Signals and Systems 2

Giovanni Geraci — Universitat Pompeu Fabra, Barcelona

Lab #1: Introduction to Matlab

High-level objectives:

(a) Use the help system, to learn about Matlab commands and syntax.

(b) Explore and modify examples, to understand third-party code and to write your own.

(c) Get familiar with basic Matlab commands.

(d) Analyze examples of code and relate them with the theory.

(e) Introduce good working habits for creating and presenting reports.

The deliverable for each group consists of a **single .zip file**, including all the materials required at the different exercises, and well as the required manual calculations in paper. The document should be named (please, **be strict** on that, it will be processed automatically):

SiS2_Lab1_NameSurnameMember1_NameSurnameMember2.zip

Submit only **one document per group** to Aula Global.

1. **Basic commands**

   (a) Explore Matlab's *help* functionality. Type each of the following lines to learn about these commands:
   ```
   help
   help sqrt
   help plot
   help clear
   ```

   (b) Use Matlab as a calculator. Try the following:
   ```
   pi*pi-10
   sin(pi/4)
   ans^2
   ```

   (c) Store values in variables. Try the following:
   ```
   x = sin(pi/5)
   cos(pi/5) % assigned to what?
   y = sqrt(1-x*x)
   ans
   ```

   (d) Explore complex numbers. Try the following:
   ```
   z = 3+4j
   conj(z)
   abs(z)
   angle(z)
   real(z)
   imag(z)
   exp(sqrt(-1)*pi)
   exp(j*(pi/4))
   ```

   (e) Explore vectorization and plotting. Try the following:
   ```
   x = -3:0.5:3
   y = x.^2+x
   plot(x,y)
   ```

   (f) Create executable scripts in Matlab:

      i. Click *New Script* in the menu.

      ii. In the new editor window, type the following code:
      ```
      function test
      t = -4*pi:0.1:4*pi;
      y = sin(t);
      plot(t,y,'r');
      xlabel('t (sec.)');
      ylabel('y(t)');
      axis([-4*pi 4*pi -2 2]);
      ```

      iii. Click on *Save As...* in the menu, to save the code as a script **test.m** (.m is Matlab's extension for scripts). Execute the script by typing ***test*** in the command window. **Analyze** the code (Matlab's *help* of each new command, try code variants, etc.) to **understand** every aspect of it.

There is no deliverable for this first exercise.

2. **Symbolic signals in Matlab**

Analyze and understand the following code:

```
clear
cos(pi/6)
cos(sym(pi/6)) % Appreciate the difference between this and the previous command

syms x u t
x = cos(2*pi*t)
u = heaviside(t)
subplot(311); fplot(x); axis([-5 5 -2 2]); xlabel('t (sec.)'); ylabel('x(t)');
subplot(312); fplot(u); axis([-5 5 -2 2]); xlabel('t (sec.)'); ylabel('u(t)');
subplot(313); fplot(x*u); axis([-5 5 -2 2]); xlabel('t (sec.)'); ylabel('x(t)u(t)');
```

Based on what you've learnt, **write the code**[1] that **creates and plots**, in an **explanatory way** (as a composition of subfunctions), the symbolic functions below

(a)

$$x(t) = \begin{cases} e^{-t}, & t \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

(b)

$$x(t) = \begin{cases} 1, & 1 \leq t < 3 \\ 0, & \text{otherwise} \end{cases}$$

(c)

$$x(t) = \begin{cases} \sin(2\pi t), & -2 \leq t \leq 2 \\ 0, & \text{otherwise} \end{cases}$$

(d)

$$x(t) = \begin{cases} t+1, & -1 \leq t \leq 0 \\ -t+1, & 0 \leq t \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

[1] Create **one script per case**, named as **2a.m, 2b.m**, etc. Add them to the final .zip file.

3. **Symbolic operations in Matlab**

Analyze and understand the following code:

```
clear
syms x t a
diff(sin(a*t),t)
diff(sin(a*t),t,t)
diff(heaviside(t),t)
int(cos(t),t)
int(cos(t),t,0,pi/2)
int(dirac(t-pi)*cos(t),t,-Inf,Inf)
```

Based on what you've learnt, **write the code**[2] for computing the expressions below, and **compare the result with hand-made calculations.**[3].

(a) $\frac{\mathrm{d}}{\mathrm{d}t}x(t)$, where $x(t) = e^{-(t-1)}u(t-1)$

(b) $\int_{-1}^{1} \delta(t)\mathrm{d}t$

(c) $\int_{-\infty}^{t} \delta(t)\mathrm{d}t$

(d) $\int_{-\infty}^{\infty} \delta(t-a)\cos(t)\mathrm{d}t$

[2]Write the code for all the cases in a **single script, named 3.m, one line per case, no semicolon at the ends**. Add it to the final .zip file.

[3]Do the **manual calculations in a paper sheet (to be delivered at the end of the session)**, taking a picture for your personal use.

4. **Exercise**

   Analyze and understand the following code:

```
clear
syms x h t tau
x = exp(-tau)*heaviside(tau);
subplot(511); fplot(x); axis([-5 5 -1 2]); title('x(tau)');
h = heaviside(tau);
subplot(512); fplot(h); axis([- 5 5 -1 2]); title('h(tau)');
h1 = subs(h,tau,-tau)
subplot(513); fplot(h1); axis([-5 5 -1 2]); title('h(-tau)');
h2 = subs(h,tau,1-tau);
subplot(514); fplot(h2); axis([-5 5 -1 2]); title('h(1-tau)');
subplot(515); fplot(x*h2); axis([-5 5 -1 2]); title('x(tau)h(1-tau)');
```

   Based on what you've learnt all along the lab, and the **definition of CT convolution**, **write the code**[4] for **computing and plotting** the expression below, and **compare the result with hand-made calculations.**[5]

   (a) $x(t) * h(t)$, where: $x(t) = e^{-t}u(t); \quad h(t) = u(t) - u(t-5)$
   and the symbol $*$ represents the **CT convolution** operation.

[4] Write the code as a script, named **4.m**. Add it to the final .zip file.

[5] Do the **manual calculations in a paper sheet (to be delivered at the end of the session)**, taking a picture for your personal use.