

A Biproportional Construction Algorithm for Correctly Calculating Fourier Series of Aperiodic Non-Sinusoidal Signal

Zicheng Li*, Mingwei Ren, Zhaoling Chen, Guohai Liu

School of Electrical and Information Engineering, Jiangsu University, Zhenjiang, China
Email: *lifengli9222@ujs.edu.cn

How to cite this paper: Li, Z.C., Ren, M.W., Chen, Z.L. and Liu, G.H. (2021) A Biproportional Construction Algorithm for Correctly Calculating Fourier Series of Aperiodic Non-Sinusoidal Signal. *Engineering*, 13, 503-525.
<https://doi.org/10.4236/eng.2021.1310036>

Received: August 31, 2021

Accepted: October 11, 2021

Published: October 14, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The Fourier series (FS) applies to a periodic non-sinusoidal function satisfying the Dirichlet conditions, whereas the being-processed function $f(t)$ in practical applications is usually an aperiodic non-sinusoidal signal. When $f(t)$ is aperiodic, its calculated FS is not correct, which is still a challenging problem. To overcome the problem, we derive a direct calculation algorithm, a constant iteration algorithm, and an optimal iteration algorithm. The direct calculation algorithm correctly calculates its Fourier coefficients (FCs) when $f(t)$ is periodic and satisfies the Dirichlet conditions. Both the constant iteration algorithm and the optimal iteration algorithm provide an idea of determining the states of $f(t)$. From the idea, we obtain an algorithm for determining the states of $f(t)$ based on the optimal iteration algorithm. In the algorithm, the variable iteration step is introduced; thus, we present an algorithm for determining the states of $f(t)$ based on the variable iteration step. The presented algorithm accurately determines the states of $f(t)$. On the basis of these algorithms, we build a biproportional construction theory. The theory consists of a first and a second proportional construction theory. The former correctly calculates the FCs of $f(t)$ at the present sampling time, and the latter creates a precondition for correctly calculating the FCs of $f(t)$ at the next sampling time. From the biproportional construction theory, we propose a biproportional construction algorithm. The proposed biproportional construction algorithm correctly calculates its FCs whether $f(t)$ is periodic or aperiodic, and thus its FS.

Keywords

Fourier Coefficients (FCs), Fourier Series (FS), Iteration Algorithm,

1. Introduction

The Fourier series (FS) was proposed by Jean Baptiste Joseph Fourier (1768-1830, a French mathematician and physicist) to solve the heat equation in a metal plate in 1807. The FS since then has been continually developed and perfected by many mathematicians, such as Dirichlet, Dini, Gibbs, Kolmogorov [1], Carleson [2], and Iserles [3]. The FS is a branch of Fourier analysis nowadays. The FS has been applied to almost all disciplines, including physics [4] [5], architectonics [6] [7], electronics [8] [9], electrical engineering [10] [11] [12], and signal processing [13] [14] [15].

The FS applies to a periodic non-sinusoidal function satisfying the Dirichlet conditions: The function must have a finite number of maxima, minima, and discontinuities in one period, and be absolutely integrable over a period. However, the being-processed function $f(t)$ in practical applications is usually an aperiodic non-sinusoidal signal defined as follows. $f(t)$ is aperiodic for $t \in (t_s, t_s + mT)$, and is T -periodic and satisfies the Dirichlet conditions for $t \in [0, t_s] \cup [t_s + mT, +\infty]$, where $t_s > 0$, t is time, and m is a sufficiently small positive integer like 1 or 2. For instance, $g(t)$ is illustrated in **Figure 1**, where $g(t)$ is aperiodic for $t \in (0.17, 0.21)$, and is periodic and satisfies the Dirichlet conditions for $t \in [0, 0.17] \cup [0.21, 0.34]$.

According to the FS, the FS of the being-processed function $f(t)$ can be expressed as

$$f(t) \sim a_0^* + \sum_{k=1}^{\infty} [a_k^* \cos(k\omega t) + b_k^* \sin(k\omega t)], \tag{1}$$

where

$$\begin{cases} a_0^* = \frac{1}{T} \int_0^T f(t) dt = \frac{1}{T} \int_{t_1-T}^{t_1} f(t) dt \\ a_k^* = \frac{2}{T} \int_0^T f(t) \cos(k\omega t) dt = \frac{2}{T} \int_{t_1-T}^{t_1} f(t) \cos(k\omega t) dt, \quad k = 1, 2, 3, \dots \end{cases} \tag{2}$$

and

$$b_k^* = \frac{2}{T} \int_0^T f(t) \sin(k\omega t) dt = \frac{2}{T} \int_{t_1-T}^{t_1} f(t) \sin(k\omega t) dt, \quad k = 1, 2, 3, \dots \tag{3}$$

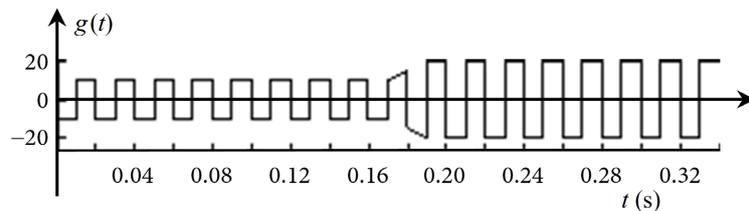


Figure 1. An aperiodic non-sinusoidal signal $g(t)$.

In Equations (1)-(3), a_0^* , a_k^* , and b_k^* are all called the Fourier coefficients (FCs) of $f(t)$; T is the period of $f(t)$, and $\omega = 2\pi/T$; and $t_1 > 0$.

From the FS, a precondition for Equation (1) is that $f(t)$ is a periodic non-sinusoidal function satisfying the Dirichlet conditions.

When $f(t)$ is periodic and satisfies the Dirichlet conditions, the precondition for Equation (1) is satisfied, and thus it is correct that $f(t)$ is expressed as Equation (1). Hence, a_k^* and b_k^* are correctly calculated by Equations (2) and (3), respectively, and thus its FS is correctly calculated by Equation (1). However, when $f(t)$ is aperiodic, the precondition for Equation (1) is not satisfied, and thus it is not correct that $f(t)$ is expressed as Equation (1). Therefore, a_k^* and b_k^* are not correctly calculated by Equations (2) and (3), respectively, and thus its FS is not correctly calculated by Equation (1). At present, how to calculate its FS correctly when $f(t)$ is aperiodic is still a problem necessary to solve.

The FS of $f(t)$ is correctly calculated if the FCs of $f(t)$ are. Therefore, numerous methods have been researched and reported as follows: many methods and algorithms for calculating the FCs [15]-[22], some estimates of the FCs [23] [24], a few methods for determining the FCs of the particular functions [25] [26], various formulae for computing the FCs of the special functions [27] [28] [29], and other methods. Each of these methods has its own applicability and merits, whereas how to calculate its FS correctly when $f(t)$ is aperiodic is still a challenging problem.

In practical applications, myriad signals in various disciplines (like the nonlinear load current in electrical engineering) can be considered as the being-processed function defined above. The being-processed function is denoted by $f(t)$ unless a specific description is given in this article. Although its FS is correctly calculated when $f(t)$ is periodic and satisfies the Dirichlet conditions, its FS is not correctly calculated when $f(t)$ is aperiodic. Nowadays, how to calculate its FS correctly when $f(t)$ is aperiodic is an important engineering calculation problem difficult to solve, and also an interesting mathematical-physical problem. The problem has to be solved because of numerous practical applications. To solve the problem, we will achieve the following main contributions:

- 1) Four basic concepts will be correctly defined.
- 2) A direct calculation algorithm, a constant iteration algorithm, and an optimal iteration algorithm will be derived.
- 3) A biproportional construction theory will be built, and then a biproportional construction algorithm will be proposed from the theory.
- 4) An algorithm for determining the states of $f(t)$ based on the variable iteration step will be presented.

The proposed biproportional construction algorithm correctly calculates its FCs whether $f(t)$ is periodic or aperiodic, and thus its FS. Therefore, the problem is solved. The rest of this article will be organized as follows. Section 2 will define four basic concepts and prove one theorem. In Sections 3-5, a direct

calculation algorithm, a constant iteration algorithm, and an optimal iteration algorithm will be derived, respectively. A biproportional construction theory, a biproportional construction algorithm, and an algorithm for determining the states of $f(t)$ based on the variable iteration step will be proposed in Section 6. The proposed biproportional construction algorithm will be validated by the simulation results and the experimental results. Finally, Section 7 will summarize the main conclusions.

2. Definitions and Theorem

It is assumed that $(3N - 2)$ samples of $f(t)$ and those of $\cos(k\omega t)$ are defined as $f(i) = f(t_1 + iT/(N - 1))$ and $\cos_k(i) = \cos(k\omega(t_1 + iT/(N - 1)))$, respectively, for $i = 0, 1, \dots, (3N - 3)$ in any three consecutive periods $[t_1, t_1 + 3T]$, where $t_1 > 0$.

Definition 2.1. If $f(t) = f(t - T)$ for all t in the period $[t_1, t_1 + T]$, then $f(t)$ changes periodically in the period. That is, $f(t)$ is in stable states in the period. Thus, $a_k^* = \frac{2}{T} \int_{t_1 - T}^{t_1} f(t) \cos(k\omega t) dt = \frac{2}{T} \int_{t_1}^{t_1 + T} f(t) \cos(k\omega t) dt$ in the period.

Definition 2.2. If $f(t) \neq f(t - T)$ for any t in the period $[t_1, t_1 + T]$, then $f(t)$ changes nonperiodically in the period. That is, $f(t)$ is in unstable states in the period.

