Lab 3: Introduction to Signal Processing – Transfer Functions, Responses, and Further Transforms

This laboratory requires the following equipment:

- Matlab

The laboratory duration is approximately 3 hours. Although this laboratory is not graded, we encourage you to take your own personal notes as the examinations might leverage results acquired during this laboratory session. For any questions, please contact us at **sis-ta@groupes.epfl.ch**

1.1 Information

In the following text you will find several exercises and questions.

- The notation **S** means that the question can be solved using only additional simulation.
- The notation **Q** means that the question can be answered theoretically, without any simulation.
- The notation **I** means that the problem has to be solved by implementing a piece of code and performing a simulation.
- The notation **B** means that the question is optional and should be answered if you have enough time at your disposal.

Outline

This lab continues on the subject of signal processing, and in particular, reviews the transfer functions, responses and, mainly, Laplace and Z-transforms. In Part 1, you will apply different continuous transforms to a first-order continuous system in order to find the step response and to analyze it. In Part 2, you will be given the discrete version of the system. You will apply discrete transformations to obtain the step response and analyze it. Every part includes both theoretical and MATLAB related questions. You can find the Laplace and Z-transform tables in the appendix if you need them.

Getting Started

To start with this lab, you will need to download the material available on Moodle. Download `lab03.tar.gz` or `lab03.zip` and extract it in your home directory (you can type `tar xvfz lab03.tar.gz`). Now start Matlab and change your "Current directory" to be `lab03/part01/`.

Please install Control System Toolbox if you have not installed it already. You can do so by pressing Add-Ons button in MATLAB main window.

Part 1: Continuous-Time Transforms

In this part, you will apply Continuous-Time (CT) transformations (Fourier and Laplace transforms) to a first-order system in order to obtain its step response and analyze the response. Open the script `part_01.m` in the folder `part_01`, read the explanations of this template and check the corresponding questions.

Step response of a system is defined as the time evolution of its output when its input is a step function. Fig. 1 illustrates such concept. Here, *s(t)* is the step function and $x_s(t)$ is step response.

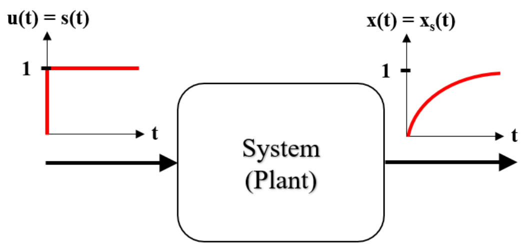

Figure 1. Step input s(t) and step response $x_s(t)$

Consider the analog circuitry given in Fig. 2. This circuitry consists of an independent voltage source v_s, a resistor *R,* and a capacitor *C,* whose voltage is shown by v_c.

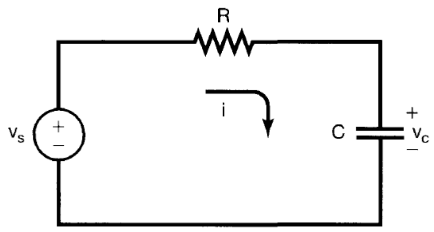

Figure 2. RC circuitry [1]

This system can be represented by a continuous first-order differential equation, if we select input as v_s and output as v_c. The equation can be written as follows:

$$\frac{dv_c(t)}{dt} = \frac{1}{RC}\,(v_s(t) - v_c(t))$$

If we define *u(t)* as input, *x(t)* as state and the multiplication of *R* and *C* as Γ, the equation has a new form as follows:

$$\frac{dx(t)}{dt} = \frac{1}{\Gamma}\,(u(t) - x(t))$$

Here Γ is known as time constant of the system and it is an indication of how fast the system responds. It is the duration where the system reaches (approximately) 63% of its steady state (final) value. Pay attention to this definition, you will use this to extract time constants from the plots by adding data points on it.

