AN IMPROVED DISCRETE FOURIER TRANSFORM METHOD OF FREQUENCY MEASUREMENT IN POWER SYSTEMS

Vivek Bargate¹, R. M. Potdar²

ABSTRACT
In this present work we elegantly avoid the errors that arise when frequency deviates from the nominal frequency. The speed of this method is high and very easy to implement, so it is very suitable for use in real-time. With the help of an algorithm, we can obtain exact solution when frequency deviates from nominal frequency. In this paper we also show the simulation results.

KEYWORDS: Discrete Fourier Transforms (DFT), frequency estimation, frequency Relay, power system measurement.

1. INTRODUCTION
Frequency is one of the most important quantities in power system operation because it can reflect the dynamic energy balance between load and generating power. So frequency is always regarded as an index of the operating practices, and utilities can know the system energy balance situations by observing frequency variations. Frequency may vary very fast in the transient events such that it is difficult to track it accurately. In addition, there are many devices, such as power electronic equipment’s and arc furnaces, etc. generating lots of harmonics and noise in modern power systems. It is therefore essential for utilities to seek and develop a reliable method that can measure frequency in presence of harmonics and noise.

With the advent of the microprocessor, more and more microprocessor-based equipment’s have been extensively used in power systems. Using such equipment’s is known to provide accurate, fast responding, economic, and flexible solutions to measurement problems. Therefore, all we have to do is to find the best algorithm and implement it. There have been many digital algorithms applied to estimating frequency during recent years, for example Modified Zero Crossing Technique, Level Crossing Technique, Least Squares Error Technique, Newton method, Kalman Filter , Prony Method, and Discrete Fourier Transform (DFT), etc. For real-time use, most of the aforementioned methods have trade-off between accuracy and speed. A precise digital algorithm, namely Smart Discrete Fourier Transform (SDFT) is presented and tries to meet the real-time use. SDFT has the advantages that it can obtain exact solution when frequency deviates from nominal frequency, its speed is even faster than DFT, and it can get exact solution in the presence of harmonics.

2. PROPOSED DIGITAL ALGORITHM
This section presents the algorithm of the SDFT that calculates the frequency from a voltage/current signal.

Consider a sinusoidal input signal of frequency \( \omega = 2\pi f \) with \( m^{th} \) harmonic given by:

\[
x(t) = X_\cos \left( \omega + \Theta \right) + X_\cos \left( m\omega + \Theta \right)
\]

The signal \( x(t) \) is conventional represented by a phasor (a complex number) \( \mathbf{x} \)

\[
\mathbf{x} = Xe^{j\theta} = Xe^{j\phi} + jXe^{j\theta}
\]

The signal \( x(t) \) can be expressed as

\[
x(t) = \frac{\mathbf{x}_1 e^{j\omega t} + \mathbf{x}_2 e^{-j\omega t}}{2} + \frac{\mathbf{x}_3 e^{j\omega t} + \mathbf{x}_4 e^{-j\omega t}}{2}
\]

Where * denotes complex conjugate. Moreover, the fundamental frequency (60 Hz) component if DFT of \( x(k) \) is given by

\[
\mathbf{x}_r = \frac{2}{N} \sum_{k=0}^{N-1} x(k + r) e^{-j\frac{2\pi knk}{N}}
\]

Combining (4) and (5) and taking frequency deviation \([\omega = 2\pi (60 + \Delta)]\) into consideration, we obtain:

\[
\mathbf{x}_r = \frac{2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) k} e^{-j\frac{2\pi k n k}{N}} + \frac{\mathbf{x}_2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) m(k + r) \frac{0}{N}} e^{-j\frac{2\pi k n k}{N}}
\]

We rearrange (6) as the following

\[
\mathbf{x}_r = \frac{2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) k} e^{-j\frac{2\pi k n k}{N}} + \frac{\mathbf{x}_2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) m(k + r) \frac{0}{N}} e^{-j\frac{2\pi k n k}{N}}
\]

\[
\frac{\mathbf{x}_2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) m(k + r) \frac{0}{N}} e^{-j\frac{2\pi k n k}{N}}
\]

\[
\mathbf{x}_r = \frac{2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) k} e^{-j\frac{2\pi k n k}{N}}
\]

ABSTRACT—
In this present work we elegantly avoid the errors that arise when frequency deviates from the nominal frequency. The speed of this method is high and very easy to implement, so it is very suitable for use in real-time. With the help of an algorithm, we can obtain exact solution when frequency deviates from nominal frequency. In this paper we also show the simulation results.

KEYWORDS: Discrete Fourier Transforms (DFT), frequency estimation, frequency Relay, power system measurement.

