**Theory of filtering and time series processing**

L. Zotov

Laboratory work №1

Purpose: Studying discrete Fourier transform with use of Matlab.

Matlab software – powerful tool for matrix and vector calculations.

*advantages* – huge amount of functions and toolboxes, easy matrix and vector computations, it’s possible to write programs and incorporate them into C++ programs

*disadvantages* – calculation speed, programs written in Matlab perform longer than C or Fortran programs. They need optimization. Data format conversion is sometimes tricky.

If you are not familiar with Matlab, the best way is to run it and press F1 for Help. Tens of thousands of pages of information on its use, functions and toolboxes will be at your disposal.

1) First part of the lab – generating the signal and it’s fast Fourier transform.

Download program for the 1 lab at

<http://lnfm1.sai.msu.ru/grav/english/lecture/filtr/>

%---------------------------------------------------

% program is written 14.01.2009 by L.V. Zotov

clear;

N\_signal=1024\*4;

% generating two-sine signal

for (k=1:1:N\_signal)

signal(k)=sin(2\*pi/10\*(k-1))+sin(2\*pi/100\*(k-1));

end;

plot(signal);

%fast Fourier transform calculation

Ftrns\_signal=fft(signal);

%amplitude spectrum calculation

apl\_spectrum=abs(Ftrns\_signal);

plot(apl\_spectrum);

%-----------------------------------------------------

2) Second part of the lab – reading and processing the real signal

Download EOP C01 bulletin with Earth Orientation Parameters from

<http://hpiers.obspm.fr/eop-pc/>

<http://hpiers.obspm.fr/iers/eop/eopc01/eopc01.1900-now.dat>

to inputdata folder

the data should be evenly sampled

cd 'D:/Wuhan18/filtering/lab/';

%from inputdata

fin=fopen('inputdata/eopc01.1900-now.dat','rt');

fgetl(fin)%skip the first line

A=fscanf(fin,'%f',[11 inf]);% A - array of data

fclose(fin);

%determining the size of the signal

l=size(A);

N=l(2);

%selecting the rows of the Array

Date=A(1,1:N);

X\_pole=A(2,1:N);

Y\_pole=A(4,1:N);

UT1TAI=A(6,1:N);

dt=Date(2)-Date(1);

plot3(X\_pole,Y\_pole,Date)

%selecting the length of the part of the signal to be trasformed

N\_ft=N;

%fast Fourier transform calculation

Ftrns\_X=fft(X\_pole,N\_ft);

% frequency calculation

% for the half of the spectrum not counting the first coefficient (the mean)

% N\_ft is odd or even - ?

N\_half=round((N\_ft-1)/2);

freq=(1:N\_half)/N\_ft/dt;

%amplitude spectrum calculation

ampl\_spectrum\_X=abs(Ftrns\_X)/N\_ft;

% we plot spectrum only for positive frequencyes,

%since it is symmetric for real signal

%it is multiplied by two to take into account all the energy

plot(freq, 2\*ampl\_spectrum\_X(2:N\_half+1));

title('amplitude spectrum - module of Fourier-transformation ')

xlabel('frequency, cycles per year')

3) Third part of lab – calculating spectrum of the complex polar motion trajectory

using function spect\_fftn.m written by L. Zotov

the program should see it in the './functions' folder

%------------ now let's calculate the complex spectrum

%functions path

% slow Fourier transform

addpath( './functions')

XY=X\_pole-i\*Y\_pole;

[ spectrXY, freqXY] = spect\_fftn(Date,XY);

plot(freqXY', abs(spectrXY)')

title('amplitude spectrum - module of Fourier-transformation ')

xlabel('frequency, cycles per year')

%Co L.V. Zotov

4) Try to make Fourier analysis of your own data

See example – Lab1\_J2.m with data

http://lnfm1.sai.msu.ru/grav/english/lecture/filtr/inputdata/deltaJ2\_1976-2018.dat