

PROJECT TITLE:

FREQUENCY ESTIMATION WITH LINEAR ALGEBRA AND APPLICATIONS

SUBMITTED BY -

[TEAM-11]:

SABARINATH J [21141]

SHYAM GANESH [21149]

SIDESH SUNDAR [21150]

SASANK SAMI [21160]

BHARADWAJ V [21165]

**ABSTRACT**

Estimation of the frequency (f) of a noisy sinusoidal wave has been one of the main problems in the field of signal processing and communications, due to its vast applications. For getting accurate and instant frequency estimation from noisy sinusoidal waves, we used an algorithm called non uniform fast fourier transform [FFT]. Fourier Transform (FFT) is any efficient algorithm for calculating the DFT.

**INTRODUCTION**

The problem of estimating the frequencies of sinusoidal components from a finite number of noisy discrete-time measurements has attracted a great deal of attention and still is an active research area to date, because of its wide applications in science and engineering.Many theoretical techniques have been proposed to solve this problem; examples include discrete Fourier transform, least squares methods and phase-locked loops. All of the proposed methods are focused on speed and accuracy of the estimation.

**DEFINITIONS**

**Linear Algebra:** It is a branch of mathematics that is concerned with mathematical structures closed under the operations of addition and scalar multiplication and that includes the theory of systems of linear equations, matrices, determinants, vector spaces, and linear transformations.

**DFT:** Discrete Fourier Transform (DFT) is the discrete version of the Fourier Transform (FT) that transforms a signal (or discrete sequence) from the time domain representation to its representation in the frequency domain.

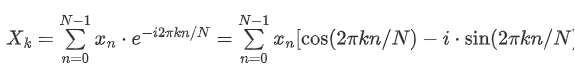
**FFT:** Fast Fourier Transform (FFT) is any efficient algorithm for calculating the DFT.

* Computing a DFT of n n points by using only its definition, takes Θ(n^2) time , whereas an FFT can compute the same result in only Θ(n log n) steps.

**THEORY**

**DFT**

The DFT can transform a sequence of evenly spaced signals to the information about the frequency of all the sine waves needed to sum to obtain the time-domain signal. It is defined as



*N* = number of samples

*n* = current sample

*k* = current frequency, where

k∈[0,N−1]

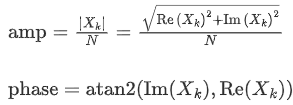
xn = the sine value at sample *n*

Xk = the DFT that includes information of both amplitude and phase

Also, the last expression in the above equation is derived from the *Euler's formula*, which links the trigonometric functions to the complex exponential function: e^(i⋅x)=cos⁡x+i⋅sin⁡x

Note that

Xk, is a complex number that encodes both the amplitude and phase information of a complex sinusoidal component e^(i⋅2πkn/N) of function xn. The amplitude and phase of the signal can be calculated as



where Im(Xk) and Re(Xk) are the imaginary and real parts of the complex number, and atan2 is the two-argument form of the arctan function.

**FFT**

The "Fast Fourier Transform" (FFT) is an important measurement method in the science of audio and acoustics measurement. It converts a signal into individual spectral components and thereby provides frequency information about the signal. FFTs are used for fault analysis, quality control, and condition monitoring of machines or systems. This article explains how an FFT works, the relevant parameters and their effects on the measurement result.

Strictly speaking, the FFT is an optimized algorithm for the implementation of the "Discrete Fourier Transformation" (DFT). A signal is sampled over a period of time and divided into its frequency components.

**METHODOLOGY**

STEP BY STEP:

In the first step, a section of the signal is scanned and stored in the memory for further processing. Two parameters are relevant:

1. The sampling rate or sampling frequency fs of the measuring system (e.g. 48 kHz). This is the average number of samples obtained in one second (samples per second).
2. The selected number of samples; the blocklength BL. This is always an integer power to the base 2 in the FFT (e.g., 2^10 = 1024 samples)

From the two basic parameters fs and BL, further parameters of the measurement can be determined.

**Bandwidth fn** (= Nyquist frequency). This value indicates the theoretical maximum frequency that can be determined by the FFT.

*fn = fs / 2*

For example at a sampling rate of 48 kHz, frequency components up to 24 kHz can be theoretically determined. In the case of an analog system, the practically achievable value is usually somewhat below this, due to analog filters - e.g. at 20 kHz.

**Measurement duration D**. The measurement duration is given by the sampling rate fs and the blocklength BL.

*D = BL / fs.*

At fs = 48 kHz and BL = 1024, this yields 1024/48000 Hz = 21.33 ms

**Frequency resolution df**. The frequency resolution indicates the frequency spacing between two measurement results.

*df = fs / BL*

At fs = 48 kHz and BL = 1024,

This gives a df of 48000 Hz / 1024 = 46.88 Hz.

In practice, the sampling frequency fs is usually a variable given by the system. However, by selecting the blocklength BL, the measurement duration and frequency resolution can be defined. The following applies:

* A small blocklength results in fast measurement repetitions with a coarse frequency resolution.
* A large blocklength results in slower measuring repetitions with fine frequency resolution.

CODE

% Assume we capture 8192 samples at 1kHz sample rate

Nsamps = 8192;

fsamp = 1000;

Tsamp = 1/fsamp;

t = (0:Nsamps-1)\*Tsamp;

% Assume the noisy signal is exactly 123Hz

fsig = 123;

signal = sin(2\*pi\*fsig\*t);

noise = 1\*randn(1,Nsamps);

x = signal + noise;

% Plot time-domain signal

subplot(2,1,1);

plot(t, x);

ylabel('Amplitude'); xlabel('Time (secs)');

axis tight;

title('Noisy Input Signal');

% Choose FFT size and calculate spectrum

Nfft = 1024;

[Pxx,f] = pwelch(x,gausswin(Nfft),Nfft/2,Nfft,fsamp);

% Plot frequency spectrum

subplot(2,1,2);

plot(f,Pxx);

ylabel('PSD'); xlabel('Frequency (Hz)');

grid on;

% Get frequency estimate (spectral peak)

[~,loc] = max(Pxx);

FREQ\_ESTIMATE = f(loc)

title(['Frequency estimate = ',num2str(FREQ\_ESTIMATE),' Hz']);

**FUTURE DEVELOPMENT**

We are looking to develop this estimation tool for different applications (but not only in the case of noisy signals).