs. Noisy')

xlabel('Time (sec)')

ylabel('Joint Angle Position')

legend('Actual','Noisy')

for i = 2:length(t)

VelJointEst(i) = (Th2\_n(i) - Th2\_n(i-1))/dt;

end

for i = 2:length(t)-1

AccJointEst(i) = (Th2\_n(i+1) - 2\*Th2\_n(i) + Th2\_n(i-1))/(dt^2);

end

figure

plot(t,VelJointEst)

hold on;

plot(t,ActualJointVel)

xlabel('Time')

ylabel('Velocity')

legend('Estimate','Actual')

title('Actual/Estimate Velocities')

AccJointEst(30) = 0;

figure

plot(t,AccJointEst)

hold on;

plot(t,ActualJointAcc)

xlabel('Time (sec)')

ylabel('Accleration')

title('Noisy vs. Actual Acceleration')

legend('Estimate','Actual')