**PRACTICE 5**

**CONTINUOUS-TIME FOURIER TRANSFORM (FT)**

**AND**

**DISCRETE-TIME FOURIER TRANSFORM (DTFT)**

**OBJECTIVES:**

1. Calculate the Fourier Transform of aperiodic signals x(t).
2. Calculate the DT Fourier Transform of aperiodic sequences x[n].

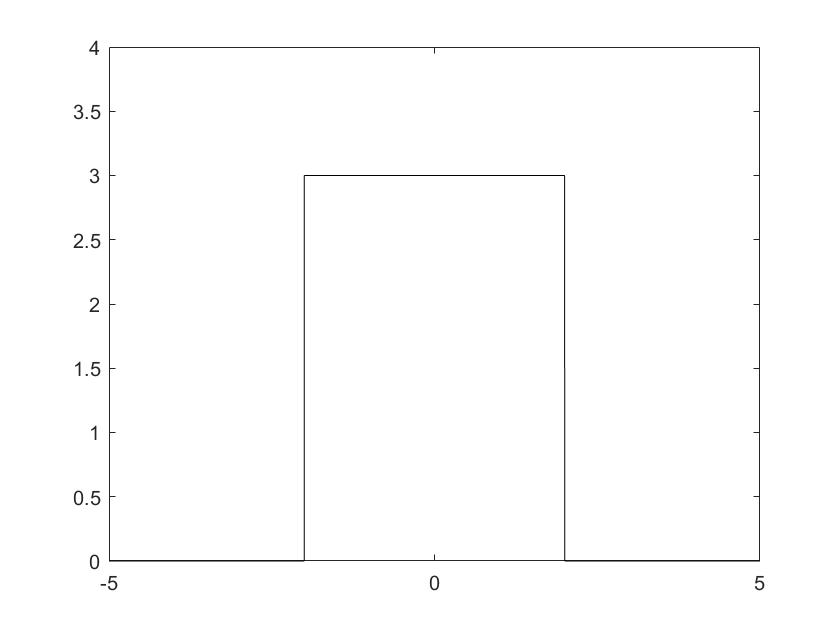
|  |  |
| --- | --- |
| **Name 1:** | Alejandro Serrano López |
| **Name 2:** | Mihnea-Iulian Popescu |

1. **Fourier Transform of aperiodic signals x(t)**

Given an integrable function x(t), the Fourier Transform of x(t) is defined as:

and the inverse Fourier Transform of is:

The following MATLAB script let you calculate the Fourier Transform of a rectangular pulse:



% Fourier Transform of a rectangular pulse

syms t w

a=2;

A=3;

x(t)= A\*heaviside(t+a)-A\*heaviside(t-a);

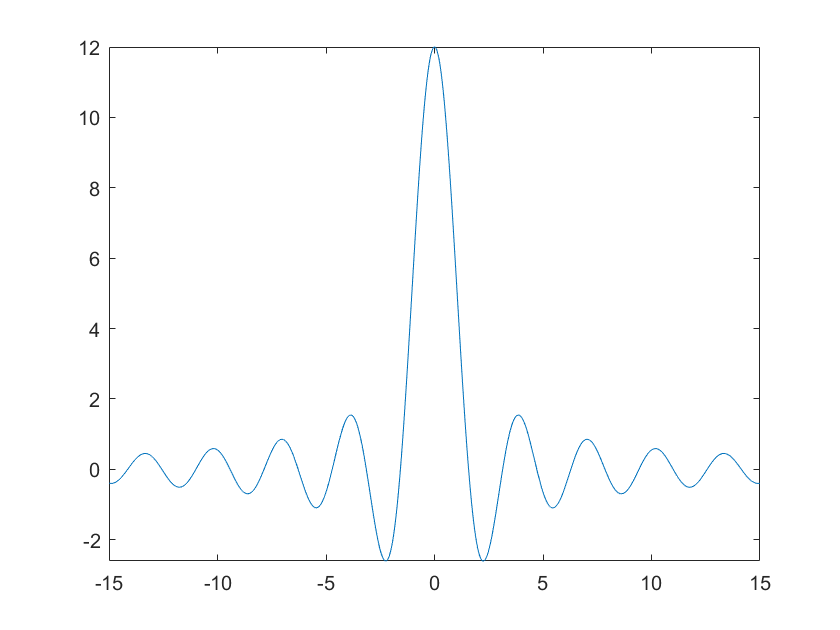
F(w) = fourier(x);

clc

disp('Fourier Transform: ')

F(w)=simplify(F(w))

fplot(F,[-15,15])



1. Define a MATLAB function that obtains the **Fourier Transform** of a symbolic signal x(t) and plot in the interval [a,b] (both module and phase).

function [F] = MyFourier(x,a,b)

|  |
| --- |
| **MyFourier.m** |
| **function [F] = MyFourier(x,a,b)**  % Fourier Transform  syms w  % the output and input of fourier is symbolic  F(w) = fourier(x);  clc  disp('Fourier Transform: ')  F(w)=simplify(F(w))  subplot(2,1,1);  % Module  fplot(abs(F),[a,b])  subplot(2,1,2);  % System Identification Toolbox is needed for phase()  fplot(phase(F),[a,b])  **end** |

