

Fractional-order linear digital 1D and 2D filter response calculation using Matlab

Piotr W. Ostalczyk¹

Received: 2 November 2015 / Revised: 12 January 2016 / Accepted: 16 January 2016 / Published online: 8 February 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract In this paper a simple method of the fractional-order linear digital filter response calculation is proposed. The fractional-order linear digital filter description is based on the fractional-order linear time-invariant difference equation and related fractional-order discrete transfer function.

Keywords Fractional calculus · Linear discrete system · Signal processing · Image processing

1 Introduction

Digital filters (DF) are a very important part of digital signal processing [1]. It refers to signals varying with time (considered further as 1D signals) as well as varying in space (considered further as 2D signals). The mentioned signals usually include errors due to different reasons from which one can mention the imperfection of measuring devices or sensors so the separation of the signal from the errors is necessary. In 1D signals case many different synthesis methods of DFs are developed [2,3]. Similar methods are used to filter 2D signals represented mainly by digital images [4–6]. The mentioned techniques base on the classical calculus in the discrete version, where derivatives are replaced by differences and integrals by sums. In 1D, as well as in 2D DF description, the discrete convolution is used.

Successful applications of the fractional calculus [7–11] in the era of universal digitalization cause the expansion of discrete applications in various fields of science and technology. A substitution of the left fractional-order derivative by

a fractional-order backward difference (FOBD) leads to the fractional-order (FO) difference equation (DE) [12]. In the linear time-invariant FODE case one can evaluate the FO discrete transfer function (FODTF). Such FODEs and FODTFs will be used in the 1D and 2D FODFs description.

The main result of the paper is the simplicity of mentioned filters outputs evaluation by Matlab. No specialized toolbox is needed. First, the 1D-FODF is synthesised and applied to filter the measured real signal. The signal filtration effects are shown via their discrete frequency characteristics. Then, it is shown that the 1D-FODF is helpful in FO image filtering. 2D-FODF characterised by multidirectional mask supported by 1D-FODF may be useful in the discrete image filtering. An immense progress in the digital processors calculation abilities proves that one can cross the usually used 3×3 and 5×5 masks in the pixel convolution evaluation [13]. A supplementary advantage of the paper is the indication that there is an additional possibility to shape the FODF frequency characteristics by releasing integer orders. Supplementary order coefficients lead to better filtration effects.

The paper is organised as follows. In Sect. 2 fundamental notions related to the FOBD, coefficients function $a^{(\mu)}(k)$ of one discrete-variable k , FODE and FOTF are given. Section 3 contains a proposal of the 1D-FODF synthesis. Considerations are supported by numerical examples in Sect. 4 where real measured signal filtration effects are presented due to the very simple calculation method. The next section contains a definition of multidirectional mask serving to filter a digital image. The 2D-FODF output for a given image is presented.

2 Mathematical preliminaries

The fundamental notion used in the FOF-1DF description is the FOBD in the so-called Grünvald–Letnikov form [12]. Its definition is given below.

✉ Piotr W. Ostalczyk
postalcz@p.lodz.pl

¹ Institute of Applied Computer Science, Lodz University of Technology, 90-924 Lodz, Poland

$$\begin{aligned}
 {}^{GL}\Delta_k^{(\mu)} f(k) &= \sum_{i=k_0}^k a^{(\mu)}(i - k_0) f(k + k_0 - i) \\
 &= \sum_{i=0}^{k-k_0} a^{(\mu)}(i) f(k - i),
 \end{aligned} \tag{1}$$

where

$$a^{(\mu)}(k) = \begin{cases} 0 & \text{for } k < 0 \\ 1 & \text{for } k = 0 \\ (-1)^k \frac{(\mu)((\mu)-1)\dots((\mu)-k+1)}{k!} & \text{for } k = 1, 2, 3, \dots \end{cases} \tag{2}$$

