Digital Signal Processing 2018/2019.

Lab. 1. An Introduction to Time-Discrete Signals and Systems.

Aim: The aim of this practical session is setting some basic but relevant concepts related to time-discrete signals and systems. First, some examples of basic signals will be introduced, that will be used to understand how shifting and scaling properties affect signals.

Then, combinations of these signals will be used to feed the input of different systems, in order to determine whether they fulfil or not the properties of linearity, time invariance and/or causality. Finally, some examples of system interconnection will be introduced.

1 An Introduction to Signals

1.1 Basic Signals

List 1 shows the code of a MATLAB®script used to plot different time-discrete signals. To run this code, just write down the name of the script file in MATLAB®console, and then press Enter.

List 1: *ap11Eng.m*

```
%ap11Eng  Figure plots for time-discrete signals

% Close figures and clear memory
close all;clear;
% Definition of n axis (discrete time)
n=-10:10;

% Discrete delta function
x1=delta(n);
figure;stem(n,x1,'b','linewidth',2);xlabel('discrete time, n'); ylabel('x1[n]');
    axis([-10 10 -1.2 1.2]);grid

% Unit step function u(n)
x2=escalon(n);
figure;stem(n,x2,'b','linewidth',2);xlabel('discrete time, n'); ylabel('x2[n]');
    axis([-10 10 -1.2 1.2]);grid

% Pulse function of width "w" samples (w>0)
w=4;
x3=pulsow(n,w);
figure;stem(n,x3,'b','linewidth',2);xlabel('discrete time, n'); ylabel('x3[n]');
    axis([-10 10 -1.2 1.2]);grid

% Time-discrete sinusoid
x4=sin(pi*n/5-pi/3);
figure;stem(n,x4,'b','linewidth',2);xlabel('discrete time, n'); ylabel('x4[n]');
    axis([-10 10 -1.2 1.2]);grid

% Exponential signal
a=1/2;
x5=a.^n;
```

```
28 figure;stem(n,x5,'b','linewidth',2);xlabel('discrete time, n'); ylabel('x5[n]');
      axis([-10 10 -1.2 2^10]);grid
```

The first two commands in line 4 are used to close all previously opened figure plots, as well as to clear the variables stored in memory. The vector defined in line 6 will be used as a time-discrete axis for the signals to be calculated and plotted.

(a) Is it possible to represent the signal $x_1[n]$ by using the unit step function (i.e. the signal *escalon*)? Change the code in the script and include a new figure that illustrates this possibility, and check your response with the graphical representation.

(b) From the code, determine the values of the angular frequency ω_0 and the period N_0 for the discrete-time signal $x_4[n]$. Check that these values coincide with those deduced from the figure plot.

$$\omega_0 = \qquad\qquad\qquad\qquad N_0 =$$

(c) Change the code of the script and include a new plot for the signal $x_S[n] = \sin(n/2 - \pi/3)$. Is this signal periodic? Confirm your guess with the observation of the figure plot.

(d) Change the code of the script so that the signal $x_5[n]$ is nulled for $n < 0$, and represent the result in a new figure. What's the mathematical expression for the new signal?

1.2 Amplitude scaling and shifting

Starting from the previous script, create a new one named *ShiftScaleAmp* in which the following set of signals (built from amplitude scaling and shifting over the basic set of signals previously introduced) are represented:

- $e_1[n] = -escalon[n]$
- $e_2[n] = 2 \cdot escalon[n] - 1$
- $p_1[n] = -2 \cdot pulsow[n, 3] + 1$
- $p_2[n] = 2 \cdot delta[n] - 2$

(a) Run the script *ShiftScaleAmp* and verify that the figure plots correspond with the expected results.