Definition 2.3. If $f(t)$ is in stable states in the period $[t_1, t_1 + T]$ and if $f(N) = f(1)$, then $f(t)$ changes periodically at the sampling time $t = t_1 + NT/(N - 1)$. That is, $f(t)$ is in a stable state at the sampling time. Thus, $a_k^{(N)*} = \frac{2}{T} \int_{t_1 - T}^{t_1} f(t) \cos(k\omega t) dt = \frac{2}{T} \int_{t_1}^{t_1 + T} f(t) \cos(k\omega t) dt$, where $a_k^{(N)*}$ denotes a_k^* of $f(t)$ at the sampling time.

Definition 2.4. If $f(t)$ is in stable states in the period $[t_1, t_1 + T]$ and if $f(N) \neq f(1)$, then $f(t)$ changes nonperiodically at the sampling time $t = t_1 + NT/(N - 1)$. That is, $f(t)$ is in an unstable state at the sampling time. Thus, $a_k^{(N)*} = \frac{2f(N)}{Tf(1)} \int_{t_1 - T}^{t_1} f(t) \cos(k\omega t) dt = \frac{2f(N)}{Tf(1)} \int_{t_1}^{t_1 + T} f(t) \cos(k\omega t) dt$.

Definitions 2.1-2.3 are correctly defined from the FS of $f(t)$. Besides, $f(t)$ is in stable states in the period $[t_1, t_1 + T]$ and $f(N) \neq f(1)$. Thus, we infer that $f(t)$ should change in the same proportion $f(N)/f(1)$ for all t in the period $[t_1 + NT/(N - 1), t_1 + NT/(N - 1) + T]$. For this reason, Definition 2.4 is defined above.

Theorem 2.1. If $f(t)$ is a periodic non-sinusoidal function satisfying the Dirichlet conditions, then $\int_{t_1 - T}^{t_1} [f(t) - a_k \cos(k\omega t)]^2 dt$ attains its absolute minimum at $a_k = a_k^*$ for $k = 0, 1, 2, \dots$, where T is the period of $f(t)$ and $\omega = 2\pi/T$.

Proof. It is assumed that $h(a_k) = \int_{t_1 - T}^{t_1} [f(t) - a_k \cos(k\omega t)]^2 dt$. Then, it follows from $h'(a_k) = 0$ and Equation (2) that

$$\begin{cases} a_0 = \frac{1}{T} \int_{t_1-T}^{t_1} f(t) dt = \frac{1}{T} \int_0^T f(t) dt = a_0^*, \\ a_k = \frac{2}{T} \int_{t_1-T}^{t_1} f(t) \cos(k\omega t) dt = \frac{2}{T} \int_0^T f(t) \cos(k\omega t) dt = a_k^*, \quad k = 1, 2, 3, \dots \end{cases}$$

Moreover, $h''(a_0^*) = 2T > 0$ and $h''(a_k^*) = T > 0$ for $k = 0, 1, 2, \dots$. Therefore, $h(a_k)$ attains its absolute minimum at $a_k = a_k^*$ for $k = 0, 1, 2, \dots$. Thus, Theorem 2.1 is proved.

3. Direct Calculation Algorithm

3.1. Direct Calculation Algorithm

It is assumed that $p(a_k^{(N-1)}) = \sum_{i=0}^{N-1} [f(i) - a_k^{(N-1)} \cos_k(i)]^2$. Then, it is obtained from $p'(a_k^{(N-1)}) = 0$ that

$$a_k^{(N-1)} = \frac{\sum_{i=0}^{N-1} [f(i) \cos_k(i)]}{\sum_{i=0}^{N-1} [\cos_k(i)]^2}, \quad k = 0, 1, 2, \dots \tag{4}$$

At the sampling time $t = t_1 + T$, $a_k^{(N-1)}$ is calculated by Equation (4), and then $a_k^* \cos(k\omega t) = a_k^{(N-1)} \cos_k(N-1)$. Similarly, at the next sampling time, $a_k^* \cos(k\omega t) = a_k^{(N)} \cos_k(N)$ This algorithm is designated as a direct calculation algorithm for calculating $a_k^* \cos(k\omega t)$ because a_k is directly calculated by Equation (4).

Similarly, a direct calculation algorithm for calculating $b_k^* \sin(k\omega t)$ can be derived. The direct calculation algorithm for calculating $a_k^* \cos(k\omega t)$ and that for calculating $b_k^* \sin(k\omega t)$ are collectively designated as a direct calculation algorithm for calculating the FS of $f(t)$, or a direct calculation algorithm for short.

3.2. Calculation Complexity Analysis

Although the FS of $f(t)$ consists of infinite terms, it is only necessary to calculate one or a few of the infinite terms in practical applications. Thus, taking calculating the term $a_1^* \cos(\omega t)$ as an example, we conduct the calculation complexity analysis, calculation accuracy analysis, simulation, and experiment.

From the direct calculation algorithm for calculating $a_1^* \cos(\omega t)$, it is appropriate to utilize a DSP (digital signal processing) to implement it. Its calculation complexity is much reduced by using *space in exchange for time*. The block diagram of the memory cells and calculation process of this algorithm is presented in **Figure 2**. At the beginning, N consecutive memory cells from 0 to $(N-1)$, having the same number of words, store $f(i) \cos_1(i)$ for $i = 0, 1, \dots, (N-1)$, respectively. A first product pointer points to Memory cell 0. $tot_{N-1} = \sum_{i=0}^{N-2} [f(i) \cos_1(i)]$, and $tot_N = \sum_{i=0}^{N-1} [f(i) \cos_1(i)]$. The calculation process of this algorithm is as follows. When the present samples $f(p)$ and $\cos_1(p)$ arrive, $f(p) \cos_1(p)$ is obtained by multiplying $\cos_1(p)$ by $f(p)$, tot_N by

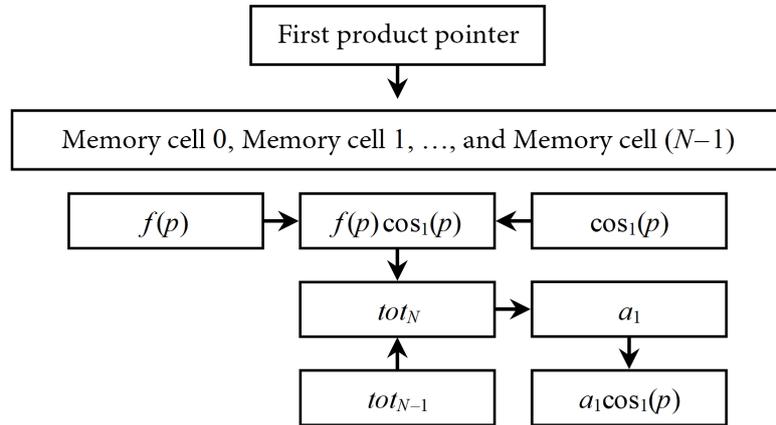


Figure 2. The block diagram of the memory cells and calculation process of the direct calculation algorithm for calculating $a_1^* \cos(\omega t)$.

adding tot_{N-1} and $f(p)\cos_1(p)$, a_1 by dividing tot_N by $\sum_{i=0}^{N-1} [\cos_1(i)]^2$, and $a_1 \cos_1(p)$ by multiplying $\cos_1(p)$ by a_1 . Finally, tot_{N-1} is obtained by subtracting the product, stored in the memory cell being pointed to by the first product pointer, from tot_N . Then, $f(p)\cos_1(p)$ is stored in the memory cell, and the pointer points to the next memory cell. If the pointer is pointing to Memory cell N , then it points to Memory cell 0 immediately. Moreover, $\sum_{i=0}^{N-1} [\cos_1(i)]^2$ is a constant because N is. Thus, its calculation complexity per iteration is one addition operation, one subtraction operation, two multiplication operations, and one division operation. Therefore, it is very low and does not change with N .

3.3. Calculation Accuracy Analysis

When $f(t)$ changes periodically, the precondition for Equation (1) is satisfied. Thus, the larger N , the higher the accuracy of the calculated a_1 , according to the direct calculation algorithm for calculating $a_1^* \cos(\omega t)$, the FS of $f(t)$, and the sampling theorem. For this reason, the calculated a_1 is correct within an allowable error because N can be large enough because of the very low calculation complexity of this algorithm and the very high calculation speed of existing DSPs.

When $f(t)$ changes nonperiodically, the precondition for Equation (1) is not satisfied. Thus, the calculated a_1 is not correct, and its accuracy depends on the tracking performance of this algorithm. From the following simulation results and experimental results, the calculated a_1 tracks a_1^* smoothly.

3.4. Simulation Results and Experimental Results

The experimental system block diagram is shown in **Figure 3**, where, the system consists of a zero-crossing comparator, an UA206, and a computer.

The zero-crossing comparator is implemented by an operational amplifier

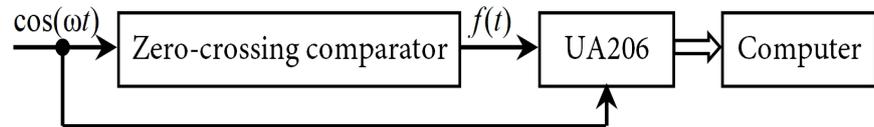


Figure 3. The experimental system block diagram.