The left and right subscripts in ${}^{GL}\Delta_k^{(\mu)} f(k)$ denote the FOBD evaluation range, respectively. The left and right superscripts stand for the FOBD form (Grünwald–Letnikov) and order (μ) , respectively. (μ) indicates that the FOBD is not raised to the μ power. Without loss of generality from this point on one assumes $k_0 = 0$. It can be proved [12] that for $0 < (\mu) < 2$

$$\lim_{k \rightarrow +\infty} a^{(\mu)}(k) = 0 \text{ and } \sum_{i=0}^{+\infty} a^{(\mu)}(i) = 0 \tag{3}$$

Assuming $f(k) = 0$ for $k < 0$ and zero initial conditions one can prove that the one-sided \mathcal{Z} -Transform of the FOBD is as follows

$$\mathcal{Z} \left\{ {}^{GL}\Delta_k^{(\mu)} f(k) \right\} = (1 - z^{-1})^\mu F(z) = \sum_{i=0}^{+\infty} a^{(\mu)}(i) z^{-i} \tag{4}$$

where $\mathcal{Z} \{ f(k) \} = F(z)$.

The description of linear fractional-order digital 1D and 2D filters considered in this paper bases on the FOBD. One considers the FO commensurate difference equation (FODE) of the form

$$\sum_{i=0}^p A_i {}^{GL}\Delta_k^{(i\mu)} y(k) = \sum_{j=0}^q B_j {}^{GL}\Delta_k^{(j\mu)} u(k) \tag{5}$$

where $0 < \mu \leq 1$ —FO, $p \geq q$, A_i, B_j —constant coefficients, $A_p = 1, y(k), u(k)$ —filter output and filtered signal, respectively. The considered form means that the processed signal sample depends only on present and past samples. No forward shift of samples is admitted. One assumes

$$\sum_{i=0}^p A_i = A^{-1} \neq 0. \tag{6}$$

FODE (5) can be equivalently expressed in a vector form

$$\begin{aligned}
 \sum_{i=0}^p A_i [1 \ a^{(i\mu)}(1) \ \dots \ a^{(i\mu)}(k)] \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(0) \end{bmatrix} \\
 = \sum_{j=0}^q B_j [1 \ a^{(j\mu)}(1) \ \dots \ a^{(j\mu)}(k)] \begin{bmatrix} u(k) \\ u(k-1) \\ \vdots \\ u(0) \end{bmatrix}
 \end{aligned} \tag{7}$$

from which one immediately gets the solution

$$\begin{aligned}
 y(k) = A \left\{ - \sum_{i=0}^p A_i [1 \ a^{(i\mu)}(1) \ \dots \ a^{(i\mu)}(k)] \begin{bmatrix} y(k-1) \\ y(k-2) \\ \vdots \\ y(0) \end{bmatrix} \right. \\
 \left. + \sum_{j=0}^q B_j [1 \ a^{(j\mu)}(1) \ \dots \ a^{(j\mu)}(k)] \begin{bmatrix} u(k) \\ u(k-1) \\ \vdots \\ u(0) \end{bmatrix} \right\}.
 \end{aligned} \tag{8}$$

Applying the one-sided \mathcal{Z} -Transform with zero initial conditions to both sides of (7) one evaluates the FO discrete transfer function (FODTF)

$$\begin{aligned}
 G(z) &= \frac{Y(z)}{U(z)} = \frac{\sum_{i=0}^q B_i (1 - z^{-1})^{i\mu}}{\sum_{i=0}^p A_i (1 - z^{-1})^{i\mu}} \\
 &= \frac{\sum_{i=0}^q B_i \sum_{l=0}^{+\infty} a^{(i\mu)}(l) z^{-l}}{\sum_{i=0}^p A_i \sum_{l=0}^{+\infty} a^{(i\mu)}(l) z^{-l}} \\
 &= \frac{\sum_{i=0}^{+\infty} \bar{B}_i z^{-i}}{\sum_{i=0}^{+\infty} \bar{A}_i z^{-i}} \approx \frac{\sum_{i=0}^L \bar{B}_i z^{-i}}{\sum_{i=0}^L \bar{A}_i z^{-i}}
 \end{aligned} \tag{9}$$