(b) Fill the blanks in the script *ap12* included below, so that the resulting signals coincide with those in Figure 1 (see next page). For this purpose, use only the following set of signals: delta, unit step, pulse, sine and constant values, *without resorting to discrete-time shifting.*

List 2: *ap12.m*

```
%ap12 Figure plots for time-discrete signals

% Close figures and clear memory
close all;clear ;
% Definition of n axis (discrete time)
n=-10:10;

% Combination of unit step and pulse signals
```

```
x1=      ;
subplot(221);stem(n,x1,'b','linewidth',2);xlabel('discrete time, n'); ylabel(' x1[n] ');axis([-10 10 -1.2 2.2]);grid

% Combination of pulse and delta signals
x2=      ;
subplot(222);stem(n,x2,'b','linewidth',2);xlabel('discrete time, n'); ylabel(' x2[n] ');axis([-10 10 -1.2 1.2]);grid

% Combination of basic signals and constant values
x3=      ;
subplot(223);stem(n,x3,'b','linewidth',2);xlabel('discrete time, n'); ylabel(' x3[n] ');axis([-10 10 -1.2 1.2]);grid

% Combination of time disecrete sinusoid and constant amplitude shift
A=      ;
B=      ;
x4=A*sin(pi*n/6)+B;
subplot(224);stem(n,x4,'b','linewidth',2);xlabel('discrete time, n'); ylabel(' x4[n]');axis([-10 10 -3.2 1.2]);grid
```

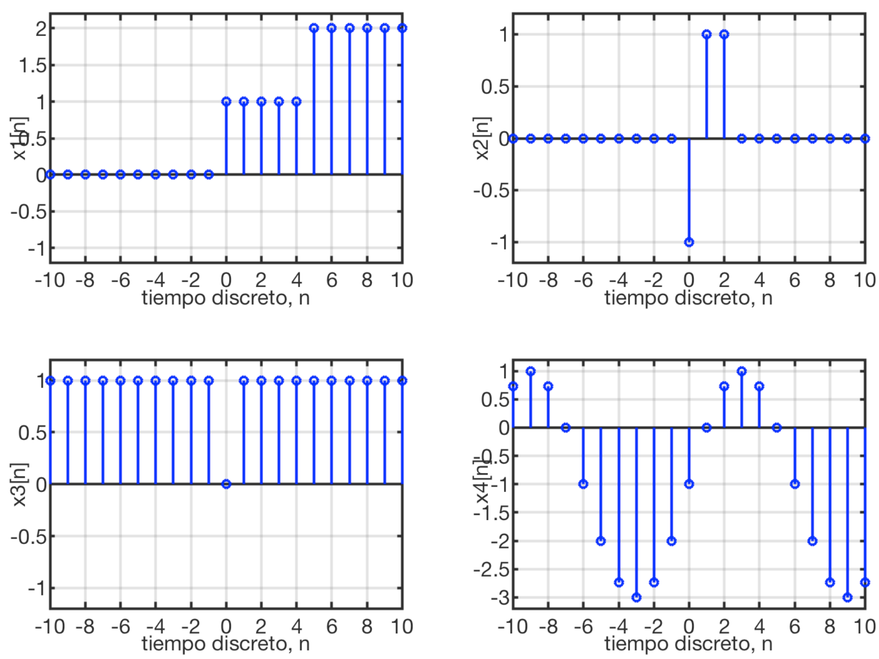

Figure 1: Amplitude Scaling and Shifting.

(c) Build the script *ap12*, run it, and verify that the figure plots coincide with the expected outcome.

1.3 Discrete-time scaling and shifting

Starting from the previous scripts, build a new one named *ShiftScaleAxis* on which the following set of signals (corresponding to time scaling and shifting over the set of basic signals) is represented:

- $d_1[n] = delta[n+2]$
- $d_2[n] = delta[n-2)]$
- $e_1[n] = escalon[-n-1]$
- $e_2[n] = escalon[-(n-1)]$
- $p_1[n] = pulsow[2*n+3, 8]$
- $p_2[n] = pulsow[-2*(n+1), 6]$

1. Run the script *ShiftScaleAxis* and verify that the figure plots coincide with the expected outcome from theory.
2. Fill the blanks in the script *ap13* included below, so that the resulting signals coincide with those in Figure 2 (see next page). For this purpose, use only combinations of the basic signals (delta, unit step, pulse, sine and constant values), with any amplitude value and conveniently shifted/scaled.