In order to find the step response (i.e. the state of the system when the input is a step function) in time domain, we should substitute the step function *s(t)* in place of *u(t)*. The definition of *s(t)* is given as follows:

$$s(t) = \begin{cases} 1\,, if\ t \geq 0 \\ 0\ \ ,otherwise \end{cases}$$

Considering the complementary and particular solutions of the first-order differential equations, the solution (step response) of this differential equation can be found as follows:

$$x_s(t) = \left(-e^{\left(-\frac{t}{\Gamma}\right)} + 1\right) \cdot s(t)$$

1. **(I):** Impulse response *h(t)* is defined as the output of the system when the input is impulse function *δ(t)*. Given that R = 1 Ω and C = 0.1 F, find the impulse response *h(t)* of the system from the step response $x_s(t)$ given. Plot both the step response and impulse response using MATLAB and observe the figure. Can you identify the time-constant from the figure?

Hint-1: Since the system is linear, the impulse response of the system is the derivative of the step response with respect to time since the impulse function is the derivative of unit step function.
Hint-2: Write the functions and variables symbolically using `syms`. *You can use* `heaviside()` *to represent the unit step input. You can use* `fplot()` *to plot symbolic functions.*
Hint-3: To have more accurate unit step function use `sympref('HeavisideAtOrigin',1)` *before using step function.*
Hint-4: You can plot all responses between the interval [0,0.6].

2. **(I):** Since you know the unit step input and found the impulse response in time domain you can find the step response by using convolution of both analytically. However, there is no symbolic convolution function in MATLAB. Instead, we can use the definition of continuous convolution to calculate it.

$$x_s(t) = s(t) * h(t) = \int_{-\infty}^{\infty} s(\tau) \cdot h(t-\tau)\, d\tau$$

Obtain the step response, plot and compare it with the one that is given in Question **1**. Do the overall shape and the time-constant differ?

Hint-1: Use `int()` *function to calculate symbolic integration.*
Hint-2: You can use `subs()` *function to substitute any value to a symbolic function.*
Hint-3: You can use `inf, -inf` *variables to define integral bounds.*
Hint-4: Use `simplify()` *function to simplify the symbolic expression calculated by above functions. Sometimes, even if you use* `simplify()`, *you can obtain complicated expressions from symbolic tools. This is because symbolic tools consider the entire time axis. In order to further simplify expression for the time greater than zero, before calculation of the expression add* `assume(t>0)` *where t is your time variable. In order to remove assumption from the variable use* `assume(t,'clear')`.

3. **(I):** Analyze the frequency content of the step response $x_s(t)$ by applying the CT Fourier transform in MATLAB. Plot the magnitude of the response and observe it. Pay attention to the unit of frequency axis. Is it in Hertz or radians/sec?

Hint-1: Use `fourier()` *function to calculate the Fourier transform symbolically.*
Hint-2: Do not forget to obtain the absolute value of transform.

Another way to obtain system responses (e.g., impulse and step responses) is to leverage the Laplace transform. The associated frequency domain of this transform is also continuous and the Laplace transform can be considered a more generic version of the CT Fourier transform. Indeed, the convergence of the Laplace transform is broader than that of the CT Fourier transform, i.e. it includes unstable systems. Therefore, while the Laplace transform is mostly used in continuous system analysis, the CT Fourier transform is mostly used in continuous signal processing.

As explained in the lecture, the complex variable s is used instead of the ordinary frequency ξ or the angular frequency ω. Multiple properties valid for the Fourier transform also apply to the Laplace transform. For instance, a convolution in time domain corresponds to a multiplication in the frequency domain.

$$x_s(t) = s(t) * h(t) \leftrightarrow S(s) \cdot \mathrm{H(s)} = X_s(s)$$

where $S(s)$ and $H(s)$ are the Laplace transform of $s(t)$ and $h(t)$. Therefore, by taking the inverse Laplace transform of the multiplication of $S(s)$ and $H(s)$, we can obtain the step response in time domain.

$$x_s(t) = \mathcal{L}^{-1}(S(s) \cdot \mathrm{H(s)})$$

where *H(s)* $= \frac{X_s(s)}{S(s)}$ or, more generically for an arbitrary input *U(s)* and a corresponding response *X(s), H(s)* $= \frac{X(s)}{U(s)}$ is also known as the transfer function of the CT system. Notice that the Laplace transform of the impulse response *h(t)* is the transfer function *H(s)*.