OP07, its input is $\cos(\omega t)$, and its output is a square-wave signal acting as $f(t)$. The UA206 is an A/D data acquisition board, developed by a company in Beijing; and it has sixteen 16-bit A/D converters. Its highest sampling frequency is 500 kHz, and its analog input signals range up to ± 10 V. Moreover, a VB source code framework demonstrating that signals are sampled and displayed is provided by the company. Thus, from the framework and the direct calculation algorithm for calculating $a_1^* \cos(\omega t)$, the experimental program is easily designed. When the experiments are done, the two analog signals $f(t)$ and $\cos(\omega t)$ are the inputs of the UA206 and are converted into two digital signals by two of the A/D converters. Then, the two digital signals are sent to the computer via its PCI port. When the experimental program is executed, $f(t)$, $\cos(\omega t)$, the calculated a_1 , and the computed $a_1 \cos(\omega t)$ are displayed on the computer screen. Thus, it is convenient to observe and record the experimental waveforms.

To validate the direct calculation algorithm for calculating $a_1^* \cos(\omega t)$, the simulations and the experiments are carried out. The simulation results are depicted in **Table 1** and **Figure 4**, and the experimental results in **Figure 5**. In

Table 1, $f(t) = \begin{cases} -10, & 5 \leq t < 15 \\ +10, & 15 \leq t < 25 \end{cases}$ and $f(t+20) = f(t)$; and a_1 and a_1^* are

calculated by Equations (4) and (2), respectively. **Table 1** indicates that the calculated a_1 is correct within an allowable error when $f(t)$ changes periodically and when N is large enough. In **Figure 4**, MATLAB 7.0 is applied, $N = 500$, and $T = 0.02$ s. In **Figure 5**, $N = 200$ and $T = 0.02$ s. **Figure 4** and **Figure 5** indicate that the calculated a_1 is a constant and thus correct, when $f(t)$ changes periodically. The calculated a_1 tracks a_1^* smoothly and thus is not correct, and all response times are 0.02 s (one period), when $f(t)$ changes nonperiodically. Particularly, the calculated a_1 has a considerably large error when $f(t)$ step changes. In addition, the further experiments indicate that this algorithm is stable and reliable, and its calculation results are very slightly affected by small changes in the frequency of $f(t)$.

3.5. Brief Summary

We have derived a direct calculation algorithm. This algorithm correctly calculates its FCs when $f(t)$ changes periodically, but not when $f(t)$ changes nonperiodically. Particularly, this algorithm has a considerably large calculation error when $f(t)$ step changes. Thus, it is necessary to find a method for correctly calculating its FCs, when $f(t)$ changes nonperiodically.

Table 1. The datasheet of N , a_1 , and a_1^* .

N	100	500	1000	5000
a_1	12.728206	12.732228	12.732354	12.732394
a_1^*	12.732395	12.732395	12.732395	12.732395

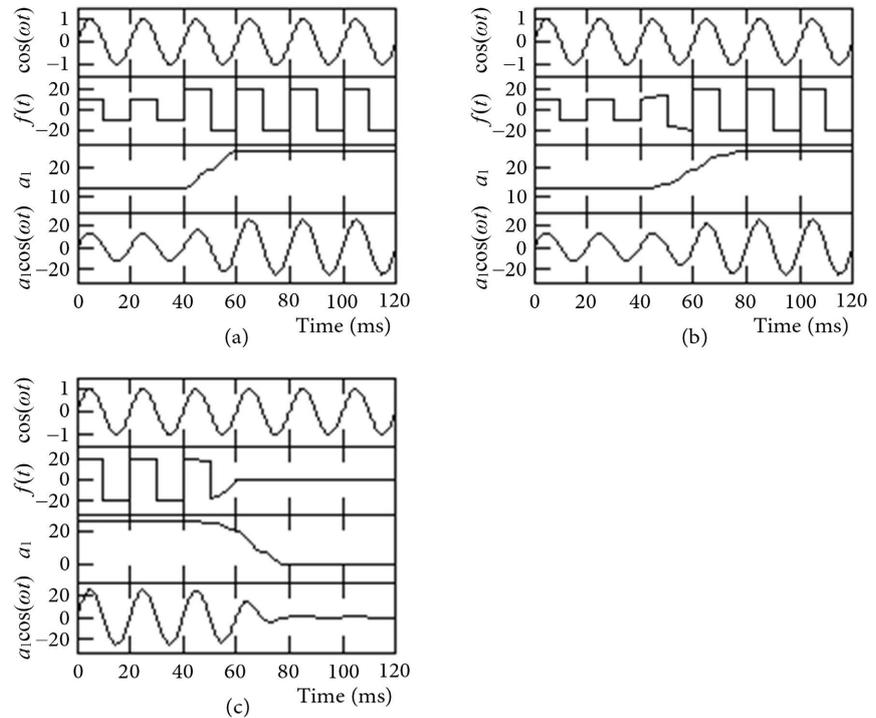


Figure 4. (a)-(c) The simulation results of the direct calculation algorithm for calculating $a_1^* \cos(\omega t)$ with a step increase, a linear increase, and an exponential decrease, respectively, in $f(t)$.

4. Constant Iteration Algorithm

4.1. Constant Iteration Algorithm

It is assumed that $a_k^{(N-1)}$ denoting $a_k^{(N-1)*}$, a_k^* of $f(t)$ at the sampling time $t = t_1 + T$, has been calculated iteratively; and that $a_k^{(N)}$ denoting $a_k^{(N)*}$, a_k^* of $f(t)$ at the next sampling time, will be calculated iteratively. On the basis of $a_k^{(N-1)}$, $a_k^{(N)}$ will approach $a_k^{(N)*}$ when $f(t)$ changes periodically, and track when $f(t)$ changes nonperiodically. However, whether $a_k^{(N-1)} \geq a_k^{(N)*}$ cannot be determined; thus, a tentative comparison is adopted to determine the iterative calculation direction according to Theorem 2.1.

It is assumed that h_k represents the iteration step. Then, it follows from Theorem 2.1 that

$$A_{k0} = \sum_{i=1}^N [f(i) - a_k^{(N-1)} \cos_k(i)]^2, \tag{5}$$

$$A_{k1} = \sum_{i=1}^N \left\{ f(i) - [a_k^{(N-1)} + h_k] \cos_k(i) \right\}^2, \tag{6}$$

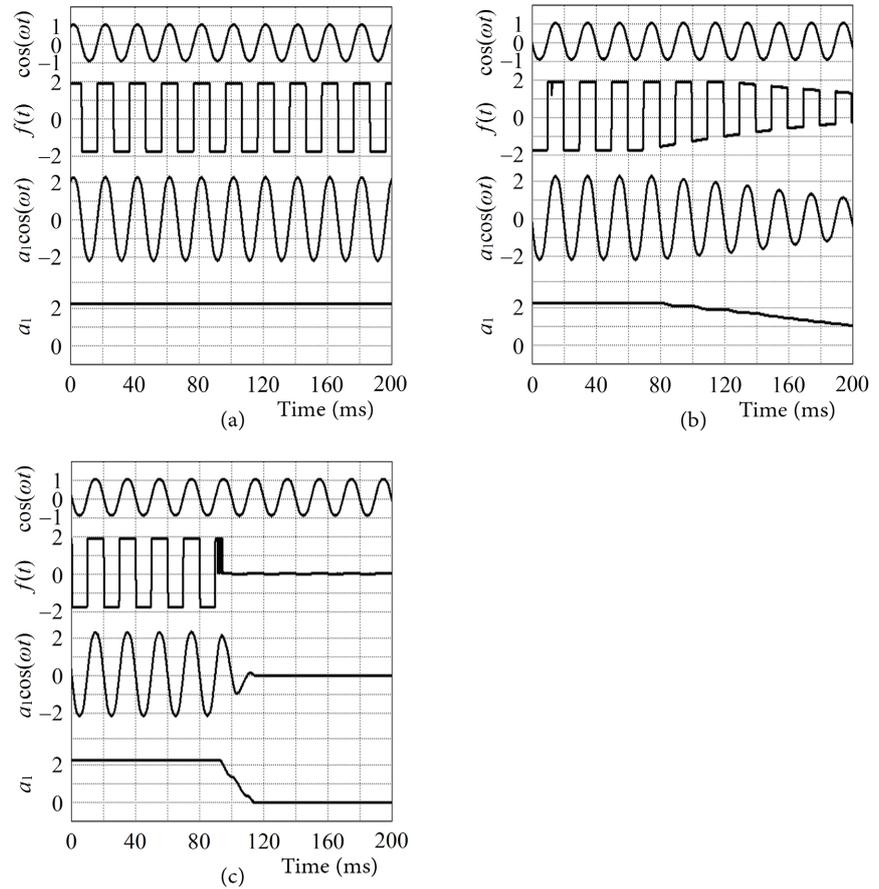


Figure 5. (a)-(c) The experimental results of the direct calculation algorithm for calculating $a_k^* \cos(\omega t)$ with no change, a slow decrease, and a step decrease, respectively, in $f(t)$.

and

$$A_{k2} = \sum_{i=1}^N \left\{ f(i) - [a_k^{(N-1)} - h_k] \cos_k(i) \right\}^2. \tag{7}$$

From Equations (5)-(7), it is obtained that

$$\begin{aligned} A_{k12} = A_{k1} - A_{k2} &= -4h_k \sum_{i=1}^N [f(i) \cos_k(i)] + 4h_k a_k^{(N-1)} \sum_{i=1}^N [\cos_k(i)]^2 \\ &= -4h_k \sum_{i=1}^N [f(i) \cos_k(i)] + 4h_k a_k^{(N-1)} C_{SS} = 4h_k A_{k00}, \end{aligned} \tag{8}$$

$$A_{k10} = A_{k1} - A_{k0} = 2h_k A_{k00} + h_k^2 C_{SS}, \tag{9}$$

and

$$A_{k20} = A_{k2} - A_{k0} = -2h_k A_{k00} + h_k^2 C_{SS}. \tag{10}$$

In Equations (8)-(10),

$$C_{SS} = \sum_{i=1}^N [\cos_k(i)]^2 \tag{11}$$

and

$$A_{k00} = -\sum_{i=1}^N [f(i) \cos_k(i)] + a_k^{(N-1)} C_{SS}. \tag{12}$$

According to Equations (8), (11), and (12), A_{k00} and A_{k12} have the same sign; and from the signs of A_{k00} , A_{k10} , and A_{k20} , the absolute minimum is determined. Thus, from Theorem 2.1, an algorithm for calculating $a_k^{(N)}$ is obtained and illustrated in **Figure 6**.