1. INTRODUCTION
Frequency is one of the most important quantities in power system operation because it can reflect the dynamic energy balance between load and generating power. So frequency is always regarded as an index of the operating practices, and utilities can know the system energy balance situations by observing frequency variations. Frequency may vary very fast in the transient events such that it is difficult to track it accurately. In addition, there are many devices, such as power electronic equipment’s and arc furnaces, etc. generating lots of harmonics and noise in modern power systems. It is therefore essential for utilities to seek and develop a reliable method that can measure frequency in presence of harmonics and noise.

With the advent of the microprocessor, more and more microprocessor-based equipment’s have been extensively used in power systems. Using such equipment’s is known to provide accurate, fast responding, economic, and flexible solutions to measurement problems. Therefore, all we have to do is to find the best algorithm and implement it. There have been many digital algorithms applied to estimating frequency during recent years, for example Modified Zero Crossing Technique, Level Crossing Technique, Least Squares Error Technique, Newton method, Kalman Filter , Prony Method, and Discrete Fourier Transform (DFT), etc. For real-time use, most of the aforementioned methods have trade-off between accuracy and speed. A precise digital algorithm, namely Smart Discrete Fourier Transform (SDFT) is presented and tries to meet the real-time use. SDFT has the advantages that it can obtain exact solution when frequency deviates from nominal frequency, its speed is even faster than DFT, and it can get exact solution in the presence of harmonics.

2. PROPOSED DIGITAL ALGORITHM
This section presents the algorithm of the SDFT that calculates the frequency from a voltage/current signal.

Consider a sinusoidal input signal of frequency \( \omega = 2\pi f \) with \( m^{th} \) harmonic given by:

\[
x(t) = X_\cos \left( \omega + \Theta \right) + X_\cos \left( m\omega + \Theta \right)
\]

The signal \( x(t) \) is conventional represented by a phasor (a complex number) \( \mathbf{x} \)

\[
\mathbf{x} = Xe^{j\theta} = Xe^{j\phi} + jXe^{j\theta}
\]

The signal \( x(t) \) can be expressed as

\[
x(t) = \frac{\mathbf{x}_1 e^{j\omega t} + \mathbf{x}_2 e^{-j\omega t}}{2} + \frac{\mathbf{x}_3 e^{j\omega t} + \mathbf{x}_4 e^{-j\omega t}}{2}
\]

Where * denotes complex conjugate. Moreover, the fundamental frequency (60 Hz) component if DFT of \( x(k) \) is given by

\[
\mathbf{x}_r = \frac{2}{N} \sum_{k=0}^{N-1} x(k + r) e^{-j\frac{2\pi knk}{N}}
\]

Combining (4) and (5) and taking frequency deviation \([\omega = 2\pi (60 + \Delta)]\) into consideration, we obtain:

\[
\mathbf{x}_r = \frac{2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) k} e^{-j\frac{2\pi k n k}{N}} + \frac{\mathbf{x}_2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) m(k + r) \frac{0}{N}} e^{-j\frac{2\pi k n k}{N}}
\]

We rearrange (6) as the following

\[
\mathbf{x}_r = \frac{2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) k} e^{-j\frac{2\pi k n k}{N}} + \frac{\mathbf{x}_2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) m(k + r) \frac{0}{N}} e^{-j\frac{2\pi k n k}{N}}
\]

\[
\frac{\mathbf{x}_2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) m(k + r) \frac{0}{N}} e^{-j\frac{2\pi k n k}{N}}
\]

\[
\mathbf{x}_r = \frac{2}{N} \sum_{k=0}^{N-1} e^{j2\pi n (6 + \Delta) k} e^{-j\frac{2\pi k n k}{N}}
\]
We use the following identify to simplify (7)
\[ \sum_{i=0}^{N-1} (e^{i\theta})^i = \frac{\sin N\theta_0}{\sin \frac{\theta_0}{2}} e^{i((N-1)\theta_0/2)} \]  

(8)

