Lab Session 02
Apply elementary operations on signals

Operation with signals means to add, subtract, multiply, divide, scale, exponentiation, shift, delay or advance and to flip signals. MATLAB allows all these operations but we need to be careful in computation because the vector representation of the signals should have the same time origins and the same number of elements.

Basic Signal Operations:
Given the signals x1 and x2, let us see how six basic operations on the signals can be implemented in MATLAB. If x1 = 5 sin(pi/4 t) and x2 = 3 cos(pi/7 t) then the required operations are listed below as y1 through y6 with the respective plots shown in Fig. 2.1.

Operations: **y1** = x1 + x2; **y2** = x1 - x2; **y3** = x1 * x2; **y4** = x1 / x2; **y5** = 2 x1; **y6** = $(x1)^3$

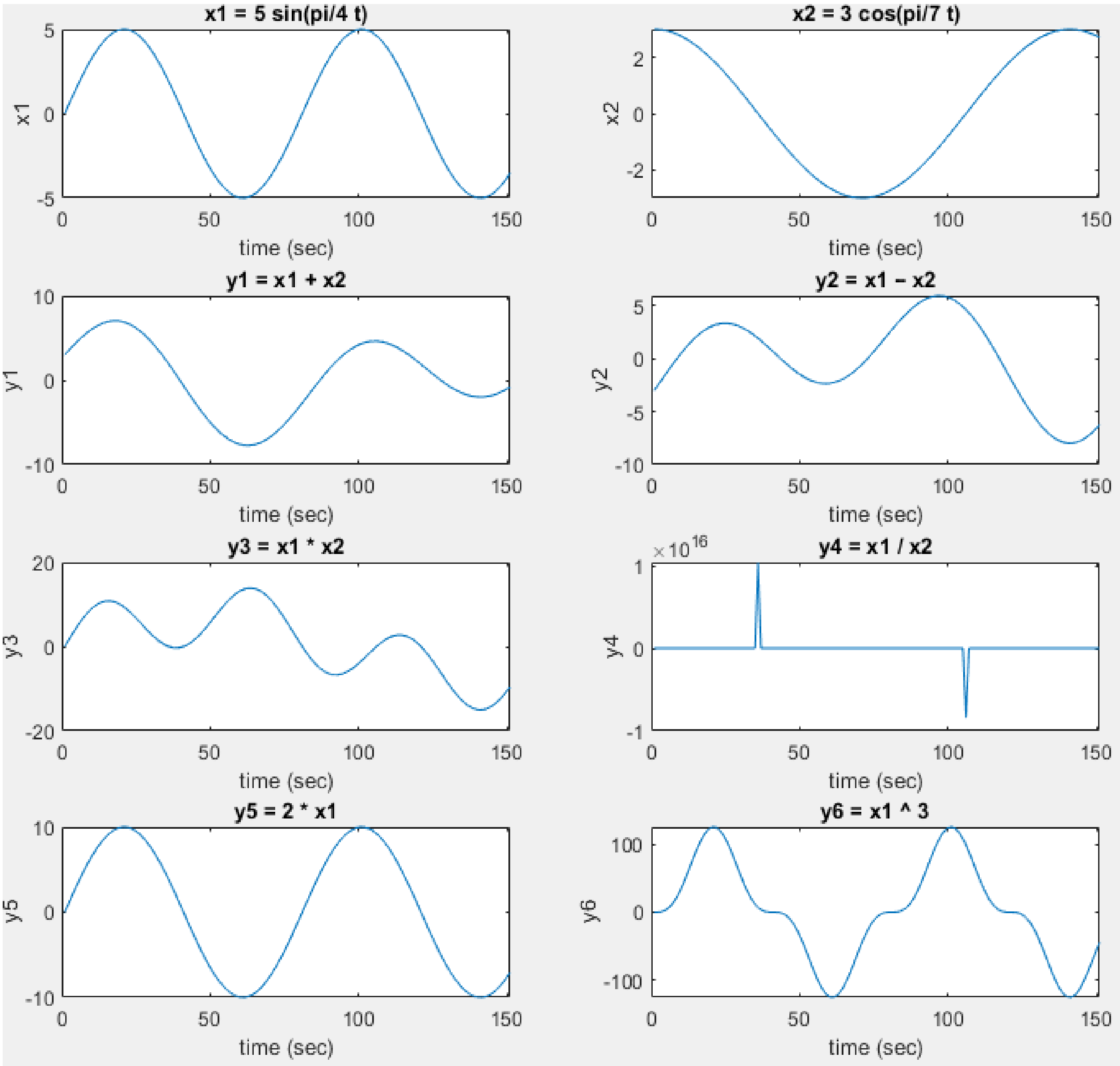

Fig. 2.1 Subplot showing basic signal operations on signals x1 and x2

Given below is the MATLAB code for plotting original sinusoids x1, x2 and the six basic operations.

```
x1 = 5*sin((pi/4)*[0:0.1:15])
subplot(4,2,1), plot(x1)
title('x1 = 5 sin(pi/4 t)')
xlabel('time (sec)')
ylabel('x1')
```

```
x2 = 3*cos((pi/7)*[0:0.1:15])
subplot(4,2,2), plot(x2)
title('x2 = 3 cos(pi/7 t)')
xlabel('time (sec)')
ylabel('x2')
```