List 3: *ap13.m*

```
%ap13 Figure plots for time-discrete signals

% Close figures and clear memory
close all;clear ;
% Definition of n axis (discrete time)
n=-10:10;

% Combination of pulse signals using time shifting
x1=     ;
subplot(221);stem(n,x1,'b','linewidth',2);xlabel('discrete time, n'); ylabel(' x1[n] ');axis([-10 10 -1.2 2.2]);grid

% Combination of delta signals using time shifting
x2=     ;
subplot(222);stem(n,x2,'b','linewidth',2);xlabel('discrete time, n'); ylabel(' x2[n] ');axis([-10 10 -1.2 1.2]);grid

% Combination of unit step functions using time scaling and shifting
x3=     ;
subplot(223);stem(n,x3,'b','linewidth',2);xlabel('discrete time, n'); ylabel(' x3[n] ');axis([-10 10 -1.2 1.2]);grid

% Combination of a discrete time sinusoid and time scaling/shifting
A=     ;
B=     ;
x4=A*sin(pi*n/2+B);
subplot(224);stem(n,x4,'b','linewidth',2);xlabel('discrete time, n'); ylabel(' x4[n]');axis([-10 10 -1.2 1.2]);grid
```

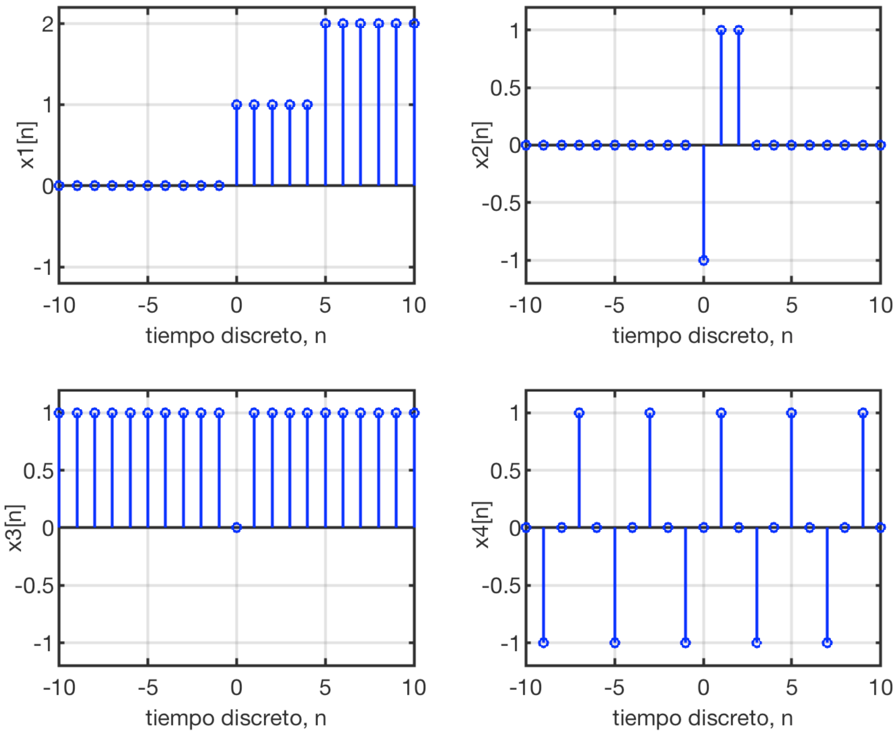

Figure 2: Amplitude Scaling and Shifting.

3. Build the script *ap13*, run it, and verify that the figure plots coincide with the expected outcome.

2 An Introduction to Systems

In this section, we will study a set of systems by characterizing their input/output relationship. The aim is to verify whether they are linear, time-invariant, causal or not. We must note that to rigorously prove that a system does not fulfil one property, it suffices with finding just one example on which such property is not met (counterexample). However, in order to prove that the property is fulfilled, it must be guaranteed for all possible values of the input.

Therefore, in this practical session we will study the response of a number of systems to some specific inputs, so this does not prove that a system fulfils such property (we can only be sure in those cases on which the property is not fulfilled).