where $\bar{A}_j = \sum_{i=0}^p A_i a^{(i\mu)}(p-j)$ for $j = 0, 1, \dots, p$ and $\bar{B}_j = \sum_{i=0}^q B_i a^{(i\mu)}(q-j)$ for $j = 0, 1, \dots, q$. The approximation in (9) represented by the substitution of $+\infty$ by relatively large number L . This is justified noting that coefficients $a^{(i\mu)}(k)$ strongly diminish for growing k . Performing a long division of numerator and denominator polynomials one gets the form

$$G(z) = g(0) + g(1)z^{-1} + g(2)z^{-2} + \dots \tag{10}$$

with $g(0), g(1), g(2), \dots$ being the consecutive values of the filter Dirac pulse response.

3 Linear time-invariant digital FOF frequency characteristics

For $p = 1, q = 0$ from (9) one gets

$$F_{\mu_1}(z) = \frac{Y(z)}{U(z)} = \frac{B_0}{(1 - z^{-1})^{\mu_1} + A_0} \approx \frac{B_0}{1 + A_0 + a^{(\mu_1)}(1)z^{-1} + \dots + a^{(\mu_1)}(L)z^{-L}} \tag{11}$$

It is worth mentioning that for the above TFs there are related DEs

$${}^G L \Delta_k^{(i)} y(k) + A_0 y(k) = B_0 u(k), \quad i = 1, \mu \tag{12}$$

The FOF can be considered as l connected in series simple filters (11)

$$F_{\mu_i}(z) = \frac{Y_i(z)}{U_i(z)} = \frac{B_{0,i}}{(1 - z^{-1})^{\mu_i} + A_{0,i}} \approx \frac{B_{0,i}}{1 + A_{0,i} + a^{(\mu_i)}(1)z^{-1} + \dots + a^{(\mu_i)}(L)z^{-L_i}}$$

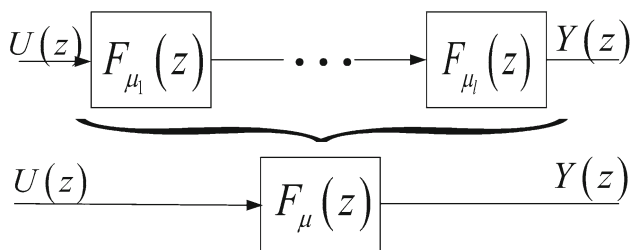


Fig. 1 FODFs connection

where $i = 1, 2, \dots, l$ and $0 < \mu_i \leq 1$ resulting in the FOFTF

$$F(z) = \frac{Y(z)}{U(z)} = \prod_{i=1}^l F_{\mu_i}(z) = \prod_{i=1}^l \left[\frac{B_{0,i}}{(1 - z^{-1})^{\mu_i} + A_{0,i}} \right] \tag{13}$$

A serial connection is depicted in Fig. 1.

The FODF properties are characterized by the same mathematical tools as the classical integer-order ones. Here, two fundamental ones can be mentioned: the FODF unit step response and the discrete frequency characteristics set [2,3]. In the following numerical example one presents an example of the FODF characteristics.

Consider the FODFs of the FOs $i\mu = 1.2$ for $\mu = 1.2$ and $i = 1, 2, 3$ with $A_0 = 0.01$ and $B_0 = A_0 + \sum_{i=0}^L a^{(\mu)}(i)$. Their fundamental frequency characteristics of the FOFs: the discrete Bode magnitude (left) and phase (right) plots are presented in Fig. 2 together with the IOF ones simulated for $\mu = m = 1$, respectively. In Fig. 3 (left) appropriate Nyquist plots are presented.

The considered FODFs characteristics relations with the colors of plots are indicated in the Table 1 given below.

The selected FO Nyquist and related Bode plots reveal growing maximum of amplitudes due to the order $i\mu$ for $i = 1, 2, 3$. One can see that very small extrema in the Bode magnitude plots are no more visible in the Nyquist characteristics. Moreover, this shows a commonly known relation to the unit step responses of the integer order systems.

