Article

Fast 10-Point DFT Algorithm for Power System Harmonic Analysis

Janusz P. Paplinski * and Aleksandr Cariow

Faculty of Computer Science and Information Technology, West Pomeranian University of Technology,
Żołnierska 49, 71-210 Szczecin, Poland; acariow@wi.zut.edu.pl

* Correspondence: janusz.paplinski@zut.edu.pl

Abstract: This article presents an efficient algorithm for computing a 10-point DFT. The proposed algorithm reduces the number of multiplications at the cost of a slight increase in the number of additions in comparison with the known algorithms. Using a 10-point DFT for harmonic power system analysis can improve accuracy and reduce errors caused by spectral leakage. This paper compares the computational complexity for an $L \times 10^M$ -point DFT with a 2^M -point DFT.

Keywords: FFT; 10-point DFT; harmonic analysis; power system

1. Introduction

Non-linear elements are a source of distorted waveforms in power systems. Typical examples are variable-speed drives (VSD) [1,2], high-voltage direct currents (HVDC), frequency converters in wind, and solar power plants [3,4]. The resulting non-sinusoidal periodic waveforms consist of a basic component and unwanted harmonics, which can disrupt the power system. Due to the large increase in the applications of power electronics, harmonics and their reduction have been major problems and the topic of research in recent decades. A power system uses a frequency of 50 Hz or 60 Hz, and the harmonic components are multiples of the fundamental frequency.

One of the basic methods of harmonic analysis is the Discrete Fourier Transform. Various fast Fourier transform algorithms have been developed to optimize the time and resources needed for the determination of harmonics [5,6]; most of these are based on a signal with length N being a power of 2 or 4. When the harmonic analysis method based on $N = 2^M$ -point DFTs is selected for a power system, errors appear due to the method used. For non-coherent signals, when a non-integral number of periods is present in the sampled data set, a spectral leakage occurs. This effect is because the sequence of data collected must have a finite duration and does not occur with coherent sampling in which the sampling rate is an integer multiple of period of the fundamental harmonic. The spectral leakage has two forms: short-range leakage and long-range leakage [7–11]. The first is caused by a non-flat upper lobe of the main window. An error in the frequency estimate occurs and the amplitude and phase estimates are attenuated and shifted, respectively. The second kind of leakage is caused by the fact that all harmonic components make a nonzero contribution to the evaluation of the n -th harmonic component. If zero padding is used, the picket fence effect might occur, and some important peaks may be ignored. These phenomena affect the accuracy of the harmonic analysis [10]. Similar effects occur when there are interharmonics in the signal.

Various frequency estimation methods have been proposed and adapted to non-coherent and non-stationary signals. These include Phase-Locked Loop (PLL) algorithms based on the ability to generate a sinusoidal output whose amplitude and phase are equal to those of the fundamental component of the input signal [12–15]; methods based on Prony's theory, which approximates a signal by a sum of exponential functions [16,17]; Adaptive Notch Filters (ANFs), which are frequently used to eliminate narrow-band or sine

Citation: Paplinski, J.P.; Cariow, A. Fast 10-Point DFT Algorithm for Power System Harmonic Analysis. *Appl. Sci.* **2021**, *11*, 7007. <https://doi.org/10.3390/app11157007>

Academic Editor: Sergio Toscani

Received: 24 June 2021

Accepted: 28 July 2021

Published: 29 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

wave disturbances with unknown frequencies from observed time series and to estimate the frequencies of measured sine wave signals [18,19]; variants of the Extended Kalman Filter (EKF), which works as recursive estimation method based on a linear state-space model [20,21]; and even neural network-based techniques [22–24]. All these algorithms have constraints to their use [25]. The drawbacks associated with the PLL are sensitivity to grid voltage variations, difficulty to obtain an accurate tuning of the PLL parameters, and slow action because of the proportional–integral block. The drawbacks of Prony’s method include its sensitivity to the noise contained in the signal, stability problems, and difficulties in inverting high-dimensional matrices [16]. An ANF implemented with a cascaded structure is perfectly able to track the fundamental frequency when the number of sub-parts is greater than the number of harmonics supposed to be in the signal; however, this can lead to significant computation costs to decrease the convergence speed and generate some oscillations in the estimation. In addition, the estimated frequency easily falls into local minima due to the non-quadratic error surface. The EKF approach is sensitive to the fact that the amplitudes of the harmonics are small compared to the fundamental amplitude, resulting in a steady-state error.

There are also many methods that are complementary to the FFT that allow the improvement of the efficiency of this algorithm, such as spectrum interpolation methods [26–30] and the zero padding method combined with zero-sampling elimination (pruning FFT) [31,32]. Both methods increase computational requirements and only reduces the error caused by the discrete nature of the DFT spectrum, without reducing the error caused by the effect of spectrum leakage from other components. Time windows with a shape other than rectangular are also used; however, they reduce the frequency resolution of the analysis and amplify the noise in the DFT spectrum [33–36].

For the satisfactory operation of a power system, the fundamental frequency should remain constant. In such a case, it can be assumed that there is a coherent signal; then, the FFT remains the best solution because of its efficient and simple structure. However, as mentioned above when a standard radix 2 or 4 FFT is used for the harmonic analysis of a power system, the problem of spectrum leakage remains.

