

# Similar basis function algorithm for numerical estimation of Fourier integrals

Dmitriy Melkonian

Received: 14 September 2008 / Accepted: 29 July 2009 /  
Published online: 19 August 2009  
© Springer Science + Business Media, LLC 2009

**Abstract** The methodological difficulties of estimating Fourier integrals using the fast Fourier transform (FFT) algorithm have intensified the interest in an alternative approach based on the Filon’s method of computing the trigonometric integrals. Following this approach, we introduce in this paper a similar basis function (SBF) algorithm that decomposes the function to be transformed into the sum of finite elements termed “similar basis functions”. Due to a simple analytical form of SBF, the reassignment of the SBFs’ similarity relationships into the transformation domain reduces the estimation of the Fourier integrals to a number of standard computational procedures. The SBF algorithm is capable to deal with both uniform and non-uniform samples of the function under analysis. Using this opportunity, we extend a general SBF algorithm by a fast SBF algorithm which deals with exponentially increasing sampling intervals. The efficiency and the accuracy of the method are illustrated by computer experiments with frequency characteristics and transient responses of a typical dynamic system.

**Keywords** Fourier integral · Fast Fourier transform · Filon quadrature · Similar basis functions · Frequency response characteristics · Transient waveforms

## 1 Introduction

The Fourier series and integrals are the basic theoretical and computational tools in almost every branch of science and engineering. One of the classical

---

D. Melkonian (✉)  
Brain Dynamics Centre, Westmead Millenium Institute and Westmead Hospital,  
Acacia House, Westmead Hospital, Westmead NSW 2145, Australia  
e-mail: dmitri@psychiat.usyd.edu.au

applications of the Fourier integrals is the frequency domain analysis of dynamic systems [8]. As in almost all fields of Fourier analysis, thinking in terms of sine functions is central to this application. The frequency response methods draw on the fact that it is very easy to describe what happens with the sine function when it passes through a linear time-invariant system. If the input of the system is a sine wave, the output is another sine wave of the same frequency, with different amplitude and a phase shift. The ratio of the output amplitude to the input amplitude is the gain. Consequently, at a given frequency the transfer of the input sine function by the system can be described by just two parameters: gain and phase. The dependencies of gain and phase on the frequency constitute a frequency transfer function. The ability of Fourier integral to provide this frequency domain counterpart of the time domain transient is important because it converts calculus: differentiation and integration in the time domain – into algebra: multiplication and division in the frequency domain. On the basis of this principle, the frequency domain methods of linear system analysis have been developed [2]. The methods constitute a significant part of the theories of control and communication systems, circuit analysis, signal and image processing, biological cybernetics. Many aspects of the theory and methods have been modified for analysis and design of wider classes of dynamic systems, particularly some types of time-varying [32] and non-linear systems [9].

With the use of modern computing technology, the frequency domain methods became not only the theoretical framework of signal and system analysis, but also a digital data processing tool for two important problems: (1) the estimation of system's frequency characteristics from transient responses, (2) the estimation of transient waveforms from the frequency characteristics of the system. One of the latest developments of these methods is a time-frequency analysis of non-stationary signals and systems [10].

The most readily available technique used for estimation of Fourier integrals in these applications is a computationally efficient algorithm of fast Fourier transform (FFT) [28]. The critical aspect of this approach is that the FFT is supported by a Fourier series model of the data [11]. This distinction with Fourier integral poses a number of limitations, the most important of which is that the FFT is applied to  $2^n$  equidistant points of the function to be transformed. The integer “ $n$ ” and the sampling interval define the set of equidistant points at which the transformation results are computed. A strict grid of computational points creates a number of limitations and methodological difficulties. For example, it is incompatible with widely accepted logarithmic frequency scales for computations and display of frequency characteristics. The distinction of the FFT with Fourier integral is also the source of specific computational errors [6].

To improve the numerical estimation of Fourier integrals, a number of recent studies focus their attention on alternative numerical methods. A specific aspect of the problem is that Fourier integrals deal with sets of strongly oscillatory trigonometric terms and consequently belong to a category of oscillatory integrals [15]. The standard methods of numerical integration

usually give inaccurate results for oscillatory components. Classical approach that provides a maximum precision in the estimation of Fourier integrals is based on the separation of the function to be transformed from an oscillatory kernel, i.e. a trigonometric function [12]. The original Filon's method consists of dividing the integration interval into equal size panels and interpolating non-oscillatory function on each panel using a Simpson's rule (quadratic interpolation). In each panel the resulting integral is expressed analytically. This basic approach supports a number of Filon-type methods which use interpolation polynomials of different degrees [15, 26]. The interpolant is made up piecewise of segments of Lagrange interpolation polynomials. It has discontinuous derivatives at the data points, where the pieces join, because the set of points used in interpolation changes discretely. A high-accuracy method of numerical Fourier transform in  $d$ -dimensions improves accuracy of approximation using interpolation with polynomial splines and re-establishes the continuity of original function and its derivatives [5].

An appealing feature of Filon-type methods is their inherent ability to support a number of different algorithmic implementations [17]. Particularly, the Filon's quadrature may be used as a tool to improve the accuracy of conventional FFT [28, 31]. The algorithms designed in such a way using polynomials of first [30] and second degrees [18] significantly improved accuracy of the frequency domain measures, as compared with conventional FFT, in a number of physical applications of Fourier transform spectroscopy. However, this approach does not eliminate restrictions posed by a Fourier series model of the data, particularly a strict grid of computational points.

The objective guided the development of the similar basis function (SBF) algorithm [19] was to a full extent retain the conceptual framework of the Fourier integral as a tool addressed to continuous functions both in the primary and transformation domains. Computationally, the idea of the SBF algorithm is to express the function under the analysis as the sum of finite elements (basis function) with a similar form of spectral characteristics. This principle reduces the numerical integration to a number of relatively simple standard time and frequency domain calculations and provides an opportunity to construct computationally efficient algorithms.

The SBF algorithm has been successfully applied to a digital spectral analysis of a number of biomedical signals such as event related potentials [23], electro-myogram [20], electroencephalogram [24] and electrocardiogram [3]. The estimation of both the amplitude and the phase frequency characteristics of the signal in these applications provided means to support the interpretation of the spectral analysis results by a model-based approach named "the fragmentary decomposition" [7, 21, 22]. Providing a dynamic model of a non-stationary signal, the fragmentary decomposition heavily relies on the Fourier methods of dynamic system analysis. In this context, the SBF algorithm is a means of a digital spectral analysis on the signal segments of arbitrary length.

In this study we present an extended SBF algorithm as a universal tool for estimation of Fourier integrals in the context of the abovementioned problems and dynamic system analysis. We also present a new fast SBF algorithm, which

provides an effective method of computing the time domain counterparts of frequency domain characteristics.

This paper is organized in the following way. In Section 2 we present the SBF algorithm. We start with general relationships and introduce the notions of SBF and similarity relationships. Then, we describe the interpolation procedure and formulas for computing transforms for the three different types of given function. In Section 3 we present the fast SBF algorithm. In Section 4 we discuss the computer experiments that compare the numerical solutions of the algorithm with theoretical data. The paper is finalized with concluding remarks in Section 5. We also attach an Appendix 1 with major relationships supporting the applications of Fourier integrals to linear system analysis and Appendix 2 with the description of theoretical dynamic characteristics of harmonic oscillator used in the computer experiments.

## 2 The SBF algorithm

### 2.1 General relationships

As in most texts, we shall throughout represent the Fourier transform of a function written in lower-case by the same symbol as the function itself, but written in upper-case. The major forms of Fourier integrals employed for computing frequency response characteristics from transient waveforms and vice versa are given in Appendix 1. The transformations of both time functions and frequency characteristics bear dual relationships. This allows us to reduce the major transforms to the form of the following cosine and sine Fourier integrals,

$$F_C(u) = \mathfrak{F}_C\{f(x)\} = \int_0^{\infty} f(x) \cos uxdx, \quad (1)$$

$$F_S(u) = \mathfrak{F}_S\{f(x)\} = \int_0^{\infty} f(x) \sin uxdx. \quad (2)$$

Conceptually,  $f(x)$  in these integrals is a continuous function that extends over a semi-infinite interval from 0 to  $\infty$ . However, every function we use to construct a numerical algorithm must be of finite extent. The application of Fourier transforms to finite-duration functions requires care. The point is that discontinuities at the boundaries of the finite interval may have distorting impact on the transforms. In Fourier series analysis the discontinuities are responsible for the spectral leakage. A number of different windows and weighting functions have been developed to reduce the spectral leakage associated with finite observation intervals [14].

In the context of Fourier integral transforms, the discontinuities at the beginning and the end of the integration interval may reflect the physical nature

of the processes under the investigation. For example, current approaches to the transient analysis of biological processes define “transient” as a function that has discontinuities at the beginning and the end of the observation interval [13].

Filon-type methods have a potential to deal with such kind of signals. The interpolation with uniform sampling using connected splines, involve the FFT as a mere tool, but overcome the flaws of the FFT [4, 5]. This solves the problem of discontinuities without windowing or weighting functions.

In the context of boundary values, the design of numerical procedures needs to be supported by a clear understanding of the principle used to replace  $f(x)$  in integrals (1) and (2) by its finite approximant. To avoid ambiguity in dealing with this problem, we use a special symbol, an overhat ( $\hat{\cdot}$ ), to denote in a primary domain a function which has nonzero values over only a finite interval. In the transformation domain, the same symbol indicates the Fourier transform of the corresponding finite extent function. It is important to remember that if a function is finite in the primary domain it must be infinite in the transformation domain.