In the main folder, there are two scripts named *system1.m* and *system2.m*, which include two MATLAB®functions that characterize the input/output relationship of two discrete-time systems. If we run the command *y=system1(x)*, we obtain the output signal y when the input signal x is fed into the system *system1*.

2.1 Time invariance

List 4 examines the response of the system *system1* to different input signals, as indicated in Figure 3. Run the *script Invariance.m* and answer the following questions:

1. After the inspection of the figure plots: what can we say about the time invariance of *system1*?
2. Modify the previous script in order to run the same test over the system *system2*. Run this new script *Invariance2.m*: what can we now say about the time invariance of *system2*?

   Note: In the code, change the axis parameters to (-10,10,-1.5,1.5) for the case of using subplot(212).

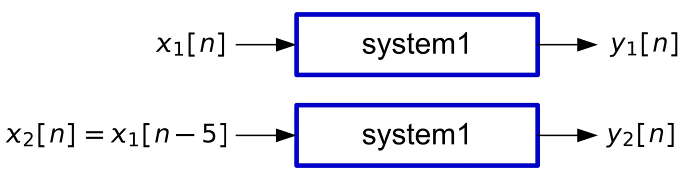

Figure 3: Input/output relationship in List 4

List 4: *Invariance.m*

```
% Time invariance. Response of system1 to two different inputs

% Close figures and clear memory
close all;clear;
% Definition of n axis (discrete time)
n=-10:10;

figure(1);
% Input signal
x1=delta(n)+delta(n-1)+delta(n-2);
subplot(211); % upper subfigure in figure 1
stem(n,x1,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' x1[n] ');
axis([-10 10 -1.5 1.5]);grid
% Output signal
```

```
y1=system1(x1);
subplot(212); % lower subfigure in figure 1
stem(n,y1,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' y1[n] ');
axis([-10 10 -0.5 25]);grid

figure(2);
% Input signal (shifted by 5 samples)
x2=delta(n-5)+delta(n-6)+delta(n-7);
subplot(211); % upper subfigure in figure 2
stem(n,x2,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' x2[n] ');
axis([-10 10 -1.5 1.5]);grid
% System response to shifted input
y2=system1(x2);
subplot(212); % lower subfigure in figure 2
stem(n,y2,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' y2[n] ');
axis([-10 10 -0.5 25]);grid
```

2.2 Linearity

List 5 examines the response of *system1* to different input signals, as indicated in Figure 4.

1. Run the script *Linearity.m*. After the inspection of the figure plots, what can we say about the linearity of *system1*?
2. Modify the previous script in order to run the same test over the system *system2*. Run this new script *Invariance2.m*: what can we now say about the linearity of *system2*?