In [37], a solution to this problem by using $N = L \times 10^M$ -point DFTs was proposed to reduce spectral leakage and provide high-accuracy harmonic parameters. An original approach to computing a 10-point DFT was described in [38,39]. Developing this method, we propose a new 10-point DFT algorithm with less multiplicative complexity.

2. Preliminaries

For a sequence of N complex numbers $\{\mathbf{x}_n\} := x_0, x_1, \dots, x_{N-1}$, $\{\mathbf{X}_k\} := X_0, X_1, \dots, X_{N-1}$ the Discrete Fourier Transform is defined by

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{j2\pi}{N}kn}, \quad k = 0, 1, \dots, N-1 \quad (1)$$

where e is Euler’s number.

If we assume $w_N = e^{-\frac{j2\pi}{N}}$, the DFT can be expressed as

$$X_k = \sum_{n=0}^{N-1} x_n w_N^{kn}, \quad k = 0, 1, \dots, N-1 \quad (2)$$

For $N = 10$, the DFT has the following matrix notation:

$$\mathbf{X}_{10 \times 1} = \mathbf{W}_{10} \mathbf{x}_{10 \times 1} \quad (3)$$

where

$$\begin{aligned} \mathbf{X}_{10 \times 1} &= [X_0 \ X_1 \ X_2 \ X_3 \ X_4 \ X_5 \ X_6 \ X_7 \ X_8 \ X_9]^T, \\ \mathbf{x}_{10 \times 1} &= [x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9]^T, \end{aligned}$$

$$\mathbf{W}_{10} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w_{10}^1 & w_{10}^2 & w_{10}^3 & w_{10}^4 & w_{10}^5 & w_{10}^6 & w_{10}^7 & w_{10}^8 & w_{10}^9 \\ 1 & w_{10}^2 & w_{10}^4 & w_{10}^6 & w_{10}^8 & w_{10}^{10} & w_{10}^{12} & w_{10}^{14} & w_{10}^{16} & w_{10}^{18} \\ 1 & w_{10}^3 & w_{10}^6 & w_{10}^9 & w_{10}^{12} & w_{10}^{15} & w_{10}^{18} & w_{10}^{21} & w_{10}^{24} & w_{10}^{27} \\ 1 & w_{10}^4 & w_{10}^8 & w_{10}^{12} & w_{10}^{16} & w_{10}^{20} & w_{10}^{24} & w_{10}^{28} & w_{10}^{32} & w_{10}^{36} \\ 1 & w_{10}^5 & w_{10}^{10} & w_{10}^{15} & w_{10}^{20} & w_{10}^{25} & w_{10}^{30} & w_{10}^{35} & w_{10}^{40} & w_{10}^{45} \\ 1 & w_{10}^6 & w_{10}^{12} & w_{10}^{18} & w_{10}^{24} & w_{10}^{30} & w_{10}^{36} & w_{10}^{42} & w_{10}^{48} & w_{10}^{54} \\ 1 & w_{10}^7 & w_{10}^{14} & w_{10}^{21} & w_{10}^{28} & w_{10}^{35} & w_{10}^{42} & w_{10}^{49} & w_{10}^{56} & w_{10}^{63} \\ 1 & w_{10}^8 & w_{10}^{16} & w_{10}^{24} & w_{10}^{32} & w_{10}^{40} & w_{10}^{48} & w_{10}^{56} & w_{10}^{64} & w_{10}^{72} \\ 1 & w_{10}^9 & w_{10}^{18} & w_{10}^{27} & w_{10}^{36} & w_{10}^{45} & w_{10}^{54} & w_{10}^{63} & w_{10}^{72} & w_{10}^{81} \end{bmatrix} \quad (4)$$

In a general case, w_N is a periodic function, and some simplifications can be made: $w_N^{mN} = 1$, $w_N^{mN+N/2} = -1$, $w_N^m = w_N^{\text{mod}(m,N)}$, $w_N^{-m} = w_N^{N-m}$, where m is an integer. For $N = 10$, the matrix $\mathbf{W}_{10}$ (4) can be simplified to

$$\mathbf{W}_{10} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w_{10}^1 & w_{10}^2 & w_{10}^3 & w_{10}^4 & -1 & w_{10}^{-4} & w_{10}^{-3} & w_{10}^{-2} & w_{10}^{-1} \\ 1 & w_{10}^2 & w_{10}^4 & w_{10}^6 & w_{10}^{-2} & 1 & w_{10}^2 & w_{10}^4 & w_{10}^{-4} & w_{10}^{-2} \\ 1 & w_{10}^3 & w_{10}^{-4} & w_{10}^{-1} & w_{10}^2 & -1 & w_{10}^{-2} & w_{10}^1 & w_{10}^4 & w_{10}^{-3} \\ 1 & w_{10}^4 & w_{10}^{-2} & w_{10}^2 & w_{10}^{-4} & 1 & w_{10}^4 & w_{10}^{-2} & w_{10}^2 & w_{10}^{-4} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & w_{10}^{-4} & w_{10}^2 & w_{10}^{-2} & w_{10}^4 & 1 & w_{10}^{-4} & w_{10}^2 & w_{10}^{-2} & w_{10}^4 \\ 1 & w_{10}^{-3} & w_{10}^4 & w_{10}^{-1} & w_{10}^{-2} & -1 & w_{10}^2 & w_{10}^{-1} & w_{10}^{-4} & w_{10}^3 \\ 1 & w_{10}^{-2} & w_{10}^{-4} & w_{10}^4 & w_{10}^2 & 1 & w_{10}^{-2} & w_{10}^{-4} & w_{10}^4 & w_{10}^2 \\ 1 & w_{10}^{-1} & w_{10}^{-2} & w_{10}^{-3} & w_{10}^{-4} & -1 & w_{10}^4 & w_{10}^3 & w_{10}^2 & w_{10}^1 \end{bmatrix} \quad (5)$$

