**PRACTICE 6**

**DISCRETE FOURIER TRANSFORM (DFT)**

**FAST FOURIER TRANSFORM**

**OBJECTIVE:**

1. Calculate the Discrete Fourier Transform of sequences x[n].
2. Calculate the Fast Fourier Transform of sequences x[n].

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

1. **Discrete Fourier Transform of sequences x[n]**

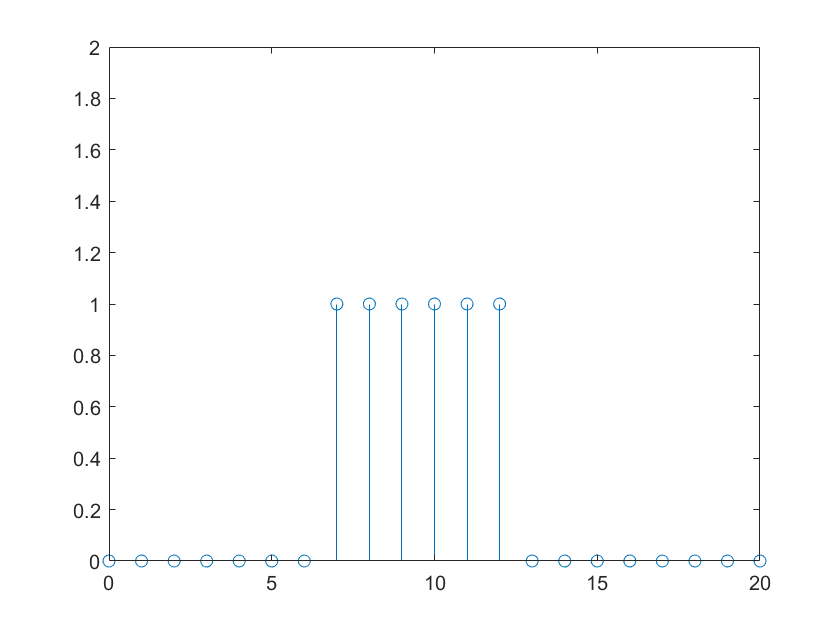
Given a finite sequence x[n] for n=0, …, N-1, the Discrete Fourier Transform of x[n] is defined as:

|  |
| --- |
| Remember Fourier coefficients of a T-periodic signal x(t): |

and the inverse Discrete Fourier Transform of is:

|  |
| --- |
| Remember the N-order partial sum of Fourier series of T-periodic x(t): |

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



% Discrete Fourier Transform of a rectangular pulse

[x,n] = rectan(7,12,0,20);

N=length(x);

syms k

X(k)= sum(x.\*exp(-1i\*k.\*2\*pi\*n/N));

subplot(2,1,1); stem(n,abs(X(n)))

subplot(2,1,2); stem(n,angle(X(n)))

Imagen que contiene Gráfico

Descripción generada automáticamente

syms w

F(w)= sum(x.\*exp(-1i\*w.\*n));

subplot(2,1,1); fplot(sqrt(real(F)^2+imag(F)^2),[0,2\*pi])

subplot(2,1,2); fplot(angle(F),[0,2\*pi])

Gráfico

Descripción generada automáticamente

If X[k] is x-axis scaled and plotted together with F(w):

N=length(x);

m = n\*2\*pi/N;

subplot(2,1,1)

stem(m,abs(X(n)))

hold on

fplot(sqrt(real(F)^2+imag(F)^2),[0,2\*pi])

axis([0 2\*pi 0 6])

subplot(2,1,2)

stem(m,angle(X(n)))

hold on

fplot(angle(F),[0,2\*pi])

axis([0 2\*pi -4 4])

Gráfico, Histograma

Descripción generada automáticamente

1. Define a MATLAB function that obtains the symbolic **Discrete Fourier Transform** of a sequence x[n] defined in [n1,n2] and plot the in the interval [] (both module and phase).

function [X] = MyDFT(x,n)

|  |
| --- |
| **MyDFT.m** |
| **function [X] = MyDFT(x,n)**  N=length(x);  syms k  X(k)= sum(x.\*exp(-1i\*k.\*2\*pi\*n/N));  subplot(2,1,1); stem(n,abs(X(n))) %stem(n, sqrt(real(X(n))^2+imag(X(n))^2)  subplot(2,1,2); stem(n,angle(X(n)))  **end** |

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

**SAMPLED EXPONENTIAL FUNCTION in the interval [-5,5] with sample period T=0.3**

syms t; A=3; b=1;