Addition

```
y1 = x1 + x2 % addition
subplot(4,2,3), plot(y1)
title('y1 = x1 + x2')
xlabel('time (sec)')
ylabel('y1')
```

Subtraction

```
y2 = x1 - x2 % subtraction
subplot(4,2,4), plot(y2)
title('y2 = x1 – x2')
xlabel('time (sec)')
ylabel('y2')
```

Multiplication

```
y3 = x1 .* x2 % multiplication
subplot(4,2,5), plot(y3)
title('y3 = x1 * x2')
xlabel('time (sec)')
ylabel('y3')
```

Division

```
y4 = x1 ./ x2 % division
subplot(4,2,6), plot(y4)
title('y4 = x1 / x2')
xlabel('time (sec)')
ylabel('y4')
```

Scaling

```
y5 = 2*x1 % scaling
subplot(4,2,7), plot(y5)
title('y5 = 2*x1')
xlabel('time (sec)')
ylabel('y5')
```

Exponentiation

```
y6 = x1 .^3 % exponentiation
subplot(4,2,8), plot(y6)
title('y6 = x1 ^ 3')
xlabel('time (sec)')
ylabel('y6')
```

More on Signal Operations

Time Shifting

One of the most basic operations in a DSP system is to shift the time reference. The shift may be considered either a delay (- or right shift) and an advance (+ or left shift). If δ[n] is the representation of unit impulse signal in discrete domain, then δ[n + N] and δ[n - N] represents the signal advanced and delayed by N samples respectively. The unit impulse signal, its advanced and delayed forms with N=1 are shown below in Fig. 2.2.