Given bounded  $\hat{y}(x)$  which has nonzero values over a finite interval from 0 to  $L$  and vanishes outside the interval, the Fourier integrals (1) and (2) from  $\hat{y}(x)$  have forms of the following finite cosine and sine Fourier integrals:

$$\hat{Y}_C(u) = \mathfrak{F}_C \{ \hat{y}(x) \} = \int_0^L \hat{y}(x) \cos u x dx, \tag{3}$$

$$\hat{Y}_S(u) = \mathfrak{F}_S \{ \hat{y}(x) \} = \int_0^L \hat{y}(x) \sin u x dx. \tag{4}$$

These integrals are the point of departure for the SBF algorithm. The transcription of  $\hat{y}(x)$  into a digital form specifies  $\hat{y}(x)$  by its sampled values  $\hat{y}_i = \hat{y}(x_i)$  in a finite number of nodal points  $x_i (i = 0, 1, \dots, N)$  with  $x_0 = 0$  and  $x_N = L$ . The nodes need not be spaced equally.

The calculations are based on the interpolation of  $\hat{y}(x)$  over its samples by a piece-wise linear polynomial  $\hat{h}(x)$ . The basic approach of the SBF algorithm is to decompose the interpolant into the sum of similar basis functions

$$\hat{h}(x) = \sum_{i=0}^{N-1} a_i \hat{\theta}_i(x), \tag{5}$$

where  $a_i$  are the interpolation coefficients and  $\hat{\theta}_i(x)$  is a similar basis function. The SBF is defined by the similarity relationship as

$$\hat{\theta}_i(x) = \hat{r}(x/x_{i+1}),$$

where

$$\hat{f}(x) = \begin{cases} 1 - x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

is a basic element in the form of a unit right-angled triangle (Fig. 1a) termed “triangular basis function” (TBF).

The SBF vanishes everywhere except on the interval  $[0, x_{i+1}]$ ,

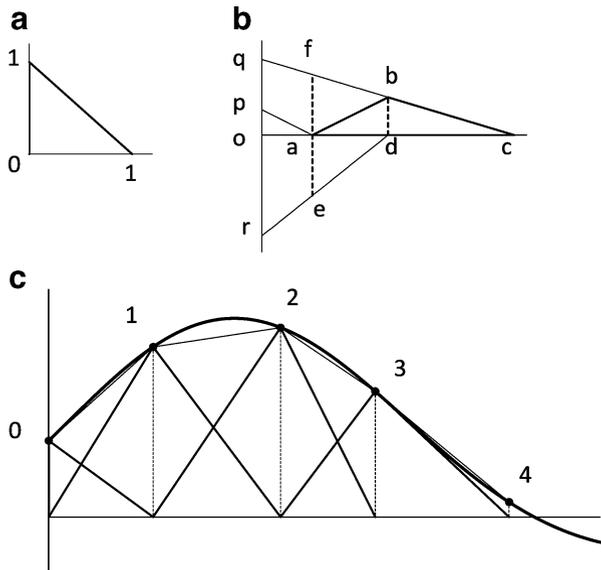
$$\hat{\theta}_i(x) = \begin{cases} 1 - \frac{x}{x_{i+1}}, & \text{if } 0 \leq x \leq x_{i+1} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

An important aspect of the interpolation is the necessity to have a similar behavior of  $\hat{h}(x)$  and  $\hat{y}(x)$  at the ends of the interpolation interval. Given  $x = 0$  and  $x = L$ ,  $\hat{h}(x)$  is defined at the ends of the interpolation interval by the following values:

$$\hat{h}(0) = \sum_{i=0}^{N-1} a_i \quad \text{and} \quad \hat{h}(L) = 0.$$

To avoid a disagreement of  $\hat{h}(x)$  and  $\hat{y}(x)$  at these points, we presume that  $\hat{y}(0)$  may have an arbitrary value while  $\hat{y}(L) = 0$ .

**Fig. 1** **a** The triangular basis function. **b** Shows construction of the hat function (*triangle abc*) as the sum of the three SBFs (*triangles oqc, ord and opa*). **c** Exemplifies a piece-wise linear interpolation of the segment of the function shown by the bold solid line. The interpolant (*straight lines* connecting data points from 0 to 4) represents the sum of the SBF (vertex at the data point 0) and the hat functions with vertexes at the data points from 1 to 4



In the context of numerical transformations, the advantage of the decomposition (5) is that Fourier integrals (1) and (2) from TBF are expressed in an analytical form as

$$\hat{R}_C(u) = \mathfrak{S}_C \{ \hat{r}(x) \} = \frac{1 - \cos u}{u^2}, \tag{7}$$

$$\hat{R}_S(u) = \mathfrak{S}_S \{ \hat{r}(x) \} = \frac{u - \sin u}{u^2}. \tag{8}$$

These functions are illustrated by Fig. 2.

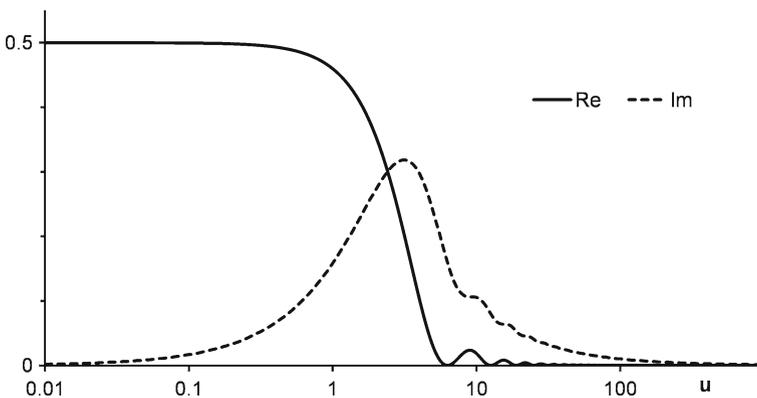
According to the similarity theorem of the theory of Fourier transforms, the compression of the abscissa in the primary domain corresponds to the expansion of the abscissa plus contraction of the ordinate in the transform domain. These operations establish the following similarity relationships between Fourier integrals of SBF and TBF:

$$\hat{\Theta}_{Ci}(u) = \mathfrak{S}_C \{ \hat{\theta}_i(x) \} = x_{i+1} \hat{R}_C(x_{i+1}u), \tag{9}$$

$$\hat{\Theta}_{Si}(u) = \mathfrak{S}_S \{ \hat{\theta}_i(x) \} = x_{i+1} \hat{R}_S(x_{i+1}u). \tag{10}$$

These relationships allow us to reduce the entire issue of the transform calculations to some standard manipulations with relatively simple functions (7) and (8).

The description of numerical techniques of the SBF algorithm is split in two parts. The first part (Subsection 2.2) presents the interpolation technique. The second part (Subsection 2.3) is devoted to the numerical estimation of Fourier integrals after presentation of  $\hat{y}(x)$  by the formula (5) with defined interpolation coefficients.



**Fig. 2** The curves Re and Im show functions (7) and (8), respectively

### 2.2 Interpolation using similar basis functions

Given  $\hat{y}(L) = 0$ , it is assumed that  $\hat{y}(x)$  is represented in the interval  $[0, L]$  by its samples  $\hat{y}_i = \hat{y}(x_i)$  in the nodal points  $x_i (i = 0, 1, 2, \dots, N)$  the choice of which presumes that  $x_0 = 0, x_N = L$  and  $x_0 < x_1 < \dots < x_i < \dots < x_N$ .

The interpolant,  $\hat{h}(x)$ , is created as a piece-wise linear polynomial which satisfies the interpolation condition:

$$\hat{h}_i = \hat{y}_i \quad \text{for } i = 0, 1, \dots, N, \tag{11}$$

where  $\hat{h}_i = \hat{h}(x_i)$ .

In the interval  $[x_i, x_{i+1}] (i = 0, \dots, N - 1)$  the interpolant is equivalent to a straight line connecting the values  $\hat{y}_i$  and  $\hat{y}_{i+1}$ ,

$$\hat{h}(x) = \hat{y}_i + (\Delta\hat{y}_{i+1}/\Delta x_{i+1}) \cdot x,$$

where  $\Delta\hat{y}_{i+1} = \hat{y}_{i+1} - \hat{y}_i$  and  $\Delta x_{i+1} = x_{i+1} - x_i$ .

At the end of the interval  $\hat{h}(L) = 0$ , i.e. the very form of the interpolant provides its coincidence with  $\hat{y}(x)$  at the last interpolation node. Therefore, we need  $N$  interpolation coefficients to support the condition (11) for data points from  $i = 0$  to  $N - 1$ .

A general principle of determining the interpolation coefficients is a substitution of values of  $\hat{h}_i$  from (11) into (5) for each data point from  $i = 0$  to  $i = N - 1$ . The coefficients can be determined by solving the resulting system of  $N$  linear equations. A substantial simplification of calculations, as compared with general procedures, comes from a triangular form of a matrix supporting the system. Note, that each term in the interpolant (5), i.e.  $a_i\hat{\theta}_i(x)$ , represents a rescaled TBF which has been rescaled on the abscissa axis by a scaling coefficient  $x_{i+1}$  and multiplied on the ordinate axis by a factor  $a_i$ . According to (6),  $\hat{\theta}_i(x)$  vanishes everywhere except on the interval  $[0, x_{i+1}]$ . Given a nodal point  $x_j$ ,  $\hat{\theta}_i(x_j) = 0$  if  $j \geq i + 1$ . Therefore, a substitution of  $x = x_j$  into (5) removes from the sum all the terms with indexes from  $i = 0$  to  $i = j - 1$ . The value of the interpolant is

$$\hat{h}_j = \sum_{i=j}^{N-1} a_i\hat{\theta}_i(x_j).$$

A substitution of values  $x = x_j$  for  $j$  from 0 to  $N - 1$  into (5) and replacement of  $\hat{h}_j$  by  $\hat{y}_j$  according to the interpolation condition (11) results in the following triangular form of the system of  $N$  linear equations:

$$\begin{aligned} \hat{y}_0 &= a_0\hat{\theta}_0(x_0) + a_1\hat{\theta}_1(x_0) + \dots + a_i\hat{\theta}_i(x_0) + \dots + a_{N-1}\hat{\theta}_{N-1}(x_0), \\ \hat{y}_1 &= a_1\hat{\theta}_1(x_1) + \dots + a_i\hat{\theta}_i(x_1) + \dots + a_{N-1}\hat{\theta}_{N-1}(x_1), \\ &\dots\dots\dots \\ \hat{y}_j &= a_j\hat{\theta}_j(x_j) + \dots + a_{N-1}\hat{\theta}_{N-1}(x_j), \\ &\dots\dots\dots \\ \hat{y}_{N-1} &= a_{N-1}\hat{\theta}_{N-1}(x_{N-1}), \end{aligned}$$

To solve these equations using the conventional matrix notation, we collect the signal samples and the interpolation coefficients into N-by-1 vectors  $\hat{\mathbf{y}} = [\hat{y}_0, \dots, \hat{y}_j, \dots, \hat{y}_{N-1}]$  and  $\mathbf{a} = [a_0, \dots, a_i, \dots, a_{N-1}]$ . In these terms the whole system of linear equations is  $\hat{\mathbf{y}} = \mathbf{W} \cdot \mathbf{a}$ , where  $\mathbf{W}$  is a square N-by-N matrix of upper triangular form.