Then (7) can be expressed as
\[ \bar{x}_r = \frac{\bar{x}_1}{N} e^{i\left(2\pi N \frac{\Delta f (2N-1)}{2}ight)} + \frac{\bar{x}_1}{N} e^{i\left(2\pi N \frac{\Delta f (2N-1) + 120\theta_0}{2}ight)} \]
\[ + \frac{\bar{x}_2}{N} e^{i\left(2\pi N \frac{\Delta f (2N-1) + 120\theta_0}{2}ight)} + \frac{\bar{x}_2}{N} e^{i\left(2\pi N \frac{\Delta f (2N-1) + 120\theta_0}{2}ight)} \]
\[ + \frac{\bar{x}_2}{N} e^{i\left(2\pi N \frac{\Delta f (2N-1) + 120\theta_0}{2}ight)} + \frac{\bar{x}_2}{N} e^{i\left(2\pi N \frac{\Delta f (2N-1) + 120\theta_0}{2}ight)} \]
\[ + \frac{\bar{x}_2}{N} e^{i\left(2\pi N \frac{\Delta f (2N-1) + 120\theta_0}{2}ight)} + \frac{\bar{x}_2}{N} e^{i\left(2\pi N \frac{\Delta f (2N-1) + 120\theta_0}{2}ight)} \]

(9)

Where \( \theta_1 = \frac{2\Delta f}{60N} \),
\[ \theta_2 = -\frac{2\pi (2 + \frac{\Delta f}{60})}{N}, \]
\[ \theta_3 = \frac{2\pi (m - 1 + \frac{\Delta f}{60})}{N}, \]
\[ \theta_4 = \frac{2\pi (-m - 1 + \frac{\Delta f}{60})}{N}. \]

Rearranging (9) further, we obtain
\[ \bar{x}_r = \frac{\bar{x}_1}{N} \sin \frac{\theta_1}{2} e^{i\left(\frac{\pi}{60} \Delta f (2N-1) + 120\theta_0\right)} + \frac{\bar{x}_1}{N} \sin \frac{\theta_2}{2} e^{i\left(-\frac{\pi}{60} \Delta f (2N-1) + 120\theta_0\right)} \]
\[ + \frac{\bar{x}_2}{N} \sin \frac{\theta_3}{2} e^{i\left(\frac{2\pi}{60} \Delta f (2N-1) + 120\theta_0\right)} + \frac{\bar{x}_2}{N} \sin \frac{\theta_4}{2} e^{i\left(-\frac{2\pi}{60} \Delta f (2N-1) + 120\theta_0\right)} \]

(10)

If we define \( A_r, B_r, C_r, \) and \( D_r \), as
\[ A_r = \frac{\bar{x}_1}{N} \sin \frac{\theta_1}{2} e^{i\left(\frac{\pi}{60} \Delta f (2N-1) + 120\theta_0\right)} \]
\[ B_r = \frac{\bar{x}_1}{N} \sin \frac{\theta_2}{2} e^{i\left(-\frac{\pi}{60} \Delta f (2N-1) + 120\theta_0\right)} \]
\[ C_r = \frac{\bar{x}_2}{N} \sin \frac{\theta_3}{2} e^{i\left(\frac{2\pi}{60} \Delta f (2N-1) + 120\theta_0\right)} \]
\[ D_r = \frac{\bar{x}_2}{N} \sin \frac{\theta_4}{2} e^{i\left(-\frac{2\pi}{60} \Delta f (2N-1) + 120\theta_0\right)} \]

(11)

(12)

(13)

(14)

Then (9) can be expressed as
\[ \bar{x}_r = A_r + B_r + C_r + D_r \]

(15)

Expect the parts of \( m^{th} \) harmonic, so far the development of the algorithm of SDFT are the same as the conventional DFT method. So the SDFT can keep all advantages of DFT such as recursive computing manner. But in the DFT, it assumes that the frequency deviation is small enough to be ignored, and it always considers \( \Delta f \approx \Delta f \), so traditional DFT based methods incur error in estimating frequency and phasor when frequency deviates from nominal from nominal frequency \( (60 \text{ Hz}) \). If we want to get exact solution, we must take \( B_r, C_r, \) and \( D_r \), into consideration. So we define
\[ a = e^{i\left(\frac{\pi}{60} \Delta f + 120\theta_0\right)} \]

(16)

And from (10), we will find the following relations
\[ A_{r+1} = A_r a \]
\[ B_{r+1} = B_r a^{-1} \]
\[ C_{r+1} = C_r a^m \]
\[ D_{r+1} = D_r a^{-m} \]
Then
\[ \hat{x}_{r+1} = A_r a + B_r a^{-1} + C_r a^m + D_r a^{-m} \]
\[ \hat{x}_{r+2} = A_{r+1} a + B_{r+1} a^{-1} + C_{r+1} a^m + D_{r+1} a^{-m} \]  

If we multiply "a^m" on both sides of (21) and (22), respectively, then we get
\[ a^m \hat{x}_{r+1} = A_r a^{1+m} + B_r a^{-1+m} + C_r a^{2m} + D_r \]  
\[ a^m \hat{x}_{r+2} = A_{r+1} a^{1+m} + B_{r+1} a^{-1+m} + C_{r+1} a^{2m} + D_{r+1} a^{-1+m} \]  