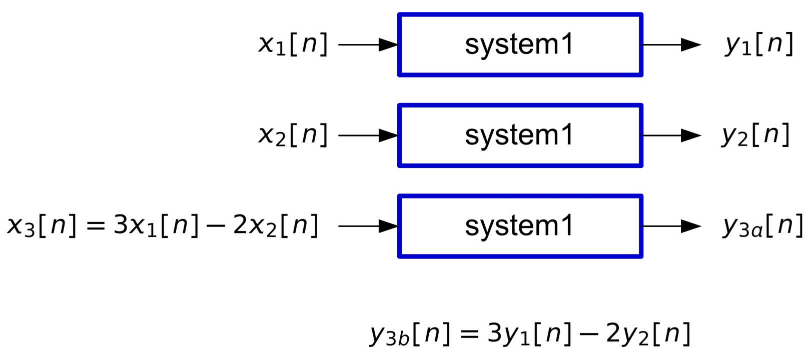

Figure 4: Input/output relationship in List 5

List 5: *Linearity.m*

```
% Linearity.  Response of system1 to different inputs

% Close figures and clear memory
close all;clear;
% Definition of n axis (discrete time)
n=-10:10;

figure(1);
% Input signal x1
x1=2*delta(n)-delta(n-1)+2*delta(n-2);
subplot(211); % upper subfigure in figure 1
stem(n,x1,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' x1[n] ');
axis([-10 10 -1.5 2.5]);grid
```

```
% Output signal y1
y1=system1(x1);
subplot(212); % lower subfigure in figure 1
stem(n,y1,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' y1[n] ');
axis([-10 10 -2.5 5]);grid

figure(2);
% Input signal x2
x2=delta(n-3)+2*delta(n-4)+delta(n-5);
subplot(211); % upper subfigure in figure 2
stem(n,x2,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' x2[n] ');
axis([-10 10 -1.5 2.5]);grid
% Output signal y2
y2=system1(x2);
subplot(212); % lower subfigure in figure 2
stem(n,y2,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' y2[n] ');
axis([-10 10 -0.5 10]);grid

figure(3);
% Input signal x3 = Linear combination of x1 and x2
x3=3*x1-2*x2;
subplot(311); % upper subfigure in figure 3
stem(n,x3,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' x3[n] ');
axis([-10 10 -8 8]);grid
% Output signal y3
y3a=system1(x3);
subplot(312); % middle subfigure in figure 3
stem(n,y3a,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' y3a[n] ');
axis([-10 10 -18 18]);grid
% Combination of y1 and y2
y3b=3*y1-2*y2;
subplot(313); % lower subfigure in figure 3
stem(n,y3b,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' y3b[n] ');
axis([-10 10 -18 18]);grid
```

2.3 Causality

List 6 examines the response of the systems *system3* and *system4* to the input $x_1[n]$.

1. Run the *script Causality.m*.After the inspection of the figure plots, what can we say about the causality of *system3* and *system4*?

List 6: *Causality.m*

```
% Causality. Response of system3 and system4 to different inputs

% Close figures and clear memory
close all;clear;
% Definition of n axis (discrete time)
n=-10:10;

figure(1);
% Input signal x1
x1=delta(n)+2*delta(n-2);
subplot(211); % upper subfigure in figure 1
stem(n,x1,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' x1[n] ');
axis([-10 10 -1.5 2.5]);grid
```

```
% Output signal y1
y1=system3(x1);
subplot(212); % lower subfigure in figure 1
stem(n,y1,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' y1[n] ');
axis([-10 10 -2.5 5]);grid

figure(2);
% Input signal x1
x1=delta(n)+2*delta(n-2);
subplot(211); % upper subfigure in figure 2
stem(n,x1,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' x2[n] ');
axis([-10 10 -1.5 2.5]);grid
% Output signal y2
y2=system4(x1);
subplot(212); % lower subfigure in figure 2
stem(n,y2,'b','linewidth',2); xlabel('discrete time, n'); ylabel(' y2[n] ');
axis([-10 10 -2.5 5]);grid
```

2.4 System Combination

List 7 examines the output of the system *system5* to a given input $x[n]$. The system *system5* is defined as the combination and interconnection of a number of simpler sub-systems, as indicated in List 8.

List 7: *Combinacion.m*

```
%Combinación de sistemas

% Inicio de figuras y memoria
close all;
% Ejes de tiempos discreto
n=-10:10;

figure;
x=pulsow(n,4);
y=system5(x);
subplot(211)
stem(n,x,'b','linewidth',2);  xlabel('tiempo discreto, n'); ylabel(' x[n] ');
axis([-10 10 -1.5 1.5]);grid
subplot(212)
stem(n,y,'r','linewidth',2);  xlabel('tiempo discreto, n'); ylabel(' y[n] ');
axis([-10 10 -4.5 4.5]);grid
```

List 8: *system5.m*

```
function y=system5(x)
% System  y[n]=?

y1=systemA(x);
y2=systemB(y1);
y3=systemC(x);
y=y2+y3;
```

1. From the code in *Combinacion.m*, build a block diagram showing how the subsystems *systemA*, *systemB* and *systemC* are interconnected.

2. Run the script *Combinacion.m*. What can we say about the causality of *system5*?

3. The system *system5* is linear and time invariant. What's the expression for its impulse response? Calculate an analytical expression from the individual impulse responses of the systems *systemA*, *systemB* and *systemC*. Modify the script *Combinacion.m* and run it, so you can verify this result from the figure plots.