Let  $w_{ji}$  denote the entry of  $\mathbf{W}$  in the  $j$ th row and  $i$ th column. The entries of  $\mathbf{W}$  below the main diagonal are zero, i.e.  $w_{ji} = 0$  for  $j > i$ . For  $j \leq i$ ,

$$w_{ji} = \hat{\theta}_i(x_j) = 1 - x_j/x_{i+1},$$

where  $i$  and  $j$  take values from 0 to  $N - 1$ .

The system of linear equations with a triangular matrix is easily solved by a conventional recursive back substitution procedure. However, we support the SBF algorithm by much more efficient procedure using the inverse matrix  $\mathbf{V} = \mathbf{W}^{-1}$  in terms of which the matrix equation is  $\mathbf{a} = \mathbf{V} \cdot \hat{\mathbf{y}}$ . In the remaining part of this section we show that the matrix has a form

$$\mathbf{V} = \begin{pmatrix} \alpha_0 & \beta_1 & \gamma_2 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \alpha_1 & \beta_2 & \gamma_3 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_2 & \beta_3 & \gamma_4 & \cdots & 0 & 0 & 0 & 0 \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \alpha_{N-4} & \beta_{N-3} & \gamma_{N-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \alpha_{N-3} & \beta_{N-2} & \gamma_{N-1} \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \alpha_{N-2} & \beta_{N-1} \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \alpha_{N-1} \end{pmatrix},$$

where

$$\begin{aligned} \alpha_i &= \frac{x_{i+1}}{\Delta x_{i+1}} & (0 \leq i \leq N - 1), \\ \beta_i &= x_i \frac{\Delta x_{i+1} + \Delta x_i}{\Delta x_{i+1} \Delta x_i} & (1 \leq i \leq N - 1), \\ \gamma_i &= \frac{x_{i-1}}{\Delta x_i} & (2 \leq i \leq N - 1), \end{aligned} \tag{12}$$

Accordingly, the formulas for the estimation of the interpolation coefficients are:

$$\begin{aligned} a_i &= \alpha_i \hat{y}_i - \beta_{i+1} \hat{y}_{i+1} + \gamma_{i+2} \hat{y}_{i+2}, & 0 \leq i < N - 3 \\ a_{N-2} &= \alpha_{N-2} \hat{y}_{N-2} - \beta_{N-1} \hat{y}_{N-1}, \\ a_{N-1} &= \alpha_{N-1} \hat{y}_{N-1}. \end{aligned} \tag{13}$$

An essential aspect of these formulas is that interpolation coefficients are defined by the local properties of the function  $\hat{y}(x)$ . Note that each interpolation coefficient,  $a_i$ , ordered by index  $i$  from 0 to  $N-3$  is defined by the three successive samples,  $\hat{y}_i, \hat{y}_{i+1}$  and  $\hat{y}_{i+2}$ . The last two coefficients,  $a_{N-2}$  and  $a_{N-1}$ , are defined by the two samples ( $\hat{y}_{N-2}$  and  $\hat{y}_{N-1}$ ) and one sample ( $\hat{y}_{N-1}$ ), respectively.

To make this aspect of the interpolation as intuitive as possible, we deduce the interpolation coefficients using a geometrical principle. Figure 1b shows how the sum of three successive SBFs composes a “hat-function”. Widely used by a finite-element method [25], the hat function is defined as

$$\hat{\phi}_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & \text{if } x_{i-1} \leq x < x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & \text{if } x_i \leq x < x_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

on the mesh  $x_0 < x_1 < \dots < x_i < \dots < x_N$ .

The interpolation capability of the hat function is supported by the two properties:

1.  $\hat{\phi}_i(x)$  vanishes everywhere except on the two subintervals to which  $x_i$  belongs.
2.  $\hat{\phi}_i(x)$  is unity at the node  $i$  and zero at all other nodes, i.e.

$$\hat{\phi}_i(x_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

Using the hat functions, we represent the interpolant (5) in the form

$$\hat{h}(x) = \hat{y}_0 \hat{\theta}_0(x) + \sum_{i=1}^{N-1} \hat{y}_i \hat{\phi}_i(x). \quad (14)$$

Figure 1c exemplifies this decomposition. Given  $\hat{y}(x)$  (bold solid line), the approximating function,  $\hat{h}(x)$ , is created by joining the data points 0, 1, 2, 3 and 4 by the straight lines (light traces). In the same fashion the approximation can be continued for any number of the following nodal points. The interpolation condition at the first interpolation node  $x_0$  is supported by the coincidence of the vertex of  $\hat{y}_0 \hat{\theta}_0(x)$  with the data point 0. The following data points coincide with vertexes of the hat functions with indexes from 1 to 4.

To find the interpolation coefficients using (14), we need to express the hat function in terms of TBFs. Referring to Fig. 1b, let  $\mathbf{oa} = x_{i-1}$ ,  $\mathbf{od} = x_i$ ,  $\mathbf{oc} = x_{i+1}$  and  $\mathbf{bd} = 1$ . Therefore, the triangles  $\mathbf{oqc}$ ,  $\mathbf{ord}$  and  $\mathbf{opa}$  may be regarded as the functions  $\alpha_i \hat{\theta}_i(x)$ ,  $\beta_i \hat{\theta}_{i-1}(x)$  and  $\gamma_i \hat{\theta}_{i-2}(x)$ , respectively.  $\alpha_i$ ,  $\beta_i$  and  $\gamma_i$  are the weighting coefficients which correspond to the segments  $\mathbf{oq}$ ,  $\mathbf{or}$  and  $\mathbf{op}$ , respectively. Since  $\mathbf{ae} = \mathbf{af}$ ,

$$\hat{\phi}_i(x) = \alpha_i \hat{\theta}_i(x) - \beta_i \hat{\theta}_{i-1}(x) + \gamma_i \hat{\theta}_{i-2}(x). \quad (15)$$

The weighting coefficients are defined by the following similarity relationships of relevant triangles.

From the similarity of triangles **oqc** and **dbc**,  $\mathbf{oq}/\mathbf{db} = x_{i+1}/\Delta x_{i+1}$  where  $\Delta x_{i+1} = x_{i+1} - x_i$ . Since **oq** =  $\alpha_i$  and **db** = 1,  $\alpha_i = x_{i+1}/\Delta x_{i+1}$ .

From the similarity of triangles **afc** and **dbc**,  $\mathbf{af}/\mathbf{db} = (x_{i+1} - x_{i-1})/\Delta x_{i+1}$ , i.e. **af** =  $(x_{i+1} - x_{i-1})/\Delta x_{i+1}$ . Since **ae** = **af**, the similarity of triangles **ord** and **aed** shows that **or** =  $x_i(\Delta x_{i+1} + \Delta x)/(\Delta x_{i+1}\Delta x_i)$ . Accordingly,  $\beta_i = x_i(\Delta x_{i+1} + \Delta x)/(\Delta x_{i+1}\Delta x_i)$ .

From the segments on the ordinate axis it is seen that **op** = **or** - **oq**, i.e.  $\gamma_i = \beta_i - \alpha_i$ . Substitution of the values of  $\beta_i$  and  $\alpha_i$  gives  $\gamma_i = x_{i-1}/\Delta x_i$ .

For the hat function  $\hat{\phi}_1(x)$  in (14) the first subinterval  $[x_0, x_1]$  begins from  $x_0 = 0$ , i.e. **oa** = 0. Therefore,  $\gamma_1 = 0$  and

$$\hat{\phi}_1(x) = \alpha_1 \hat{\theta}_1(x) - \beta_1 \hat{\theta}_0(x),$$

where  $\beta_1 = \alpha_1$ .

Substitution of  $\hat{\phi}_1(x)$  and  $\hat{\phi}_i(x)$  according to (15) into (14) gives

$$\begin{aligned} \hat{h}(x) &= \hat{y}_0 \hat{\theta}_0(x) + \hat{y}_1 \left[ \alpha_1 \hat{\theta}_1(x) - \beta_1 \hat{\theta}_0(x) \right] \\ &+ \sum_{i=2}^{N-1} \hat{y}_i \left[ \alpha_i \hat{\theta}_i(x) - \beta_i \hat{\theta}_{i-1}(x) + \gamma_i \hat{\theta}_{i-2}(x) \right]. \end{aligned}$$

The right-hand side of this equation is equal to the right-hand side of the equation (5). Re-ordering terms, yields

$$\begin{aligned} \hat{h}(x) &= \sum_{i=0}^{N-3} \left[ (\alpha_i \hat{y}_i - \beta_{i+1} \hat{y}_{i+1} + \gamma_{i+2} \hat{y}_{i+2}) \hat{\theta}_i(x) \right] \\ &+ (\alpha_{N-2} \hat{y}_{N-2} - \beta_{N-1} \hat{y}_{N-1}) \hat{\theta}_{N-2}(x) + \alpha_{N-1} \hat{y}_{N-1} \hat{\theta}_{N-1}(x), \end{aligned}$$

where  $\alpha_0 = 1$ .