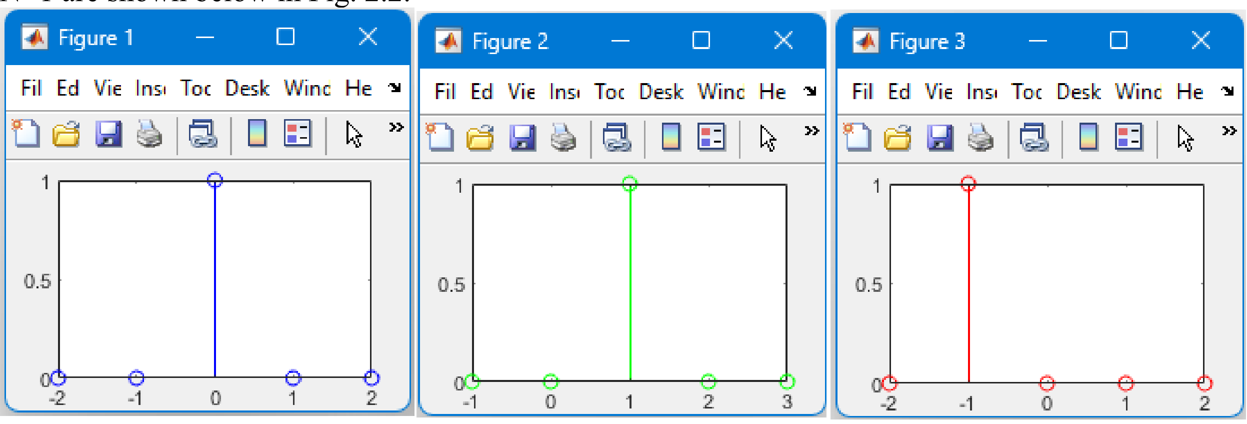

Fig. 2.2 Plots for δ[n] and shifted δ[n] right and left

A custom function in MATLAB is given below. This function takes x, n, k as inputs and returns y, n as outputs. This function will shift the signal x for k samples and save the shifted signal as y.

```
function [y, n] = SigShift(x, n, k)
n = n + k;
y = x;
```

Example:

$$x(t) = \cos(\pi t) \qquad -0.5 \le t \le 0.5$$

Shift x (t) by 0.5 units on the right

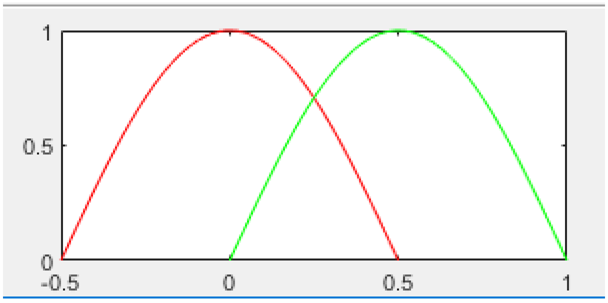

Fig. 2.3 Plots for x(t) and shifted (delayed) x(t)

Time Scaling

If f(t) is compressed in time by a factor of 'a' where (a>1) then the resulting signal is given by:

ϴ (t) = f(at)

Similarly if f(t) is expanded in time by a factor of 'a' where (a>1) then the resulting signal is given by:

ϴ (t) = f(t/a)

Fig. 1.4 shows the original signal and how it is scaled in time to give its compressed and expanded versions. The figure is followed by its MATLAB code.

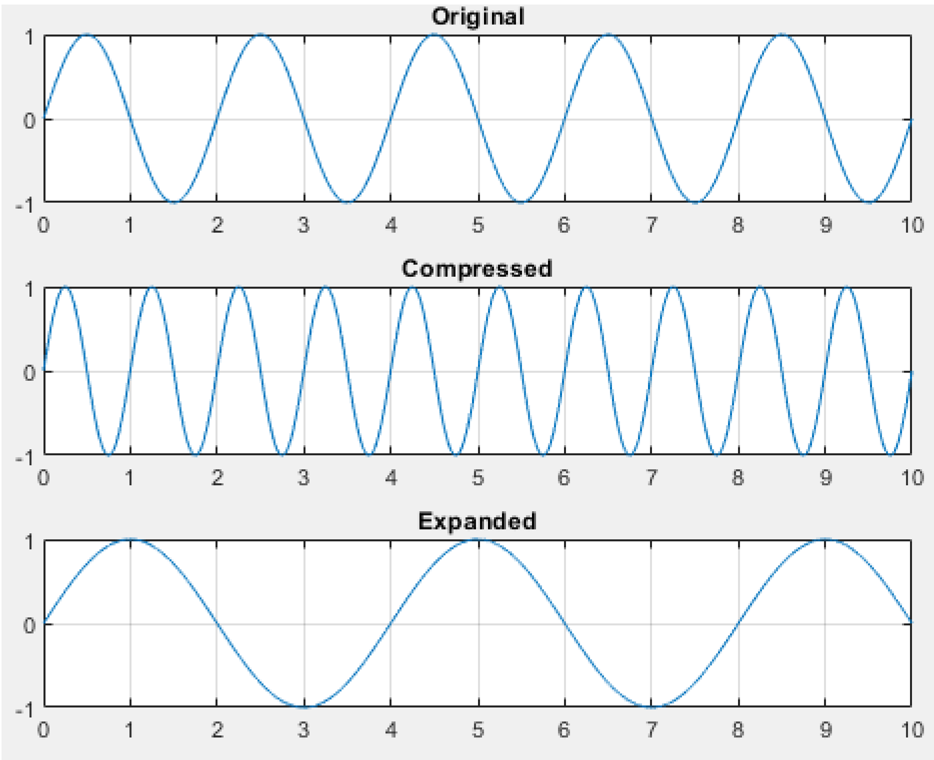

Fig. 2.4 Time scaling - Plots against original signal, its compressed and expanded forms