Let us assume, according to the procedure given in [38], new variables $A_0 = X_0$, $A_i = (X_i + X_{10-i})/2$, and $A_{10-i} = (X_i - X_{10-i})/2$ for $i = 1, 2, 3, 4$. Then, we can write the following:

$$\mathbf{A}_{10 \times 1} = \mathbf{V}_{10} \mathbf{X}_{10 \times 1} \quad (6)$$

where

$$\mathbf{A}_{10 \times 1} = [A_0 \ A_1 \ A_2 \ A_3 \ A_4 \ A_5 \ A_6 \ A_7 \ A_8 \ A_9]^T, \quad \mathbf{V}_{10} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/2 \end{bmatrix} \quad (7)$$

Using the inverse operation $\mathbf{X}_{10 \times 1} = \mathbf{V}_{10}^{-1} \mathbf{A}_{10 \times 1}$, the 10-point DFT (3) can be rewritten as

$$\mathbf{X}_{10 \times 1} = \mathbf{V}_{10}^{-1} \mathbf{W}_{10}^{(1)} \mathbf{x}_{10 \times 1} \quad (8)$$

where

$$\mathbf{V}_{10}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix},$$

$$\mathbf{W}_{10}^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & C_1 & C_2 & C_3 & C_4 & -1 & C_4 & C_3 & C_2 & C_1 \\ 1 & C_2 & C_4 & C_4 & C_2 & 1 & C_2 & C_4 & C_4 & C_2 \\ 1 & C_3 & C_4 & C_1 & C_2 & -1 & C_2 & C_1 & C_4 & C_3 \\ 1 & C_4 & C_2 & C_2 & C_4 & 1 & C_4 & C_2 & C_2 & C_4 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & -jS_4 & jS_2 & -jS_2 & jS_4 & 0 & -jS_4 & jS_2 & -jS_2 & jS_4 \\ 0 & -jS_3 & jS_4 & jS_1 & -jS_2 & 0 & jS_2 & -jS_1 & -jS_4 & jS_3 \\ 0 & -jS_2 & -jS_4 & jS_4 & jS_2 & 0 & -jS_2 & -jS_4 & jS_4 & jS_2 \\ 0 & -jS_1 & -jS_2 & -jS_3 & -jS_4 & 0 & jS_4 & jS_3 & jS_2 & jS_1 \end{bmatrix},$$

and coefficients C_m, jS_m , for $m = 1, 2, 3, 4$, are equal:

$$C_m = (w_{10}^m + w_{10}^{-m})/2,$$

$$-jS_m = (w_{10}^m - w_{10}^{-m})/2.$$

Using Euler's formula, C_m and jS_m are reduced to [38]

$$C_m = \left(e^{-\frac{j2\pi}{10}m} + e^{\frac{j2\pi}{10}m} \right) / 2 = \cos(m\pi/5) \quad (9)$$

$$jS_m = \left(e^{-\frac{j2\pi}{10}m} - e^{\frac{j2\pi}{10}m} \right) / 2 = j \sin(m\pi/5) \quad (10)$$

We create auxiliary variables $a_0 = x_0$, $a_5 = x_5$, $a_i = (x_i + x_{10-i})/2$, and $a_{10-i} = (x_i - x_{10-i})/2$ for $i = 1, 2, 3, 4$, as in [38]:

$$\mathbf{a}_{10 \times 1} = \mathbf{V}_{10} \mathbf{x}_{10 \times 1},$$

where matrix $\mathbf{V}_{10}$ is defined in (7) and

$$\mathbf{a}_{10 \times 1} = [a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9]^T.$$

Then, the 10-point DFT algorithm (8) takes the following form:

$$\mathbf{X}_{10 \times 1} = \mathbf{V}_{10}^{-1} \mathbf{W}_{10}^{(2)} \mathbf{V}_{10}^{-1} \mathbf{x}_{10 \times 1} \quad (11)$$

where

$$\mathbf{W}_{10}^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & C_1 & C_2 & C_3 & C_4 & -1 & 0 & 0 & 0 & 0 \\ 1 & C_2 & C_4 & C_4 & C_2 & 1 & 0 & 0 & 0 & 0 \\ 1 & C_3 & C_4 & C_1 & C_2 & -1 & 0 & 0 & 0 & 0 \\ 1 & C_4 & C_2 & C_2 & C_4 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & jS_4 & -jS_2 & jS_2 & -jS_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & -jS_2 & jS_1 & jS_4 & -jS_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & jS_2 & jS_4 & -jS_4 & -jS_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -jS_4 & -jS_3 & -jS_2 & -jS_1 \end{bmatrix}.$$