Subtracting (15) and (23) and subtracting (21) from (24), respectively, we can erase D_r and obtain
\[ \hat{y}_r = a^m \hat{x}_{r+1} - \hat{x}_r \]
\[ \hat{y}_{r+1} = a^m \hat{x}_{r+2} - \hat{x}_{r+1} \]

Repeat similar operation to erase the C_r and B_r, then the equation will become
\[ a^2 \hat{z}_{r+1} - \hat{z}_r = A_r (a^2 - 1)(a^{1-m} - 1) \]
\[ a^2 \hat{z}_{r+2} - \hat{z}_{r+1} = A_{r+1} (a^2 - 1)(a^{1-m} - 1) \]

Where \( \hat{z} = a \hat{u}_{r+1} - \hat{u}_r \)

Dividing (28) by (27), we get
\[ \frac{A_{r+1}}{A_r} = a \]  

Then expand (29), and use numerical method to find the solution of "a". And from the definition of "a" in (16), we can get the exact solution of the frequency
\[ f = 60 + \Delta f = \cos^{-1}\left( \frac{\text{Re}(a)}{2\pi} \right) \]

From (29), it is observed that SDFT can exact frequency using \( \hat{x}_r, \hat{x}_{r+1}, \hat{x}_{r+2}, \hat{x}_{r+3} \) and \( \hat{x}_{r+4} \) in the presence of harmonics. Moreover, we can estimate phasor after getting exact "f" by the following equations:
\[ A_r = \frac{2 \hat{z}_{r+1} - \hat{z}_r}{(a^2 - 1)(a^{1-m} - 1)} \]
\[ X_i = \text{abs}(A_r) \]
\[ \angle \theta_i = \angle(A_r) - \frac{\pi}{60}(N - 1) \]

It appears that SDFT can take integral order harmonic into consideration. To distinguish easily, SDFT means calculating frequency for m = 1 and we add suffix to the others, for example SDFT_3 and SDFT_35 calculate frequency for m = 3 and m = 3.5, respectively. And here we offer the polynomial equation of SDFT_3 (m = 3)

\[ a^4 + p_3 a^3 + p_2 a^2 + p_1 a + p_0 = 0 \]

Where
\[ p_3 = -\hat{x}_{r+1} + \hat{x}_{r+3} \]
\[ p_2 = -\frac{3}{4} \hat{x}_{r+2} \]
\[ p_1 = -\frac{4 \hat{x}_{r+3}}{16 \hat{x}_{r+2}} \]
\[ p_0 = -\frac{\hat{x}_r + 2 \hat{x}_{r+2} + \hat{x}_{r+4}}{16 \hat{x}_{r+2}} \]

Actually, if we assume that \( x(t) = X_1 \cos(\omega_1 t + \phi_1) + X_2 \cos(\omega_2 t + \phi_2) \) from the beginning of development of the algorithm, we will derive a polynomial equation similar to (29) that provide exact frequency in the presence of nonintegral harmonic. We add suffix "n" to SDFT means that has taken nonintegral harmonic into consideration. Although we can take the entire harmonic into consideration, we still need a digital filter to decay noise and high order harmonics. Since, in SDFT the harmonics taken into consideration, the more CPU time needed in computing. The advantage of digital filtering are no voltage drop, no temperature drift, no noise addition, and don’t have any analog filter element features, like aging. Besides these, digital filter can be implemented in microprocessor-equipment. These make us chose a digital filter to filter noise and high order harmonics. There are many digital filters that we can choose e.g., Hanning, Haming and Blackman windows. In our simulation we will use the Blackman window for filtering.

3. SIMULATION OUTPUT
MATLAB OUTPUT: -
Find out the frequency of selected signal and to compute error in power system.

**Input Data:**
- Enter the value of power frequency = 50 Hz
- Enter the value of Vmax = 200
- Enter the value of sampling frequency = 500 Hz
- Enter the value of frequency deviation = 1

**Output Data:**
- Calculated Power frequency = 51
- Percentage error = 0
- Sampling Frequency = 500 Hz
- Elapsed time is 17.613908 seconds.

**4. CONCLUSION**
In this paper we calculate the frequency deviates from the nominal frequency. The above technique deals with the basis of frequency deviation errors, while attractive harmonics into consideration. This method is used to designing digital meters and relays that need to measure system parameter accurately over a large frequency range.

**REFERENCES**

*Note: This Paper/Article is scrutinised and reviewed by Scientific Committee, BITCON-2015, BIT, Durg, CG, India*