4 FOF response calculation

The 1D-FODF solution in Matlab is particularly simple. Consider any measured type of a digital signal. It may be voltage,

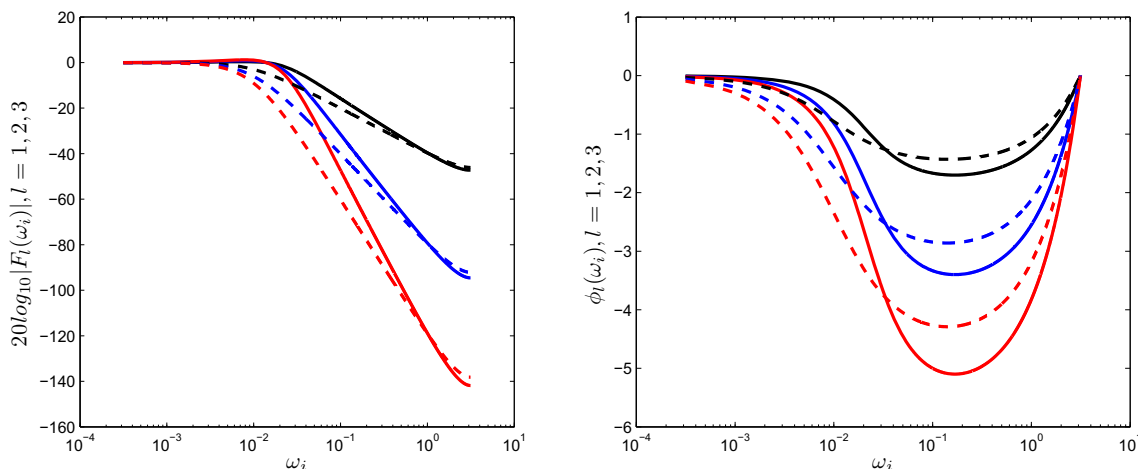


Fig. 2 Discrete Bode magnitude (left) and phase plots (right)

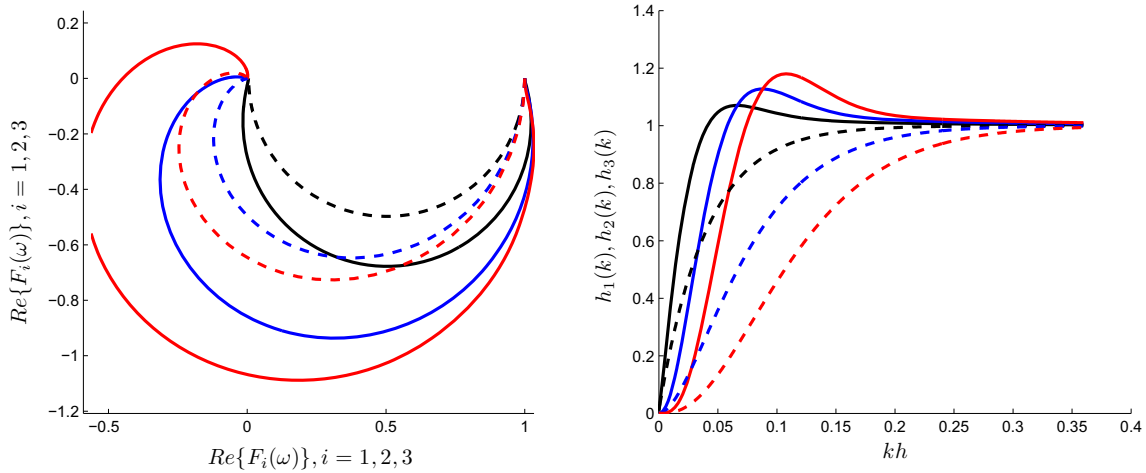


Fig. 3 Discrete Nyquist (*left*) and step responses (*right*) plots

Table 1 Relations between indices

| l | Color μ | Color m |
|----------|-------------------------------|-----------------------------|
| 1 | Black solid-line | Black dotted-line |
| 2 | Blue solid-line | Blue dotted-line |
| 3 | Red solid-line | Red dotted-line |

current, position or speed. For its measured values sequence $u = [u(0)u(1) \cdots u(k_{max})]$ and chosen L one evaluates two polynomials represented by their coefficients

$$n1 = [b_0], \quad d1 = [1 + a_0 a^{(\mu)}(1) a^{(\mu)}(2) \cdots a^{(\mu)}(L)] \tag{14}$$