At the sampling time $t = t_1 + NT/(N - 1)$, $a_k^{(N)}$ is calculated by the algorithm illustrated in **Figure 6**, and then $a_k^* \cos(k\omega t) = a_k^{(N)} \cos_k(N)$. Similarly, at the next sampling time, $a_k^* \cos(k\omega t) = a_k^{(N+1)} \cos_k(N + 1)$ This algorithm is designated as a constant iteration algorithm for calculating $a_k^* \cos(k\omega t)$ because it has the constant iteration step h_k .

Similarly, a constant iteration algorithm for calculating $b_k^* \sin(k\omega t)$ can be derived. The constant iteration algorithm for calculating $a_k^* \cos(k\omega t)$ and that for calculating $b_k^* \sin(k\omega t)$ are together designated as a constant iteration algorithm for calculating the FS of $f(t)$, or a constant iteration algorithm for short.

4.2. Calculation Complexity Analysis

From the constant iteration algorithm for calculating $a_1^* \cos(\omega t)$, its calculation complexity depends mainly on calculating A_{100} and A_{110} , or A_{100} and A_{120} . The calculation complexity of A_{110} is the same as that of A_{120} . Moreover, $2h_1$, $-2h_1$, $\sum_{i=1}^N [\cos_1(i)]^2$, and $h_1^2 \sum_{i=1}^N [\cos_1(i)]^2$ are constants because h_1 and N are. Thus, similarly, from the calculation complexity analysis of the direct calculation algorithm for calculating $a_1^* \cos(\omega t)$, the calculation complexity of A_{100} is one addition operation, one subtraction operation, and two multiplication operations. That of A_{110} is one addition operation and one multiplication operation. Thus, its main calculation complexity per iteration is two addition operations, one subtraction operation, and three multiplication operations. Therefore, it is very low and does not change with N .

4.3. Calculation Accuracy Analysis

When $f(t)$ changes periodically, the precondition for Equation (1) is satisfied.

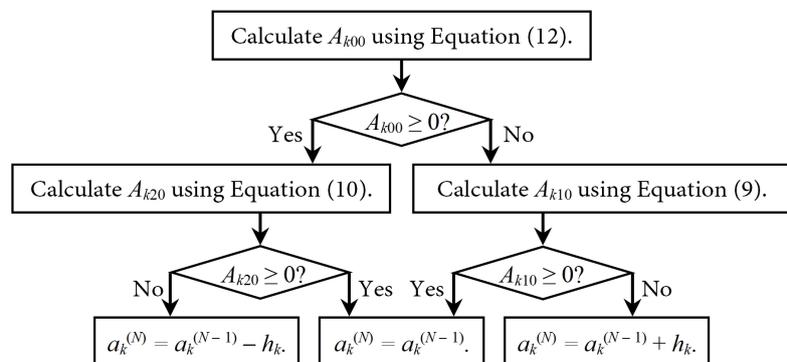


Figure 6. The constant iteration algorithm for calculation $a_k^{(N)}$.

Thus, the larger N and the smaller h_1 , the higher the accuracy of the calculated a_1 , according to the constant iteration algorithm for calculating $a_1^* \cos(\omega t)$, the FS of $f(t)$, and the sampling theorem. For this reason, the calculated a_1 is correct within an allowable error because h_1 can be small enough and N large enough because of the very low calculation complexity of this algorithm and the very high calculation speed of existing DSPs.

When $f(t)$ changes nonperiodically, the precondition for Equation (1) is not satisfied. Thus, the calculated a_1 is not correct, and its accuracy depends on the tracking performance of this algorithm. From the following simulation results, the calculated a_1 tracks a_1^* smoothly.

4.4. Simulation Results

To validate the constant iteration algorithm for calculating $a_1^* \cos(\omega t)$, the simulations are done, and the simulation results are presented in **Figure 7**. In **Figure 7**, $N = 500$, $T = 0.02$ s, and $h_1 = 0.025$. **Figure 7** indicates that the calculated a_1 is a constant and thus correct, when $f(t)$ changes periodically. The calculated a_1 tracks a_1^* smoothly and thus is not correct, and all response times are about 0.02 s (one period), when $f(t)$ changes nonperiodically. Besides, h_1 gravely affects the calculation performance of this algorithm, and should be selected appropriately.

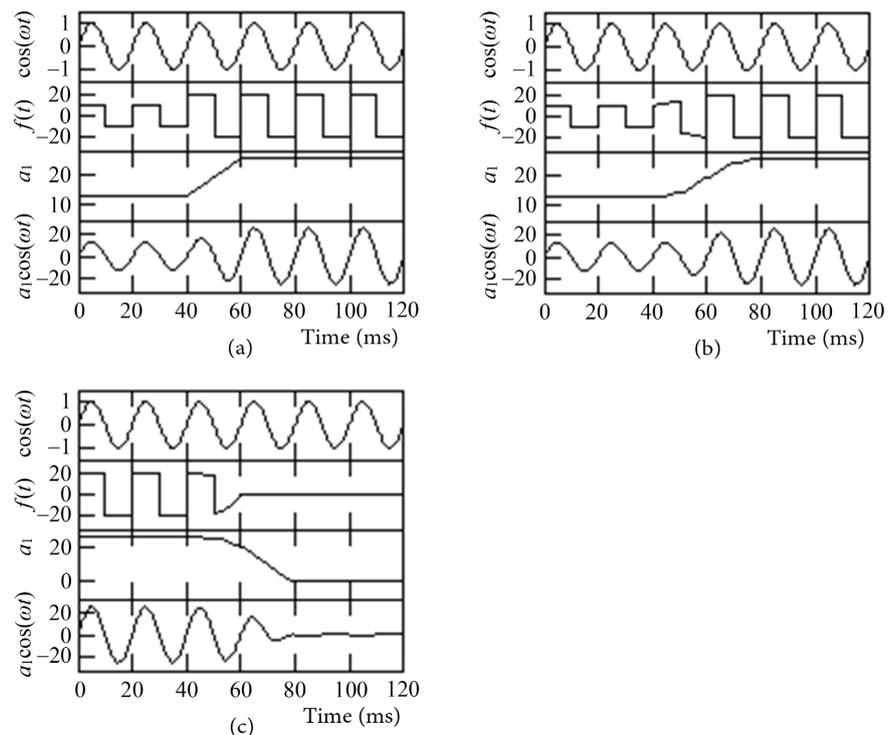


Figure 7. (a)-(c) The simulation results of the constant iteration algorithm for calculating $a_1^* \cos(\omega t)$ with a step increase, a linear increase, and an exponential decrease, respectively, in $f(t)$.

4.5. Brief Summary

We have derived a constant iteration algorithm. This algorithm has the constant iteration step seriously affecting its calculation performance. Therefore, the constant iteration step should be neither too large nor too small, and be selected appropriately. This algorithm does not perform as well as the direct calculation algorithm does; however, it provides an idea of determining the states of $f(t)$. The idea may have significant applications in further research.

5. Optimal Iteration Algorithm

5.1. Optimal Iteration Algorithm

The iteration step of the constant iteration algorithm for calculating $a_k^* \cos(k\omega t)$ is the constant h_k , and thus cannot accurately reflect the actual conditions. Under the actual conditions, the iteration step should be large when $|a_k^{(N-1)} - a_k^{(N)*}|$ is large, and small when $|a_k^{(N-1)} - a_k^{(N)*}|$ is small. Then, how do we determine the optimal iteration step h_{k-opt} ?

In **Figure 6**, if $A_{k10} < 0$, then $a_k^{(N)} = a_k^{(N-1)} + h_k$; and A_{k10} can be regarded as a function of h_k , according to Equation (9). If it is assumed that $q(h_k) = A_{k10} = 2h_k A_{k00} + h_k^2 C_{ss}$, then $q(h_k)$ attains its absolute minimum at $h_k = -A_{k00}/C_{ss}$. Thus, the optimal iteration step is $h_{k-opt} = -A_{k00}/C_{ss}$.

Similarly, the optimal iteration step is $h_{k-opt} = A_{k00}/C_{ss}$, according to Equation (10).

Thus, from **Figure 6**, an algorithm for calculating $a_k^{(N)}$ is obtained and depicted in **Figure 8**.

At the sampling time $t = t_1 + NT/(N - 1)$, $a_k^{(N)}$ is calculated by the algorithm depicted in **Figure 8**, and then $a_k^* \cos(k\omega t) = a_k^{(N)} \cos_k(N)$. Similarly, at the next sampling time, $a_k^* \cos(k\omega t) = a_k^{(N+1)} \cos_k(N + 1)$ This algorithm is designated as an optimal iteration algorithm for calculating $a_k^* \cos(k\omega t)$ because it has the optimal iteration step h_{k-opt} .

