UNIVERSITY COLLEGE DUBLIN

SCHOOL OF ELECTRICAL & ELECTRONIC ENG.

EEN30110 SIGNALS AND SYSTEMS

Experiment 3SS1

Linear, constant coefficients ODEs and System Analysis

1. Objective:

To investigate linear, constant coefficient ODEs and analysis of LTI systems using Matlab.

2. Background Information/Practice Problem:

See information in introductory laboratories and module notes.

Linear, constant coefficient ODEs offer commonly acceptable models for the behaviour of systems close to some desired operating point. We will investigate a toy example:

$$\ddot{x} + 0.4\dot{x} + 4x = 0.2\dot{f} + f\ .$$

Following the lecture notes the associated polynomials are $D(s) = s^2 + 0.4s + 4$ and $N(s) = 0.2s + 1$. These are presented to Matlab as vectors of the coefficients in descending order of powers of *s*, with *all* coefficients included (i.e. zero coefficients produce a component 0 in the vector).

```
>> N = [0.2 1]
>> D = [1 0.4 4]
```

In this case the transfer function is

$$H(s) = \frac{N(s)}{D(s)} = \frac{0.2s + 1}{s^2 + 0.4s + 4}\ .$$

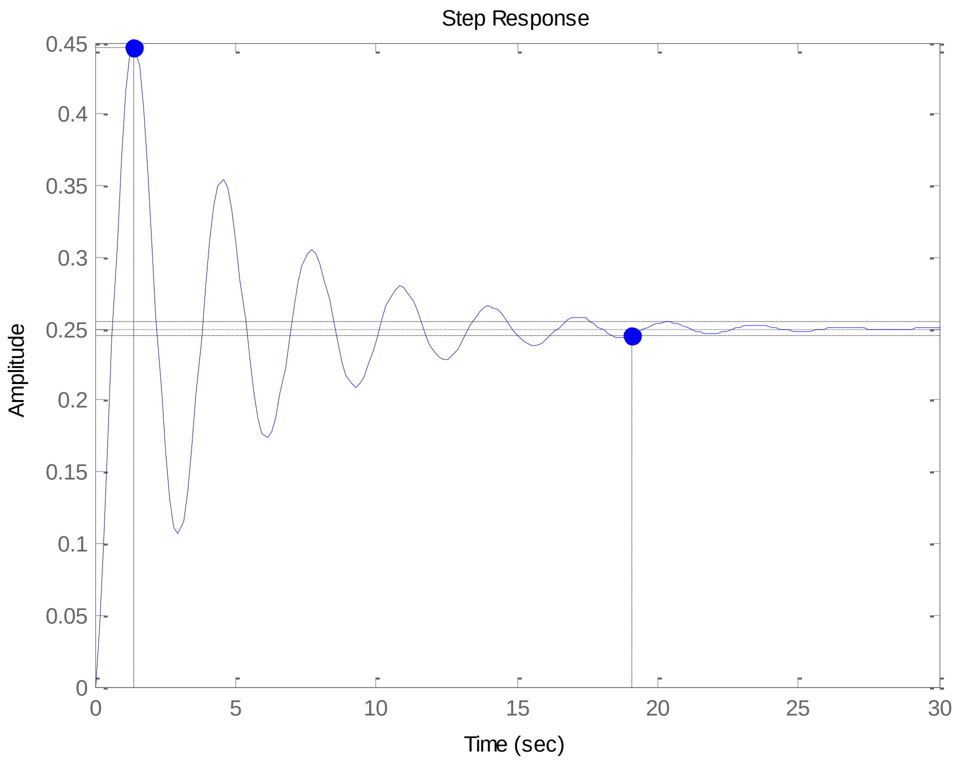
We may carry on to create a *system variable* corresponding to this transfer function.

```
>> H = tf(N,D)

Transfer function:
       0.2 s + 1
-----------------------
  s^2 + 0.4 s + 4
```

Of interest commonly is the response when the forcing term *f*(*t*) equals the unit step. The Matlab command **step** will calculate and plot the step response as follows.