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

fplot(x,[-5,5])

t = -5:0.3:5;

sef=x(t);

n=-17:16;

|  |  |
| --- | --- |
| **PLOT x[n]** | **PLOT X(w) in [-pi,pi]** |
|  | [F] = MyDTFT(sef,n); |
| **PLOT x[n]** | **PLOT X[k] in [-17,16]** |
|  | [X] = MyDFT(sef,n); |
| N=length(sef);  m = n\*2\*pi/N;  subplot(2,1,1)  stem(m,abs(X(n)))  hold on  fplot(abs(F),[-pi,pi])  axis([-pi pi 0 20])  subplot(2,1,2)  stem(m,angle(X(n)))  hold on  fplot(angle(F),[-pi,pi])  axis([-pi pi -1 1]) |  |

**SAMPLED FORCED OSCILLATION in the interval [0,5] with sample period T=0.1**

syms t; b=1; w0=6;

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

fplot(x,[0,5])

t = 0:0.1:5;

sfo=x(t);

n=0:50;

|  |  |
| --- | --- |
| **PLOT x[n]** | **PLOT X(w) in [-pi,pi]** |
|  | [F] = MyDTFT(sfo,n);  **Gráfico, Gráfico de líneas  Descripción generada automáticamente** |
| **PLOT x[n]** | **PLOT X[k] in [0,50]** |
|  | [X] = MyDFT(sfo,n);  **Gráfico, Histograma  Descripción generada automáticamente** |
| N=length(sfo);  m = n\*2\*pi/N; % done so X(ω) and X(k) are plotted together  subplot(2,1,1) % MODULE  stem(m,abs(X(n))) % DFT  hold on  fplot(sqrt(real(F)^2+imag(F)^2), [0,2\*pi]) % DTFT  axis([0 2\*pi 0 20])  subplot(2,1,2) % phase  stem(m,angle(X(n)))  hold on  fplot(angle(F),[0,2\*pi])  axis([0 2\*pi -1.5 1.5]) | Here I decided to plot it on [0, pi] so we can clearly see it.  Gráfico, Histograma  Descripción generada automáticamente |

1. Define a MATLAB function that obtains **the inverse** **Discrete Fourier Transform** .

function [x] = MyiDFT(X,k)

|  |
| --- |
| **MyiDFT.m** |
| **function [x] = MyiDFT(X,k)** % X is the symbolic function of X(k)  N=length(k); % lengths of X(omega) and X(k) match  syms n  x(n) = (1/N) \* sum(X(k).\*exp(1i.\*n.\*2.\*pi.\*k/N));  **end** |

1. Use the previous MATLAB function to calculate **the inverse DFT** in the case of:

**SAMPLED EXPONENTIAL FUNCTION**

**Gráfico, Histograma

Descripción generada automáticamente**

|  |  |
| --- | --- |
| **PLOT X[k] in [-17,16]** | **PLOT x[n] in [-75,75]** |
| [X] = MyDFT(sef,n); | k=-17:16;  [y] = MyiDFT(X,k);  m = -75:75; stem(m,y(m)) |

**SAMPLED FORCED OSCILLATION**

**Gráfico, Histograma

Descripción generada automáticamente**

|  |  |
| --- | --- |
| **PLOT X[k] in [0,50]** | **PLOT x[n] in [-100,100]** |
| [X] = MyDFT(sfo,n);  **Gráfico, Histograma  Descripción generada automáticamente** | k= 0:50;  [y2] = MyiDFT(X,k);  m = -100:100; stem(m,y2(m))  Gráfico  Descripción generada automáticamente |

1. **Fast Fourier Transform of sequences x[n]**

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

Gráfico, Histograma

Descripción generada automáticamente

% Fast Fourier Transform of a rectangular pulse

[x,n] = rectan(7,12,0,20);

X= fft(x);

subplot(2,1,1); stem(n,abs(X))

subplot(2,1,2); stem(n,angle(X))

1. Use the MATLAB function “**fft**” to calculate **the Fast Fourier Transforms** of the following sequences:

**SAMPLED EXPONENTIAL FUNCTION in the interval [0,10] with sample period T=0.3**

syms t; A=3; b=1;

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

t = 0:0.3:10;

sef=x(t);

n=0:33;

X= fft(double(sef))

subplot(2,1,1); stem(n,abs(X))

subplot(2,1,2); stem(n,angle(X))

X =

Columns 1 through 6

11.5745 + 0.0000i 8.8239 - 4.4193i 5.5469 - 4.8007i 3.8409 - 4.0468i 2.9910 - 3.2988i 2.5317 - 2.7038i

Columns 7 through 12

2.2617 - 2.2393i 2.0919 - 1.8695i 1.9792 - 1.5671i 1.9015 - 1.3129i 1.8462 - 1.0938i 1.8063 - 0.9005i

