ECE 3723 - Electric Circuits II - Fall 2021
Project 3: Signals and Transfer Functions

Due: 10/05/21 on Canvas

Introduction

In this project some elemental signal waveforms will be explored. How to set up a transfer function in MATLAB® will be investigated. Finally, systems will be implemented and analyzed. Some of these systems are going to be broken into cascades consisting of multiple components or transfer functions. Such systems are very commonly used in a wide range of engineering solutions. Examples include control circuitry, transmitter and receivers, electrical power distribution, and many more.

Finally, a method for doing partial fraction expansion for Laplace transform will be introduced.

1 Unit Step Function

The unit step function is commonly used to add discrete behavior to a signal, such as turning it on or turning it off. The definition of the unit step function is:

$$Ku(t) = \begin{cases} K & t > 0 \\ 0 & t < 0 \end{cases}$$

where K is a constant. In general this function can be viewed as a switch, which is only on if the input parameter is greater than zero. Two main methods can be used to implement the unit step function. 1) using conditional statements and 2) using a built in function in MATLAB® called heaviside. For this project either method is acceptable. Please be aware of the number of points and how MATLAB® interpolates between points when plotting.

- Create a time vector, `t`, where $-5 \leq t \leq 5$ seconds. The number of elements can be chosen arbitrarily, however, it will affect the remainder of this section, so give it some thought.
- Now create a vector, `u1`, that represents the unit step function, $u(t)$, with $K = 1$.
- Create two new vectors, `u2` and `u3`, which should represent a shifted version of the unit step function by 1 second and 2 seconds, respectively, or $u(t-1)$ and $u(t-2)$.
- Look up the function `subplot`. It can be used to create a single figure that contains multiple graphs.
  Show a figure that contains graphs of the all three vectors as subplots in a 3x1 configuration (5 points).
- Perform the following operations.

$$\begin{aligned} y_1 &= u(t) - u(t-1) \\ y_2 &= u(t) + u(t-1) + u(t-2) \\ y_3 &= u(t)u(t-2) \\ y_4 &= u(t) + u(t-1) - 2u(t-2). \end{aligned}$$

  Plot these functions in a single figure with the graphs in a 2x2 configuration (10 points).

2 Signals in Time-Domain

Amongst very commonly used signals are sinusoidal signals such as $x(t) = A\cos(\omega t + \theta)$, where A is the amplitude, ω is the frequency in radians per second, and θ is the phase.

- Start by creating a time vector, `t`, where $-2 \leq \mathrm{t} \leq 2$ seconds.
- Create the vectors for the following functions:

$$\begin{aligned} v_0(t) &= \cos(2\pi t)u(t) \\ v_1(t) &= \cos(2\pi t + \pi/4)u(t) \\ v_2(t) &= \cos(0.5\pi t)u(t) \\ v_3(t) &= v_0(t)v_2(t) \\ v_4(t) &= e^{-|t|}v_0(t). \end{aligned}$$

Plot the following in a single plot, 2x2 configuration: 1) $v_0(t)$ **and** $v_1(t)$**, 2)** $v_0(t)$ **and** $v_2(t)$**, 3)** $v_0(t)$ **and** $v_3(t)$**, and 4)** $v_0(t)$ **and** $v_4(t)$**. There should be four graphs, each with two traces (10 points).**

3 Transfer Functions

Transfer functions are used extensively in electrical engineering and systems are seldom built without doing a full signal analysis. In most ideal cases frequency domain analysis can be used, but there are some cases where the time-domain manipulation is needed. For example when dealing with experimental data which does not have an analytic representation. In this project a brief introduction on handling frequency domain transfer functions in MATLAB®.

- The function `tf` can be used to create a transfer function. Look up the syntax for the function. Do not turn in a printout of the syntax.
- Define

$$H_1(s) = \frac{1}{s+1}$$

- Now define

$$H_2(s) = \frac{2s^2 + 3s + 2}{s^3 + 2s + 1}$$

- Multiply the functions together. What do you get? **Show the new transfer function (5 points).**

4 Partial Fraction Expansion and Laplace Transform

Some times we prefer to keep the numerator and denominator separate. This is the case when we deal with Laplace transform in MATLAB®.