1. Use the previous MATLAB function to calculate **the Fourier Transforms** of the following signals:

**TRIANGULAR PULSE**

syms t

x(t)=(1+t)\*(heaviside(t+1)-heaviside(t))+(1-t)\*(heaviside(t)-heaviside(t-1));

|  |  |
| --- | --- |
| **PLOT x(t) in [-5,5]** | **PLOT F(w) in [-10,10]** |
| >> fplot(x, [-5, 5], 'r')  Gráfico, Gráfico de líneas  Descripción generada automáticamente | **F(w) =** -(2\*cos(w) - 2)/w^2  % we called this “FourierA” in our console |

%General triangular pulse, a curiosity

A=3; a=2;

x(t)=(A+(A/a)\*t)\*(heaviside(t+a)-heaviside(t))+(A-+(A/a)\*t)\*(heaviside(t)-heaviside(t-a));

**EXPONENTIAL FUNCTION**

syms t

A=3; b=1;

x(t) = A\*exp(-b\*abs(t));

|  |  |
| --- | --- |
| **PLOT x(t) in [-5,5]** | **PLOT F(w) in [-10,10]** |
| >> fplot(x, [-5, 5], 'r')  Gráfico, Histograma  Descripción generada automáticamente | **F(w) =** 6/(w^2 + 1)  % we called this “FourierB” in our console |

**FORCED OSCILLATION**

syms t

b=1; w0=6;

x(t) = exp(-b\*t).\*cos(w0\*t).\* heaviside(t);

|  |  |
| --- | --- |
| **PLOT x(t) in [0,5]** | **PLOT F(w) in [-10,10]** |
| >> fplot(x, [0, 5], 'r') | **F(w) =** (1 + w\*1i)/(- w^2 + w\*2i + 37)  % we called this “FourierC” in our console |

**DIRAC FUNCTION**

syms t

x(t) = dirac(t);

fplot(x,[0,5])

|  |  |
| --- | --- |
| **PLOT x(t) in [0,5]** | **PLOT F(w) in [-10,10]** |
| >> fplot(x, [0, 5], 'r')  The value of dirac is infinite in 0, and 0 everywhere else!  There’s a trick, telling MATLAB to draw a line by telling it to represent inf as a capped value (f.ex. 3). | **F(w) =** 1  % we called this “FourierD” in our console  Integration between -inf and inf for dirac delta is this value. |

**COSINE FUNCTION**

syms t

x(t) = cos(t);

|  |  |
| --- | --- |
| **PLOT x(t) in [0,5]** | **PLOT F(w) in [-10,10]** |
| >> fplot(x, [0, 5], 'r')  Gráfico, Histograma  Descripción generada automáticamente | **F(w) =** pi\*(dirac(w - 1) + dirac(w + 1))  % we called this “FourierE” in our console  Gráfico, Gráfico de líneas  Descripción generada automáticamente |

1. Use the MATLAB function “ifourier” to calculate **the inverse Fourier Transforms** of the previous Fourier transforms. Plot the .

|  |  |
| --- | --- |
| **TRIANGULAR PULSE**  **Instructions**  >> x1r = ifourier(FourierA)  -(2\*pi\*x\*sign(x) + 2\*fourier(cos(w)/w^2, w, -x))/(2\*pi)  >> x1rs = simplify(x1r)  -(pi\*x\*sign(x) + fourier(cos(w)/w^2, w, -x))/pi  >> syms w t  >> x1rIntegrate = 1/(2\*pi) \* int(FourierA \* exp(i \* w \* t), -inf, inf)    x1rIntegrate =    (5734161139222659\*int(-(exp(t\*w\*1i)\*(2\*cos(w) - 2))/w^2, w, -Inf, Inf))/36028797018963968  MATLAB CANNOT RECOVER THE ORIGINAL SYMBOLIC EXPRESSION  x(t) = MATLAB CANNOT RECOVER THE ORIGINAL SYMBOLIC EXPRESSION  (but we know it’s the Triangular pulse)  >> fplot(x1r, [-5, 5], 'r') | **EXPONENTIAL FUNCTION**  **Instructions**  >> x2r = ifourier(FourierB)  **x(t) =** 3\*exp(-abs(x))  >> fplot(x2r, [-5, 5], 'r') |
| **PLOT x(t)** | **PLOT x(t)**  **Gráfico, Histograma  Descripción generada automáticamente** |

|  |  |
| --- | --- |
| **FORCED OSCILLATION**  **Instructions**  >> x3r = ifourier(FourierC)  **x(t) =** ((pi\*exp(x\*(- 1 - 6i))\*(sign(x) + 1))/2 + (pi\*exp(x\*(- 1 + 6i))\*(sign(x) + 1))/2 + (pi\*exp(x\*(- 1 - 6i))\*dirac(x)\*1i)/6 - (pi\*exp(x\*(- 1 + 6i))\*dirac(x)\*1i)/6)/(2\*pi)  >> fplot(x3r, [-5, 5], 'r') | **COSINE FUNCTION**  **Instructions**  >> x5r = ifourier(FourierE)  **x(t) =** exp(-x\*1i)/2 + exp(x\*1i)/2  >> fplot(x5r, [-5, 5], 'r') |
| **PLOT x(t)**  Gráfico  Descripción generada automáticamente | **PLOT x(t)Gráfico, Gráfico de líneas  Descripción generada automáticamente** |