The interpolation coefficients are now readily found to be in the form of the formulas (13) with weighting coefficients (12). This proves the form of matrix **V**.

The formulas may be simplified in case of samples which are ordered according to some rule. In a most common situation the function to be transformed,  $\hat{y}(x)$ , represents the succession of equidistant samples  $\hat{y}_n = \hat{y}(n\Delta x)$  for  $n = 0, 1, \dots, N$  taken with the sampling rate  $\Delta x$ . In this case,  $x_i = i\Delta x$  ( $i = 0, 1, \dots, N$ ), where  $\Delta x = L/N$ .

Under these conditions, the weighting coefficients in (13) degenerate into the following integers:

$$\begin{aligned} \alpha_i &= i + 1 \quad (0 \leq i \leq N - 1), \\ \beta_i &= 2i \quad (1 \leq i \leq N - 1), \\ \gamma_i &= i - 1 \quad (2 \leq i \leq N - 1). \end{aligned}$$

The corresponding form of computational formula is

$$\begin{aligned} a_i &= (i + 1) (\hat{y}_i - 2\hat{y}_{i+1} + \hat{y}_{i+2}), & 0 \leq i < N - 3 \\ a_{N-2} &= (N - 1) (\hat{y}_{N-2} - 2\hat{y}_{N-1}), \\ a_{N-1} &= N\hat{y}_{N-1}. \end{aligned}$$

A similar situation with exponentially increasing intervals between samples is described in the Section 3.

### 2.3 The frequency domain calculations

Given the values of the interpolation coefficients in the interpolant (5), we are in a position to estimate the Fourier integrals from  $\hat{h}(x)$ . Using the similarity relationships (9) and (10), we may write

$$\hat{H}_C(u) = \mathfrak{S}_C \left\{ \hat{h}(x) \right\} = \sum_{i=0}^{N-1} a_i x_{i+1} \hat{R}_C(x_{i+1}u), \quad (16)$$

$$\hat{H}_S(u) = \mathfrak{S}_S \left\{ \hat{h}(x) \right\} = \sum_{i=0}^{N-1} a_i x_{i+1} \hat{R}_S(x_{i+1}u). \quad (17)$$

Since  $\hat{h}(x)$  is an interpolant to  $\hat{y}(x)$ , these formulas provide estimates of finite Fourier integrals (3) and (4).

To apply these formulas for carrying out the transformations (1) and (2), we note that function  $\hat{y}(x)$  extends over a finite interval  $[0, L]$  and is zero outside the interval. Meanwhile,  $f(x)$  in the integrals (1) and (2) is a function of an infinite extend which may have non-zero values at  $x = L$  and  $x > L$ . Consequently, we need to establish the relationships between  $\hat{y}(x)$  and  $f(x)$  over the whole infinite range of  $x$  from 0 to  $\infty$ . In this context, we consider the following three cases of the calculations of the transforms (1) and (2).

1. The function  $f(x) \approx 0$  for  $x \geq L$ . This allows us to assume that  $f(x) \approx \hat{y}(x)$  for an infinite range of  $x$  from 0 to  $\infty$ . Therefore,

$$F_C(u) \approx \hat{Y}_C(u) \quad \text{and} \quad F_S(u) \approx \hat{Y}_S(u).$$

Once we have an appreciation of  $\hat{h}(x)$  as an acceptable approximation of  $\hat{y}(x)$ , then we use (16) and (17) as the estimates of desired Fourier integrals (1) and (2):

$$F_C(u) \approx \hat{H}_C(u) \quad \text{and} \quad F_S(u) \approx \hat{H}_S(u). \quad (18)$$

- The function  $f(x) \approx f_L$  for  $x \geq L$ . Such type of function is characteristic for step responses of stable dynamic systems (solid curve 1 in Fig. 5b is an example of such process). In this case we define  $\hat{y}(x)$  in the interval  $[0, L]$  as

$$\hat{y}(x) = f(x) - f_L. \tag{19}$$

Given the approximate representation of  $\hat{y}(x)$  in the form of interpolant  $\hat{h}(x)$ , we define  $f(x)$  on a semi-infinite interval as  $f(x) = \hat{h}(x) + f_L\sigma(x)$ , using a unit step function

$$\sigma(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

A unit step function does not satisfy the conditions which are necessary for its representation by the Fourier integral. A familiar method to overcome this restriction is to interpret  $\sigma(x)$  as the limit for  $\alpha \rightarrow 0$  of an one-sided decaying exponential  $\exp(-\alpha x)$ . This gives,  $\mathfrak{F}_C\{\sigma(x)\} = 0$  and  $\mathfrak{F}_S\{\sigma(x)\} = 1/u$ . Therefore,

$$F_C(u) \approx \hat{H}_C(u) \quad \text{and} \quad F_S(u) \approx \hat{H}_S(u) + \frac{f_L}{u}. \tag{20}$$

- The function  $f(x) = 0$  for  $x > L$  while  $f(L) = f_L$ . Consequently,  $f(x)$  has a discontinuity at the end of the interpolation interval. As in the previous case, we define  $\hat{y}(x)$  in  $[0, L]$  interval by the formula (19). Given  $\hat{h}(x)$ , the  $f(x)$  is defined on a semi-infinite interval as

$$f(x) = \hat{h}(x) + f_L[\sigma(x) - \sigma(x - L)] = \hat{h}(x) + f_L\hat{\rho}(x),$$

where  $\hat{\rho}(x)$  is a rectangle with a non-zero unit amplitude over the interval  $[0, L]$ . Transformation of this function into the frequency domain gives:

$$\begin{aligned} \mathfrak{F}_C\{\hat{\rho}(x)\} &= \hat{P}_C(u) = \frac{\sin uL}{u}, \\ \mathfrak{F}_S\{\hat{\rho}(x)\} &= \hat{P}_S(u) = \frac{1 - \cos uL}{u}. \end{aligned}$$

Therefore, the desired Fourier integrals are:

$$\begin{aligned} F_C(u) &\approx \hat{H}_C(u) + f_L \frac{\sin uL}{u}, \\ F_S(u) &\approx \hat{H}_S(u) + f_L \frac{1 - \cos uL}{u}. \end{aligned} \tag{21}$$

Major calculations for these three cases consist in an estimation of functions  $\hat{H}_C(u)$  and  $\hat{H}_S(u)$ , i.e. the weighted sums of sampled values of functions (7) and (8).

The amount of computations may be significantly reduced by specific ordering of sampling intervals in both primary and transformation domains. An effective algorithm constructed in such a way is presented in the next section.

### 3 Fast SBF algorithm

One major difference of the above algorithms compared to the Fourier series is that the conventional algorithms of numerical harmonic analysis, particularly the FFT, presume uniformly spaced grids of computational points. By contrast, the SBF algorithm accepts non-evenly sampled functions in both primary and transformation domains.

The advantages of non-uniform sampling in the analysis of dynamic systems are associated with the possibility of parsimonious digital representation of dynamic characteristics. In this context, it is widely accepted to use the logarithmic frequency representation of the transfer functions, i.e. to transcribe the frequency characteristics into a digital form using exponentially increasing sampling intervals. The employment of the logarithmic scales is an essential aspect of a number of fundamental frequency domain design methods, for example, the Bode diagrams. The problem of computing the Fourier transforms from exponentially increasing points attracted much attention in the engineering applications [1]. To our knowledge, the SBF algorithm for the first time provides a computationally efficient solution.

We define the sequence of nodal points for interpolation as:

$$x_0 = 0, x_i = x_1 C^{i-1} \quad (i = 1, \dots, N)$$

where  $C$  is a real constant which must satisfy the condition  $C > 1$ .

For  $i > 0$  the points  $x_i$  constitute a geometric progression naturally connected with the employment of the logarithmic scales. Indeed, for  $i$  from 1 to  $N$  we may consider  $\hat{y}(x_i)$  as a sequence of evenly spaced samples in  $x$  logarithmic scale. Accordingly, the data points  $\{x_i, \hat{y}_i\}$  for  $i = 1, \dots, N$  represent the information with which the algorithm deals. The parameter  $C$  establishes the sampling rate which is convenient to define by the number of samples per decade,  $\nu$ . Given desired  $\nu$ ,  $C = \sqrt[\nu]{10}$ .

The first interpolation node  $x_0$  does not belong to the exponentially distributed sequence of the samples of given function. However, this point is important because it corresponds to the beginning of the interval on which the interpolant (5) is defined. The fast SBF algorithm takes  $\hat{y}_0 = \hat{y}_1$ . This implicitly presumes that  $\hat{y}_1$  is close to the actual value of  $\hat{y}(x)$  at  $x = 0$ . In most applications  $\hat{y}(0)$  is zero or finite at  $x = 0$ . The situation with function which goes to infinity as  $x$  approaches zero is exemplified in the next section.

