A New Family of Measurement Technique for Tracking Voltage Phasor, Local System Frequency, Harmonics and DC offset

Jun-Zhe Yang Chih-Wen Liu

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

Abstract - A series of precise digital algorithms based on Discrete Fourier Transforms (DFT) to calculate the frequency and phasor in real-time are proposed. These algorithms, we called the Smart Discrete Fourier Transforms (SDFT) family, succeeded in overcoming several problems of DFT, which include frequency deviation, harmonics and DC offset. Moreover, if using smoothing windows to filter noise, SDFT family is not affected by phase shift and amplitude decay. Factional cycle computing is allowed. Also, the advantages of DFT are still be reserved in SDFT family. These make the SDFT family more accurate than conventional DFT. Besides, SDFT family is very easy to implement, so it is very suitable for use in power systems. We provide the general form of SDFT family and simulation results compared with conventional DFT method to verify the claimed benefits of SDFT family.

Keywords: Discrete Fourier Transform (DFT), Frequency estimation, phasor measurement

I. Introduction

Frequency and phasor are the most important quantities in power system operation because they can reflect the whole power system situation. Frequency reflects the dynamic energy balance between load and generating power, while operators use phasor to constitute the state of system and, moreover, phasor based line relays are currently used in most power systems. So frequency and phasor are regarded as indices for the operating power systems in practice.

However, utilities have difficulty in calculating those quantities precisely. There are many devices, such as power electronic equipment and arc furnaces, etc. generating lots of integral, non-integral harmonics, flicker and noise in modern power systems. Moreover, when a fault occurs, DC offset component is presented in fault current waveforms. Therefore, it is essential for utilities to seek and develop a flexible and reliable method that can measure frequency and phasor in presence of harmonics, DC offset and noise.

There have been many digital algorithms applied to estimating frequency or phasor during recent years, for example Modified Zero Crossing Technique [1], Level Crossing Technique [2], Least Squares Error Technique [3], Newton method [4], Kalman Filter [5], Prony Method [6], and Discrete Fourier Transform (DFT) [7], etc. Among the aforementioned methods, DFT is the most used method in phasor and frequency estimation [8]. When fundamental frequency is 60Hz, DFT has following advantages:

1. Highly precise: under the assumption of system frequency is very close 60Hz in normal, leakage error of DFT is very small.
2. Fast computation: recursive and simple computation process make DFT faster than other methods.
3. Strong immunity: DFT immunes to harmonics that are multiple of 60Hz.

A series of precise digital algorithms that we called SDFT family offer utilities a flexible solution for estimation. SDFT family can take integral, non-integral harmonics and DC offset into consideration by very simple rules. Besides flexibility, SDFT family keeps advantages of DFT and adds these new advantages:

1. No Leakage error: the range of frequency, which SDFT family can estimate, is 0 to half of sampling frequency.
2. Arbitrary window size: there are only two types of DFT using sampled data: full-cycle DFT and half-cycle DFT (HDFT). But in SDFT family, except zero, how many sampled data use to compute is arbitrarily.
3. Complete immunity: both integral and non-integral harmonics can be estimated while system frequency is 60Hz or not.

Therefore, SDFT family is a flexible and reliable solution for utilities to fit need in practice.

This paper is organized into four sections, the first of which is introduction. In section II, we present the rules to produce every member of SDFT family. Furthermore, we prove that SDFT family can avoid the phase shift and amplitude decay caused by smoothing windows. Fifteen members of SDFT family are tested by fifteen examples in section III. Finally, we give a conclusion in section IV.

II. The Proposed Digital Algorithm

In this section, we presented how we found the rules of SDFT family to take harmonics and DC offset into consideration at same time. First, we only consider the fundamental component of waveform.

$$x(t) = X_1 \cos(\omega t + \phi_1) \qquad (1)$$

where X_1 : the amplitude,

ϕ_1 : the phase angle.

The signal $x(t)$ is conventionally represented by a phasor (a complex number) $\bar{x}_1$

$$\bar{x}_1 = X_1 e^{j\phi_1} = X_1 \cos\phi_1 + jX_1 \sin\phi_1 \qquad (2)$$

Then $x(t)$ can be expressed as

$$x(t) = \frac{\bar{x}_1 e^{jwt} + \bar{x}_1^* e^{-jwt}}{2} \tag{3}$$

where * denotes complex conjugate.