Then, in the Matlab program, the FODF responses for $l = 1, 2, 3$ have the form

$$\begin{aligned} y1 &= filter(n1, d1, u) \\ y2 &= filter(n1, d1, filter(n1, d1, u)) \\ y3 &= filter(n1, d1, filter(n1, d1, filter(n1, d1, u))) \end{aligned} \tag{15}$$

The three filters unit step responses are plotted in Fig. 3 (right). Very small (almost invisible), but growing maxima (due to growing l) in the Bode magnitude plots in Fig. 2 (left) induce the overshoots in the unit step responses.

Now one presents the filtration effects of the FOFs considered in the previous section. For the discrete signal presented in Fig. 4 one obtains responses presented in Fig. 5. Even better information concerning the filtration effects is given in Figs. 6 and 7 where the Bode magnitude characteristics of the measured signal (left) and the first filter output (right) are plotted, respectively. It can be seen clearly that the filtered signal includes information related to the velocity-position

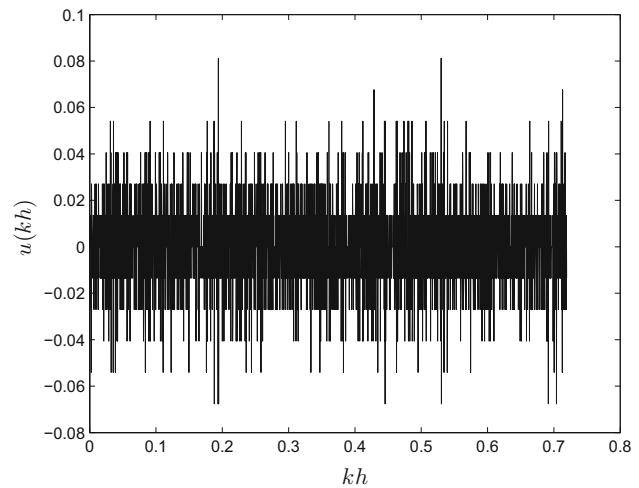


Fig. 4 Measured discrete-time signal

effect. The algorithm is valid for any measured signals, for instance: mechanical, electrical, biological [14].

The satisfactory filtration effect is obtained for three FOFs connected in series. The high frequency noise is eliminated. The classical integer-order filters show weaker filtration effects. This is due to the restrictions to the integer orders only.

5 2D-FOF definition

5.1 2D-FOF directions definition

On a discrete plane (2D discrete space) one defines so-called Bresenham circles [15]. A circle is characterised by center $c(k_1, k_2)$ and radius r . All discrete points (pixels) containing the continuous circle centered in the middle of $x(k_1, k_2)$