The first sampling interval between nodal points  $x_0$  and  $x_1$  is  $\Delta x_1 = x_1$ . The sampling intervals between the following nodal points are:

$$\Delta x_i = x_i - x_{i-1} = x_1 C^{i-2} (C - 1) \quad (i = 2, \dots, N - 1)$$

The substitution of values of the nodal points and the segmentation intervals into (12) gives

$$\begin{aligned} \alpha_0 &= 1, \\ \alpha_i &= \frac{C}{C-1} \quad (1 \leq i \leq N-1), \\ \beta_1 &= \frac{C}{C-1}, \\ \beta_i &= \frac{C+1}{C-1} \quad (2 \leq i \leq N-1), \\ \gamma_i &= \frac{1}{C-1} \quad (2 \leq i \leq N-1). \end{aligned}$$

We see that with the exception of  $\alpha_0$  and  $\beta_1$ , the interpolation coefficients are no more dependent on the ordering index  $i$  and may be regarded as constants. Using the notations

$$\alpha_C = \frac{C}{C-1}, \quad \beta_C = \frac{C+1}{C-1}, \quad \text{and} \quad \gamma_C = \frac{1}{C-1},$$

we present the computational formulas in the form

$$\begin{aligned} a_0 &= \hat{y}_0 - \beta_1 \hat{y}_1 + \gamma_C \hat{y}_2, \\ a_i &= \alpha_C \hat{y}_i - \beta_C \hat{y}_{i+1} + \gamma_C \hat{y}_{i+2} \quad 1 \leq i < N-3, \\ a_{N-2} &= \alpha_C \hat{y}_{N-2} - \beta_C \hat{y}_{N-1}, \\ a_{N-1} &= \alpha_C \hat{y}_{N-1}. \end{aligned}$$

The corresponding matrix is

$$\mathbf{V} = \begin{pmatrix} 1 & \beta_1 & \gamma_C & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & \alpha_C & \beta_C & \gamma_C & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_C & \beta_C & \gamma_C & \cdots & 0 & 0 & 0 & 0 \\ \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & \alpha_C & \beta_C & \gamma_C & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \alpha_C & \beta_C & \gamma_C \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \alpha_C & \beta_C \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \alpha_C \end{pmatrix}$$

The computational formulas are:

$$\hat{H}_C(u) = x_1 \sum_{i=0}^{N-1} a_i C^i \hat{R}_C(x_1 C^i u), \tag{22}$$

$$\hat{H}_S(u) = x_1 \sum_{i=0}^{N-1} a_i C^i \hat{R}_S(x_1 C^i u). \tag{23}$$

A major computational work consists in the estimation of functions  $\hat{R}_C(u)$  and  $\hat{R}_S(u)$  for different values of  $u$ . To compute  $\hat{H}_C(u)$  (or  $\hat{H}_S(u)$ ) at a single point, it is necessary to compute  $\hat{R}_C(u)$  (or  $\hat{R}_S(u)$ ) at  $N$  points. Consequently, to estimate  $\hat{H}_C(u)$  (or  $\hat{H}_S(u)$ ) at  $M$  points, we need to compute values of  $\hat{R}_C(u)$  (or  $\hat{R}_S(u)$ ) at  $MN$  points.

The idea behind the fast algorithm is to reduce the amount of these calculations via a choice of exponentially spaced values  $u_m = u_1 C^m$  ( $u_1 > 0$ ,  $m = 1, \dots, M$ ) with the same sampling rate as the initial succession of  $x_i$  points. The insertion of these values into (22) and (23) results in

$$\hat{H}_C(u_m) = x_1 \sum_{i=0}^{N-1} a_i C^i \hat{R}_C(z C^{m+i}), \quad (24)$$

$$\hat{H}_S(u_m) = x_1 \sum_{i=0}^{N-1} a_i C^i \hat{R}_S(z C^{m+i}), \quad (25)$$

where  $z = x_1 u_1$ .

These formulas use the values of functions  $\hat{R}_C(u)$  and  $\hat{R}_S(u)$  in the points  $z C^j$  where an integer  $j$  takes values from 0 to  $N + M + 1$ . Compared with the general case, the number of values of  $\hat{R}_C(u)$  (or  $\hat{R}_S(u)$ ) involved in the calculations is reduced from  $NM$  to  $N + M + 1$ . Typically, a reduction in calculations may be quite significant. For example, given typical numbers  $N = 500$  and  $M = 500$  from the following examples, the reduction is 250 fold.

#### 4 Computer experiments

The goal of the computer experiments is to ensure that the algorithms worked properly and the numerical solutions were consistent with the theory. We compared the numerical solutions with precisely known Fourier integrals in order to test the algorithms and evaluate their accuracy. The experiments have been supported by a specially designed computer program written in Borland Delphi. The demonstration version of the program is available from the author by request.

The theoretical solutions to which we refer in the computer experiments represent the dynamic characteristics calculated from the theoretical equations of a harmonic oscillator (see Appendix 2). The numerical solutions represent the estimates of relevant Fourier transforms (see Appendix 1) calculated by the SBF algorithm. Given the data points, the interpolation coefficients have been calculated by the (13) and followed by the frequency domain transformations of defined SBFs using (18), (20) or (21). In the case of the fast SBF the (24) and (25) were employed.

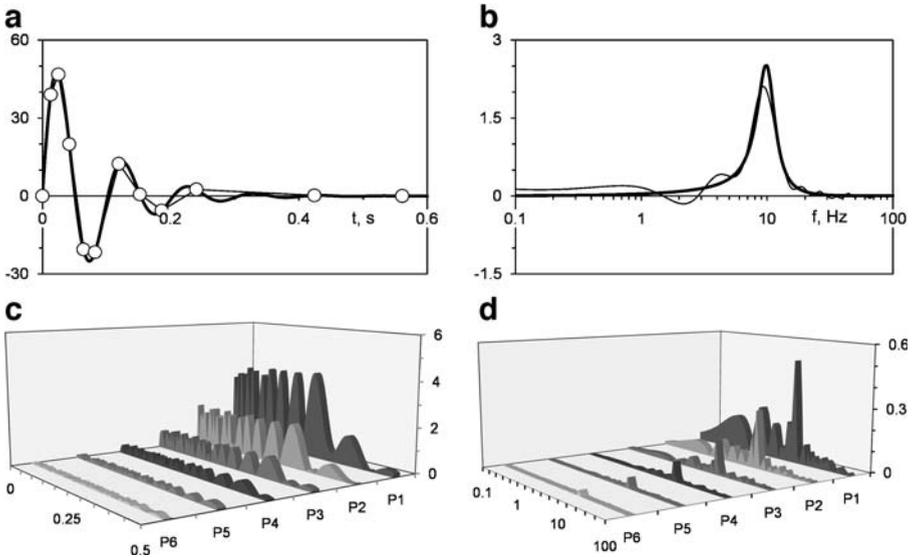
The first series of experiments was focused on the forward Fourier transform (FT), i.e. the conversion of the transient responses from the time domain to the frequency domain. Using the impulse response of the system with parameters

$v_0 = 10$  Hz are  $\zeta = 0.2$  (a dimensionless unit) as the theoretical solution (bold solid curve in Fig. 3a), we examined computational schemes supported by non-uniform and uniform samples of the waveform.

The non-uniform nodal points have been estimated by a data compression algorithm [29] which performs an adaptive segmentation in such a way that the modulus of the error function (the residual between the theoretical and numerical solutions) does not exceed the value of the threshold,  $P$ .

The time series to which the compression algorithm has been applied represented 561 samples of the impulse response (36) on the interval from 0 to 0.56 s (sampling interval 0.001 s). To illustrate the character of the computational errors, we refer to the following six decreasing values of the threshold,  $P_1 = 4$ ,  $P_2 = 2$ ,  $P_3 = 1$ ,  $P_4 = 0.5$ ,  $P_5 = 0.25$  and  $P_6 = 0.125$ . The corresponding numbers of the interpolation nodes (non-redundant points) are: 12, 16, 23, 31, 46 and 65.

The numerical estimation of the forward FTs has shown a similar character of the error distributions for both  $G(\omega)$  and  $B(\omega)$  (equations (34) and (35), respectively). For illustration purposes, we use  $B(\omega)$  which is shown by the bold line in Fig. 3b.



**Fig. 3** The transform of the impulse response (36) to the frequency domain. **a** Theoretical impulse response and piece-wise linear interpolant are shown by the bold and light curves, respectively. **b** The imaginary part (35) of the frequency transfer function. The bold and light curves show the theoretical and numerical solutions, respectively. **c** The ensemble of the absolute values of error functions for different values  $P_1, \dots, P_6$  of the threshold. The absolute value of the difference between the curves displayed in **a** corresponds to the error function for threshold value  $P_1$ . **d** The ensemble of the absolute values of error functions in the frequency domain for different values of the threshold. The error function for the threshold  $P_1$  corresponds to the modulus of the difference between the curves displayed in **b**

In the scales of the plots in Fig. 3a and b, the numerical solutions are indistinguishable from the theoretical ones for threshold values below 1. To give a visual impression of the differences in the curves, we show the numerical solutions in Fig. 3a and b (light traces) for a large threshold value P1.

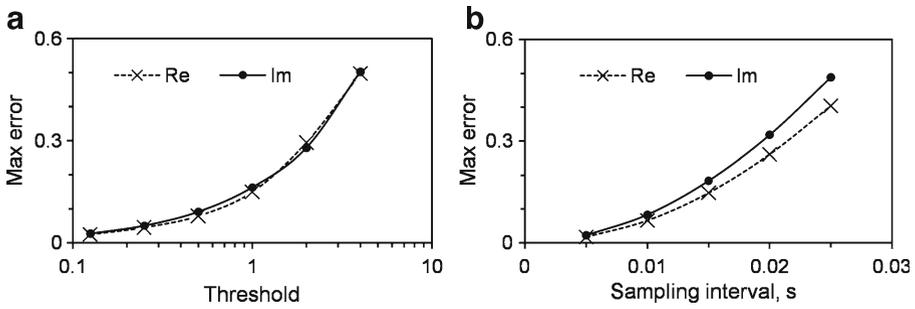
The threshold was varied from P1 to P6 and a consistent decrease of error functions both in the time and frequency domains is illustrated by Fig. 3c and d. The Fig. 3c shows for each value of the threshold, the absolute value of the time domain error function on the interval from 0 to 0.5 s. In a similar manner, the Fig. 3d shows the absolute value of the frequency domain error function in the interval from 0.1 to 100 Hz for each value of the threshold.

Given a particular value of the threshold, a qualitative difference between the error functions in the time and frequency domains is that the time domain errors are characterized by a nearly uniform distribution of the amplitude values across the time interval while the maximum errors in the frequency domain are linked to the maximum values of the frequency characteristic. The time domain aspect of this difference is relevant to the method of adaptive segmentation which keeps the error function in the intervals between the subsequent nodes under the threshold level. With respect to the frequency domain, a larger absolute error for larger value of estimated functions indicates that relative rather than absolute errors have a tendency of a uniform distribution over the frequency scale. This is a desirable property of a robust numerical algorithm. In this context it is important to note that errors of spectral estimates provided by the FFT have a strong tendency to increase with increase of the frequency [6]. By contrast, the Filon's quadrature is free of this highly undesirable trend and may even improve with accuracy as the frequency increases [15].