4. **(B):** Compute *S(s)* and *H(s)* by using the definition of Laplace transform or Laplace transform tables in appendix. Find the step response $x_s(t)$ in time domain by using the definition of inverse Laplace transform or inverse Laplace transform tables.

5. **(I):** Apply the same procedure in MATLAB symbolically and find the step response $x_s(t)$. Plot and compare the result with the one you found in Questions **1** and **2**.

   Hint-1: Use `laplace()` *function to calculate Laplace transform symbolically.*
   Hint-2: Use `ilaplace()` *function to calculate inverse Laplace transform symbolically.*
   Hint-3: Use `simplify()` *function to simplify the symbolic expression calculated by above functions. Sometimes, even if you use* `simplify()`, *you can obtain complicated expressions from symbolic tools. This is because symbolic tools consider the entire time axis. In order to further simplify expression for the time greater than zero, before calculation of the expression, add* `assume(t>0)` *where t is your time variable. In order to remove assumption from the variable use* `assume(t,'clear')`.

6. **(I):** Observe *H(s)* and represent it in MATLAB using `tf()` function. Use `step()` function together with the resulting transfer function to obtain the step response. Plot and compare the result with the one you found in Questions **1, 2** and **5**. Do the overall shape and time-constant differ?

   Hint-1: After you obtain H(s) symbolically, take note of the numerator and denominator coefficients and enter this into `tf()`. *For example if your H(s)=0.1/(0.4s-0.3) use H_s =* `tf(0.1,[0.4,-0.3])`.
   Hint-2: `step(H_s)` *directly gives and plots the step response where* `H_s` *is your transfer function.*

Part 2: Discrete-Time Transforms

In this part, you will apply some Discrete-Time (DT) transformations (Z-transform and DT Fourier Transform) to a first-order system in order to obtain step response of a discrete system and analyze the response. Open the script `part_02.m` in the folder `part_02`, read the explanations of this template and check the corresponding questions.

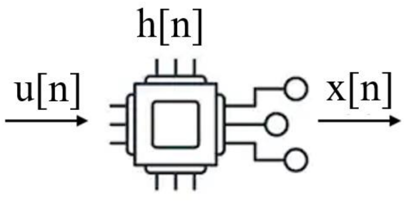

Figure 3. Digital system

Now, we would like to represent our continuous-time system (RC circuitry) in an embedded digital computer as in Fig. 3. By using an appropriate discretization method, we can find the difference equation of the system as follows:

$$x[n] = \left(1 - \frac{T}{\Gamma}\right) x[n-1] + \left(\frac{T}{\Gamma}\right) u[n-1]$$

where T is the sampling period and Γ is defined in the previous questions. Take T as 0.01 seconds.

In order to find the system response *x[n]* (solution to the difference equation), we first need to find the transfer function *H(z)* of the system (in the Z continuous frequency domain, or in short Z-domain). We will use the Z-transform to find this transfer function, since the system is represented in discrete time. Note that the Z-transform has a similar purpose for discrete-time functions and systems as the Laplace transform for continuous-time functions and systems.

7. **(B):** Take the Z-transform of the both sides of the difference equation by using Z-transform tables given in appendix. Manipulate the equation to obtain the transfer function *H(z)* = $\frac{X(z)}{U(z)}$ symbolically.

Hint: Use "linearity" and "time shift" properties of the Z-transform while solving the problem analytically.

