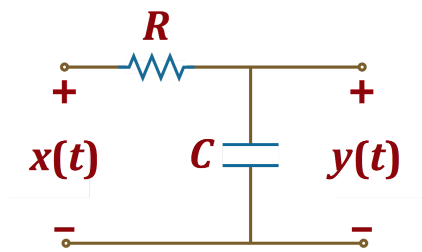
Computing Exercises

Exercise SS-C1: Understanding Convolution



[20pts] Consider the RC circuit shown to the right, with $R = 10\Omega$ and $C = 2\text{mF}$. The "input" voltage is $x(t)$, and the "output" voltage is $y(t)$.

The input voltage is given by:

$$x(t) = V_0 \cdot \text{rect}\left(\frac{t-\tau}{2\tau}\right)$$

where $V_0 = 12\text{V}$ and the time constant $\tau = RC$. We are going to solve this circuit in two ways.

First, without using convolution, we can show that the corresponding output voltage is:

$$y(t) = \begin{cases} V_0 \cdot (1 - e^{-t/\tau}) & 0 < t < 2\tau \quad \text{(charging; pulse is "on")} \\ V_0 \cdot (1 - e^{-2}) \cdot e^{-(t-2\tau)/\tau} & t > 2\tau \quad \text{(discharging; pulse is "off")} \end{cases}$$

We will use this analytic solution to check to make sure our other (numerical) solution is correct.

Second, using convolution, we start with the impulse response:

$$h(t) = \frac{1}{\tau} \cdot e^{-t/\tau}$$

and write the output voltage as the convolution of h and x:

$$y(t) = h(t) * x(t) = \int_0^\infty h(r) \cdot x(t-r) \cdot dr$$

We will solve for this output voltage numerically, by performing the convolution integral using MATLAB, at different values of time t. Because $x(t)$ is a pulse, its value is either 0 or V_0, and we only need to integrate over a relatively short period of time. However, because the argument is "$t - r$", and the integral begins at $r = 0$, you will need to think about what the limits of r should be as functions of t.

Design Specifications

1. Sketch $x(t)$, then derive an expression for the convolution integral that is suitable to be coded in your script. Specifically, the integral equation above should be rewritten using the given $x(t)$ and $h(t)$, with correct limits of integration as functions of time t. Include the hand calculation when you submit the rest of your solution. [5]
2. Use MATLAB to construct the output voltage $y(t)$ as an array suitable to be plotted, and using your expression derived in the hand calculation, i.e., without using special functions. The script should agree with the hand calculation. Print the M file to PDF with line numbers and a proper header. [5]
3. Plot the two analytic solutions, given above, each over the full time period (0 to 8τ), and each as a dotted line. They should cross each other at $t = 2\tau$. On the same figure, plot your numerical solution for $y(t)$ also from $t = 0$ to $t = 8\tau$. Adjust the vertical axis to range from 0 to a value slightly larger than V_0. Include the figure in your solution. [5]

[continued]

Design Specifications (continued)

4. There are three checks: (1) Show that the two analytic solutions are equal to each other at $t = 2\tau$. (This is a check that you transcribed the two given solutions correctly.) Show also that (2) the maximum value of the numerical solution is approximately the same as the value of the two (given) analytic solutions at $t = 2\tau$, and that (3) this maximum value occurs at approximately $t = 2\tau$. (These are checks that the numerical solution is approximately equal to the analytic solution.) Include your output from the Command Window. [5]
5. Your script should be well organized and easy to understand. Include your name, a context (e.g., ECE 213, Exercise C1), the date you started, and a description. Add in-line comments to help the reader, and add blank lines and sectioning to help organize your script. [+1]
6. The output and figure should be meaningful. For instance, there should be a legend, with expressions wherever possible and meaningful titles, descriptions, and legend text elsewhere. Use ms for the time axis and make the font sizes and line widths appropriate for the size of the figure. Clear the Command Window before your last run. [+1]
7. Your script should be robust and efficient. Define R, C, and V_0, then define τ, and use variables throughout the rest of the script. Further, in the legend, any known quantities should be constructed rather than hardwired. The upper Y limit should depend on the value of V_0. Because C is in mF, $\tau = RC$ is in ms, and it turns out that you can use ms throughout, i.e., you don't need to convert C to farads, and you never need to convert anything to seconds or work in seconds. [+2]

SUGGESTIONS:

- You will need around $N = 4000$ intervals in your time array in order to have enough points in your convolution integral.
- You will need two nested FOR loops, one that steps through the 4000 intervals in your time array, and the other to perform the (numerical) convolution integral at each value of time t.
- You should be able to write an expression to indicate when $x(t)$ is non-zero, i.e., $t_{\min} < t < t_{\max}$, which you can turn into limits of r when you insert $x(t - r)$ into the convolution integral, i.e., $r_{\min} < r < r_{\max}$. Mathematically, the convolution becomes:

$$y(t) = \int_{r_{\min}}^{r_{\max}} h(r) \cdot V_0 \cdot dr$$

NOTES:

- Successful completion of Design Specs #1–4 is worth 20 points.
- Design Specs #5–7 are worth Bonus points added onto your score. However, they will be awarded only when you are close to a perfect score on Specs #1–4.
- You are doing the numerical integral yourself, i.e., yt = yt + h(r) * V0 * dr.
- Make sure to use the GIVEN form of the convolution integral, i.e., you may not transform the integral before finding the numerical solution.