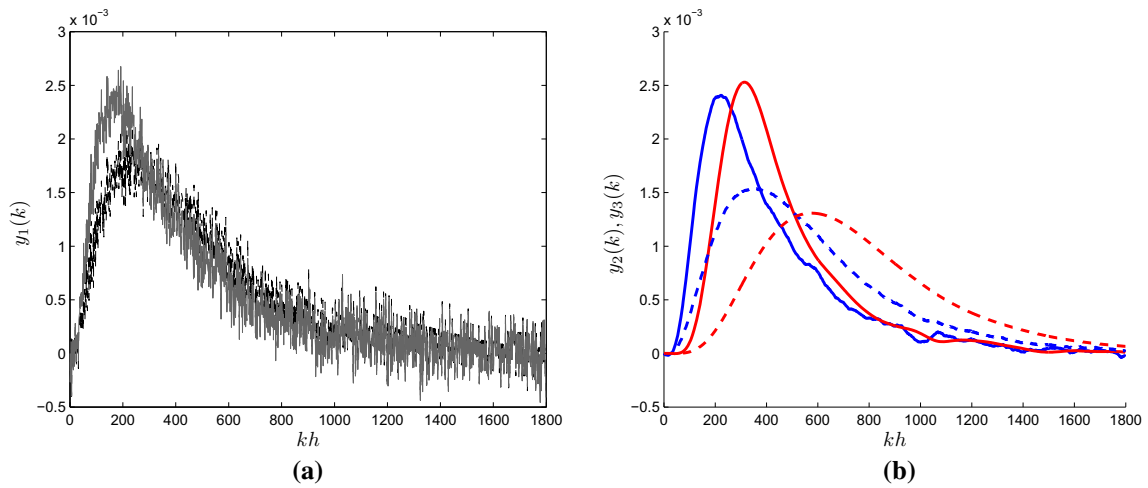


Fig. 5 FOFs and IOFs 1 (a) and 2, 3 (b) responses

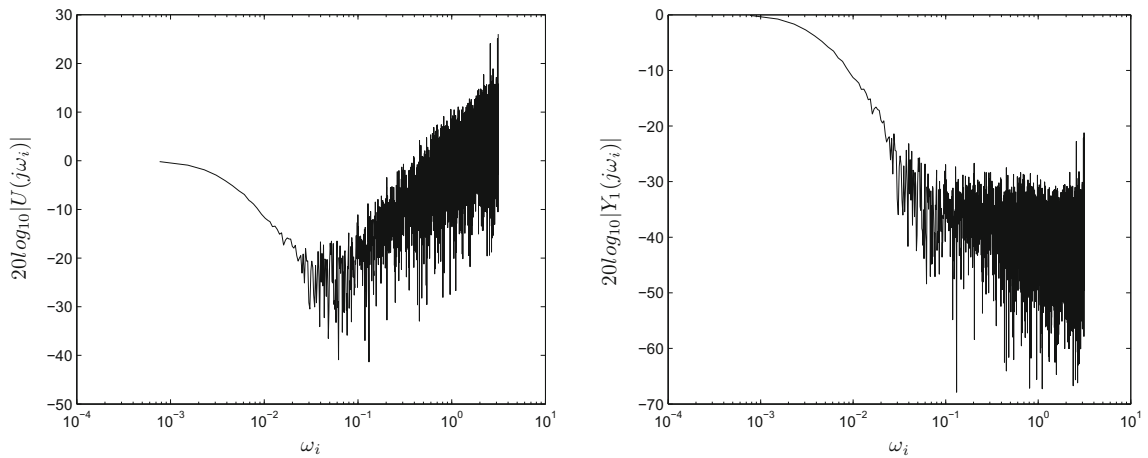


Fig. 6 Bode magnitude plot of the measured signal (left) and first filter output (right)

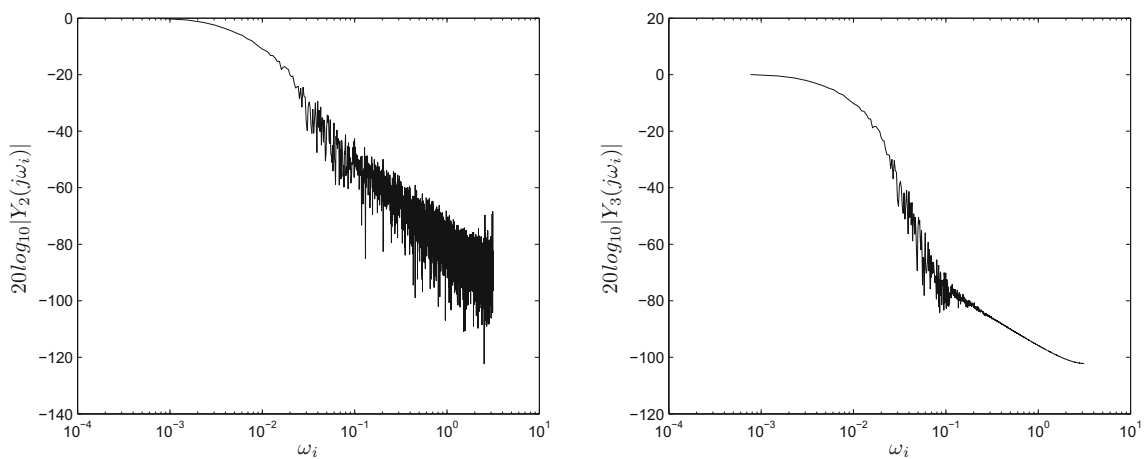


Fig. 7 Bode magnitude plot of the second filter output (left) and the third filter output (right)