We take frequency deviation ($\omega = 2\pi(60+\Delta f)$) into consideration, and suppose that $x(t)$ is sampled with a sampling rate (60*N) Hz to produce the sample set $\{x(k)\}$. Then, we define A_1, B_1, and a as

$$A_1 = \frac{\bar{x}_1}{2} \tag{4}$$

$$B_1 = \frac{\bar{x}_1^*}{2} \tag{5}$$

$$a = e^{j(\frac{2\pi}{60N}(60+\Delta f))} \tag{6}$$

Then, the sample set $\{x(k)\}$ can be expressed as

$$x(r) = A_1 + B_1 \tag{7}$$

$$x(r+1) = A_1 a + B_1 a^{-1} \tag{8}$$

$$x(r+2) = A_1 a^2 + B_1 a^{-2} \tag{9}$$

We multiply a on both sides of (8) and (9), respectively, then, subtract (7) from (8) and subtract (8) from (9). We can erase B_1 and obtain

$$x(r) - ax(r+1) = A_1(a^2-1) \tag{10}$$

$$x(r+1) - ax(r+2) = A_1 a(a^2-1) \tag{11}$$

Dividing (11) by (10), we get

$$\frac{x(r+1) - ax(r+2)}{x(r) - ax(r+1)} = a \tag{12}$$

There is only one unknown variable in (12), and after some algebraic manipulations we obtain:

$$a^2 x(r) - a(x(r) + x(r+2)) + x(r+1) = 0 \tag{13}$$

And we define z_1 as

$$z_1 = \text{Real}(a) = \cos(\frac{2\pi}{60N}(\Delta f + 60)) \tag{14}$$

Then (13) can be expressed as:

$$\begin{array}{lcl} 2\times & x(r+1) & \times\ z_1 \\ -1\times & (x(r)+x(r+2)) & =\quad 0 \end{array} \tag{15}$$

Now, we add one integral harmonic into the signal.

$$x(t) = X_1\cos(\omega t+\phi_1) + X_m\cos(m\omega t+\phi_m) \tag{16}$$

where X_1, X_m : the amplitude,
ϕ_1, ϕ_m : the phase angle.

As the same steps as above, we define z_m

$$z_m = \text{Real}(a^m) = \cos(\frac{2\pi m}{60N}(\Delta f + 60)) \tag{17}$$

And the result is

$$\begin{array}{lcl} -4\times & x(r+2) & \times\ z_1 z_m \\ +2\times & (x(r+1)+x(r+3)) & \times\ (z_1+z_m) \\ -1\times & (x(r)+2x(r+2)+x(r+4)) & =\quad 0 \end{array} \tag{18}$$

Once again, we add non-integral harmonic into the signal.

$$x(t) = X_1\cos(\omega t+\phi_1) + X_m\cos(m\omega t+\phi_m) + X_n\cos(\omega_n t+\phi_n) \tag{19}$$

where X_1, X_m, X_n : the amplitude,
ϕ_1, ϕ_m, ϕ_n : the phase angle.

Also, we define a_n and z_n as

$$a_n = e^{j(\frac{2\pi}{60N}(60+\Delta f_n))} \tag{20}$$

$$z_n = \text{Real}(a_n) = \cos(\frac{2\pi}{60N}(60+\Delta f_n)) \tag{21}$$

And the result is

$$\begin{array}{lcl} 8\times & x(r+3) & \times\ z_1 z_m z_n \\ -4\times & (x(r+2)+x(r+4)) & \times\ (z_1 z_m + z_1 z_n + z_m z_n) \\ +2\times & (x(r+1)+2x(r+3)+x(r+5)) & \times\ (z_1+z_m+z_n) \\ -1\times & (x(r)+3x(r+2)+3x(r+4)+x(r+6)) & =\quad 0 \end{array} \tag{22}$$

Finally, we add DC offset into the signal.

$$x(t) = X_1\cos(\omega t+\phi_1) + X_m\cos(m\omega t+\phi_m) + X_n\cos(\omega_n t+\phi_n) + X_d e^{-\alpha t} \tag{23}$$

where X_1, X_m, X_n : the amplitude,
ϕ_1, ϕ_m, ϕ_n : the phase angle.
$\frac{1}{\alpha} = \tau$: the time constant of the signal
$X_d e^{-\alpha t}$: DC offset