The ability of the algorithm to provide the required accuracy of the numerical estimates using an appropriate choice of computational parameters is illustrated by Fig. 4a which displays the dependencies of the maximum absolute error in the frequency domain on the value of the threshold for both  $G(\omega)$  and  $B(\omega)$ . It is clear that the improvement in the accuracy of the interpolation in the time domain is reflected in the frequency domain by convergence of the numerical solutions to theoretical solutions.

The computation of the forward FT using equal sampling intervals may be regarded as a limiting case of non-uniform sampling with the threshold for data compression so small that the initial time series representing the given time function remains unchanged. Given the equally spaced samples of the transient waveform, the computer experiments have shown that a decrease in the sampling interval improves the accuracy of the numerical solutions. Figure 4b shows the dependencies of the maximum of the frequency domain error function on the value of the sampling interval for both the real and imaginary parts of the frequency characteristic.

A similar range of values of maximum errors in Fig. 4a and b allows us to compare the effectiveness of the procedures of adaptive and uniform sampling. It appears that the adaptive sampling provides an approximately two-fold reduction in the number of nodal points for interpolation. Considerably higher



**Fig. 4** The errors of numerical estimation of the real (Re) and imaginary (Im) parts of the frequency transfer function [(34) and (35), respectively] from the impulse response (36) are plotted as functions of computational parameters. **a** and **b** display cases of an adaptive and uniform segmentation, respectively

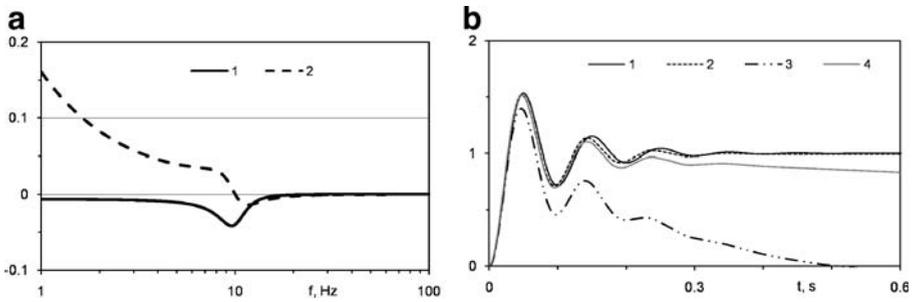
levels of redundancy reduction are achieved if the transient waveform is composed from several components with different time constants.

In the second series of computer experiments, we examined the restoration of transient responses from real and/or imaginary parts of the frequency transfer function. The computations of the inverse FTs have been performed using the fast SBF algorithm. We take values of given frequency characteristic at the points  $f_i = f_1 C^i$  ( $i = 1, \dots, N$ ). Therefore, we deal with  $N$  uniformly spaced samples of frequency characteristic on a logarithmic scale of frequency. According to the construction of the interpolation procedure, the fast algorithm universally adds a point  $f_0 = 0$  and prescribes the value  $F(f_0) = F(f_1)$  to the corresponding sample ( $F$  stands for a symbol of given frequency characteristic).

With respect to the accuracy of computations, the general outcomes of the application of the fast SBF algorithm to the estimation of (30–33) are similar to the previous results in a sense that an improvement of the interpolation accuracy in the primary domain improves the accuracy of the numerical solutions in the transformation domain. However, the implementation of (33) to the step response with a non-zero resting value requires some additional comments.

The problem is illustrated by an example shown in Fig. 5. In terms of (32) and (33), the curves 1 and 2 in Fig. 5a correspond to functions  $-B(\omega)/\omega$  and  $G(\omega)/\omega$ . The time domain counterpart of these functions is the step response (37), shown by the curve 1 in Fig. 5b.

The frequency interval from 1 to 100 Hz is sufficient to account for the major characteristic features of  $-B(\omega)/\omega$ . This conclusion is supported by a high accuracy with which a step response  $v_\sigma(t)$  has been restored from 101 sample of  $-B(\omega)/\omega$  on the interval from 1 to 100 Hz (50 samples per decade). The numerical solution which closely coincides with the theoretical solution is shown by curve 2.



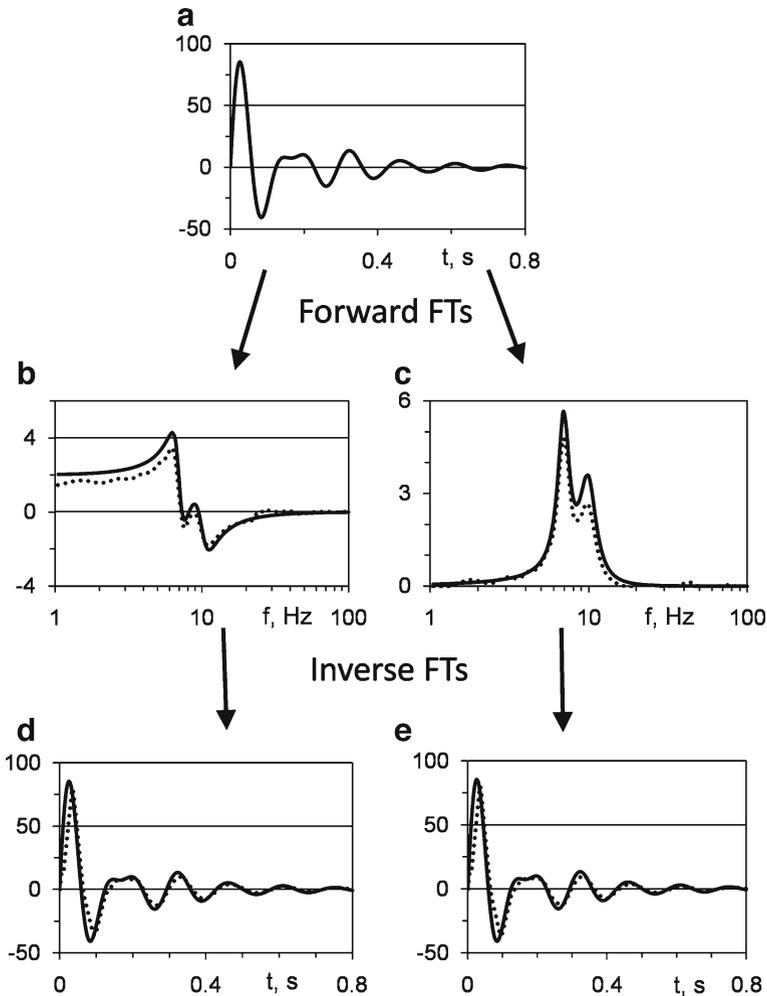
**Fig. 5** **a** The curves 1 and 2 show theoretical functions  $-B(\omega)/\omega$  and  $G(\omega)/\omega$ . **b** The curve 1 is the step response of the system calculated from (37). The curves 2–4 show numerical solutions computed under different sets of computational parameters

The problem with (33) is that  $G(\omega)/\omega$  goes to infinity as  $\omega$  approaches zero. However, the initial value of this function in numerical calculations is finite and corresponds to  $G(\omega_0) = G(\omega_1)$ . This means that we prescribe to  $G(\omega)$  a constant value  $G(\omega_1)$  on the interval from 0 to  $\omega_1 = 2\pi f_1$ . The opportunity to get rid of the errors produced by such a discrepancy from the numerical solutions stems from the fact that the asymptotic behavior of frequency characteristics at  $f \rightarrow 0$  defines the asymptotic behavior of the corresponding time function at  $t \rightarrow \infty$ . Since we search the time domain solution on a finite time interval  $[0, T]$ , we can choose  $f_1$  small enough to remove the error from the interval of interest. To test the sensitivity of the solution to the choice of the frequency  $f_1$ , we calculated step responses from  $G(\omega)/\omega$  given on the intervals of different length. The curve 3 in Fig. 5b depicts the numerical solutions computed from 101 sample of  $G(\omega)/\omega$  on the interval from 1 to 100 Hz, i.e. the same parameters as those used for treatment of  $B(\omega)/\omega$ . The curve progressively deviates from the theoretical solution as the time increases. Such character of disagreement clearly indicates the range of low frequencies as the source of discrepancy. The calculations were repeated for an extended interpolation interval from 0.1 to 100 Hz (151 sample). The result is depicted by curve 4 and shows a considerable improvement in the accuracy of the numerical solution. An excellent match (a practical coincidence of numerical and theoretical solutions in the time scale of Fig. 5) has been achieved by a further extension of the interpolation interval to the range from 0.01 to 100 Hz (201 sample). The encouraging result of these experiments is the capability of the SBF algorithm to provide reasonably accurate numerical solutions when treating a function with singularity. However, this solution's sensitivity to the frequency from which the exponentially distributed samples start, may necessitate a number of trials with different computational parameters. Fortunately, there is a simple way to avoid these complications using (32) instead of (33).

In dealing with theoretical and numerical solutions we are involved in a simultaneous consideration of both the time and frequency domains and in the interpretation of error in these two domains. These numerical experiments

show that we can make the error as small as we wish by an appropriate selection of computational parameters. This provides a basis for a universal way to test the algorithms using the forward FT followed by the inverse FT.