Using C_m and S_m , as defined in (9), and (10), the reduction formulas of trigonometric functions $\sin(\pi - a) = \sin(a)$ and $\cos(\pi - a) = -\cos(a)$, we obtain $C_3 = -C_2$, $C_4 = -C_1$, $S_3 = S_2$, and $S_4 = S_1$. This results in a new matrix form:

$$\mathbf{W}_{10}^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & C_1 & C_2 & -C_2 & -C_1 & -1 & 0 & 0 & 0 & 0 \\ 1 & C_2 & -C_1 & -C_1 & C_2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -C_2 & -C_1 & C_1 & C_2 & -1 & 0 & 0 & 0 & 0 \\ 1 & -C_1 & C_2 & C_2 & -C_1 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & jS_1 & -jS_2 & jS_2 & -jS_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -jS_2 & jS_1 & jS_1 & -jS_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & jS_2 & jS_1 & -jS_1 & -jS_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -jS_1 & -jS_2 & -jS_2 & -jS_1 \end{bmatrix} \quad (12)$$

Now, according to [38], we introduce the new variables $B_i = (A_i + A_{5-i})/2$, $B_{5-i} = (A_i - A_{5-i})/2$ for $i = 0, 1, 2$, and $B_{5+k} = (A_{5+k} + A_{10-k})/2$, $B_{10-k} = (A_{5+k} - A_{10-k})/2$ for $k = 1, 2$; then, we can write the following:

$$\mathbf{B}_{10 \times 1} = 0.5 \hat{\mathbf{V}}_{10} \mathbf{A}_{10 \times 1},$$

where

$$\mathbf{B}_{10 \times 1} = [B_0 \ B_1 \ B_2 \ B_3 \ B_4 \ B_5 \ B_6 \ B_7 \ B_8 \ B_9]^T,$$

$$\hat{\mathbf{V}}_{10} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}.$$

Then, the 10-point DFT (11) after simple conversions can be written as

$$\mathbf{x}_{10 \times 1} = \mathbf{V}_{10}^{-1} \hat{\mathbf{V}}_{10} \mathbf{W}_{10}^{(3)} \mathbf{V}_{10}^{-1} \mathbf{x}_{10 \times 1} \quad (13)$$

where

$$\mathbf{W}_{10}^{(3)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & C_2 & 0 & -C_1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -C_1 & 0 & C_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & -C_1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & C_1 & 0 & -C_2 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -jS_2 & 0 & -jS_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & jS_1 & 0 & -jS_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -jS_2 & 0 & jS_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & jS_1 & 0 & jS_2 & 0 \end{bmatrix}.$$

In matrix $\mathbf{W}_{10}^{(3)}$, we change the signs of all entries of the 3rd row, as well as all entries of the 4th, 5th, and 10th columns. This can be done by multiplying the entries by the matrices $\mathbf{S}_{10}^{(r)}$ and $\mathbf{S}_{10}^{(c)}$ on the left and right, respectively:

$$\mathbf{W}_{10}^{(4)} = \mathbf{S}_{10}^{(r)} \mathbf{W}_{10}^{(3)} \mathbf{S}_{10}^{(c)} = \begin{bmatrix} 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & C_2 & 0 & C_1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & C_1 & 0 & C_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & C_1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & C_1 & 0 & C_2 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -jS_2 & 0 & jS_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & jS_1 & 0 & jS_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -jS_2 & 0 & jS_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & jS_1 & 0 & jS_2 & 0 \end{bmatrix},$$

where

$$\mathbf{S}_{10}^{(r)} = \text{diag}(1, 1, -1, 1, 1, 1, 1, 1, 1, 1),$$

$$\mathbf{S}_{10}^{(c)} = \text{diag}(1, 1, 1, -1, -1, 1, 1, 1, 1, -1).$$

Then, we can write the 10-point DFT (13) in the following form:

$$\mathbf{X}_{10 \times 1} = \mathbf{V}_{10}^{-1} \hat{\mathbf{V}}_{10} \mathbf{S}_{10}^{(r)} \mathbf{W}_{10}^{(3)} \mathbf{S}_{10}^{(c)} \mathbf{V}_{10}^{-1} \mathbf{x}_{10 \times 1} \quad (14)$$

Let us denote the entries of the matrix $\mathbf{W}_{10}^{(4)}$ as w_{ik} , where $i, k = 0, 1, \dots, 9$ are the numbers of a row and a column, respectively. It is easy to see that we can extract four submatrices from the matrix $\mathbf{W}_{10}^{(4)}$ by choosing appropriate rows and columns such that they are identical in pairs:

$$\mathbf{W}_2^{(1)} = \begin{bmatrix} w_{12} & w_{14} \\ w_{22} & w_{24} \end{bmatrix} = \begin{bmatrix} w_{31} & w_{33} \\ w_{41} & w_{43} \end{bmatrix} = \begin{bmatrix} C_2 & C_1 \\ C_1 & C_2 \end{bmatrix},$$

$$\mathbf{W}_2^{(2)} = \begin{bmatrix} w_{67} & w_{69} \\ w_{77} & w_{79} \end{bmatrix} = \begin{bmatrix} w_{86} & w_{88} \\ w_{96} & w_{98} \end{bmatrix} = \begin{bmatrix} -jS_2 & jS_1 \\ jS_1 & jS_2 \end{bmatrix}.$$

The matrices $\mathbf{W}_2^{(1)}$ and $\mathbf{W}_2^{(2)}$ contain symmetric and bisymmetric structures that allow the further simplification of the number of operations [40].

