Name:	Muhammad Attiq
Section:	BCE – 4A
Reg. No:	FA23 – BCE – 060
Course:	Signals & Systems
Lab Report:	11
Course Instructor:	Dr. Bilal Qasim

LAB 11

Continuous Time Fourier Transform (CTFT)

Task-1

Compute the Fourier transform of $x(t) = \mathrm{e}^{-t^2}$. MATLAB code is given in following, run this code and compare your output using eq. (11.3). Write your code and results in following.

```
syms t w
x = exp(-t^2);
X = fourier(x, w);
disp('Fourier Transform of x(t) = exp(-t^2) is:')
pretty(X)
```

```
>> TASK_1_PART_1
Fourier Transform of x(t) = exp(-t^2) is:
                  /    2 \
                  |   w  |
sqrt(pi)  exp| - -- |
                  \    4 /
```

Using Equation (11.3):

```
syms t w real
x = exp(-t^2);
F = x * exp(-1i * w * t);
X_manual = int(F, t, -inf, inf);
disp('Fourier Transform using Equation (11.3):')
pretty(X_manual)
```

```
>> TASK_1_PART_2
Fourier Transform using Equation (11.3):
                  /    2 \
                  |   w  |
sqrt(pi)  exp| - -- |
                  \    4 /
```

Task-2

Compute the inverse Fourier transform of X = exp(-1/4*w^2)*pi^(1/2) using the command for inverse Fourier transform and also verify your result using eq. (11.7). Give your results in following.

```
syms w t real
X = exp(-1/4*w^2)*pi^(1/2);
x_ifourier = ifourier(X, w, t);
disp('Inverse Fourier Transform using ifourier():')
pretty(x_ifourier)
```

```
>> TASK_2_PART_1
Inverse Fourier Transform using ifourier():
        2
exp(- t ) 3991211251234741
--------------------------
 2251799813685248 sqrt(pi)
```

Using Equation(11.7):

```
syms w t real
X = exp(-1/4*w^2)*pi^(1/2);
F = X * exp(1i * w * t);
x_manual = (1/(2*pi)) * int(F, w, -inf, inf);
disp('Inverse Fourier Transform using Equation (11.7):')
pretty(simplify(x_manual))
```

```
>> TASK_2_PART_2
Inverse Fourier Transform using Equation (11.7):
                  2
sqrt(pi) exp(- t ) 22886248455258496714918675196319
---------------------------------------------------
          40564819207303340847894502572032
```

Task-3

Compute the inverse Fourier transform of the function $X(\Omega) = 1/(1 + j\Omega)$ using command of Fourier and then take inverse of the resultant x(t) to produce again $X(\Omega)$.

```
syms w t real
X = 1 / (1 + 1i * w);
x = ifourier(X, w, t);
disp('Inverse Fourier Transform x(t):')
pretty(x)
X_recovered = fourier(x, t, w);
disp('Recovered X(w) from Fourier of x(t):')
pretty(simplify(X_recovered))
```

```
>> TASK_3
Inverse Fourier Transform x(t):
exp(-t) heaviside(t)

Recovered X(w) from Fourier of x(t):
   1
-------
w i + 1
```

Task-4

Let x(t) = 1, compute its Fourier transform to produce X(w) and then take inverse Fourier transform of X(w) to get back x(t), using commands of Fourier transform.

```
syms t w real
x = 1;
X = fourier(x, t, w);
disp('Fourier Transform X(w) of x(t) = 1:')
pretty(X)
x_recovered = ifourier(X, w, t);
disp('Inverse Fourier Transform of X(w):')
pretty(x_recovered)
```

```
>> TASK_4
Fourier Transform X(w) of x(t) = 1:
2 pi dirac(w)

Inverse Fourier Transform of X(w):
1
```

Task-5

Let x(t) = u(t), compute its Fourier transform, take inverse Fourier transform of the resultant signal to get back x(t).

```
syms t w real
x = heaviside(t);
X = fourier(x, t, w);
disp('Fourier Transform X(w) of x(t) = u(t):')
pretty(X)
x_recovered = ifourier(X, w, t);
disp('Recovered x(t) using inverse Fourier
Transform:')
pretty(simplify(x_recovered))
```

```
>> TASK_5
Fourier Transform X(w) of x(t) = u(t):
              i
pi dirac(w) - -
              w

Recovered x(t) using inverse Fourier Transform:
heaviside(t)
```

Task-6

Let x(t) = δ(t), compute its Fourier transform, take inverse Fourier transform of the resultant signal and state whether it is possible to get back x(t) or not?

```
syms t w real
x = dirac(t);
X = fourier(x, t, w);
disp('Fourier Transform X(w) of x(t):')
pretty(X)
x_recovered = ifourier(X, w, t);
disp('Recovered x(t) using inverse Fourier Transform:')
pretty(x_recovered)
```

```
>> TASK_6
Fourier Transform X(w) of x(t):
1

Recovered x(t) using inverse Fourier Transform:
dirac(t)
```

Task-7

Prove that $x(t) = \delta(t-2)$ and $X(\Omega) = e^{-j2\Omega}$, are Fourier transform pairs of each other.

```
syms t w real
x = dirac(t - 2);
X = fourier(x, t, w);
disp('Fourier Transform X(w) of x(t) = (t - 2):')
pretty(X)
x_recovered = ifourier(X, w, t);
disp('Recovered x(t) from inverse Fourier of X(w):')
pretty(x_recovered)
```

```
>> TASK_7
Fourier Transform X(w) of x(t):
exp(-w 2 i)

Recovered x(t) from inverse Fourier of X(w):
dirac(t - 2)
```

Task-8

Prove that $x(t) = u(t-2)$ and $X(\Omega) = \exp(-2 * j * w) * (pi * \delta(w) - j/w)$, are Fourier transform pairs of each other.

```
syms t w real
x = heaviside(t - 2);
X = fourier(x, t, w);
x_rec = ifourier(X, w, t);
disp('Fourier Transform X(w) of x(t):')
disp(X)
disp('Recovered x(t) using inverse Fourier
Transform:')
disp(simplify(x_rec));
```

```
>> TASK_8
Fourier Transform X(w) of x(t):
exp(-w*2*i)*(pi*dirac(w) - i/w)

Recovered x(t) using inverse Fourier Transform:
heaviside(t - 2)
```

Post Lab Task

Critical Analysis

This lab is very useful because it helps us understand the concept of the Continuous Time Fourier Transform (CTFT) in a better way. In class, we learn the theory, but in the lab, we actually apply it using MATLAB. This makes it easier to see how signals change from time domain to frequency domain. By using MATLAB functions like **fourier** and **ifourier**, we can find the transform and also recover the original signal. This helps us check if our results are correct. Doing these kinds of labs also improves our understanding of signal behavior, helps us solve problems. Overall, this lab helps us connect theory with practical work and prepares us for more advanced topics in signals and systems.

<table>
<tr><th colspan="3">Lab Assessment</th></tr>
<tr><td>Lab Task Evaluation</td><td>/6</td><td rowspan="2">/10</td></tr>
<tr><td>Lab Report</td><td>/4</td></tr>
<tr><td colspan="3">Instructor Signature and Comments</td></tr>
</table>