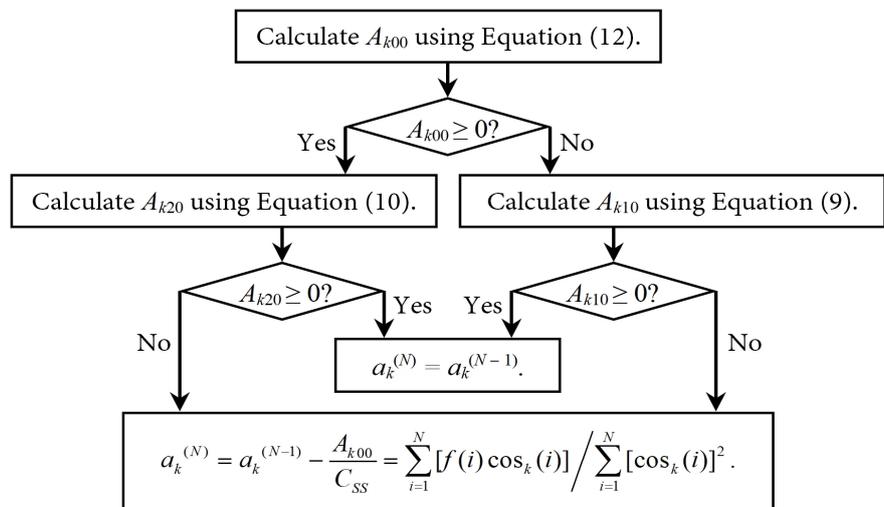


Figure 8. The optimal iteration algorithm for calculating $a_k^{(N)}$.

Similarly, an optimal iteration algorithm for calculating $b_k^* \sin(k\omega t)$ can be derived. The optimal iteration algorithm for calculating $a_k^* \cos(k\omega t)$ and that for calculating $b_k^* \sin(k\omega t)$ are collectively designated as an optimal iteration algorithm for calculating the FS of $f(t)$, or an optimal iteration algorithm for short.

5.2. Brief Summary

We have derived an optimal iteration algorithm. This algorithm is essentially consistent with the direct calculation algorithm. It has the optimal iteration step, and thus performs better than the constant iteration algorithm does. Besides, this algorithm, like the constant iteration algorithm, provides an idea of determining the states of $f(t)$. The idea may be important in discovering an algorithm for determining the states of $f(t)$.

6. Biproportional Construction Algorithm

6.1. Biproportional Construction Theory

If $f(t)$ is in stable states in the period $[t_1, t_1 + T]$, then $a_k^{(N-1)}$ calculated by Equation (4) equals $a_k^{(N-1)*}$ within an allowable error when N is large enough, according to Definition 2.1.

If $f(t)$ is in a stable state at the sampling time $t = t_1 + NT/(N - 1)$: $f(N) = f(1)$, then it follows from Equation (4) that

$$a_k^{(N)} = \frac{\sum_{i=1}^N [f(i) \cos_k(i)]}{\sum_{i=1}^N [\cos_k(i)]^2}. \tag{13}$$

From Definition 2.3, $a_k^{(N)} = a_k^{(N)*}$ within an allowable error when N is large enough. On the contrary, if $f(t)$ is in an unstable state at the sampling time $t = t_1 + NT/(N - 1)$: $f(N) \neq f(1)$, then $a_k^{(N)} = a_k^{(N)*}$ cannot be ensured within an allowable error even though N is large enough. However, $f_1(t)$ is constructed from $f(1)$, $f(N)$, and $f(t)$ in stable states in the period $[t_1, t_1 + T]$. The constructed $f_1(t)$ in the period satisfies

$$f_1(t)/f(t) = f(N)/f(1) = \lambda, \quad f(1) \neq 0. \tag{14}$$

That is, $f_1(t)$ is proportional to $f(t)$ in the period. If it is assumed that N discrete values of $f_1(t)$ are defined as $f_1(i) = f_1(t_1 + iT/(N - 1))$, then $f_1(i) = \lambda f(i)$ from Equation (14), for $i = 0, 1, \dots, (N - 1)$. Thus, it is obtained from Equation (4) that

$$a_k^{(N)'} = \frac{\sum_{i=0}^{N-1} [f_1(i) \cos_k(i)]}{\sum_{i=0}^{N-1} [\cos_k(i)]^2} = \lambda \frac{\sum_{i=0}^{N-1} [f(i) \cos_k(i)]}{\sum_{i=0}^{N-1} [\cos_k(i)]^2} = \lambda a_k^{(N-1)}. \tag{15}$$

According to Equation (15) and Definition 2.4, $a_k^{(N)'} = a_k^{(N)*}$ within an allowable error when N is large enough. This discovery is designated as a first proportional construction theory discovered with the aims of calculating $a_k^{(N)'} and ensuring that $a_k^{(N)'} = a_k^{(N)*}$ within an allowable error when N is large enough.$

At the sampling time $t = t_1 + NT/(N - 1)$, the precondition for the first proportional construction theory correctly calculating $a_k^{(N)^\prime}$ is that $f(t)$ is in stable states in the period $[t_1, t_1 + T]$. Here, the term “stable states” mean that $f(t)$ is in “stable states” in the period if $a_k^{(N-1)}$ calculated by Equation (4) equals $a_k^{(N-1)*}$ within an allowable error when N is large enough.

When the next sample $f(N + 1)$ arrives, $f(t)$ is in unstable states in the period $[t_1 + T/(N - 1), t_1 + NT/(N - 1)]$. Thus, the precondition for correctly calculating $a_k^{(N+1)*}$, a_k^* of $f(t)$ at the sampling time $t = t_1 + (N + 1)T/(N - 1)$, does not exist. For this reason, although $a_k^{(N+1)^\prime}$ denoting $a_k^{(N+1)*}$ can be calculated similarly to $a_k^{(N)^\prime}$, $a_k^{(N+1)^\prime} = a_k^{(N+1)*}$ cannot be ensured within an allowable error even though N is large enough. Nevertheless, $f_2(t)$ is constructed from $a_k^{(N)}$, $a_k^{(N)^\prime}$, and $f(t)$ in unstable states in the period $[t_1 + T/(N - 1), t_1 + NT/(N - 1)]$. The constructed $f_2(t)$ in the period fulfils

$$f_2(t)/f(t) = a_k^{(N)^\prime}/a_k^{(N)} = \eta. \tag{16}$$

That is, $f_2(t)$ is proportional to $f(t)$ in the period. If it is assumed that N discrete values of $f_2(t)$ are defined as $f_2(i) = f_2(t_1 + iT/(N - 1))$, then $f_2(i) = \eta f(i)$ from Equation (16), for $i = 1, 2, \dots, N$. Thus, from Equations (4), (13), and (16), and the first proportional construction theory, it is deduced that

$$a_k^{(N)^\prime} = \frac{\sum_{i=1}^N [f_2(i) \cos_k(i)]}{\sum_{i=1}^N [\cos_k(i)]^2} = \frac{\eta \sum_{i=1}^N [f(i) \cos_k(i)]}{\sum_{i=1}^N [\cos_k(i)]^2} = \eta a_k^{(N)} = a_k^{(N)^\prime} = a_k^{(N)*}. \tag{17}$$

From Equation (17), $f_2(t)$ is in stable states in the period $[t_1 + T/(N - 1), t_1 + NT/(N - 1)]$. Therefore, the period is considered as the period in front of $f(N + 1)$, and thus a precondition for correctly calculating $a_k^{(N+1)*}$ is created. This discovery is designated as a second proportional construction theory discovered with the goal of creating the precondition.

The first and the second proportional construction theory are together designated as a biproportional construction theory for calculating a_k^* , which correctly calculates a_k^* even though $f(t)$ is in an unstable state.

Similarly, a biproportional construction theory for calculating b_k^* can be obtained, which correctly calculates b_k^* even though $f(t)$ is in an unstable state. The biproportional construction theory for calculating a_k^* and that for calculating b_k^* are collectively designated as a biproportional construction theory for calculating the FCs of $f(t)$, or a biproportional construction theory for short. The biproportional construction theory correctly calculates its FCs even though $f(t)$ is in an unstable state.

6.2. Biproportional Construction Algorithm

It is assumed that $f_d(i)$ corresponding to $f(i)$ represent the data processed by the biproportional construction theory for calculating a_k^* for

$i = 1, 2, \dots, N$; and that the calculated $a_k^{(N-1)'}$ equals $a_k^{(N-1)*}$ within an allowable error at the sampling time $t = t_1 + T$.

The direct calculation algorithm for calculating $a_k^* \cos(k\omega t)$ is used to calculate a_k^* accurately when $f(t)$ changes periodically, and the biproportional construction theory for calculating a_k^* when $f(t)$ changes nonperiodically. Therefore, an algorithm for calculating $a_k^{(N)'}$ is obtained and shown in **Figure 9**.

At the sampling time $t = t_1 + NT/(N-1)$, $a_k^{(N)'}$ is calculated by the algorithm shown in **Figure 9**, and then $a_k^* \cos(k\omega t) = a_k^{(N)'}$ $\cos_k(N)$. Similarly, at the next sampling time, $a_k^* \cos(k\omega t) = a_k^{(N+1)'}$ $\cos_k(N+1)$ This algorithm is designated as a biproportional construction algorithm for calculating $a_k^* \cos(k\omega t)$ because it is based on the biproportional construction theory for calculating a_k^* .

Similarly, a biproportional construction algorithm for calculating $b_k^* \sin(k\omega t)$ can be obtained. The biproportional construction algorithm for calculating $a_k^* \cos(k\omega t)$ and that for calculating $b_k^* \sin(k\omega t)$ are together designated as a biproportional construction algorithm for calculating the FS of $f(t)$, or a biproportional construction algorithm for short.