The structure of the $\mathbf{W}_2^{(1)}$ matrix can be symbolically represented as

$$\mathbf{W}_2^{(1)} = \begin{bmatrix} a & b \\ b & a \end{bmatrix} = \mathbf{H}_2 \mathbf{D}_2 \mathbf{H}_2,$$

where $\mathbf{H}_2$ is the order two Hadamard matrix, $\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$,

$$\mathbf{D}_2 = \frac{1}{2} \text{diag}(a+b, a-b).$$

The structure of the $\mathbf{W}_2^{(2)}$ matrix can be represented as

$$\mathbf{W}_2^{(2)} = \begin{bmatrix} -a & b \\ b & a \end{bmatrix} = \mathbf{T}_{2 \times 3} \mathbf{D}_3 \mathbf{T}_{3 \times 2},$$

where

$$\mathbf{T}_{2 \times 3} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{D}_3 = \text{diag}(a+b, a-b, b), \quad \mathbf{T}_{3 \times 2} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

Consequently, the final algorithm for the 10-point DFT (14), which minimizes numerical complexity, can be written as

$$\mathbf{X}_{10 \times 1} = \mathbf{V}_{10}^{-1} \hat{\mathbf{V}}_{10} \mathbf{S}_{10}^{(r)} \mathbf{T}_{10 \times 14} \mathbf{W}_{14 \times 12} \mathbf{T}_{12 \times 10} \mathbf{S}_{10}^{(c)} \mathbf{V}_{10}^{-1} \mathbf{x}_{10 \times 1} \quad (15)$$

where

$$\mathbf{T}_{12 \times 10} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{T}_{10 \times 14} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix},$$

and

$$\mathbf{W}_{14 \times 12} = \begin{bmatrix} \mathbf{W}_{8 \times 6} & \mathbf{0}_{8 \times 6} \\ \mathbf{0}_6 & \mathbf{W}_6 \end{bmatrix},$$

where $\mathbf{0}_{8 \times 6}$ and $\mathbf{0}_6$ are zero matrices,

$$\mathbf{W}_{8 \times 6} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & (C_2 + C_1)/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & (C_2 - C_1)/2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & (C_2 + C_1)/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & (C_2 - C_1)/2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{W}_6 = \begin{bmatrix} jS_2 + jS_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & jS_2 - jS_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & jS_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & jS_2 + jS_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & jS_2 - jS_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & jS_1 \end{bmatrix}.$$

We can substitute real values into C_1 , C_2 , and S_1 , S_2 using relations (9) and (10):

$$C_1 = \cos(\pi/5) = 0.8090, \quad jS_1 = j \sin(\pi/5) = j0.5878, \\ C_2 = \cos(2\pi/5) = 0.3090, \quad jS_2 = \sin(2\pi/5) = j0.9511.$$

The matrices $\mathbf{W}_{8 \times 6}$ and $\mathbf{W}_6$ take the following form:

$$\mathbf{W}_{8 \times 6} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5590 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.2500 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5590 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.2500 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{W}_6 = \begin{bmatrix} 1.5388 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3633 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5878 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5388 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3633 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5878 \end{bmatrix}.$$

The signal flow graph of the proposed algorithm is presented in Figure 1. The Matlab code of the 10-point DFT and the code verifying the each step of obtaining the 10-point DFT are included in the Supplementary Materials.

Figure 1. The signal flow graph of the proposed algorithm for a 10-point DFT, where $b_0 = (C_2 + C_1)/2 = 0.5590$, $b_1 = S_2 + S_1 = 1.5388$, and $b_2 = S_2 - S_1 = 0.3633$.

The multiplication by 0.25 is performed with a 2 bit shift to the right. From an examination of Figure 1, it can be seen that the algorithm contains two multiplications by the number 1/4, which is a power of two. This operation is reduced to the usual shift of the multiplicand by two positions to the right. Due to the simplicity of implementation, such operations are usually not considered when evaluating computational complexity [41,42]. Therefore, the proposed algorithm requires only 16 real multiplications and 92 real additions. Thus, in comparison with the base method [38,39], which required 20 real multiplications and 84 additions, we decrease the number of real multiplications by 4 and increase the number of real additions by 8. Table 1 compares the number of real multiplication and addition operations for the proposed algorithm, the baseline algorithm [38], DFT, the mixed-radix fast Fourier transform (MFFT), prime factor algorithm (PFA), Winograd Fourier transform (WFTA), and SWIFT algorithms [43,44]. Since multiplication is a more complicated operation from the point of view of its implementation [41,42], the algorithm we propose has an advantage over the presented algorithms.

Table 1. The number of real multiplications and additions for the proposed algorithm, the baseline algorithm [38], DFT [43], MFFT [43], PFA [44], WFTA [44], and SWIFT algorithms [43].

	DFT	MFFT	SWIFT	WFTA	PFA	Baseline [38]	Proposed
Number of real multiplications	400	48	32	20	20	20	16
Number of real additions	380	92	84	88	88	84	92

3. Computing $L \times 10$ -Point DFT

The $N = 2^M$ -point DFT can be quickly computed with radix-2 or radix-4. There are many applications where the sequence length varies from 2^m or 4^m . Abnormal radix-based DFTs are often proposed in this case [45–47]. When $N = L \times 10^M$ -point DFT, in which L and M are integers, several algorithms can be used, such as the prime-factor algorithm (PFA) [48], double factors algorithm (DFA) [49], or Winograd Fourier transform algorithm

(WFTA) [50]. The length N of DFT can be subdivided into smaller transforms of length $N = N_1 N_2$.

