Project #5

Even and Odd Signals

Even and odd signals bear some important symmetry properties. Under reversal of independent variable, these signals either remain the same (even signal) or get reflected or flipped (odd signal) about the horizontal axis. Equations or definitions (5.1) and (5.2) mathematically express these properties for both continuous and discrete time cases.

Even Signals: $x(t) = x(-t)$, $x[n] = x[-n]$ (5.1)

Odd Signals: $x(t) = -x(-t)$, $x[n] = -x[-n]$ (5.2)

To explore even and odd signals in continuous time, let us create one even and one odd signal. The even one will be a cosine and the odd will be a sine, both will have frequency of 1 Hz, defined over a time axis extending from –1 to 1 with a sampling interval of 0.01s, each totaling 201 points or samples.

```
» f=1;                                    MCL 1
» t=-1:0.01:1;                            MCL 2
» xe=cos(2*pi*f*t);                       MCL 3
» xo=sin(2*pi*f*t);                       MCL 4
```

We can now see what happens to our functions under time reversal. We first examine the even one, and then the odd one.

```
» subplot(2,1,1);plot(t,xe);              MCL 5
» subplot(2,1,2);plot(-t,xe);             MCL 6
» figure                                  MCL 7
» subplot(2,1,1);plot(t,xo);              MCL 8
» subplot(2,1,2);plot(-t,xo);             MCL 9
```

What do you observe? How do you interpret the pictures you got? As you have realized the *figure* command at `MCL 7` opens a new figure window.

We will now see that it is possible to decompose any signal into its even and odd parts, as expressed by equations (5.3) through (5.5).

$$x_e(t) = \text{Even}\{x(t)\} = \frac{1}{2}[x(t) + x(-t)] \tag{5.3}$$

$$x_o(t) = \text{Odd}\{x(t)\} = \frac{1}{2}[x(t) - x(-t)] \tag{5.4}$$

$$x(t) = x_e(t) + x_o(t) \tag{5.5}$$

An exactly similar decomposition is possible in discrete time and we will do our demonstration in discrete time. Consider the following simple signal.

```
» n =-3:3;xn = [0 0 0 1 1 1 1];                          MCL 10
```

This signal is neither even nor odd. We can write the time reversed version of $x[n]$, i.e. $x[-n]$, as follows.

```
» xN = [1 1 1 1 0 0 0];                                  MCL 11
```

We could have obtained `xN` by using the *fliplr* (short for "flip left-right") function as follows.

```
» xN = fliplr(xn);
```

For longer vectors (signals) this method would be definitely better than writing the whole signal in reverse order. If we did not know that such a utility function exists, we could have reversed our vector by using Matlab's indexing or rearrangement features[1].

```
» xN = xn(length(xn):-1:1);
```

Back to our decomposition, even and odd parts of $x[n]$ can be computed as follows.

```
» xe =(xn+xN)/2;                                         MCL 12
```

[1] For more information about the colon notation that is frequently used in rearranging matrices, type "help colon" at the Matlab prompt.

```
» xo =(xn-xN)/2;                                              MCL 13
```

Can you now compute `xn-(xe+xo)` in Matlab, to verify our decomposition?

We can finally do a figure to see how our signals look. Figure 5.1 shows the discrete time signal that we operated on and its even and odd parts.

```
» subplot(3,1,1);stem(n,xn);axis([-4 4 -1 1.5])               MCL 14
» grid;ylabel('x[n]')                                          MCL 15
» title('Demonstration of Even-Odd Decomposition')             MCL 16
» subplot(3,1,2);stem(n,xe);axis([-4 4 -1 1.5])               MCL 17
» grid;ylabel('xe[n]')                                         MCL 18
» subplot(3,1,3);stem(n,xo);axis([-4 4 -1 1.5])               MCL 19
» grid;ylabel('xo[n]');xlabel('n')                             MCL 20
```

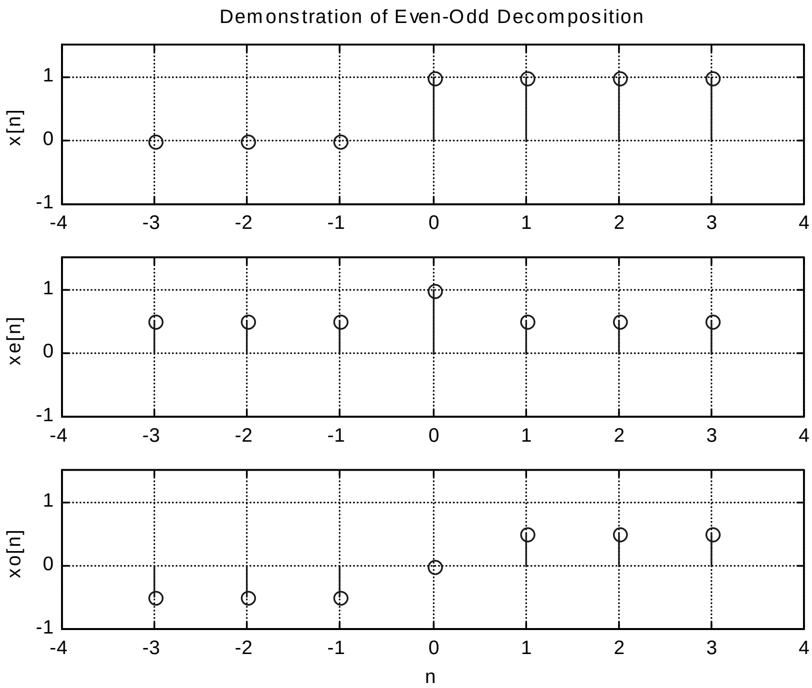

Figure 5.1 Demonstration of even-odd decomposition a discrete time signal. A discrete time signal (upper panel), its even (middle panel) and odd (lower panel) parts.