6.3. Algorithm for Determining the States of $f(t)$

From the biproportional construction algorithm for calculating $a_k^* \cos(k\omega t)$, it

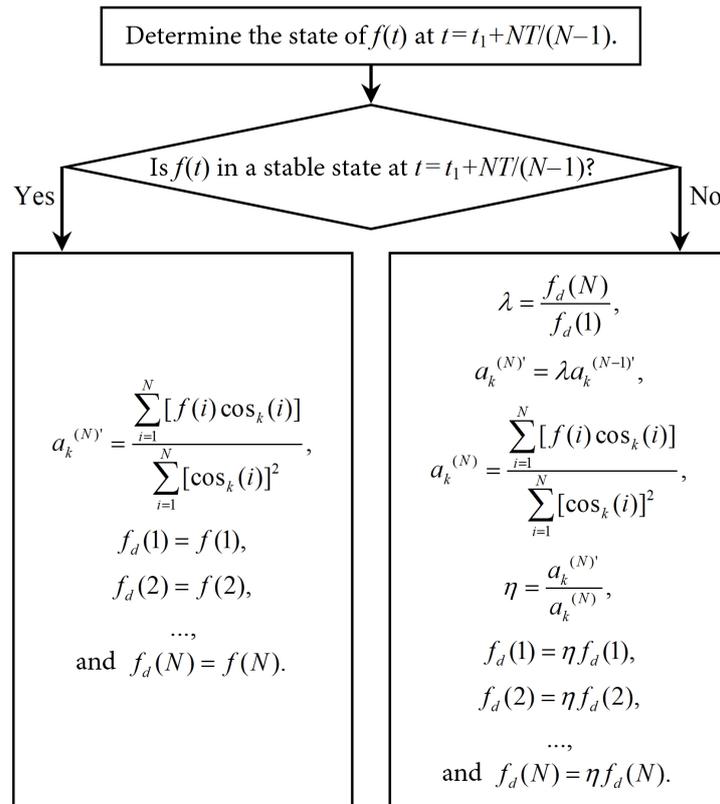


Figure 9. The biproportional construction algorithm for calculating $a_k^{(N)'}$.

is required to determine the states of $f(t)$ accurately. Thus, how to determine the states of $f(t)$ accurately is certainly a key problem that has to be solved.

In the optimal iteration algorithm for calculating $a_k^* \cos(k\omega t)$, at the sampling time $t = t_1 + NT/(N - 1)$, $f(t)$ may be in a stable state when $a_k^{(N)} = a_k^{(N-1)}$, and must be in an unstable state when

$$a_k^{(N)} = \frac{\sum_{i=1}^N [f(i) \cos_k(i)]}{\sum_{i=1}^N [\cos_k(i)]^2}. \text{ Thus, this algorithm provides an idea}$$

of determining the states of $f(t)$. From the idea, an algorithm for determining the states of $f(t)$ based on the optimal iteration algorithm is obtained; and the algorithm for determining the state of $f(t)$ at the sampling time $t = t_1 + NT/(N - 1)$ is illustrated in **Figure 10**.

The constant h_k is the iteration step of the algorithm, and thus must seriously affect its determination accuracy. For this reason, h_k should be selected appropriately to determine the states of $f(t)$ accurately, whereas it is very difficult to do so. Therefore, it is necessary to find the variable iteration step that can help in accurately determining the states of $f(t)$ and that can change with $f(t)$.

The computation formulae of $a_k^{(N-1)}$, $a_k^{(N)}$, A_{k0} , A_{k1} , and A_{k2} are Equations (4), (13), (5), (6) and (7), respectively. If $f(t)$ is in a stable state at the sampling time $t = t_1 + NT/(N - 1)$, then $A_{k0} \leq A_{k1}$ and $A_{k0} \leq A_{k2}$. From $A_{k0} \leq A_{k1}$ and $A_{k0} \leq A_{k2}$, it follows that

$$h_k \geq 2[a_k^{(N)} - a_k^{(N-1)}] \tag{18}$$

and

$$h_k \geq 2[a_k^{(N-1)} - a_k^{(N)}], \tag{19}$$

respectively. If $f(t)$ is in an unstable state at the sampling time $t = t_1 + NT/(N - 1)$, then $A_{k0} \geq A_{k1}$ or $A_{k0} \geq A_{k2}$. From $A_{k0} \geq A_{k1}$ and $A_{k0} \geq A_{k2}$, it is obtained that

$$h_k \leq 2[a_k^{(N)} - a_k^{(N-1)}] \tag{20}$$

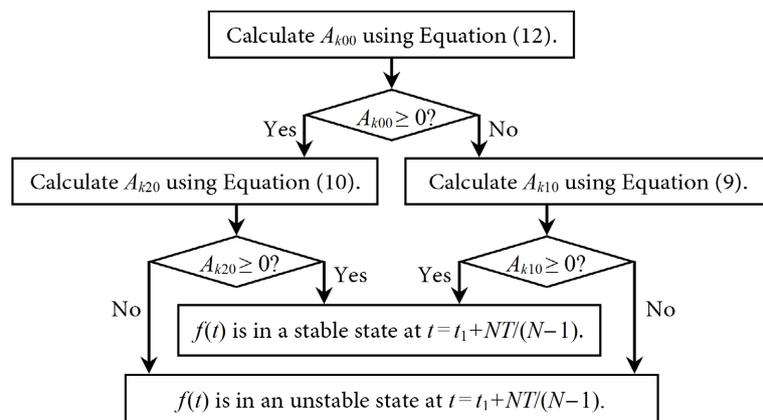


Figure 10. The algorithm for determining the state of $f(t)$ at the sampling time $t = t_1 + NT/(N - 1)$ based on the optimal iteration algorithm.

and

$$h_k \leq 2 \left[a_k^{(N-1)} - a_k^{(N)} \right], \tag{21}$$

respectively.

From inequalities (18)-(21), $h_k = 2 \left| a_k^{(N)} - a_k^{(N-1)} \right| = 2x_k$ must be a critical value to determine the states of $f(t)$, where $x_k = \left| a_k^{(N)} - a_k^{(N-1)} \right|$.

In the algorithm for determining the states of $f(t)$ based on the optimal iteration algorithm, if it is assumed that $h_k > 2x_k$, $h_k = 2x_k$, or $h_k < 2x_k$, then a function of $2x_k$ can be studied. To discover the function, the simulations are conducted. The simulation results indicate that $f(t)$ is in stable states at all sampling times when $h_k > 2x_k$, and in stable states at some sampling times and in unstable states at the others when $h_k = 2x_k$. Fortunately, this algorithm accurately determines the states of $f(t)$ when $h_k < 2x_k$.

It is assumed that $h_{k-va} = 2\lambda_k x_k = 2\lambda_k \left| a_k^{(N)} - a_k^{(N-1)} \right|$, where $0 < \lambda_k < 1$. Then, substituting h_{k-va} for h_k in Equations (9) and (10) yields

$$A_{k10v} = 4\lambda_k \left| a_k^{(N)} - a_k^{(N-1)} \right| A_{k00} + \left(2\lambda_k \left| a_k^{(N)} - a_k^{(N-1)} \right| \right)^2 C_{SS} \tag{22}$$

and

$$A_{k20v} = -4\lambda_k \left| a_k^{(N)} - a_k^{(N-1)} \right| A_{k00} + \left(2\lambda_k \left| a_k^{(N)} - a_k^{(N-1)} \right| \right)^2 C_{SS}, \tag{23}$$

respectively.

From the algorithm for determining the states of $f(t)$ based on the optimal iteration algorithm, an algorithm for determining the states of $f(t)$ based on the variable iteration step h_{k-va} is obtained. The algorithm for determining the state of $f(t)$ at the sampling time $t = t_1 + NT/(N - 1)$ is presented in **Figure 11**.

6.4. Calculation Complexity Analysis

According to the biproportional construction algorithm for calculating $a_1^* \cos(\omega t)$,

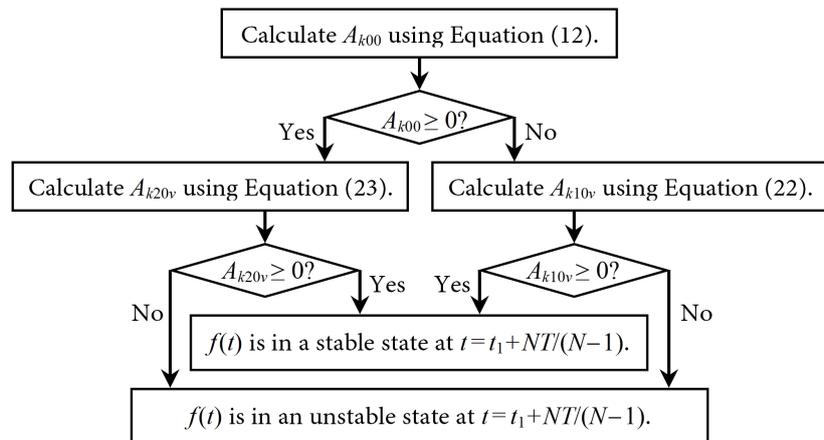


Figure 11. The algorithm for determining the state of $f(t)$ at the sampling time $t = t_1 + NT/(N - 1)$ based on the variable iteration step.

its main calculation complexity per iteration depends on that the N data are updated. That is, $f(1)' = \eta f(1)'$, $f(2)' = \eta f(2)'$, ..., and $f(N)' = \eta f(N)'$. Thus, it is N multiplication operations.