8. **(I):** By applying the method in Question 7, it can be found that $H(z) = \frac{0.1}{z-0.9}$, represent it as MATLAB transfer function using `tf()`. Use the `step()` function together with the resulting transfer function to obtain the step response. Plot and compare the result with the one you found in questions **1, 2** and **5**. Is there any difference between the CT and DT counterparts? Do the overall shape and the time-constant differ?

Hint-1: After you obtain H(z) symbolically, take note of the numerator and denominator coefficients and enter this into `tf()`*. For example if your H(z)=0.1/(0.4z-0.3) use H_z =* `tf(0.1,[0.4,-0.3],T)` *where T is your sample time for discrete system.*
Hint-2: `step(H_z)` *directly gives and plots the step response where H_z is your transfer function.*

9. **(I):** Similar to previous questions, take input *u[n]* as unit step input *s[n]* whose definition is given as

$$s[n] = \begin{cases} 1, if\ n \geq 0 \\ 0\ , otherwise \end{cases}$$

Similar to the Laplace transform, a convolution in time domain corresponds to a multiplication in the Z-domain.

$$x_s[n] = s[n] * h[n] \leftrightarrow S(z) \cdot \mathrm{H(z)} = X_s(z)$$

where *S(z)* and *H(z)* are the Z-transforms of *s[n]* and *h[n]*. Therefore, by taking the inverse Z-transform of the multiplication, we can obtain the step response in time domain.

$$x_s[n] = Z^{-1}(S(z) \cdot \mathrm{H(z)})$$

Apply this procedure analytically by using transform tables **(B)** and in MATLAB to obtain the step response $x_s[n]$. Plot and compare the result with the one you found in Questions **1, 2, 5** and **8**. Do the overall shape and the time-constant differ?

Hint-1: You already obtained H(z). First, write it symbolically. Find the Z-transform of s[n] by using `ztrans()`*. You can use* `heaviside()` *to represent the unit step input s[n]. Multiply Z-transform of s[n] with H(z) and take the inverse transform symbolically by using* `iztrans()`.
Hint-2: You can use `fplot()` *to plot symbolic functions.*

For sake of completeness, we would like you to get a feeling for the Discrete-Time Fourier Transform (DTFT). This transform is associated with a continuous frequency domain and should not be confused with the Discrete Fourier Transform (DFT) that has been exercised in Labs 1 and 2. Moreover, the DT Fourier transform is a special case of the Z-transform. The convergence of the Z-transform is broader than that of the DT Fourier transform, i.e. it includes unstable systems. Therefore, while the Z-transform is mostly used in discrete system analysis, the DT Fourier transform is mostly used in digital signal processing.

10. **(I):** Define the time series for *n* using the code below:

```
n = 0:1:60;
```

Represent the step response $x_s[n]$ (found in Question **9**) as time-series by using *n.* There is no symbolic function to calculate the DT Fourier transform in MATLAB. However, we can use `fft()` to calculate it. Use `calculate_DTFT_and_FFT()` to plot transforms in in the folder `part_02/`. Input your response series to the functions. Observe how `calculate_DTFT_and_FFT()` is utilizing MATLAB's built in function `fft()` to calculate the DT Fourier transform. Observe the differences between FFT and DT Fourier transform plots. Observe the frequency content of the step response and compare it with the one you found in Question **3**.

Appendix: Transform tables

Laplace Transform [1]