Also, we define a_d and z_d as

$$a_d = e^{\frac{-\alpha}{60N}} \tag{24}$$

$$z_d = \cosh(a_d) = \frac{e^{\frac{-\alpha}{60N}} + e^{\frac{\alpha}{60N}}}{2} \tag{25}$$

And the result is

$$\begin{array}{llll}
-16\ \times & x(r+4) & \times\ z_1 z_m z_n z_d & \leftarrow k=4 \\
+8\ \times & (x(r+3)+x(r+5)) & \times\ (z_1 z_m z_n + z_1 z_m z_d + z_1 z_d z_n + z_d z_m z_n) & \leftarrow k=3 \\
-4\ \times & (x(r+2)+2x(r+4)+x(r+6)) & \times\ (z_1 z_m + z_1 z_n + z_1 z_d + z_m z_n + z_m z_d + z_n z_d) & \leftarrow k=2 \\
+2\ \times & (x(r+1)+3x(r+3)+3x(r+5)+x(r+7)) & \times\ (z_1 + z_m + z_n + z_d) & \leftarrow k=1 \\
-1\ \times & (x(r)+4x(r+2)+6x(r+4)+4x(r+6)+x(r+8)) & = \quad 0 & \leftarrow k=0 \\
\uparrow & \uparrow & \uparrow & \\
\text{part1} & \text{part2} & \text{part3} &
\end{array} \tag{26}$$

Now, it is very easy to find the rules from (15), (18), (22) and (26). There are three parts of these equations. The first part of rules is $(-1)^{k+1}2^k$, k=0... i . Where i is dependent on the number of components considered in signal. The second part is similar to Pascal's triangle, and third part is a type of C_k^i. Using these rules, we can easily express any type of waveform containing harmonics and DC offset. Then, we combine these rules with DFT. As (23), we consider integral harmonic, non-integral harmonic and DC offset at the same time.

$$x(t) = X_1 \cos(\omega t + \phi_1) + X_m \cos(m\omega t + \phi_m) + X_n \cos(\omega_n t + \phi_n) + X_d e^{-\alpha t} \tag{27}$$

Suppose that $x(t)$ is sampled with a sampling rate (60*N) Hz to produce the sample set $\{x(k)\}$. Moreover, the fundamental frequency (60Hz) component of DFT of $\{x(k)\}$ is given by

$$\hat{x}_r = \frac{2}{M}\sum_{k=0}^{M-1} x(k+r) e^{-j\frac{2\pi k}{N}} \tag{28}$$

M is the window size used in DFT. Taking frequency deviation ($\omega = 2\pi(60+\Delta f)$) into consideration, at last, we obtain:

$$\begin{aligned}
\hat{x}_r = {} & \frac{\bar{x}_1}{M}\frac{\sin M\theta_{1a}}{\sin\theta_1} e^{j(M-1)\theta_1} e^{j\frac{2\pi}{60N}(60+\Delta f)r} \\
& + \frac{\bar{x}_1^*}{M}\frac{\sin M\theta_{1b}}{\sin\theta_2} e^{j(M-1)\theta_2} e^{-j\frac{2\pi}{60N}(60+\Delta f)r} \\
& + \frac{\bar{x}_m}{M}\frac{\sin M\theta_{ma}}{\sin\theta_{ma}} e^{j(M-1)\theta_{ma}} e^{j\frac{2m\pi}{60N}(60+\Delta f)r} \\
& + \frac{\bar{x}_m^*}{M}\frac{\sin M\theta_{mb}}{\sin\theta_{mb}} e^{j(M-1)\theta_{mb}} e^{-j\frac{2m\pi}{60N}(60+\Delta f)r} \\
& + \frac{\bar{x}_n}{M}\frac{\sin N\theta_{na}}{\sin\theta_{na}} e^{j(M-1)\theta_{na}} e^{j\frac{2\pi}{N}(1+\frac{\Delta f_n}{60})r} \\
& + \frac{\bar{x}_n^*}{M}\frac{\sin N\theta_{nb}}{\sin\theta_{nb}} e^{j(M-1)\theta_{nb}} e^{-j\frac{2\pi}{N}(1+\frac{\Delta f_n}{60})r} \\
& + \frac{2X_d}{M}\frac{e^{-\frac{\alpha M}{60N}}-1}{e^{-\frac{\alpha}{60N}-j\frac{2\pi}{M}}-1} e^{-\frac{\alpha r}{60N}}
\end{aligned} \tag{29}$$

where $\theta_{1a} = \frac{\Delta f\pi}{60N}$, and $\theta_{1b} = -\frac{\pi}{N}(2+\frac{\Delta f}{60})$,