The calculation complexity of this algorithm is much higher than that of the direct calculation algorithm or the constant iteration algorithm for calculating $a_1^* \cos(\omega t)$. However, it is relatively low compared to the very high calculation speed of existing DSPs. Therefore, this algorithm can be implemented because N can be large enough.

6.5. Calculation Accuracy Analysis

From the biproportional construction algorithm for calculating $a_1^* \cos(\omega t)$, the direct calculation algorithm for calculating $a_1^* \cos(\omega t)$ correctly calculates a_1' when $f(t)$ changes periodically. The biproportional construction theory for calculating a_1^* correctly computes a_1' when $f(t)$ changes nonperiodically. Thus, this biproportional construction algorithm correctly calculates a_1' whether $f(t)$ changes periodically or nonperiodically.

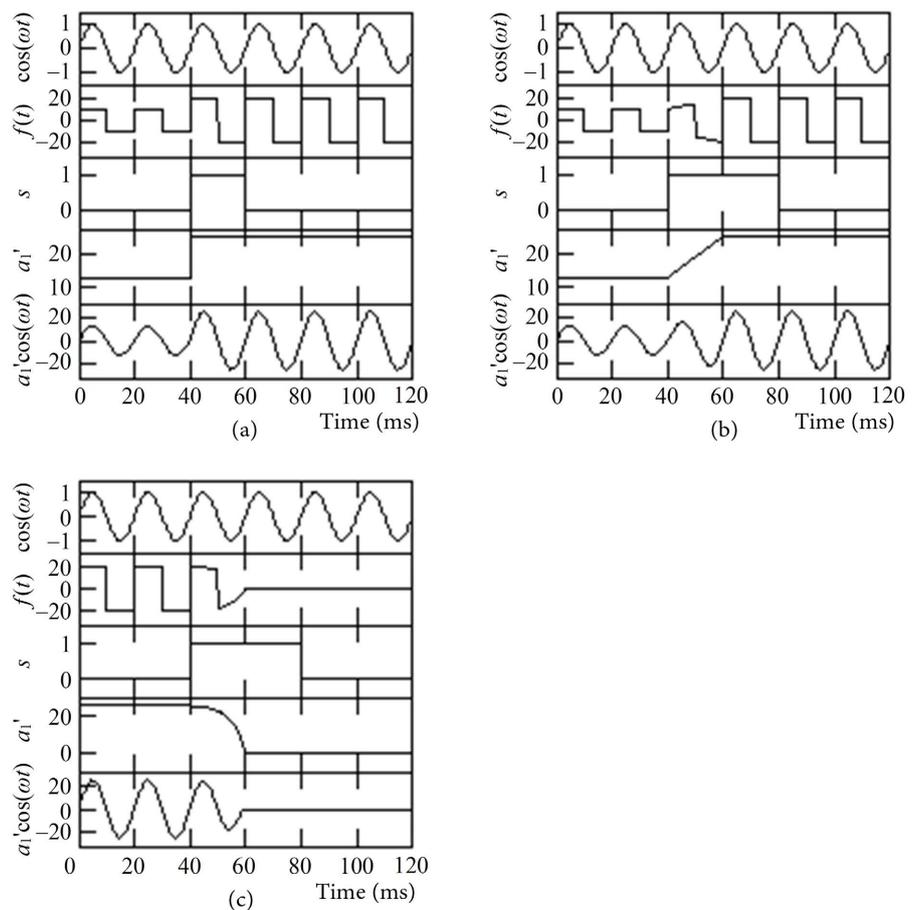


Figure 12. (a)-(c) The simulation results of the biproportional construction algorithm for calculating $a_1^* \cos(\omega t)$ with a step increase, a linear increase, and an exponential decrease, respectively, in $f(t)$.

6.6. Simulation Results and Experimental Results

To validate the biproportional construction algorithm for calculating $a_1^* \cos(\omega t)$, the simulations and the experiments are conducted. The simulation results are depicted in **Figure 12**, and the experimental results in **Figure 13**. In **Figure 12**, $N = 500$, $\lambda_1 = 0.75$, and $T = 0.02$ s. In **Figure 13**, $N = 200$, $\lambda_1 = 0.75$, and $T = 0.02$ s. In **Figure 12** and **Figure 13**, s represents the state of $f(t)$: $s = 0$ signifies that $f(t)$ is in a stable state, and $s = 1$ that $f(t)$ is in an unstable state. **Figure 12** and **Figure 13** indicate that the calculated a_1' is a constant and thus correct, when $f(t)$ changes periodically. All response times are 0 s and thus the calculated a_1' is also correct, when $f(t)$ changes nonperiodically. In addition, this algorithm accurately determines the states of $f(t)$.

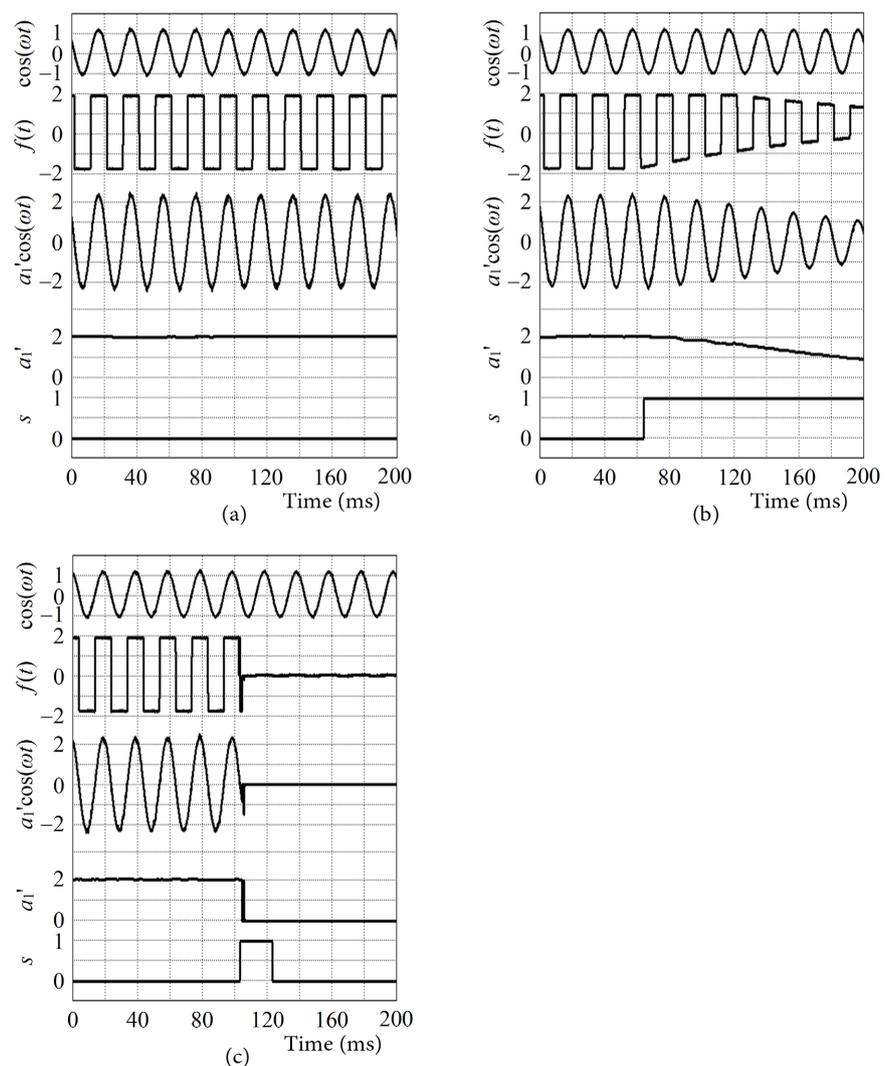


Figure 13. (a)-(c) The experimental results of the biproportional construction algorithm for calculating $a_1^* \cos(\omega t)$ with no change, a slow decrease, and a step decrease, respectively, in $f(t)$.

6.7. Brief Summary

We have built a biproportional construction theory, and then proposed a biproportional construction algorithm from the theory. From this biproportional construction algorithm, the direct calculation algorithm correctly calculates its FCs when $f(t)$ changes periodically, and the biproportional construction theory when $f(t)$ changes nonperiodically. Therefore, this biproportional construction algorithm correctly calculates its FCs whether $f(t)$ changes periodically or nonperiodically, and thus its FS.

7. Conclusions

The FS applies to a periodic non-sinusoidal function satisfying the Dirichlet conditions, whereas the being-processed function $f(t)$ in practical applications is usually an aperiodic non-sinusoidal signal. When $f(t)$ is aperiodic, its calculated FS is not correct, being still a challenging problem today.

To resolve the problem, we have correctly defined several basic concepts, such as $f(t)$ in a stable state, $f(t)$ in an unstable state, the FCs of $f(t)$ in a stable state, and the FCs of $f(t)$ in an unstable state. In particular, the definition of the FCs of $f(t)$ in an unstable state is of great significance in correctly calculating these FCs.

We have derived a direct calculation algorithm, a constant iteration algorithm, and an optimal iteration algorithm. The direct calculation algorithm correctly calculates its FCs when $f(t)$ changes periodically. Both the constant iteration algorithm and the optimal iteration algorithm provide an idea of determining the states of $f(t)$. From the idea, we have obtained an algorithm for determining the states of $f(t)$ based on the optimal iteration algorithm. In the algorithm, the variable iteration step is introduced, and thus we have presented an algorithm for determining the states of $f(t)$ based on the variable iteration step. The presented algorithm accurately determines the states of $f(t)$.