Using a mixed-radix in an electric power system is not a new solution [51]. For the harmonic analysis of a 50 Hz or 60 Hz signal, it is convenient to assume $N = 10^M N_2$ and $N_1 = 10^M$, which solves the problem of sampling a non-integral number of periods and thus spectral leakage [37,39]. If we let M_{R1} , A_{R1} , S_{R1} , and M_{R2} , A_{R2} , S_{R2} denote the number of real multiplications, additions, and assume right-shiftings by 1 bit of the N_1 and N_2 -point DFT, then the total number of real multiplications and additions of the $(N = N_1 N_2)$ -point DFT in the DFA [49] are

$$M_R = N_2 M_{R1} + N_1 M_{R2} + 4(N_1 - 1)(N_2 - 1),$$

$$A_R = N_2 A_{R1} + N_1 A_{R2} + 2(N_1 - 1)(N_2 - 1),$$

$$S_R = N_2 S_{R1} + N_1 S_{R2}.$$

The total number of real multiplications and additions of the $(N = N_1 N_2)$ -point DFT in the PFA [43] are

$$M_R = N_2 M_{R1} + N_1 M_{R2},$$

$$A_R = N_2 A_{R1} + N_1 A_{R2},$$

In Table 2, a comparison of the computational requirements for the $N = 10^M$ -point DFT for DFA and PFA algorithms and the related $N = 2^M$ -point and $N = 4^M$ -point DFTs is shown. The proposed algorithm can be used in any mixed-radix or split-radix algorithms. The best algorithm was a PFA with radix-10. The spectral leakage can be reduced, and the computed harmonic parameters are more approximate to the actual value, as shown in [37,39]. Therefore, the proposed radix-10 DFT can be efficient for use in harmonics analyses in power systems.

Table 2. The number of real multiplications M_R and additions A_R for DFT algorithms.

Length- N	Radix	FFT [52]		Radix	DFA		Radix	PFA	
		M_R	A_R		M_R	A_R		M_R	A_R
20				R-2 R-10	78	222	R-2 R-10	32	224
20							R-5 R-4 [44]	40	216
50				R-5 R-10	324	852	R-5 R-10	180	800
60				R-6 R-10	316	1002	R-6 R-10	136	912
60							R-5 R-4 R-3 [44]	200	888
64	R-2	332	964						
64	R-4	264	920						
100				R-10	644	2002	R-10	320	1840
128	R-2	908	2308						
200				R-2 R-10	1784	4402	R-2 R-10	640	4080
256	R-2	2316	5380						
256	R-4	1800	5080						
500				R-5 R-10	5804	11,302	R-5 R-10	2600	12,600
512	R-2	5644	12,292						
1000				R-10	11,604	31,002	R-10	4800	27,600
1024	R-2	13,324	27,652						
1024	R-4	10,248	25,944						
2000				R-2 R-10	28,204	66,002	R-2 R-10	9600	59,200
2048	R-2	30,732	61,444						

4. Conclusions

The algorithm proposed in this paper allows us to reduce the number of multiplication operations at the expense of additions compared to known algorithms. An approach based on 10^M -point DFT harmonic analysis in power systems seems to be a better solution than

the commonly used 2^M -point DFT. This algorithm can be used for coherent signals. It allows the reduction of errors resulting from a mismatch between the number of samples and the period of the signal under test, while reducing computational requirements.

Supplementary Materials: The Matlab code of the 10-point DFT and the code verifying the each step of obtaining the 10-point DFT are available online at <https://www.mdpi.com/article/10.3390/app11157007/s1>.

