Week4 (Time Domain Analysis) –ZIR

- Use "**ode23**" function to solve the Differential Equation (DE) in MATLAB with end result in array form.

  Loadable Function is: [t, y] = od**e23** (*fcn, t, x_o*)

  Eg: [t,y] = ode23(@cf2orderode23,[0 20],[−1;1])

- These functions (ode23) use the appropriate numerical method algorithms such Runge-Kutta, Euler, etc.

The solution is returned in the matrix **y** with each row values are the solutions to the DE corresponding to **each of the time instants** that are defined as elements of the **vector *t***. The value of first element of *t* should be the **initial time t_o** and the solution correspond to this **initial time** is the **initial state** of the system represented as **column vector $x_o = x(t_o)$**, and is supplied as arguments in 'ode23' functions so that the first **row** of the output y is x_o. The '***n***' **column** elements of the output matrix y**,** are the '***n***' linearly independent system variables (solutions to DE) (output and successive derivatives (***n −1*** times) of the output variable, where '***n***' is the order of the system) values at the corresponding instant of time.

In the right-hand side of the loadable function, the first argument is 'ode23', which calls *fcn*, which is a string, inline, or function handle to compute the '***n***' set of ***first order*** DE. The function definition must create separately of the form: y*dot* = *fcn*(*t*, *y*);
Here, the '***n***' set of ***first order*** DE should be entered inside this created function file with a name '*fcn*', and *ydot* and *y* are vectors of size equal to the order of the system '***n***'.

Finding Zero –Input Response (ZIR):

Eg: Let the differential equation, $(D^2+4D+3)\ y(t) = (3D+5)\ x(t)$ and with initial conditions: $y(0) = -3$; $dy(0)/dt = -6$.

A n^{th} order DE can be written in '***n***' **first order** DEs as (in this case ***n*** = 2):

Let $y_1 = y$ and $y_2 = Dy(t)$ which implies, $Dy_1(t) = y_2(t)$ ---- (1)

and substituting these in the given above differential equation, we get,

$$Dy_2(t) = -4y_2(t) - 3y_1(t) + 3D\ x(t) + 5\ x(t) \quad \text{---- (2a)}$$

Since x(t) = 0 (for ZIR), the Eqn. (2a) modifies as, $Dy_2(t) = -4y_2(t) - 3y_1(t)$ ---- (2)

These equations (1) and (2) should be defined in the **function file** "**fn1.m**" as:

```
function ydot = fn1(t, y)      <------ First line of the created function file with name 'fn1.m'
ydot = zeros (2,1);            <---- creating null column matrix ydot for the entry of '2' first order DEs
ydot(1) = y(2);                <---- entry of first DE, using Eqn. (1)
ydot(2) = −4*y(2)−3*y(1);      <---- entry of second DE, using Eqn. (2)
end
```

While creating the function file, "**fn1.m**" the first statement should start with the function command and the right-hand side of the equal sign of this statement should start with same name as the name given to the respective function file.

Once the function file '**fn1.m**' is created with DEs entered in it, the solution (numerical integration) for these DEs can be obtained by calling this function file "**fn1.m**", through "**ode23**" in a separate script file with initial conditions defined as:

```
>> y0 = [−3;−6];                <---- initial condition vector
>> t = [0:0.01:10];             <---- time duration for the solution
>> [t,y] = ode23(@fn1, t,y0);   <---- calling the function for solution
```

The solution thus calculated is in matrix form **y** is sketched using plot command as:

```
>> plot(t,y(:,1),t,y(:,2));     <--- sketch of two linearly independent solutions
```

y(:,1) represents y1 the output variable (one of the solution to DE) and y(:,2) represents y2, the derivative of the output variable (2nd independent variable of the solution to DE).

Finding Impulse Response:

For the above example, evaluate the new set of initial conditions due to the application of impulse at $t = 0$, as $h(0^+) = K_1$ and $Dh(0^+) = K_2$ using the method-1. Compute the values of K_1 & K_2 manually. This result in, $h(0^+) = 3$ and $Dh(0^+) = -7$ after the hand calculation. Use this as the initial conditions in MATLAB to solve the differential equation as explained above for ZIR.

Exercises

1. Following are the Differential Equations (DE) of the LTIC system,
   a) $D(D+1)y(t) = (D+2)x(t)$
   b) $(D^2+9)y(t) = (3D+2)x(t)$ AND practice $(D^2-9)y(t) = (3D+2)x(t)$
   c) $(D^2+D+13)y(t) = 4(D+2)x(t)$ AND practice $(D^2-D+13)y(t) = 4(D+2)x(t)$
   d) $D^2+1.5D+0.5625)y(t) = 4(D+2)x(t)$ AND practice $D^2-1.5D+0.5625)y(t) = 4(D+2)x(t)$
   e) $D^2\,(D+1)y(t) = (D^2+2)x(t)$

   i. Find manually the **zero input response** $\{y_o(t)\}$ given initial conditions: $y(0^-) = 2$, $\dot{y}(0^-) = -3$.
   ii. Using "**ODE**23" function in MATLAB get $y_o(t)$ in array form and sketch the same. Also sketch $\dot{y}(t)$ vs. t.
   iii. Identify the characteristic polynomial, characteristic roots, characteristic modes of the system.
   iv. Give comments on which class of signal model the output response belongs to.
   v. Find the steady state values.
   vi. Find the approximate time taken by the output response to reach steady value.
   vii. Find the period of oscillation of output response (if applicable).

2. Find manually the **unit impulse response** of a system specified by the Differential Equations, in exercise 1a) to 1c). Use "**ODE23**" function in MATLAB to solve the unit impulse response for these DEs.

3. Try using "**dsolve**" function to solve the Differential Equations in symbolic form.

NOTE: **Solve all of the above Differential Equations manually (i.e. hand calculation) before entering the lab. Weightage is mainly based on the how well you prepare and come.**