TABLE 9.1 PROPERTIES OF THE LAPLACE TRANSFORM

Section	Property	Signal	Laplace Transform	ROC
		$x(t)$ $x_1(t)$ $x_2(t)$	$X(s)$ $X_1(s)$ $X_2(s)$	R R_1 R_2
9.5.1	Linearity	$ax_1(t) + bx_2(t)$	$aX_1(s) + bX_2(s)$	At least $R_1 \cap R_2$
9.5.2	Time shifting	$x(t - t_0)$	$e^{-st_0}X(s)$	R
9.5.3	Shifting in the s-Domain	$e^{s_0 t}x(t)$	$X(s - s_0)$	Shifted version of R (i.e., s is in the ROC if $s - s_0$ is in R)
9.5.4	Time scaling	$x(at)$	$\frac{1}{\lvert a\rvert}X\left(\frac{s}{a}\right)$	Scaled ROC (i.e., s is in the ROC if s/a is in R)
9.5.5	Conjugation	$x^*(t)$	$X^*(s^*)$	R
9.5.6	Convolution	$x_1(t) * x_2(t)$	$X_1(s)X_2(s)$	At least $R_1 \cap R_2$
9.5.7	Differentiation in the Time Domain	$\frac{d}{dt}x(t)$	$sX(s)$	At least R
9.5.8	Differentiation in the s-Domain	$-tx(t)$	$\frac{d}{ds}X(s)$	R
9.5.9	Integration in the Time Domain	$\int_{-\infty}^{t} x(\tau)d(\tau)$	$\frac{1}{s}X(s)$	At least $R \cap \{\mathcal{R}e\{s\} > 0\}$

TABLE 9.2 LAPLACE TRANSFORMS OF ELEMENTARY FUNCTIONS

Transform pair	Signal	Transform	ROC
1	$\delta(t)$	1	All s
2	$u(t)$	$\frac{1}{s}$	$\mathcal{R}e\{s\} > 0$
3	$-u(-t)$	$\frac{1}{s}$	$\mathcal{R}e\{s\} < 0$
4	$\frac{t^{n-1}}{(n-1)!}u(t)$	$\frac{1}{s^n}$	$\mathcal{R}e\{s\} > 0$
5	$-\frac{t^{n-1}}{(n-1)!}u(-t)$	$\frac{1}{s^n}$	$\mathcal{R}e\{s\} < 0$
6	$e^{-\alpha t}u(t)$	$\frac{1}{s+\alpha}$	$\mathcal{R}e\{s\} > -\alpha$
7	$-e^{-\alpha t}u(-t)$	$\frac{1}{s+\alpha}$	$\mathcal{R}e\{s\} < -\alpha$
8	$\frac{t^{n-1}}{(n-1)!}e^{-\alpha t}u(t)$	$\frac{1}{(s+\alpha)^n}$	$\mathcal{R}e\{s\} > -\alpha$
9	$-\frac{t^{n-1}}{(n-1)!}e^{-\alpha t}u(-t)$	$\frac{1}{(s+\alpha)^n}$	$\mathcal{R}e\{s\} < -\alpha$
10	$\delta(t - T)$	e^{-sT}	All s
11	$[\cos \omega_0 t]u(t)$	$\frac{s}{s^2 + \omega_0^2}$	$\mathcal{R}e\{s\} > 0$
12	$[\sin \omega_0 t]u(t)$	$\frac{\omega_0}{s^2 + \omega_0^2}$	$\mathcal{R}e\{s\} > 0$
13	$[e^{-\alpha t}\cos \omega_0 t]u(t)$	$\frac{s+\alpha}{(s+\alpha)^2 + \omega_0^2}$	$\mathcal{R}e\{s\} > -\alpha$
14	$[e^{-\alpha t}\sin \omega_0 t]u(t)$	$\frac{\omega_0}{(s+\alpha)^2 + \omega_0^2}$	$\mathcal{R}e\{s\} > -\alpha$
15	$u_n(t) = \frac{d^n\delta(t)}{dt^n}$	s^n	All s
16	$u_{-n}(t) = \underbrace{u(t) * \cdots * u(t)}_{n \text{ times}}$	$\frac{1}{s^n}$	$\mathcal{R}e\{s\} > 0$

Z Transform [1]