Example
If x = sin(πt); for t = 0:0.001:10
Find: ϴ (t) = x(2*t) & ϴ (t) = x(t/2)

MATLAB Code:
```
t = 0:0.001:10;
time = pi*t;
orig = sin(time);
scale = 2;                  %scaling Factor
comp = sin(scale*time);     %Compression
exp = sin(time/scale);      %Expansion
figure;
subplot(3,1,1); plot(t, orig); title('Original'); grid on;
subplot(3,1,2); plot(t, comp); title('Compressed'); grid on;
subplot(3,1,3); plot(t, exp); title('Expanded'); grid on;
```

Time Inversion (Folding)

Folding involves the reversal of the time axis. As in Fig 2.5, it shows two signals inverted to each other.

$$\Theta(t) = f(-t)$$
$$\Theta(-t) = f(t)$$

To invert a signal, we replace t by –t.

Consider the following example:

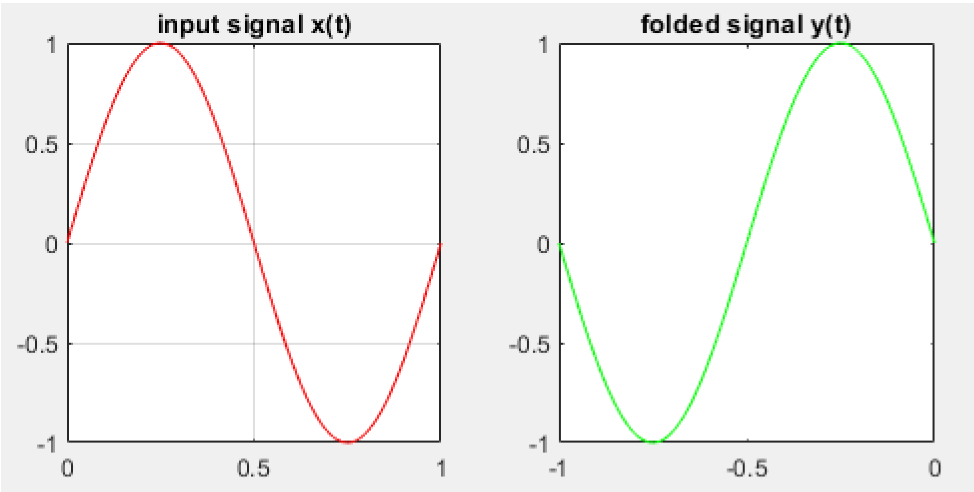

Fig. 2.5 Plots showing input signal and its folded (inverted) signal

MATLAB Code:
```
t=0:0.01:1;
x=sin(2*pi*t);
subplot(1,2,1), plot(t,x,'r-'); grid on;
title('input signal x(t)');
y=x;
t=-t;
subplot(1,2,2), plot(t,y,'g-');
title('folded signal y(t)');
```

Operations on DT Signals

Signal Addition

This is sample-by-sample addition given by:
$\{x1(n)\} + \{x2(n)\} = \{x1(n) + x2(n)\}$

Implemented in MATLAB by '+' operator. However, the lengths of $x1(n)$ and $x2(n)$ must be same.

Signal Multiplication

This is sample-by-sample multiplication given by:
$\{x1(n)\}.\{x2(n)\} = \{x1(n)\, x2(n)\}$
It is implemented in MATLAB by '$*$' operator. However, the lengths of $x1(n)$ and $x2(n)$ must be same.

Exercise

1. Write down MATLAB code to perform the signals operations discussed in lab.
2. The operations discussed in lab can also be performed on DT signals (sample by sample). Write MATLAB scripts to generate and plot each of the following signals for the intervals indicated.

a) $x(n) = n^2[u(n+5) - u(n-6)] + 10\delta(n) + 20(0.5)^n[u(n-4) - u(n-10)] \qquad -10 \le n \le 10$

b) $x(n) = (0.2)^n[u(n-5) + u(n+6)]\delta(n-5) + 20(0.5)^n[u(n+4)] \qquad -10 \le n \le 10$