The corresponding numerical procedures include the two major steps illustrated by Fig. 6. The first step is the transition from the time to frequency domain using the forward FT. The second step is the restoration of the transient response from numerical frequency characteristics using the inverse FT. We use as an example a complex oscillatory process  $w(t) = v_{\delta 1}(t) + v_{\delta 2}(t)$ , which represents the sum of the impulse responses of two harmonic oscillators



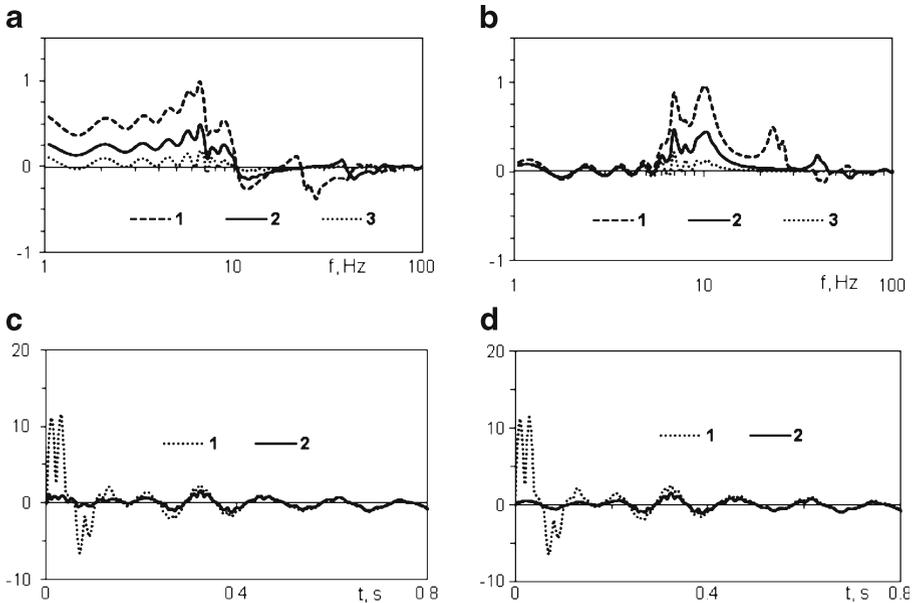
**Fig. 6** Application of the forward FT followed by inverse FT to the composite impulse response  $w(t)$  shown in **a**. The graphs of  $G_W(\omega)$  and  $B_W(\omega)$  are shown in **b** and **c**, respectively. **d** and **e** show  $w(t)$  before (solid lines) and after (dotted lines) numerical restoration

with the following parameters: (1)  $\nu_0 = 7$  Hz,  $\zeta = 0.1$ , (2)  $\nu_0 = 10$  Hz,  $\zeta = 0.15$ . The real and imaginary parts of the complex spectrum of  $w(t)$  are  $G_W(\omega) = G_{W1}(\omega) + G_{W2}(\omega)$  and  $B_W(\omega) = B_{W1}(\omega) + B_{W2}(\omega)$ , where “1” and “2” in the subscripts denote the first and second oscillators, respectively. The graphs of  $w(t)$ ,  $G_W(\omega)$  and  $B_W(\omega)$  calculated from (34), (35) and (36) are shown in Fig. 6 by the solid lines.

Forward FTs have been computed by general procedures of the SBF algorithm from uniformly sampled values of  $w(t)$  taken with the sampling rate  $\Delta t$  in the interval from 0 to 0.8 s. Inverse FTs have been computed by the fast SBF algorithm from exponentially distributed samples of  $G_W(\omega)$  and  $B_W(\omega)$  in the interval from 1 to 100 Hz. Referring to the previous series of computer experiments, we established the frequency domain sampling rate (50 samples per decade) which provides sufficiently accurate estimates of inverse FTs. This allowed us to analyse the accuracy of the whole cycle of calculations using  $\Delta t$  as a parameter. The dotted lines in Fig. 6 show numerical solutions obtained at  $\Delta t = 0.03$  s (28 samples). The character of the errors and their improvement with more concentrated sampling of  $w(t)$  are illustrated by the graphs of error functions in Fig. 7. Graphs in a and b compare the frequency domain errors for three choices of  $\Delta t$ : 0.03, 0.02 and 0.01 (s) (28, 41 and 81 samples, respectively). Increasing number of samples causes the actual error to decrease. A similar effect characterizes the errors of  $w(t)$  restoration. An important aspect of improved computational accuracy is not just the reduction of the errors but also their tendency of a more uniform distribution over the interpolation interval. This is clearly seen in Fig. 7c and d where reduction of  $\Delta t$  from 0.02 to 0.005 (curves 1 and 2, respectively) provides a nearly tenfold decrease of the maximum values of the absolute errors. An interesting point to note is the closeness of numerical estimates of  $w(t)$  separately restored from  $G_W(\omega)$  and  $B_W(\omega)$ . The similarity of the error functions depicted in C and D may be attributed to the fact that one of the functions  $G_W(\omega)$  and  $B_W(\omega)$  can be expressed in terms of the other. The relationships are established by the Hilbert transforms [27].

Complex oscillatory processes with closely spaced resonant peaks are difficult objects for digital spectral analysis using conventional discrete Fourier transform (DFT) [16]. A relatively simple and accurate solution of such a task in our example is partly explained by the effectiveness of the logarithmic frequency scales. However, the major factor is that the SBF algorithm is supported by a piece-wise linear interpolation which provides remarkably better approximation of complex waveforms than interpolation procedures underlying the DFT [33].

Applicability of described procedures of combined forward and inverse FTs to short segments of transient waveforms provides effective tools for the time-frequency analysis of non-stationary processes. Such an approach revealed different frequency domain patterns of characteristic waveforms of human electrocardiogram (ECG) [3]. A crucial finding is that initiation of the R wave of the ECG during a normal heart contraction is described by an equation of underdamped harmonic oscillator the parameters of which typically belong



**Fig. 7** The error functions (the differences between theoretical and numerical solutions) computed for different values of the time domain sampling interval  $\Delta t$ . The frequency domain error functions are shown in **a** for  $G_W(\omega)$  and in **b** for  $B_W(\omega)$ . The curves 1, 2 and 3 correspond to the sampling intervals 0.03, 0.02 and 0.01 (s), respectively. **c** and **d** show the error functions for  $w(t)$  restored from  $G_W(\omega)$  and  $B_W(\omega)$ , respectively. The curves 1 and 2 correspond to the sampling intervals 0.02 and 0.005 (s), respectively

to the following ranges: natural frequency from 12 to 17 Hz, damping ratio of heart contractions  $\zeta$  from 0.35 to 0.5. The dynamics of heart contractions varies from beat to beat. Having transformed the ECG segment to the frequency domain, the inverse Fourier transform (fast SBF algorithm) affords the means to handle these tiny aspects of the heart performance in the time domain where the ECG data are initially specified.

### 5 Concluding remarks

The SBF algorithm and its fast version are developed for estimation of sine and cosine Fourier integrals using the function decomposition into a sum of similar basis functions. We can conclude with confidence from our computer experiments that the SBF algorithms are stable in the sense that the increase of the interpolation accuracy makes the numerical solution to converge towards the theoretical one. A beneficial aspect of high (in fact arbitrary) order accuracy of the algorithm is that the error distributions in the transformation domain did not show some unexpected, “pathological” features. Meanwhile, the spectral leakage induced by the discontinuities at the boundaries of the function under

analysis is a major challenge in the applications of conventional FFT to the estimation of Fourier integrals and in the short-time spectral analysis. The usual remedies of error reduction by windowing and zero-padding introduce problems of their own.

The SBF algorithm permits an explicit treatment of discontinuities at the boundary points. The solution is sought in the very form of the interpolation function in which the first term represents a discontinuous component of the solution while the rest of the basis functions approximate smooth waveforms.

The broadest conclusions to be drawn from the comparison of the SBF algorithm with conventional FFT are as follows.

1. Conceptually, the SBF algorithm deals with a continuous Fourier spectrum instead of a discrete spectrum defined by the discrete Fourier transform and implemented by the FFT algorithm.
2. The transcription of the continuous functions into a digital form accepts non-uniform sampling intervals in both primary and transformation domains. The fast SBF algorithm draws on novel principles of the design of effective algorithms supported by these opportunities.
3. The need for windows for spectral analysis is eliminated, along with their distorting impact.
4. The algorithm may be applied to signal segments of arbitrary lengths.

### Appendix 1: Linear system analysis using Fourier Integrals

This section gives a survey of basic relationships which support the frequency domain analysis of a linear stationary system (LSS) [2, 8, 27]. Given an input force applied to the system, let the response be a continuous time function  $v(t)$  which satisfies the conditions of its representation by Fourier integral. Then it may be presented as

$$v(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} V(\omega) \exp(i\omega t) d\omega, \quad (26)$$

where

$$V(\omega) = \int_{-\infty}^{\infty} v(t) \exp(-i\omega t) dt \quad (27)$$

is a spectral density function,  $i = \sqrt{-1}$  and  $\omega$  is the angular frequency connected with the frequency  $f$  as  $\omega = 2\pi f$ .

This classical pair of Fourier transforms bears a dual relationship. Essential aspects are the infinite limits of integration and complex form of both the frequency and time domain functions. Actually,  $v(t)$  is regarded as a real time function while  $V(\omega)$  is a complex spectrum. In terms of real functions  $V(\omega) = V_C(\omega) - iV_S(\omega)$ . If LSS is physically realizable, the response of the

system on the application of the input function at  $t = 0$  appears at  $t \geq 0$ . Accordingly, the frequency domain representation of  $v(t)$  in terms of real functions is

$$V_C(\omega) = \int_0^\infty v(t) \cos(\omega t) dt, \quad V_S(\omega) = \int_0^\infty v(t) \sin(\omega t) dt. \quad (28)$$

The inversion formulas provide means to restore  $v(t)$  from only the real  $V_C(\omega)$  or imaginary  $V_S(\omega)$  parts of the complex spectrum. A common approach [27] to obtain these special forms of Fourier integrals uses decomposition of  $v(t)$  into its even and odd parts by writing  $v(t) = e(t) + o(t)$ , where  $e(-t) = e(t)$ ,  $o(-t) = -o(t)$ . Referring to the symmetry properties of cosine and sine Fourier transforms [8], we readily obtain from (27) that

$$V_C(\omega) = \int_{-\infty}^\infty e(t) \cos(\omega t) dt, \quad V_S(\omega) = \int_{-\infty}^\infty o(t) \sin(\omega t) dt,$$

where  $V_C(\omega)$  and  $V_S(\omega)$  are respectively even and odd functions of  $\omega$ . Substitution of these functions into (26) and separation into two integrals yields the following inverse transforms

$$e(t) = \frac{1}{2\pi} \int_{-\infty}^\infty V_C(\omega) \cos(\omega t) d\omega, \quad o(t) = \frac{1}{2\pi} \int_{-\infty}^\infty V_S(\omega) \sin(\omega t) d\omega.$$

Here the limits of integration may be extended only over positive values of  $\omega$  and the result multiplied by 2, since the integrands are even functions. Given that  $v(t)$  is zero for  $t < 0$ , it is clear that  $e(t)$  and  $o(t)$  cancel for  $t < 0$ , and hence must be equal for  $t > 0$ . Therefore,  $v(t)$  for  $t > 0$  is given either twice  $e(t)$  or twice  $o(t)$ , and we obtain

$$v(t) = \frac{2}{\pi} \int_0^\infty V_C(\omega) \cos(\omega t) d\omega = \frac{2}{\pi} \int_0^\infty V_S(\omega) \sin(\omega t) d\omega. \quad (29)$$

We see that the real or imaginary part of the complex spectrum is sufficient to estimate the time response.