TABLE 10.1 PROPERTIES OF THE z-TRANSFORM

Section	Property	Signal	z-Transform	ROC
		$x[n]$	$X(z)$	R
		$x_1[n]$	$X_1(z)$	R_1
		$x_2[n]$	$X_2(z)$	R_2
10.5.1	Linearity	$ax_1[n] + bx_2[n]$	$aX_1(z) + bX_2(z)$	At least the intersection of R_1 and R_2
10.5.2	Time shifting	$x[n - n_0]$	$z^{-n_0}X(z)$	R, except for the possible addition or deletion of the origin
10.5.3	Scaling in the z-domain	$e^{j\omega_0 n}x[n]$	$X(e^{-j\omega_0}z)$	R
		$z_0^n x[n]$	$X\left(\frac{z}{z_0}\right)$	$z_0 R$
		$a^n x[n]$	$X(a^{-1}z)$	Scaled version of R (i.e., $\|a\|R$ = the set of points $\{\|a\|z\}$ for z in R)
10.5.4	Time reversal	$x[-n]$	$X(z^{-1})$	Inverted R (i.e., R^{-1} = the set of points z^{-1}, where z is in R)
10.5.5	Time expansion	$x_{(k)}[n] = \begin{cases} x[r], & n = rk \\ 0, & n \neq rk \end{cases}$ for some integer r	$X(z^k)$	$R^{1/k}$ (i.e., the set of points $z^{1/k}$, where z is in R)
10.5.6	Conjugation	$x^*[n]$	$X^*(z^*)$	R
10.5.7	Convolution	$x_1[n] * x_2[n]$	$X_1(z)X_2(z)$	At least the intersection of R_1 and R_2
10.5.7	First difference	$x[n] - x[n-1]$	$(1 - z^{-1})X(z)$	At least the intersection of R and $\|z\| > 0$
10.5.7	Accumulation	$\sum_{k=-\infty}^{n} x[k]$	$\frac{1}{1 - z^{-1}}X(z)$	At least the intersection of R and $\|z\| > 1$
10.5.8	Differentiation in the z-domain	$nx[n]$	$-z\frac{dX(z)}{dz}$	R

TABLE 10.2 SOME COMMON z-TRANSFORM PAIRS

Signal	Transform	ROC
1. $\delta[n]$	1	All z
2. $u[n]$	$\frac{1}{1 - z^{-1}}$	$\|z\| > 1$
3. $-u[-n-1]$	$\frac{1}{1 - z^{-1}}$	$\|z\| < 1$
4. $\delta[n - m]$	z^{-m}	All z, except 0 (if $m > 0$) or ∞ (if $m < 0$)
5. $\alpha^n u[n]$	$\frac{1}{1 - \alpha z^{-1}}$	$\|z\| > \|\alpha\|$
6. $-\alpha^n u[-n-1]$	$\frac{1}{1 - \alpha z^{-1}}$	$\|z\| < \|\alpha\|$
7. $n\alpha^n u[n]$	$\frac{\alpha z^{-1}}{(1 - \alpha z^{-1})^2}$	$\|z\| > \|\alpha\|$
8. $-n\alpha^n u[-n-1]$	$\frac{\alpha z^{-1}}{(1 - \alpha z^{-1})^2}$	$\|z\| < \|\alpha\|$
9. $[\cos \omega_0 n]u[n]$	$\frac{1 - [\cos \omega_0]z^{-1}}{1 - [2\cos \omega_0]z^{-1} + z^{-2}}$	$\|z\| > 1$
10. $[\sin \omega_0 n]u[n]$	$\frac{[\sin \omega_0]z^{-1}}{1 - [2\cos \omega_0]z^{-1} + z^{-2}}$	$\|z\| > 1$
11. $[r^n \cos \omega_0 n]u[n]$	$\frac{1 - [r\cos \omega_0]z^{-1}}{1 - [2r\cos \omega_0]z^{-1} + r^2 z^{-2}}$	$\|z\| > r$
12. $[r^n \sin \omega_0 n]u[n]$	$\frac{[r\sin \omega_0]z^{-1}}{1 - [2r\cos \omega_0]z^{-1} + r^2 z^{-2}}$	$\|z\| > r$

References

[1] Oppenheim A.V. , Willsky A.S., Hamid S., Signals and Systems, Prentice Hall (1996)