Author Contributions: Conceptualization, A.C.; methodology, A.C. and J.P.P.; validation, J.P.P.; formal analysis, A.C. and J.P.P.; writing—original draft preparation, J.P.P.; writing—review, and editing, J.P.P.; visualization, A.C. and J.P.P.; supervision, A.C. and J.P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aarniovuori, L.; Kärkkäinen, H.; Niemelä, M.; Pyrhönen, J. PWM power distribution and switching frequency analysis in motor drives. In Proceedings of the IECON 2016—42nd Annual Conference of the IEEE Industrial Electronics Society, Florence, Italy, 23–26 October 2016; pp. 4356–4361.
2. Rangari, S.; Suryawanshi, H.; Shah, B. Effect of inverter switching frequency on the stator current harmonics of five phase induction motor. In Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 1–2 August 2017; pp. 2128–2132.
3. Nayusrizal, N.; Untari, I.; Putra, A.A.; Hudaya, C. Effects of Harmonic Distortions on the Accuracy of Energy and Power Losses Measurements in a Water Treatment Industry. In Proceedings of the 2019 IEEE 2nd International Conference on Power and Energy Applications (ICPEA), Singapore, 27–30 April 2019; pp. 277–281. [\[CrossRef\]](#)
4. Torquato, R.; Freitas, W.; Hax, G.R.T.; Donadon, A.R.; Moya, R. High frequency harmonic distortions measured in a Brazilian solar farm. In Proceedings of the 2016 17th International Conference on Harmonics and Quality of Power (ICHQP), Belo Horizonte, Brazil, 16–19 October 2016; pp. 623–627. [\[CrossRef\]](#)
5. Blahut, R.E. *Fast Algorithms for Signal Processing*; Cambridge University Press: Cambridge, UK, 2010.
6. Lu, R.T.M.A.C. *Algorithms for Discrete Fourier Transform and Convolution*; Springer: Cham, Switzerland, 1989.
7. Chen, K.F.; Jiang, J.T.; Crowsen, S. Against the long-range spectral leakage of the cosine window family. *Comput. Phys. Commun.* **2009**, *180*, 904–911. [\[CrossRef\]](#)
8. Ferrero, A.; Ottoboni, R. High-accuracy Fourier analysis based on synchronous sampling techniques. *IEEE Trans. Instrum. Meas.* **1992**, *41*, 780–785. [\[CrossRef\]](#)
9. Grandke, T. Interpolation algorithms for discrete Fourier transforms of weighted signals. *IEEE Trans. Instrum. Meas.* **1983**, *32*, 350–355. [\[CrossRef\]](#)
10. Rapuano, S.; Harris, F.J. An introduction to FFT and time domain windows. *IEEE Instrum. Meas. Mag.* **2007**, *10*, 32–44. [\[CrossRef\]](#)
11. Santamaria-Caballero, I.; Pantaleón-Prieto, C.J.; Ibanez-Diaz, J.; Gómez-Cosío, E. Improved procedures for estimating amplitudes and phases of harmonics with application to vibration analysis. *IEEE Trans. Instrum. Meas.* **1998**, *47*, 209–214. [\[CrossRef\]](#)
12. Santos Filho, R.M.; Seixas, P.F.; Cortizo, P.C.; Torres, L.A.; Souza, A.F. Comparison of three single-phase PLL algorithms for UPS applications. *IEEE Trans. Ind. Electron.* **2008**, *55*, 2923–2932. [\[CrossRef\]](#)
13. Karimi-Ghartemani, M.; Iravani, M.R. Measurement of harmonics/inter-harmonics of time-varying frequencies. *IEEE Trans. Power Deliv.* **2005**, *20*, 23–31.
14. Karimi-Ghartemani, M.; Iravani, M.R. A nonlinear adaptive filter for online signal analysis in power systems: Applications. *IEEE Trans. Power Deliv.* **2002**, *17*, 617–622. [\[CrossRef\]](#)
15. Karimi-Ghartemani, M.; Ziarani, A. Periodic orbit analysis of two dynamical systems for electrical engineering applications. *J. Eng. Math.* **2003**, *45*, 135–154. [\[CrossRef\]](#)
16. Bracale, A.; Caramia, P.; Carpinelli, G. Adaptive Prony method for waveform distortion detection in power systems. *Int. J. Electr. Power Energy Syst.* **2007**, *29*, 371–379. [\[CrossRef\]](#)
17. Mitra, S.K.; Kaiser, J.F. *Handbook for Digital Signal Processing*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1993.
18. Lee, K.J.; Lee, J.P.; Shin, D.; Yoo, D.W.; Kim, H.J. A novel grid synchronization PLL method based on adaptive low-pass notch filter for grid-connected PCS. *IEEE Trans. Ind. Electron.* **2013**, *61*, 292–301. [\[CrossRef\]](#)
19. Nehorai, A. A minimal parameter adaptive notch filter with constrained poles and zeros. *IEEE Trans. Acoust. Speech Signal Process.* **1985**, *33*, 983–996. [\[CrossRef\]](#)