$.\theta_{ma} = \frac{\pi}{N}(m-1+\frac{m\Delta f}{60})$, $\theta_{mb} = -\frac{\pi}{N}(m+1+\frac{m\Delta f}{60})$

$\theta_{na} = \frac{\pi\Delta f_n}{60N}$, $\theta_{nb} = -\frac{\pi}{N}(2+\frac{\Delta f_n}{60})$

Actually, the development from (27) to (29) is the same as the conventional DFT method. So the SDFT family can keep all advantages of DFT such as recursive computing manner, but then we use the steps of SDFT family to investigate (29). We define A_r, B_r, C_r, D_r, E_r, F_r, and G_r as

$$A_r = \frac{\bar{x}_1}{M}\frac{\sin M\theta_1}{\sin\theta_1} e^{j(M-1)\theta_1} e^{j\frac{2\pi}{60N}(60+\Delta f)r} \tag{30}$$

$$B_r = \frac{\bar{x}_1^*}{M}\frac{\sin M\theta_2}{\sin\theta_2} e^{j(M-1)\theta_2} e^{-j\frac{2\pi}{60N}(60+\Delta f)r} \tag{31}$$

$$C_r = \frac{\bar{x}_m}{M}\frac{\sin M\theta_{ma}}{\sin\theta_{ma}} e^{j(M-1)\theta_{ma}} e^{j\frac{2m\pi}{60N}(60+\Delta f)r} \tag{32}$$

$$D_r = \frac{\bar{x}_m^*}{M}\frac{\sin M\theta_{mb}}{\sin\theta_{mb}} e^{j(M-1)\theta_{mb}} e^{-j\frac{2m\pi}{60N}(60+\Delta f)r} \tag{33}$$

$$E_r = \frac{\bar{x}_n}{M}\frac{\sin M\theta_{na}}{\sin\theta_{na}} e^{j(M-1)\theta_{na}} e^{j\frac{2\pi}{60N}(60+\Delta f_n)r} \tag{34}$$

$$\dot{F}_r = \frac{\bar{x}_n^*}{M}\frac{\sin M\theta_{nb}}{\sin\theta_{nb}} e^{j(M-1)\theta_{nb}} e^{-j\frac{2\pi}{60N}(60+\Delta f_n)r} \tag{35}$$

$$G_r = \frac{2X_d}{M}\frac{e^{-\frac{\alpha M}{60N}}-1}{e^{-\frac{\alpha}{60N}-j\frac{2\pi}{M}}-1} e^{-\frac{\alpha r}{60N}} \tag{36}$$

Then (29) can be expressed as

$$\hat{x}_r = A_r + B_r + C_r + D_r + E_r + F_r + G_r \tag{37}$$

And from (29), we can find the following relations

$$\hat{x}_{r+1} = A_r a + B_r a^{-1} + C_r a^m + D_r a^{-m} + E_r a_n + F_r a_n^{-1} + G_r a_d \tag{38}$$

And the result is

$$\begin{array}{lll} -16\times & \hat{x}_{r+4} & \times z_1 z_m z_n z_d \\ +8\times & (\hat{x}_{r+3}+\hat{x}_{r+5}) & \times(z_1 z_m z_n + z_1 z_m z_d + z_1 z_n z_d + z_m z_n z_d) \\ -4\times & (\hat{x}_{r+2}+2\hat{x}_{r+4}+\hat{x}_{r+6}) & \times(z_1 z_m + z_1 z_n + z_1 z_d + z_m z_n + z_m z_d + z_n z_d) \\ +2\times & (\hat{x}_{r+1}+3\hat{x}_{r+3}+3\hat{x}_{r+5}+\hat{x}_{r+7}) & \times(z_1+z_m+z_n+z_d) \\ -1\times & (\hat{x}_r+4\hat{x}_{r+2}+6\hat{x}_{r+4}+4\hat{x}_{r+6}+\hat{x}_{r+8}) & =0 \end{array} \quad (39)$$

From (39), we can get the solutions of frequency and time constant of DC offset.

