

Project #4

Periodic Signals

In periodic signals, a basic building block of the signal repeats itself by a time separation called *period*. We will later see that periodic signals play an important role in the “Fourier Analysis” of signals. Here we will just create a continuous time and a discrete time periodic signal.

We should note here that, in actuality, periodic signals go on forever but we cannot realize this feature in Matlab. We can just create portions of periodic signals that are long enough for our purposes.

Part 1: Continuous Time Case

As an example of continuous time periodic function we will generate a cosine of frequency of 1Hz, over a time axis extending from -2 to 2 .

First we create the time axis in this range by 0.01 intervals.

```
» t=-2:0.01:2;
```

MCL 1

Can you tell the size of this t vector, i.e. how many points or samples it has? Check your answer by using the *length* or *whos*¹ command.

To simulate our continuous time signal in Matlab, we selected the sampling frequency as 100Hz, i.e. we will be getting 100 samples of the cosine signal, per second. This rate is quite high for our slowly varying cosine signal of 1Hz. While simulating a continuous time signal in Matlab, choosing the sampling frequency as 10 times the highest frequency in the signal is enough generally. Later, when we discuss Nyquist’s sampling theorem², we will see that this choice is more than enough when reconstruction of a signal from its samples is

¹ “whos” command lists all the variables and their sizes that are currently in Matlab’s workspace. Matlab workspace refers the chunk of memory that is allocated by the operating system to Matlab.

² This theorem states that a band-limited signal can be recovered from its samples as long as the sampling rate is at least two times the highest frequency in the signal.

considered. However, for the time being, we are just trying get signals that look like or resemble continuous time signals and for this purpose we are choosing very large sampling rates.

Let us now define the frequency of our cosine as a variable.

```
» f=1; MCL 2
```

This way, the frequency can be changed by just assigning a new value to f , without modifying all the lines referring to this frequency variable. This approach, i.e. using variables, is especially useful when writing Matlab script files³ or functions. We will now form our cosine signal and plot it.

```
» xt=cos(2*pi*f*t); MCL 3
```

```
» xt=cos(2*pi*f*t);subplot(2,1,1);plot(t,xt) MCL 4
```

```
» grid;xlabel('t');ylabel('x(t)') MCL 5
```

The result is shown in the upper panel of Figure 4.1. As you would expect, there are four periods of the cosine signal in the figure. Now, change the frequency, for instance make it 2Hz, and reissue the remaining lines to redo the plot. What do you observe, how many periods did you get this time?

Part 2: Discrete Time Case

To produce an example of discrete time periodic signal, let us create a template signal and repeat it. Let the period N be 3, and the template signal defined over the indices [0 1 2] be as follows.

```
» xx=[3 3 -1]; MCL 6
```

Now, we will repeat this template signal four times, to obtain a portion of the discrete time periodic signal that has xx as its one period.

³ Matlab script files called mfiles. An mfile consists of ordinary Matlab command lines that we usually issue at the Matlab prompt. In later projects, instead of putting down the command lines that we develop to do our tasks one by one, we will group and put them into mfiles. For information about writing functions in Matlab, type “help function” at the Matlab prompt.

```
» xn=[xx xx xx xx];
```

MCL 8

We will now generate the corresponding index vector and plot our signal.

```
» n=-6:5;
```

MCL 9

```
» subplot(2,1,2);stem(n,xn);
```

MCL 10

```
» grid;xlabel('n');ylabel('x[n]')
```

MCL 11

Note that we arbitrarily started our periodic signal portion at index -6 . We will finally use the *axis* utility to better see our signal.

```
» axis([-6.5 5.5 -2 4])
```

MCL 12

The result is shown in the lower panel of Figure 4.1.

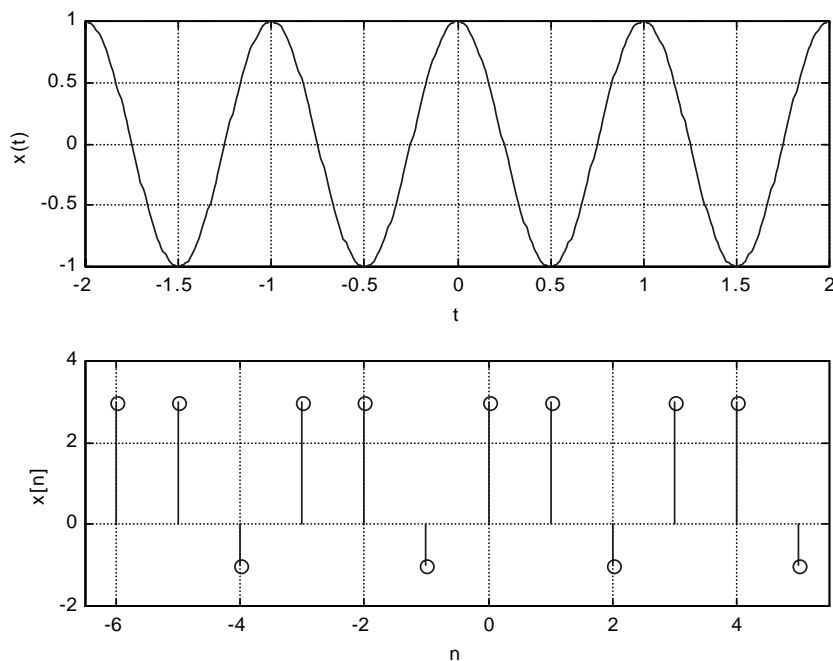


Figure 4.1 Examples of continuous time (upper panel) and discrete time (lower panel) periodic signals.