20. Dash, P.; Pradhan, A.; Panda, G. Frequency estimation of distorted power system signals using extended complex Kalman filter. *IEEE Trans. Power Deliv.* **1999**, *14*, 761–766. [\[CrossRef\]](#)
21. Girgis, A.A.; Chang, W.B.; Makram, E.B. A digital recursive measurement scheme for online tracking of power system harmonics. *IEEE Trans. Power Deliv.* **1991**, *6*, 1153–1160. [\[CrossRef\]](#)
22. Dash, P.; Swain, D.; Routray, A.; Liew, A. An adaptive neural network approach for the estimation of power system frequency. *Electr. Power Syst. Res.* **1997**, *41*, 203–210. [\[CrossRef\]](#)
23. Halbwachs, D.; Wira, P.; Mercklé, J. Adaline-based approaches for time-varying frequency estimation in power systems. *IFAC Proc. Vol.* **2009**, *42*, 31–36. [\[CrossRef\]](#)
24. Lai, L.; Chan, W.; Tse, C.; So, A. Real-time frequency and harmonic evaluation using artificial neural networks. *IEEE Trans. Power Deliv.* **1999**, *14*, 52–59. [\[CrossRef\]](#)
25. Caciotta, M.; Giarnetti, S.; Leccese, F.; Leonowicz, Z. Comparison between DFT, adaptive window DFT and EDFT for power quality frequency spectrum analysis. In Proceedings of the 2010 Modern Electric Power Systems, Wroclaw, Poland, 20–22 September 2010; pp. 1–5.
26. Aboutanios, E.; Mulgrew, B. Iterative frequency estimation by interpolation on Fourier coefficients. *IEEE Trans. Signal Process.* **2005**, *53*, 1237–1242. [\[CrossRef\]](#)
27. Agrez, D. Weighted multipoint interpolated DFT to improve amplitude estimation of multifrequency signal. *IEEE Trans. Instrum. Meas.* **2002**, *51*, 287–292. [\[CrossRef\]](#)
28. Andria, G.; Savino, M.; Trotta, A. Windows and interpolation algorithms to improve electrical measurement accuracy. *IEEE Trans. Instrum. Meas.* **1989**, *38*, 856–863. [\[CrossRef\]](#)
29. Belega, D.; Dallet, D. Amplitude estimation by a multipoint interpolated DFT approach. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 1316–1323. [\[CrossRef\]](#)
30. Borkowski, J.; Mroczka, J. Metrological analysis of the LIDFT method. *IEEE Trans. Instrum. Meas.* **2002**, *51*, 67–71. [\[CrossRef\]](#)
31. Fan, C.P.; Su, G.A. Pruning fast Fourier transform algorithm design using group-based method. *Signal Process.* **2007**, *87*, 2781–2798. [\[CrossRef\]](#)
32. Rangarajan, S.; Srinivasan, S. Generalised method for pruning an FFT type of transform. *IEE Proc. Vision Image Signal Process.* **1997**, *144*, 189–192. [\[CrossRef\]](#)
33. Belega, D.; Ciugudean, M.; Stoiciu, D. Choice of the cosine-class windows for ADC dynamic testing by spectral analysis. *Measurement* **2007**, *40*, 361–371. [\[CrossRef\]](#)
34. Cortés, C.; Mombello, E.; Dib, R.; Rattá, G. A new class of flat-top windows for exposure assessment in magnetic field measurements. *Signal Process.* **2007**, *87*, 2151–2164. [\[CrossRef\]](#)
35. Harris, F.J. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proc. IEEE* **1978**, *66*, 51–83. [\[CrossRef\]](#)
36. Okarma, K. Polynomial windows with low sidelobes' level. *Signal Process.* **2007**, *87*, 782–788. [\[CrossRef\]](#)
37. Li, H.; Yan, Q.; Ji, P. A New Strategy for Power System Harmonic Analysis Based on $N = L \times 10M$ -Point DFT. In Proceedings of the 2011 Asia-Pacific Power and Energy Engineering Conference, Wuhan, China, 25–28 March 2011; pp. 1–4.
38. Li, H.; Fei, R. An efficient algorithm of 10-point dft and its applications. In Proceedings of the 2010 International Conference on Electrical and Control Engineering, Wuhan, China, 25–27 June 2010; pp. 4642–4645.
39. Li, D.L.; Li, H.J. 10-Point DFT and its Application in Power System Harmonic Analysis. *Adv. Mater. Res.* **2012**, *433*, 7458–7464. [\[CrossRef\]](#)
40. Cariow, A. Strategies for the Synthesis of Fast Algorithms for the Computation of the Matrix-vector Products. *J. Signal Process. Theory Appl.* **2014**, *3*, 1–19. [\[CrossRef\]](#)
41. Lee, B.H.; Kuo, S. *Real-Time Digital Signal Processing: Implementations, Applications and Experiments with the TMS320C55X*; Wiley: Hoboken, NJ, USA, 2001.
42. *TMS320C54x DSP Reference Set, Volume 1: CPU and Peripherals*; Texas Instruments Inc.: Dallas, TX, USA, 2001.
43. Hicks, R.C. *Comparison of Arithmetic Requirements for the PFA (Prime Factor Algorithm), WFTA (Winograd Fourier Transform Algorithm), SWIFT, MFFT (Mixed Radix Fast Fourier Transform), FFT (Fast Fourier Transform) and DFT (Discrete Fourier Transform) Algorithms*; Technical Report; Army Missile Command Redstone Arsenal AI Advanced Sensors Directorate: Redstone Arsenal, AL, USA, 1982.
44. Burrus, C.S. *Fast Fourier Transforms (6×9 Version)*; Rice University: Houston, TX, USA, 2012.
45. Zheng, W.; Li, K. Split radix algorithm for length 6^m DFT. *IEEE Signal Process. Lett.* **2013**, *20*, 713–716. [\[CrossRef\]](#)
46. Wang, C.; Wang, N.; Lin, S.-J. Radix-2/6 and Radix-3/6 FFTs for a Length 6^m . In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 1130–1134. [\[CrossRef\]](#)
47. Li, H.; Wang, B.; Chen, X. An Efficient Algorithm for Computing $4M \times 24$ -Point DFT. In Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control, Dalian, China, 18–20 June 2008; p. 450.
48. Kolba, D.; Parks, T. A prime factor FFT algorithm using high-speed convolution. *IEEE Trans. Acoust. Speech Signal Process.* **1977**, *25*, 281–294. [\[CrossRef\]](#)
49. Li, H.; Yan, C.; Peng, W. Double factors algorithm for computing DFT. In Proceedings of the 2009 International Conference on Image Analysis and Signal Processing, Linhai, China, 11–12 April 2009; pp. 133–137.
50. Nawab, H.; McClellan, J. Bounds on the minimum number of data transfers in WFTA and FFT programs. *IEEE Trans. Acoust. Speech Signal Process.* **1979**, *27*, 394–398. [\[CrossRef\]](#)

-
51. Lu, I.D.; Lee, P. Use of mixed radix FFT in electric power system studies. *IEEE Trans. Power Deliv.* **1994**, *9*, 1276–1280. [[CrossRef](#)]
 52. Takahashi, D. An extended split-radix FFT algorithm. *IEEE Signal Process. Lett.* **2001**, *8*, 145–147. [[CrossRef](#)]