$$f = 60+\Delta f = \cos^{-1}(\mathrm{real}(z_1))\frac{60N}{2\pi} \quad (40)$$

$$f_n = 60+\Delta f_n = \cos^{-1}(\mathrm{real}(z_n))\frac{60N}{2\pi} \quad (41)$$

$$\tau = \frac{1}{60N\log a_d} \quad (42)$$

Moreover, by some algebraic operation we can get the value of A_r, B_r, C_r, D_r, E_r, F_r, and G_r after getting exact frequency and time constant of DC offset. Then phasors can be obtained by the following equations:

$$X_1 = \mathrm{abs}(A_r)\frac{M\sin(\theta_{1a})}{\sin(M\theta_{1a})} \quad (43)$$

$$\phi_1 = \mathrm{angle}[A_r e^{-j\theta_{1a}(M-1)}] \quad (44)$$

$$X_m = \mathrm{abs}(C_r)\frac{M\sin(\theta_{ma})}{\sin(M\theta_{ma})} \quad (45)$$

$$\phi_m = \mathrm{angle}[C_r e^{-j\theta_{ma}(M-1)}] \quad (46)$$

$$X_n = \mathrm{abs}(E_r)\frac{M\sin(\theta_{na})}{\sin(M\theta_{na})} \quad (47)$$

$$\phi_n = \mathrm{angle}[E_r e^{-j\theta_{na}(M-1)}] \quad (48)$$

$$X_d = \frac{C_r M}{2}\frac{e^{-\frac{\alpha}{60N}-j\frac{2\pi}{M}}-1}{e^{-\frac{\alpha M}{60N}}-1} \quad (49)$$

Conventional DFT methods incur leakage error in estimating frequency and phasor when frequency deviates from nominal frequency (60Hz). However, in SDFT family we get exact solutions of frequency and phasor.

For off-line analysis, we can take all of the harmonics into consideration, but for on-line applications, we need smoothing windows to decay noise and high order harmonics. Since the more harmonics taken into consideration will take more time in computing. The advantages of smoothing windows are no voltage drop, no temperature drift, no noise addition, and don't care any filter element feature. Besides these, smoothing windows can be easily implemented in microprocessor-based equipment. These make us choose the smoothing windows to filter noise and high order harmonics for on-line applications. There are many smoothing windows that we can choose e.g., Hanning, Hamming, and Blackman window.

Now, we take smoothing windows into consideration. Consider a sampled set $\{x(k)\}$ becomes a filtered set $\{z(k)\}$ by a smoothing window $\{SW(n) \mid s_1, s_2, \cdots s_n\}$, n is window size.

$$z(k) = s_1 x(k) + s_2 x(k+1) + \cdots + s_n x(k+n-1) \quad (50)$$

Moreover, the DFT of $\{z(k)\}$ is given by

$$\begin{aligned} \hat{z}_r &= \frac{2}{M}\sum_{k=0}^{M-1} z(k+r)e^{-j\frac{2\pi k}{M}} \\ &= \frac{2}{M}\sum_{k=0}^{M-1}\left[\sum_{i=1}^{n} s_i x(k+r+i-1)\right]e^{-j\frac{2\pi k}{M}} \\ &= \sum_{i=1}^{n} s_i\left[\frac{2}{M}\sum_{k=0}^{M-1} x(k+r+i-1)e^{-j\frac{2\pi k}{M}}\right] \\ &= \sum_{i=1}^{n} s_i \hat{x}_{r+i-1} \end{aligned} \quad (51)$$

From the (37), we can obtain:

$$\begin{aligned} \hat{z}_r = & A_r(s_1+s_2 a+\cdots+s_n a^{n-1}) \\ & +B_r(s_1+s_2 a^{-1}+\cdots+s_n a^{-n+1}) \\ & +C_r(s_1+s_2 a^{m}+\cdots+s_n a^{(n-1)m}) \\ & +D_r(s_1+s_2 a^{-m}+\cdots+s_n a^{(-n+1)m}) \\ & +E_r(s_1+s_2 a_n+\cdots+s_n a_n^{n-1}) \\ & +F_r(s_1+s_2 a_n^{-1}+\cdots+s_n a_n^{-n+1}) \\ & +G_r(s_1+s_2 a_d+\cdots+s_n a_d^{-n+1}) \end{aligned} \quad (52)$$

The relations of (37) and (38) are still kept in (52). Therefore, we can estimate frequency without modifying equations, but we have to do some change in (43-49) when we estimate phasor. For example:

$$X_1 = \mathrm{abs}\left(\frac{A_r}{(s_1+s_2 a+\cdots+s_n a^{n-1})}\right)\frac{M\sin(\theta_{1a})}{\sin(M\theta_{1a})} \quad (53)$$

$$\phi_1 = \mathrm{angle}\left[\frac{A_r}{(s_1+s_2 a+\cdots+s_n a^{n-1})}e^{-j\theta_{1a}(M-1)}\right] \quad (54)$$

The phasor getting from (53) and (54) will allay the phase shift and amplitude decay caused by smoothing windows.