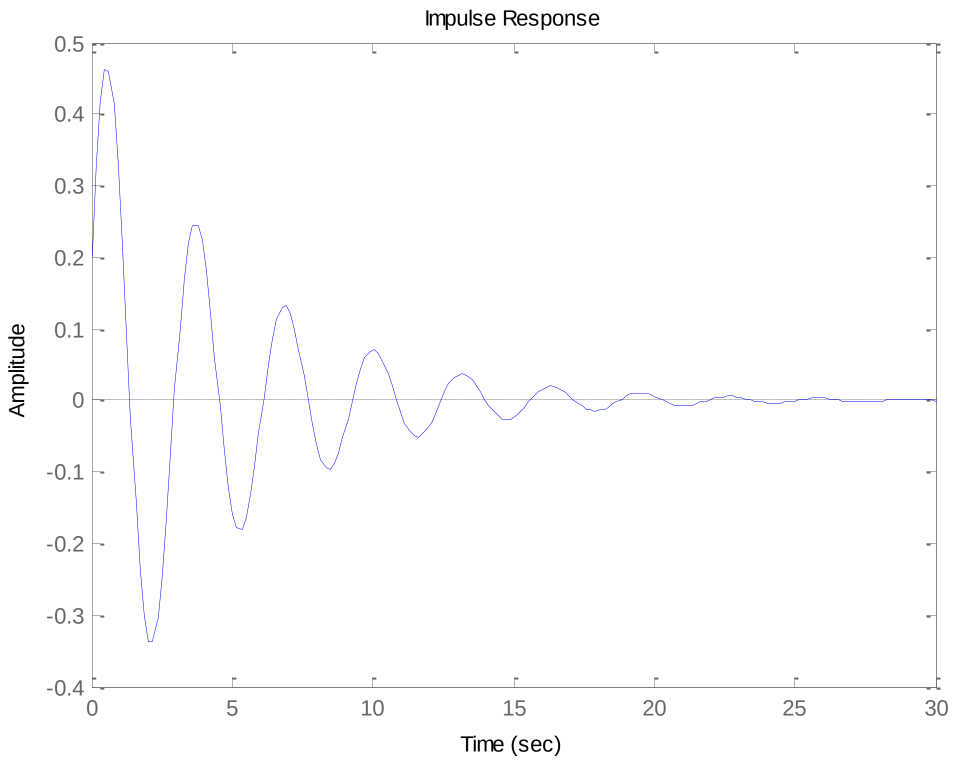
```
>>  step(H)
```



I have employed some of the interactive tools which accompany the plot. By right clicking on the figure background, selecting **characteristics** from the drop down menu and selecting both **Settling Time** and **Peak Response** I ask Matlab to place the dots shown. By moving the cursor over each of the dots in turn I am informed that the *Settling Time* is 19.1 sec and the *Percentage Overshoot* is 78.7% These quantities obviously give some measure of the size and duration of the transient component of the step response of the system.

I note that, should I desire, the **impulse** command will calculate and plot the impulse response of the system, i.e. the inverse Laplace transform of the transfer function $H(s)$.

```
>> impulse(H)
```



We find that the impulse response is rather oscillatory and that it takes about 25 to 30 sec to decay away. To confirm we may employ Matlab to actually find a formula for the impulse response. We use the partial fraction expansion method covered in lectures.

```
>>  [R,P,K] = residue(N,D)

R =

   0.1000 - 0.2412i
   0.1000 + 0.2412i


P =

  -0.2000 + 1.9900i
  -0.2000 - 1.9900i


K =

     []
```

As usual the vector K is null and we just ignore it. The poles are -0.2 +/- 1.99j. Note that the poles should equal the roots of the polynomial $D(s)$. We may confirm this by using the Matlab command **roots**.

```
>> roots(D)

ans =
```

-0.2000 + 1.9900i
-0.2000 - 1.9900i

Alternatively we may use the command **pole**.

```
>> pole(H)

ans =

  -0.2000 + 1.9900i
  -0.2000 - 1.9900i
```

All of these commands confirm that the system has two poles, that they are complex conjugates of one another, that they have negative real part (although it is not very negative) and that their imaginary part is a good bit larger in magnitude than their real part. The latter makes them rather oscillatory and explains why both the impulse response and the step response have strong oscillatory components. Now to pursue the evaluation of the impulse response it is a matter of interpreting the information given by the **residue** command to achieve:

$$H(s) = \frac{0.2s+1}{s^2+0.4s+4} = \frac{0.1-0.2412j}{s+0.2-1.99j} + \frac{0.1+0.2412j}{s+0.2+1.99j}$$