On the basis of the definitions and algorithms, we have built a biproportional construction theory. The biproportional construction theory consists of a first and a second proportional construction theory. The first proportional construction theory correctly calculates the FCs of $f(t)$ at the present sampling time. The second proportional construction theory creates a precondition for correctly calculating the FCs of $f(t)$ at the next sampling time. From the biproportional construction theory, we have proposed a biproportional construction algorithm. The proposed biproportional construction algorithm correctly calculates its FCs whether $f(t)$ is periodic or aperiodic, and thus its FS. Therefore, the problem is solved.

The biproportional construction theory and the biproportional construction algorithm extend the application range of the FS, and have considerable theoretical and practical significance.

Acknowledgements

This work was supported by the Priority Academic Program Development of

Jiangsu Higher Education Institutions (PAPD). The authors would like to thank the editors and referees for their valuable advice for the improvement of this article.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this article.

References

- [1] Kendall, D.G., Batchelor, G.K., Bingham, N.H., Hayman, W.K., Hyland, J.M.E., Lorentz, G.G., Moffatt, H.K., Parry, W., Razborov, A.A., Robinson, C.A. and Whittle, P. (1990) Andrei Nikolaevich Kolmogorov (1903-1987). *Bulletin of the London Mathematical Society*, **22**, 31-100. <https://doi.org/10.1112/blms/22.1.31>
- [2] Carleson, L. (1966) On Convergence and Growth of Partial Sums of Fourier Series. *Acta Mathematica*, **116**, 135-157. <https://doi.org/10.1007/BF02392815>
- [3] Iserles, A. and Nørsett, S.P. (2008) From High Oscillation to Rapid Approximation I: Modified Fourier Expansions. *IMA Journal of Numerical Analysis*, **28**, 862-887. <https://doi.org/10.1093/imanum/drn006>
- [4] Mozumder, M. and Tarvainen, T. (2020) Time-Domain Diffuse Optical Tomography Utilizing Truncated Fourier Series Approximation. *Journal of the Optical Society of America A*, **37**, 182-191. <https://doi.org/10.1364/JOSAA.37.000182>
- [5] Khazaeli, R., Nazari, M.R. and Zadehgo, A. (2008) Introducing Unsteady and Nonuniform Source Terms in Entropic Lattice Kinetic Models Using Fourier Series. *Physical Review E*, **98**, Article ID: 053303. <https://doi.org/10.1103/PhysRevE.98.053303>
- [6] Tsai, C.L., Chen, W.T. and Chang, C.S. (2016) Polynomial-Fourier Series Model for Analyzing and Predicting Electricity Consumption in Buildings. *Energy & Buildings*, **127**, 301-312. <https://doi.org/10.1016/j.enbuild.2016.05.083>
- [7] Niu, F., O'Neill, Z. and O'Neill, C. (2018) Data-Driven Based Estimation of HVAC Energy Consumption Using an Improved Fourier Series Decomposition in Buildings. *Building Simulation*, **11**, 633-645. <https://doi.org/10.1007/s12273-018-0431-2>
- [8] Lin, Y., Quindroit, C. and Jang, H. (2015) 3-D Fourier Series Based Digital Predistortion Technique for Concurrent Dual-Band Envelope Tracking with Reduced Envelope Bandwidth. *IEEE Transactions on Microwave Theory and Techniques*, **63**, 2764-2775. <https://doi.org/10.1109/TMTT.2015.2452271>
- [9] Gómez-Arista, I., Dávila-Pintle, J.A., Montalvo-Montalvo, N., Rubin-Alvarado, A.A., Bravo-García, Y.E. and Reynoso-Lara, E. (2020) Fourier Coefficients Applied to Improve Backscattered Signals in a Short-Range LIDAR System. *Electronics*, **9**, 390. <https://doi.org/10.3390/electronics9030390>
- [10] Tavighi, A., Marti, J.R. and Galvan, V.A. (2018) Time-Window-Based Discrete-Time Fourier Series for Electromagnetic Transients in Power Systems. *IEEE Transactions on Power Delivery*, **33**, 2551-2561. <https://doi.org/10.1109/TPWRD.2018.2794887>
- [11] Hamed, H.D. and Reza, G. (2019) A New Approach to Design an Observer for Load Current of UPS Based on Fourier Series Theory in Model Predictive Control System. *International Journal of Electrical Power & Energy Systems*, **104**, 898-909. <https://doi.org/10.1016/j.ijepes.2018.07.047>
- [12] Liu, G., Chi, C., Jin, Y., Ma, Y., Sun, L., Li, L. and Sun, Y. (2017) Analysis of Non-

- Sinusoidal Steady Electric Field of ± 500 kV Converter Transformer. *Energy and Power Engineering*, **9**, 53-62. <https://doi.org/10.4236/epe.2017.94B007>
- [13] Bolhasani, M., Ghafi, E.K., Ghorashi, S.A. and Mehrshahi, E. (2019) Waveform Covariance Matrix Design Using Fourier Series Coefficients. *IET Signal Processing*, **13**, 562-567. <https://doi.org/10.1049/iet-spr.2019.0024>
- [14] Bahaz, M. and Benzid, R. (2018) Efficient Algorithm for Baseline Wander and Powerline Noise Removal from ECG Signals Based on Discrete Fourier Series. *Australian Physical & Engineering Sciences Medicine*, **41**, 143-160. <https://doi.org/10.1007/s13246-018-0623-1>
- [15] Mastriani, M. (2018) Quantum-Classical Algorithm for an Instantaneous Spectral Analysis of Signals: A Complement to Fourier Theory. *Journal of Quantum Information Science*, **8**, 52-77. <https://doi.org/10.4236/jqis.2018.82005>
- [16] Scheibler, R. and Hurley, P. (2012) Computing Exact Fourier Series Coefficients of IC Rectilinear Polygons from Low-Resolution Fast Fourier Coefficients. *Proceedings of the International Society for Optical Engineering*, **8326**, Article ID: 83262V. <https://doi.org/10.1117/12.916360>
- [17] Zadiraka, V.K., Kolomys, E.N. and Luts, L.V. (2013) Effective with Respect to Accuracy Algorithms of Approximation of Some Classes Functions by Fourier Series. *Journal of Automation and Information Sciences*, **45**, 14-29. <https://doi.org/10.1615/JAutomatInfScien.v45.i7.30>
- [18] Maddi, A., Guessoum, A. and Berkani, D. (2013) Applying a Technique of Identification for Computing Fourier Series Coefficients. *Proceedings of the World Congress on Engineering*, London, 3-5 July 2013, Vol II.
- [19] Gruber, C. and Abrykosov, O. (2016) On Computation and Use of Fourier Coefficients for Associated Legendre Functions. *Journal of Geodesy*, **90**, 525-535. <https://doi.org/10.1007/s00190-016-0891-z>
- [20] Nakagawa, K. and Fujimori, K. (2018) Design of the RF-DC Conversion Circuit by GA Adopting Mutation Based on Fourier Coefficients on Unit Structures. 2018 *Progress in Electromagnetics Research Symposium*, Toyama, 1-4 August 2018, 1900-1904. <https://doi.org/10.23919/PIERS.2018.8597950>
- [21] Ruiz, A. (2018) Fourier Coefficients and Moments of Piecewise-Circular Curves. *Pattern Recognition Letters*, **116**, 238-245. <https://doi.org/10.1016/j.patrec.2018.10.033>
- [22] Lytvyn, O.M., Lytvyn, O.G. and Lytvyn, O.O. (2019) Method of Calculating Fourier Coefficients of Three Variable Functions Using Tomogram. *9th International Conference on Advanced Computer Information Technologies*, Kharkiv, 5-7 June 2019, 125-128. <https://doi.org/10.1109/ACITT.2019.8779938>
- [23] Kumar, A. and Ramakrishnan, B. (2018) Estimates for Fourier Coefficients of Hermitian Cusp Forms of Degree Two. *Acta Arithmetica*, **183**, 257-275. <https://doi.org/10.4064/aa170301-26-10>
- [24] Gibert, P., Panciatici, P., Losseau, R., Guironnet, A., Tromeur-Dervout, D. and Erhel, J. (2018) Speedup of EMT Simulations by Using an Integration Scheme Enriched with a Predictive Fourier Coefficients Estimator. *IEEE PES Innovative Smart Grid Technologies Conference*, Europe, 1 October 2018, 1-6. <https://doi.org/10.1109/ISGTEurope.2018.8571899>
- [25] Jiang, D. and Liu, B. (2013) On Fourier Coefficients of Automorphic Forms of $GL(n)$. *International Mathematics Research Notices*, **2013**, 4029-4071. <https://doi.org/10.1093/imrn/rns153>
- [26] Alaca, A., Alaca, S. and Aygin, Z.S. (2015) Fourier Coefficients of a Class of Eta Qu-

-
- otients of Weight 2. *International Journal of Number Theory*, **11**, 2381-2392.
<https://doi.org/10.1142/S1793042115501109>
- [27] Tang, H. (2013) Estimates for the Fourier Coefficients of Symmetric Square L-Functions. *Archiv der Mathematik*, **100**, 123-130.
<https://doi.org/10.1007/s00013-013-0481-8>
- [28] Dickson, M.J. (2015) Fourier Coefficients of Degree Two Siegel-Eisenstein Series with Trivial Character at Square Free Level. *Ramanujan Journal*, **37**, 541-562.
<https://doi.org/10.1007/s11139-014-9633-0>
- [29] Hundley, J. and Zhang, Q. (2016) Fourier Coefficients of Theta Functions at Cusps other than Infinity. *Acta Arithmetica*, **175**, 341-383.
<https://doi.org/10.4064/aa8278-4-2016>