III. Simulation Results

Simulation results presented in this section are all simulated from Matlab. Sampling frequency is 1920Hz (N=32).

Table 1. Frequency Error of SDFT family

	Case1	Case2	Case 3	Case 4	Case 5	Case 1	Case 2	Case 3	Case 4	Case 5	Case 1	Case 2	Case 3	Case 4	Case 5	Time
16 HDFT	*	*	±0.5710	±14.372	±9.124e-2	±0.2472	±0.2283	±0.6276	±13.928	±0.5333	±0.2423	±0.2396	±0.6335	±14.368	±0.5063	5.07
32 DFT	*	*	±0.9523	±3.7547	±8.895e-2	±0.4937	±0.4565	±1.0846	±3.6701	±0.2864	±0.4854	±0.4459	±1.2218	±3.7549	±0.2868	4.76
SDFT	*	*	±1.3114	±13.311	±0.1636	*	±0.2948	±1.3453	±13.357	±0.1387	±6.678e-4	±0.2844	±1.3666	±13.618	±0.1513	8.14
$SDFT_3$	*	*	±1.0321	±15.132	±9.908e-2	*	±0.1423	±1.0934	±15.686	±9.904e-2	±6.713e-4	±0.1379	±1.1079	±16.019	±9.764e-2	12.56
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$SDFT_1$	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
16 SDFT	*	*	*	*	[illegible]	*	*	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	9.15
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	*	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	±1.0949	±1.8768	±7.899e-2	*	±0.2937	±1.1117	±1.8777	±7.689e-2	±6.676e-4	±0.2853	±1.1685	±1.9091	±7.949e-2	8.14
$SDFT_3$	*	*	±0.8651	±2.1436	±5.080e-2	*	±0.1419	±0.9064	±2.1894	±4.346e-2	±6.710e-4	±0.1377	±0.9459	±2.2277	±4.747e-2	12.7
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$SDFT_{3n}$	*	*	*	*	±0.1719	*	±3.022e-3	*	*	±0.1175	±6.752e-4	±0.1268	±0.9645	±1.878e-2	±0.1044	34.10
$SDFT_{35n}$	*	*	*	*	±0.8151	*	*	*	*	±1.1158	±6.772e-4	±2.918e-3	±1.710e-2	±1.480e-3	±0.7237	29.05
$SDFT_s$	*	*	±1.705e-3	±1.9415	±1.750e-2	*	±2.738e-4	±1.696e-3	±1.8790	±2.509e-2	±6.774e-4	±8.295e-4	±2.140e-3	±1.9403	±1.714e-2	12.90
$SDFT_{3s}$	*	*	±3.818e-3	±2.1985	±1.827e-2	*	±8.909e-4	±3.892e-3	±2.1280	±2.673e-2	±6.801e-4	±1.198e-3	±4.293e-3	±2.1917	±1.761e-2	15.82
$SDFT_{35s}$	*	*	±5.244e-4	±2.3007	±1.860e-2	*	±3.688e-4	±6.059e-4	±2.2270	±2.746e-2	±6.811e-4	±8.986e-4	±1.161e-3	±2.2993	±1.814e-2	17.47
$SDFT_{357s}$	*	*	±2.016e-4	±2.3586	±1.885e-2	*	*	±2.274e-4	±2.2830	±2.786e-2	±6.816e-4	±6.814e-4	±8.508e-4	±2.3571	±1.843e-2	19.38
$SDFT_{ns}$	*	*	*	*	±1.761e-2	*	±1.497e-4	*	*	±2.517e-2	±6.725e-4	±7.414e-4	±1.381e-3	±2.441e-2	±1.713e-2	16.15
$SDFT_{3ns}$	*	*	*	*	±1.841e-2	*	±3.662e-4	*	*	±2.704e-2	±6.725e-4	±7.963e-4	±2.404e-3	±1.296e-2	±1.784e-2	18.64
$SDFT_{35ns}$	*	*	*	*	±1.880e-2	*	*	*	*	±2.777e-2	±6.725e-4	±6.989e-4	±9.140e-4	±1.069e-2	±1.837e-2	22.08
SDFT	*	*	±0.8353	±2.8845	±5.017e-2	*	±0.2930	±0.8413	±2.8819	±4.413e-2	±6.675e-4	±0.2857	±0.8690	±2.9346	±6.063e-2	8.17
$SDFT_3$	*	*	±0.6044	±3.2929	±2.960e-2	*	±0.1414	±0.6912	±3.3598	±2.745e-2	±6.707e-4	±0.1371	±0.7006	±3.4245	±3.054e-2	12.43
$SDFT_{35}$	*	*	±6.971e-3	±3.4885	±1.263e-2	*	±3.920e-3	±9.252e-3	±3.6864	±1.231e-2	±6.726e-4	±3.991e-3	±9.147e-3	±3.7635	±1.382e-2	13.38
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	[illegible]	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	*	*	[illegible]	*	*	*	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
SDFT	*	*	[illegible]	[illegible]	[illegible]	*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
$SDFT_{357s}$	*	*	±1.478e-4	±3.5568	±1.177e-2	*	*	±1.725e-4	±3.4110	±1.346e-2	±6.813e-4	±6.811e-4	±8.080e-4	±3.5440	±1.480e-2	19.07
$SDFT_{ns}$	*	*	*	*	±1.074e-2	*	±1.498e-4	*	*	±1.227e-2	±6.725e-4	±7.421e-4	±1.193e-3	±2.532e-2	±1.329e-2	16.48
$SDFT_{3ns}$	*	*	*	*	±1.140e-2	*	±3.679e-4	*	*	±1.310e-2	±6.725e-4	±7.954e-4	±1.933e-3	±1.251e-2	±1.433e-2	18.78
$SDFT_{35ns}$	*	*	*	*	±1.173e-2	*	*	*	*	±1.342e-2	±6.725e-4	±6.983e-4	±8.521e-4	±1.154e-2	±1.475e-2	21.72