4.1 Basic polynomial functions

- Create the vector `F1 = [1 5 6]`.
- Look at the polynomial by using `poly2str(F1,'s')`.

- To evaluate the function at some $s = a$, you can use `polyval(F1,a)`. **What is the value at $s = 1$ (5 points)?**
- Find the roots of the polynomial using `roots(F1)`. **Show the factored polynomial (5 points).**
- The function `poly(x)` takes in roots and creates a polynomial. For $F2 = (s+1-2j)(s+1+2j)(s+1)s^2$, use poly to create the polynomial `F2`. **Show the resulting polynomial. Be sure to account for the two roots at $s = 0$ (5 points).**
- We can use the convolution function to multiply two polynomials. Create the vector `H = conv(F1,F2)`. **Show the resulting polynomial. Also, verify by multiplying the two polynomial (F1 and F2) by hand. Show all your work (write it up on the computer, no handwritten solutions will be accepted) (10 points).**

4.2 Partial fraction expansion for simplifying Laplace transform

- Create two polynomials, `n` and `d`, where `n` is the numerator and `d` is the denominator of the function

  $$H_3(s) = \frac{s+1}{s(s+4)}.$$

  Look at the transfer function by using the `tf` function. Show the output (5 points).
- We can use the function `residue` to perform partial fraction expansion. For example if we look at $H_3(s)$

  $$H_3(s) = \frac{s+1}{s(s+4)} = \frac{K_1}{s} + \frac{K_2}{s+4}.$$

- We simply enter the numerator and denominator into the `residue` function: `[r,p,k]=residue(n,d)`. The function outputs the magnitude of the expansion terms as `r` and `p` shows the poles that are corresponding to each value of `r`. Also, the function outputs the polynomial resulting from any long division, if needed, in the vector `k`. `k` is empty if the function is a proper rational function.
- Use the residue function on the H_3. The output will be in the form:

  $$H_3(s) = \frac{\texttt{r(1)}}{s - \texttt{p(1)}} + \frac{\texttt{r(2)}}{s - \texttt{p(2)}}.$$

- Then the inverse Laplace transform can be used to transform into time-domain.
  Calculate by hand the inverse Laplace transform and show in the report (5 points).
- When dealing with repeated roots we need to be careful with the order of the outputs from the `residue` function. The output will list the repeated poles with the lowest order first. For example if we have the function

  $$H_4(s) = \frac{s^2+4}{s^2(s+4)}.$$

  The output of the `residue` function will be in the form:

  $$H_4(s) = \frac{\texttt{r(1)}}{s - \texttt{p(1)}} + \frac{\texttt{r(2)}}{s - \texttt{p(2)}} + \frac{\texttt{r(3)}}{(s - \texttt{p(3)})^2},$$

  which is inverse to what we typically do in class.

- Use the residue method in MATLAB® to perform the partial fraction expansion of

$$H_5(s) = \frac{4s^3 + s^2 + 4}{s^3(s+3)^2(s^2+2)}.$$

  Note that this function has complex roots, so the coefficients will include some complex coefficients. There are a couple of useful functions that should be kept in mind: `abs` gives the magnitude of a complex number, and `angle` gives the phase angle in radians (note that MATLAB® is set to radians by default, so all trig will need to be with angles in radians).

- Then take the inverse Laplace transform. **Show both the function in frequency-domain and time-domain (5 points).**

5 Deliverables

This project should be written up in a neatly organized report that includes the results from all the sections, answers to all the questions, and all plots that are generated. All the code generated for this project must be included in an appendix (in small font, less than 12pt).

6 Grading

- The report itself is worth 20 points, which will be given based on the structure of the report, grammar, clarity, and formatting.
- Answering the questions listed in the handout in the main body of the report is worth 70 points.
- Turning in the code in an appendix is worth 10 points.