**fft**

*Function of calculation of direct discrete Fourier transform*

**Syntax:**

*Y* **= fft***(X);*

**Arguments:**

*X* – input array containing vector elements.

**Description:**

*fft(X)* – function of calculation of discrete Fourier transform of vector *X* calculated by fast-Fourier-transform algorithm.

,

where *, N – vector size.*

Vector X size shall be power 2.

The input vector can be an array of both real and complex numbers. A complex number shall be assigned by the expression *a+bi*, where *a* and *b* are real and virtual number parts, accordingly.

The input array *X* can be assigned:

* as variables of array type determined earlier:

*Y* = **fft**(*X*);

* as arrays consisting of variables determined earlier:

*Y* = **fft**([*x1,x2,x3,x4*]);

* as constant arrays:

*Y* = **fft**([1, 6, 4, 2])*;*

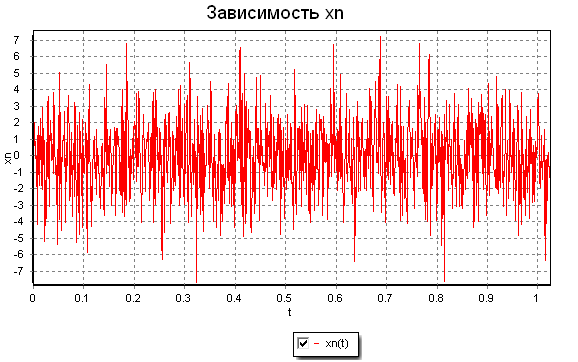
**Result:**

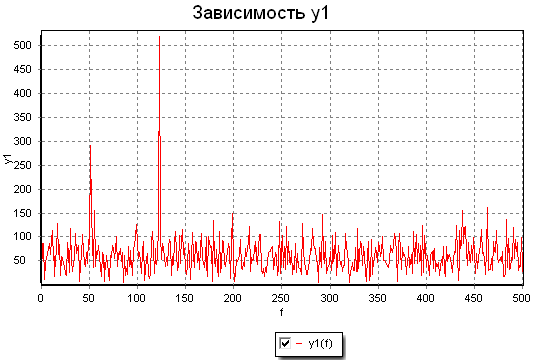
*Y* – vector of values of discrete Fourier transform of vector *X.*

**Example:**

|  |  |
| --- | --- |
|  | **var** Fs = 1000,// Frequency  Tt = 1/Fs,//Time  L = 1024;//Buffer size  **var** t:**array** = 1024#0; //Time value vector  **var** f:**array** = 1024#0; //Frequency value vector  **var** xn:**array** =1024#0;  **for**(i=0, L-1) t[i+1] = i\*Tt; //let us fill the time vector  **for**(i=0, L-1) f[i+1] = i; //let us fill the frequency vector  //Sum of 50 Hz and 120 Hz sinusoids  x = 0.7\***sin**(2\*pi\*50\*t) + **sin**(2\*pi\*120\*t);  //Let us add random noise to the signal  **for** (i=0, L-1) xn[i+1]=x[i+1]+2\***randg**(0,1);  //discrete Fourier transform  y = **fft**(xn);  y1 = **abs**(y); |

xn Relation





y1 Relation

Frequency components of the signal hardly can be defined from the signal appearance. Using the fast Fourier transform frequency components of the signal spectrum (50 Hz and 120 Hz) can be defined.