*: Exact solution (frequency error < 1e-9Hz) **: frequency changes from 60Hz to 59.5Hz during 1 second.

+: Average CPU time (sec) of Pentium 133

Case 1: $v = \cos(\omega t),\ \omega = 2\pi f$

Case 2: $v = \cos(\omega t) + 0.05\cos(3\omega t) + 0.03\cos(5\omega t) + 0.01\cos(7\omega t)$

Case 3: $v = \cos(\omega t) + 0.03\cos(2\pi f_1),\ f_1 = 321\text{Hz}$

Case 4: $v = \cos(\omega t) + 0.5e^{-50t}$

Case 5: $v = \cos(\omega t) + 0.5\%\text{noise}$

We choose fifteen members from SDFT family and use three different window sizes 16, 32 and 48 to evaluate performance. All of them are tested by three different conditions of fundamental frequency. Meanwhile, each of them includes 5 cases. Table 1 lists frequency errors (Hz) of every method under every condition and shows the average CPU time of Pentium 133. For comparison, we also listed the performances of Half-Cycle DFT (HDFT) and DFT in Table 1.

To distinguish every member of SDFT family easily, SDFT means calculating frequency for only fundamental component. We add suffix to the others, for example $SDFT_{35n}$ means takes 3rd and 5th and one non-integral harmonics into consideration. $SDFT_d$ means calculating fundamental frequency and DC offset. $SDFT_s$, which combined smoothing window in SDFT family, uses Blackman window (n=32) for filtering in simulations.

From Table 1, many topics can be discussed. Generally speaking, the accuracy of SDFT family is as well as DFT at 60Hz and no leakage error when frequency deviated from 60Hz. Moreover, SDFT family tracks frequency variation well and some members of SDFT family deal with non-integral harmonic and DC offset certainly. It is worthy to note that the performance of $SDFT_n$ is equal to $SDFT_d$, because they have to solve the same equation.

