|  |  |  |  |
| --- | --- | --- | --- |
| **Cours : Traitement du signal**  **TP n°1 : Signal generation and manipulation I** | |  | Houssem Gazzah  CIPA4 Nantes |
|  | | | |
|  | *Nom MARQUET*  *Prenom Félix* | | |

**Introduction**

A signal is a function of time.

**In math**ematical notation, it is referred to by one variable x(t)

**In Matlab**, signals are not recognized as such

Matlab is only capable of handling scalars, vectors and matrices

i.e. 1x1-dim Nx1-dim and MxN-dim grids of numbers

To accommodate signals in Matlab, one way to do is as follows:

Two vectors are created:

t (represents time indexes) and

x (the corresponding signal values)

There, one can notice that while a signal x(t) runs from –∞ to +∞,

Matlab supports vectors of finite size only. This has two consequences:

First, we should fix the time interval in which we are interested.

It will be the same for all manipulated signals

Second, within this interval, not all time instants are considered.

Only consider those separated by a regular spacing (the sampling period)

The choice of the sampling period is the result of a compromise

If too big, fine details of the signal will be lost

If too small, very large vectors will be generated

**Example**

Generate x(t)=sin(t) and y(t)=cos(t)

x(t) is periodic with zeros at 0,±π,±2π,…

y(t) is periodic with zeros at ±π/2,±3π/2,±5π/2,…

First, select a time interval 🡪 t=[0,10]

Second, select a sampling period 🡪 dt=0.1

Fill in vector t 🡪 [MATLAB] t=0:0.1:10

Fill in vector x 🡪 [MATLAB] x=sin(t)

Fill in vector y 🡪 [MATLAB] y=cos(t)

Plot x(t) 🡪 [MATLAB] plot(t,x)

Plot y(t) 🡪 [MATLAB] plot(t,y)

Reset variable dt as π. What do you notice ?

**Experiment 1**

Generate and plot the signals described below (use GRID). Insert plots here

|  |  |
| --- | --- |
| sin(2.π.t) over [0,10] at a sampling rate of 10 Hz | sin(2.π.t) over [0,10] at a sampling rate of 100 Hz |
| sin(t) over [0,10] at a sampling rate of 10 Hz | sin(t)cos(t) over [0,10] at a sampling rate of 10 Hz |
| sin(t)/exp(t) over [0,10] at a sampling rate of 10 Hz | sin(t) +sin(t2)+sin2(t) over [0,10] sampled at 10 Hz |

|  |  |
| --- | --- |
| **Experiment 2**  Generate the signal    over [0,10] at a sampling rate of 10 Hz  Insert your figure on the right |  |

**Experiment 3**

**Sin(2.pi.t/T)**

**x(t)**

**y(t)**

In order to implement the following system

Set T=1 ms

Set the sampling period to ts=T/10

Set the time interval to [ts,10T]

Set K=1/100

Generate the input signal x(t)=K sin(2.pi.t/T) /t

a/ Plot the signal x(t) in dotted line with the horizontal axis properly labeled in ms**.**

b/ Find input-output relationship between y(t) and x(t) …………………=………………

c/ Plot the output signal

|  |  |
| --- | --- |
| Input | Output |

**Experiment 4**

Load the signal from *'holiday\_offer.mat'*

It has been sampled at the rate of 11025 samples/second

i/ Plot the portion of the signal that corresponds to samples 18100 to 18300

ii/ Plot the portion of the signal that corresponds to samples 18300,18280,18260,…,18100

iii/ Plot the portion of the signal that corresponds to samples 18100,18101…18300,8100,8101…,8300

iv/ Plot the portion of the signal that corresponds to the period 1.9 sec to 2 sec

|  |  |
| --- | --- |
| i/ | ii/ |
| iii/ | iv/ |

**Experiment 5**

|  |  |
| --- | --- |
| i/ Generate and plot the signal defined as follows  *m(t)=1*, if *t in [T,2T]*  *m(t)=t/T-3*, if *t in [3T,4T]*  *m(t)=0*, otherwise | Insert plot here |
| ii/ Generate and plot the signal *m(t)*, defined as follows  *n(t)=sin(2πt/T)*, if *t in [0,T/2]*  *n(t)=0*, otherwise | Insert plot here |
| iii/ Generate a delayed version of *n(t)* above, i.e. p*(t)=n(t- d), d=T/4* | Insert plot here |

**Experiment 6**

Consider a sampling period of 10 ms. Write a MATLAB code to generate the plots below

|  |  |
| --- | --- |
|  |  |
| Your plot here | Your plot here |

**Experiment 7**

Choose an appropriate sampling period to generate

the signal x(t)= cos(2π+20000t) sin(2π 30000t)+2 cos(22000t) over [0,0.0005s]

Define the new signal y(t) as follows

y(t) = 2 x(t) if |x(t)|>0.5

y(t) = x(t) otherwise

Plot x(t) and y(t) together in the same figure