Formulas (29) are directly applicable to the estimation of an impulse response,  $v_\delta(t)$ , of a stable physically realizable LSS, i.e. the transient induced in the system by the application of a unit impulse (Dirac delta function). Let  $H(\omega) = G(\omega) - iB(\omega)$  be a complex frequency transfer function of the system. Given that the complex spectrum of a unit impulse is 1, the real and imaginary parts of the complex spectrum of  $v_\delta(t)$  are  $B(\omega)$  and  $G(\omega)$ . Therefore,

$$v_\delta(t) = \frac{2}{\pi} \int_0^\infty G(\omega) \cos(\omega t) d\omega, \tag{30}$$

$$v_\delta(t) = \frac{2}{\pi} \int_0^\infty B(\omega) \sin(\omega t) d\omega. \tag{31}$$

Another standard dynamic characteristic of LSS is a step response,  $v_\sigma(t)$ , i.e. the transient induced by a forcing unit step function the complex spectrum of which is  $1/i\omega$ . The special forms of Fourier integrals for this case are

$$v_\sigma(t) = G(0) - \frac{2}{\pi} \int_0^\infty \frac{B(\omega)}{\omega} \cos(\omega t) d\omega. \tag{32}$$

$$v_\sigma(t) = \frac{2}{\pi} \int_0^\infty \frac{G(\omega)}{\omega} \sin(\omega t) d\omega, \tag{33}$$

**Appendix 2: The dynamic characteristics of harmonic oscillator**

A harmonic oscillator with dynamic characteristics used in our computer experiments is described by a second order differential equation:

$$\frac{d^2v}{dt^2} + 2\zeta\Omega_0 \frac{dv}{dt} + \Omega_0^2 v = 0,$$

where  $\Omega_0 = 2\pi\nu_0$ ,  $\nu_0$  is the natural frequency and  $\zeta$  is the damping ratio.

The general form of the system’s transfer function is

$$H(s) = \frac{\Omega_0^2}{s^2 + 2\zeta\Omega_0 s + \Omega_0^2},$$

where  $s$  is a complex variable. After substitution of  $s = i\omega$ , the real,  $G(\omega)$ , and imaginary,  $B(\omega)$ , parts of the frequency transfer function  $H(i\omega)$  are defined as:

$$G(\omega) = \frac{1 - \left(\frac{\omega}{\Omega_0}\right)^2}{\left(1 - \left(\frac{\omega}{\Omega_0}\right)^2\right)^2 + \left(\frac{2\zeta\omega}{\Omega_0}\right)^2}, \tag{34}$$

$$B(\omega) = \frac{2\zeta \left(\frac{\omega}{\Omega_0}\right)}{\left(1 - \left(\frac{\omega}{\Omega_0}\right)^2\right)^2 + \left(\frac{2\zeta\omega}{\Omega_0}\right)^2}. \quad (35)$$

The impulse and step responses of the system at  $t \geq 0$  are:

$$v_\delta(t) = \frac{\Omega_0}{\sqrt{1 - \zeta^2}} e^{-\zeta\Omega_0 t} \sin\left(\sqrt{1 - \zeta^2}\Omega_0 t\right), \quad (36)$$

$$v_\sigma(t) = 1 - e^{-\zeta\Omega_0 t} \left( \cos \Omega_0 t + \frac{\zeta}{\sqrt{1 - \zeta^2}} \sin \Omega_0 t \right). \quad (37)$$

## References

1. Ametani, A., Imanishi, K.: Development of exponential Fourier transform and its application to electrical transients. *Proc. IEEE* **126**, 51–56 (1979)
2. Aseltine, J.A.: *Transform Method in Linear System Analysis*. McGraw-Hill, New York (1958)
3. Bahramali, H., Melkonian, D., O'Connell, O.: Self regulation of the heart: natural frequency and damping of the heart contractions. *Open Cybern. Syst. J.* **2**, 1–10 (2008)
4. Beaudoin, N.: A high-accuracy mathematical and numerical method for Fourier transform, integral, derivative, and polynomial splines of any order. *Can. J. Phys.* **76**, 659–677 (1998)
5. Beaudoin, N., Beauchemin, S.S.: A new numerical Fourier transform in d-dimensions. *IEEE Trans. Signal Process.* **51**, 1422–1430 (2003)
6. Becker, R.I., Morrison, N.: The errors in FFT estimation of the Fourier transform. *IEEE Trans. Signal Process.* **44**, 2073–2077 (1996)
7. Blumenthal, T., Melkonian, D.: A model based approach to quantitative analysis of eyeblink EMG responses. *J. Psychophysiol.* **17**, 1–11 (2003)
8. Bracewell, R.N.: *The Fourier Transform and its Applications*. McGraw-Hill, New York (1986)
9. Chua, L., Ng, C.: Frequency domain analysis of nonlinear systems: general theory, formulation of transfer functions. *IEEE J. Electron. Circuits Syst.* **3**, 165–185 (1979)
10. Cohen, L.: Time–frequency distributions—a review. *Proc. IEEE* **77**, 941–981 (1989)
11. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier Series. *Math. Comput.* **19**, 7–301 (1965)
12. Filon, L.N.G.: On a quadrature formula for trigonometric integrals. *Proc. R. Soc. Edin.* **49**, 38–47 (1929)
13. Harris, C.M.: The Fourier analysis of biological transients. *J. Neurosci. Methods* **83**, 15–34 (1998)
14. Harris, F.J.: On the use of windows for harmonic analysis with the discrete Fourier transform. *Proc. IEEE* **66**, 51–83 (1978)
15. Iserles, A.: On the numerical quadrature of highly-oscillating integrals I: Fourier transforms. *IMA J. Numer. Anal.* **24**, 365–391 (2004)
16. Kay, S.M., Marple, S.L.: Spectrum analysis—a modern perspective. *Proc. IEEE* **69**, 1380–1419 (1981)
17. Lyness, J.N.: The calculation of trigonometric Fourier coefficients. *J. Comput. Phys.* **54**, 57–73 (1984)
18. Mäkinen, S.: New algorithm for the calculation of the Fourier transform of discrete signals. *Rev. Sci. Instrum.* **53**, 6–630 (1982)
19. Melkonian, D.: *Transients in Neuronal Systems*. Armenian Academy of Sciences, Yerevan (1987, in Russian)
20. Melkonian, D., Blumenthal, T., Gordon, E.: Numerical Fourier transform spectroscopy of EMG half-waves: fragmentary-decomposition-based approach to nonstationary signal analysis. *Biol. Cybern.* **81**, 457–467 (1999)

21. Melkonian, D., Blumenthal, T., Meares, R.: High resolution fragmentary decomposition—a model based method of non-stationary electrophysiological signal analysis. *J. Neurosci. Methods* **131**, 149–159 (2003)
22. Melkonian, D., Gordon, E., Bahramali, H.: Single-event-related potential analysis by means of fragmentary decomposition. *Biol. Cybern.* **85**, 219–2 (2001)
23. Melkonian, D., Gordon, E., Rennie, C., Bahramali, H.: Dynamic spectral analysis of event-related potentials. *Electroencephalogr. Clin. Neurophysiol.* **108**, 251–259 (1998)
24. Melkonian, D., Meares, R., Bahramali, H., Harris, A., Williams, L.: Numerical Fourier transform spectroscopy of EEG waveforms using similar basis function algorithm. *Trends Appl. Spectrosc.* **6**, 69–77 (2007)
25. Noble, B.: Variational finite element methods for initial value problems. In: Whiteman, J.R. (ed.) *The Mathematics of Finite Elements and Applications*, pp. 143–151. Academic, London (1973)
26. Olver, S.: Numerical approximation of highly oscillatory integrals. PhD thesis, University of Cambridge, Trinity Hall (2007)
27. Papoulis, A.: *The Fourier Integral and its Applications*. McGraw-Hill, New York (1962)
28. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge (1993)
29. Ruttiman, U.E., Pipberger, H.V.: Compression of the ECG by prediction or interpolation and entropy encoding. *IEEE Trans. Biomed. Eng.* **26**, 613–623 (1979)
30. Schütte, J.: New fast Fourier transform algorithm for linear system analysis applied in molecular beam relaxation spectroscopy. *Rev. Sci. Instrum.* **52**, 400–404 (1981)
31. Sorella, S., Ghosh, S.K.: Improved method for the discrete fast Fourier transform. *Rev. Sci. Instrum.* **55**, 1348–1352 (1984)
32. Zadeh, L.A.: Frequency analysis of variable networks. *Proc. IRE* **38**, 1–9 (1950)
33. Zeng, P.: High-accuracy formula for discrete calculation of fourier transforms. *Appl. Math. Comput.* **106**, 117–140 (1999)