$SDFT_n$

$$\begin{array}{llll} -4 \times & \hat{x}_{r+2} & \times z_1 z_n & \\ +2 \times & (\hat{x}_{r+1} + \hat{x}_{r+3}) & \times (z_1 + z_n) & \\ -1 \times & (\hat{x}_r + 2\hat{x}_{r+2} + \hat{x}_{r+4}) & = & 0 \end{array}$$

$SDFT_d$

$$\begin{array}{llll} -4 \times & \hat{x}_{r+2} & \times z_1 z_d & \\ +2 \times & (\hat{x}_{r+1} + \hat{x}_{r+3}) & \times (z_1 + z_d) & \\ -1 \times & (\hat{x}_r + 2\hat{x}_{r+2} + \hat{x}_{r+4}) & = & 0 \end{array}$$

The only deference between $SDFT_n$ and $SDFT_d$, is the definition of z_n and z_d.

The last column of Table 1 is CPU Time of each method. Since this is the average computing time of each method under every condition, it should reflect real relation in practice. We can find there are 3 methods to increase accuracy of SDFT family in Table 1. First, increasing window size of SDFT family. Since all SDFT family use recursive computing, increasing window size doesn't increase CPU time, but reduce CPU time instead. Because increasing window size is helpful to reduce the times of iteration. Secondly, adding smoothing window to SDFT family is useful method and CPU time addition is constant. The computing time of smoothing window is about 4.8sec per 1920 data. But as the same of increasing window size, adding smoothing window is also helpful to reduce the times of iteration. It is very clear at $SDFT_n$ series. Thirdly, take more harmonics into consideration, but the more harmonics taken into consideration will take more time in computing, especially taking non-integral harmonics into consideration.

IV. Conclusion

In this paper we present the SDFT family and demonstrate their performance in Table 1. SDFT family provides flexible and reliable solutions for estimating. Not only keeps the advantages of DFT but adds new advantages in taking harmonics and DC offset into consideration, combining smoothing windows, and fractional cycle computing. These aspects make SDFT family more efficient and suitable for power systems under real-time demands.

Reference

[1] G. Missout and P. Girard, "Measurement of Bus Voltage Angle Between Montreal and Sept-Iles", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-99, No. 2, March/April 1980, pp. 536-539.

[2] C. T. Nguyen and K. Srinivasan, "A New Technique for Rapid Tracking of Frequency Deviations Based on Level Crossings", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-103, No.8, August 1984, pp.2230-2236.

[3] M. S. Sachdev and M. M. Giray, "A Least Error Squares Technique For Determining Power System Frequency", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-104, No. 2, February 1985, pp. 437-443.

[4] V. V. Terzija, M. B. Djuric, and B. D. Kovacevic, "Voltage Phasor and Local System Frequency Estimation Using Newton Type Algorithm", *IEEE Transactions on Power Delivery*, Vol. 9, No. 3, July 1994, pp. 1368-1374

[5] M. S. Sachdev, H. C. Wood, and N. G. Johnson, "Kalman Filtering Applied to Power System Measurements for Relaying", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-104, No. 12, December 1985, pp. 3565-3573.

[6] T. Lobos and J. Rezmer, "Real-Time Determination of Power System Frequency", *IEEE Transactions on Instrumentation and measurement*, Vol. 46, No. 4, August 1997, pp. 877-881.

[7] A. G. Phadke, J. S. Thorp, and M. G. Adamiak, "A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency, and Rate of Change of Frequency", *IEEE Transactions on Power Apparatus and Systems*, Vol.102, No. 5, May 1983, pp. 1025-1038.

[8] M. Meunier and F. Brouaye, "Fourier Transform, Wavelets, Prony analysis: Tools for Harmonics and Quality of Power", *Harmonics and Quality of Power Proceedings*, 1998 Proceedings. 8th International Conference On Volume: 1, 1998, pp. 71 –76.

Biographies

Chih-Wen Liu was born in Taiwan in 1964. He received the B.S. degree in Electrical Engineering from National Taiwan University in 1987, Ph.D. degree in electrical engineering from Cornell University in 1994. Since 1994, he has been with National Taiwan University, where he is associate professor of electrical engineering. His research interests include application of numerical methods to power system, motor control and GPS time transfer.

Jun-Zhe Yang was born at Tainan, Taiwan, in 1971. He received his B.S. degree in electrical engineering from Tatung Institute of Technology in 1994 and M.S. degree from National Taiwan University in 1997. He is presently a graduate student in the electrical engineering department, National Taiwan University, Taipei, Taiwan.