Now the general formula for the inverse Laplace transform of a partial fraction expansion may be applied, giving:

$$h(t) = (0.1-0.2412j)e^{-0.2t}e^{j1.99t} + (0.1+0.2412j)e^{-0.2t}e^{-j1.99t} \quad \text{for } t \geq 0.$$

As usual, whereas this is correct it is unacceptable. There are cancellations of imaginary terms which can be effected here. The answer, being the response to a real signal of a real system must be itself a real signal. The key is to find the polar expression of the complex number 0.1 – 0.2412j. This number indeed is the first element of the residue vector R created by the **residue** command above. Accordingly it is accessible to us as R(1).

```
>>  R(1)

ans =

   0.1000 - 0.2412i
```

There are many ways to convert a complex number to polar form in Matlab, but a simple way is to use the **abs** and **angle** commands.

```
>>  abs(R(1))

ans =

    0.2611

>>  angle(R(1))

ans =
```

-1.1778

I deduce that $(0.1-0.2412j)=0.2611e^{-j1.1778}$. Accordingly:

$$h(t)=0.2611e^{-0.2t}e^{j(1.99t-1.1778)}+0.2611e^{-0.2t}e^{-j(1.99t-1.1778)}=0.2611e^{-0.2t}\left(2\cos(1.99t-1.1778)\right)$$

$$h(t)=0..5222e^{-0.2t}\cos(1.99t-1.1778).\ \text{for}\quad t\geq 0.$$

I may plot this function vs time for the time-frame 0 to 30 sec over which Matlab plotted the impulse response which it calculated. I first create a vector of times at which to evaluate the signal.

```
>>  t = [0:0.01:30];
```

Do not forget the semi-colon at the end here. We do not wish to see all of this printed to the screen.

```
>>  hsamp = 0.5222*exp(-0.2*t).*cos((1.99*t)-1.1778);
```

Note the use of the *term-wise multiply* .* Note also that I was able to do this with one command, I did not need a loop.

```
>>  plot(t,hsamp)
```

You should see that the resulting plot is exactly the same as that produced by Matlab's **impulse** command except for axis labels and figure title. I may, if I wish, use the **xlabel**, **ylabel** and **title** commands to produce the required labels and title.

The dynamics of an aircraft are a little complicated. The Wikipedia article on flight dynamics (fixed-wing aircraft) provides some good visualisation. In describing the aircraft we generally choose a co-ordinate system which is called *body-fixed*. In this system the x-axis points along the longitudinal axis of the aircraft from tail to nose. The z-axis is taken to point "downwards". It is perpendicular to the plane described by the wings and the longitudinal axis. The y-axis is chosen parallel to the wings. The positive direction is picked so that the overall co-ordinate system obeys the right hand rule, i.e. the y-axis is directed from the left-hand side (port) to the right-hand side (starboard) of the aircraft. The aircraft can and does rotate somewhat about each of these body-fixed axes. Rotation about the x-axis is called *roll*. Rotation about the y-axis is called *pitch*. Rotation about the z-axis is called *yaw*. The pitch angle is important since it is determining the degree to which the nose of the aircraft is pointing up or down. To control the pitch of the aircraft there are flight control surfaces called *elevators*. These surfaces, usually located at the rear of the aircraft, produce a pitching moment and can as a result change the pitch angle. A further important angle is the *angle of attack*. Essentially it is the angle between the x-y body-fixed plane of the aircraft and the velocity vector of the oncoming air flow. It transpires, after a really impressive collection of assumptions and approximations, that if the aircraft is in steady flight at a constant velocity then the angle of attack α, the pitching angle θ and the elevator setting δ_e are related through a system of ODEs. These equations are said to describe the *longitudinal dynamics* of the aircraft.