Columns 13 through 18

1.7771 - 0.7263i 1.7559 - 0.5663i 1.7409 - 0.4166i 1.7309 - 0.2740i 1.7251 - 0.1359i 1.7233 + 0.0000i

Columns 19 through 24

1.7251 + 0.1359i 1.7309 + 0.2740i 1.7409 + 0.4166i 1.7559 + 0.5663i 1.7771 + 0.7263i 1.8063 + 0.9005i

Columns 25 through 30

1.8462 + 1.0938i 1.9015 + 1.3129i 1.9792 + 1.5671i 2.0919 + 1.8695i 2.2617 + 2.2393i 2.5317 + 2.7038i

Columns 31 through 34

2.9910 + 3.2988i 3.8409 + 4.0468i 5.5469 + 4.8007i 8.8239 + 4.4193i

Gráfico, Histograma

Descripción generada automáticamente

**SAMPLED FORCED OSCILLATION in the interval [0,5] with sample period T=0.1**

syms t; b=1; w0=6;

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

t = 0:0.1:5;

sfo=x(t);

n=0:50;

X= fft(double(sfo)) % using double makes it a pure vector instead of a vector-symbolic function mix, speeding plot and stem.

subplot(2,1,1); stem(n,abs(X))

subplot(2,1,2); stem(n,angle(X))

X =

Columns 1 through 6

0.7685 + 0.0000i 0.8023 + 0.3356i 0.9355 + 0.7458i 1.3380 + 1.3450i 2.8517 + 2.0836i 5.3967 - 1.1095i

Columns 7 through 12

2.2383 - 2.6586i 1.1674 - 1.9193i 0.8446 - 1.4367i 0.7123 - 1.1361i 0.6460 - 0.9321i 0.6081 - 0.7832i

Columns 13 through 18

0.5843 - 0.6684i 0.5685 - 0.5759i 0.5574 - 0.4988i 0.5494 - 0.4329i 0.5434 - 0.3752i 0.5388 - 0.3237i

Columns 19 through 24

0.5353 - 0.2770i 0.5325 - 0.2339i 0.5304 - 0.1938i 0.5288 - 0.1559i 0.5276 - 0.1196i 0.5267 - 0.0846i

Columns 25 through 30

0.5262 - 0.0504i 0.5259 - 0.0168i 0.5259 + 0.0168i 0.5262 + 0.0504i 0.5267 + 0.0846i 0.5276 + 0.1196i

Columns 31 through 36

0.5288 + 0.1559i 0.5304 + 0.1938i 0.5325 + 0.2339i 0.5353 + 0.2770i 0.5388 + 0.3237i 0.5434 + 0.3752i

Columns 37 through 42

0.5494 + 0.4329i 0.5574 + 0.4988i 0.5685 + 0.5759i 0.5843 + 0.6684i 0.6081 + 0.7832i 0.6460 + 0.9321i

Columns 43 through 48

0.7123 + 1.1361i 0.8446 + 1.4367i 1.1674 + 1.9193i 2.2383 + 2.6586i 5.3967 + 1.1095i 2.8517 - 2.0836i

Columns 49 through 51

1.3380 - 1.3450i 0.9355 - 0.7458i 0.8023 - 0.3356i

Gráfico, Histograma

Descripción generada automáticamente

1. Compare the CPU time used to run the MATLAB functions: MyDFT and fft (use tic … toc)

**SAMPLED EXPONENTIAL FUNCTION in the interval [0,10] with sample period T=0.3**

syms t; A=3; b=1;

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

t = 0:0.3:10;

sef=x(t);

n=0:33;

**tic**

X = MyDFT(double(sef),n);

**toc**

%Elapsed time is: 3.633981 seconds.

Elapsed time is 1.457347 seconds.

**tic**

X1 = fft(double(sef));

subplot(2,1,1); stem(n,abs(X1))

subplot(2,1,2); stem(n,angle(X1))

**toc**

%Elapsed time is: 0.261304 seconds.

Elapsed time is 0.022291 seconds.

**SAMPLED FORCED OSCILLATION in the interval [0,5] with sample period T=0.1**

syms t; b=1; w0=6;

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

t = 0:0.1:5;

sfo=x(t);

n=0:50;

**>>tic**

X = MyDFT(double(sfo),n);

**toc**

**tic**

X1 = fft(double(sfo));

subplot(2,1,1); stem(n,abs(X1))

subplot(2,1,2); stem(n,angle(X1))

**toc**

Elapsed time is 5.042453 seconds.

Elapsed time is 0.028381 seconds.