3. Problems

1. The longitudinal dynamics of an aircraft in steady flight are approximately described by the following equations:

$$\dot{\alpha} = a_{11}\alpha + a_{12}\dot{\theta} + b_1\delta_e$$
$$\ddot{\theta} = a_{21}\alpha + a_{22}\dot{\theta} + a_{23}\dot{\alpha} + b_2\delta_e$$

where α is the offset of the angle of attack (i.e. the difference between this and the desired value or operating value), θ is offset of the pitching angle (i.e. again the difference between this and the operating value) and δ_e is the elevator setting. The coefficients appearing are just parameters which depend upon the aircraft and its desired operating point (including its desired speed). Take the following values for these parameters.

$$a_{11} = -0.313\,,\ a_{12} = 1\,,\ a_{21} = -0.79\,,\ a_{22} = -0.426\,,\ a_{23} = 0\,,\ b_1 = 0.232$$

For b_2 you must generate a parameter value as follows: take the last two digits of your student card number a_1, a_0 . Calculate the number $c = 10a_1+a_0$. Execute the code

```
>>  b2 = 1.1+(fix((c+17)/14)/100)
```

to find the value of parameter b_2 which you personally are to employ.

Regarding the system described by these equations, with these parameter values as a linear, time-invariant system with input equal to elevator setting and output equal to pitching angle find the transfer function. The problem is made much easier by going into the s-domain straight away. Accordingly apply the Laplace transform directly to the given differential equations. This turns the problem into a relatively simple algebraic problem.

2. Assuming all initial conditions to be zero, i.e. that the aircraft is initially at its operating point, find a mathematical/analytic expression for the solution to the given differential equations assuming that the elevator setting is zero for $t < 0$ and is suddenly changed to 0.1 at $t = 0$ being subsequently held at this value. You must employ the partial fraction expansion method. Accordingly the Matlab command **residue** will be useful. You may certainly check your answer by using on-line tools such as Wolfram Alpha, but the grades are not given for the correct answer, they are given for the correct answer as calculated by you. Is there anything about the solution which might cause you to question its validity? This question notwithstanding, confirm the solution by employing the built-in Matlab command **step**.

3. The elevator setting is actually achieved by means of an actuator. The input to this actuator will be an elevator control command u_e (often physically a voltage). The output is the actual elevator setting δ_e. Adopt a very simple model for the actuator:

$$\tau\dot{\delta}_e + \delta_e = u_e$$

where τ is a parameter of the actuator model called the *time constant*. Assume a time constant of 0.1 sec. The actuator is therefore modelled as a linear, time-invariant system with an input equal to the elevator control signal u_e and an output equal to the actual elevator setting δ_e. Find the transfer function of this system with the given choice of time-constant.

4. We now have a new linear time-invariant system whose input is the elevator control signal u_e and whose output is the pitching angle θ. Find the transfer function of this system and find its response to an elevator control which is zero for $t < 0$ then jumps to 0.1 at $t = 0$ and remains at this value thereafter. Again the answer must be in the form of a formula and again the grades are assigned mainly to your calculation of that formula. The checking of your answer by online tools is always a good idea but will not earn any grade points. Once more

you may check your answer by employing a number of Matlab commands, but most easily by employing the command **step**.

5. At present the described elevator control signal, since it is maintained for all $t > 0$ will call for the aircraft to perform a *loop the loop* manoeuvre. Obviously there is not much call for this in a passenger aircraft. To control the pitch angle we must measure the actual pitch angle and use the difference between the pitch angle and the desired pitch angle to drive the control. We use the same kind of control loop as in the case of the robot wheel position control problem considered in lectures. Let the sensor which measures the pitch angle be modelled simply as a gain.

$$\theta_{meas} = k_{sens}\theta$$

where the sensor gain $k_{sens} = 0.2$. The effect of the control loop will be:

$$u_e = k_{sens}r - \theta_{meas} \ .$$

Draw a block diagram model for the overall control loop.

6. We now have a linear, time-invariant system with input r and output θ. Find the transfer function of this system. Write down the linear, constant coefficient ODE which describes this system. Find the response of this system to a reference input command r which is zero for $t < 0$ then jumps to 0.1 at $t = 0$ and remains at this value thereafter. Again the answer must be in the form of a formula and again the grades are assigned mainly to your calculation of that formula. Once more you may check your answer by employing the Matlab command **step**.

7. You should observe that, whereas the system response is that the pitch angle becomes equal to the desired angle as specified by the reference input r, the time which it takes for the angle to reach this value is really rather excessive, given that we are dealing with an aircraft. We may change the control law by taking:

$$u_e = k_p\left(k_{sens}r - \theta_{meas}\right).$$

The parameter k_p is said to be a *proportional gain*. Investigate the effect if any of this parameter upon the performance and in particular upon the speed of response.