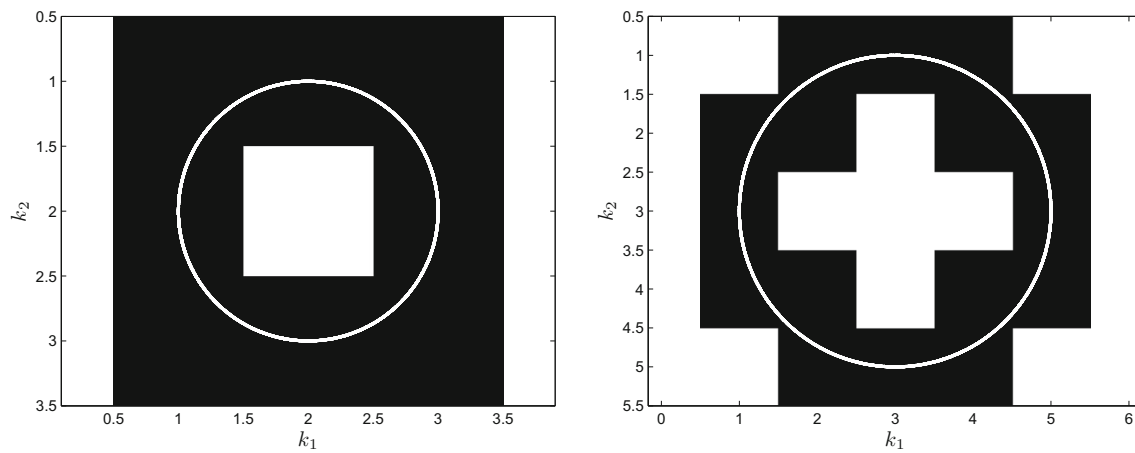


Fig. 8 Discrete circle of radius $r = 1$ (left) $r = 2$ (right)

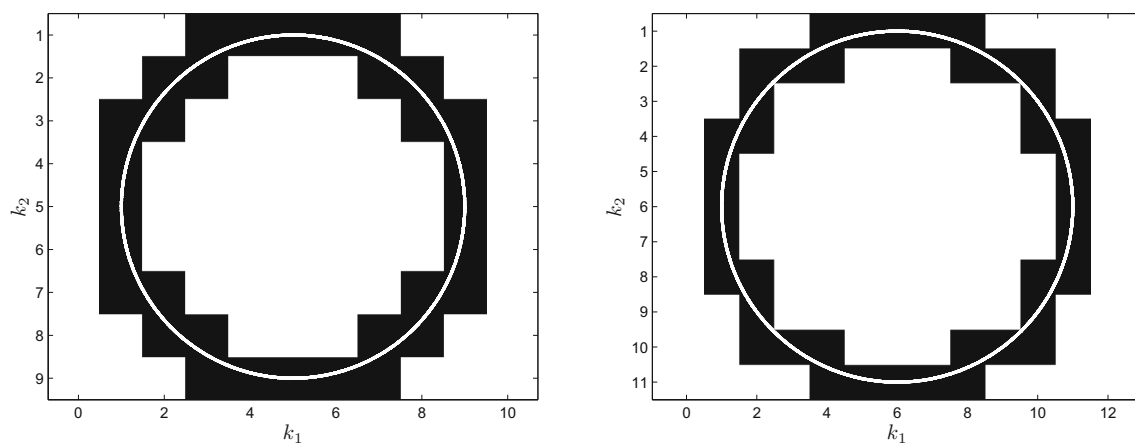


Fig. 9 Discrete circle of radius $r = 4$ (left) and $r = 5$ (right)

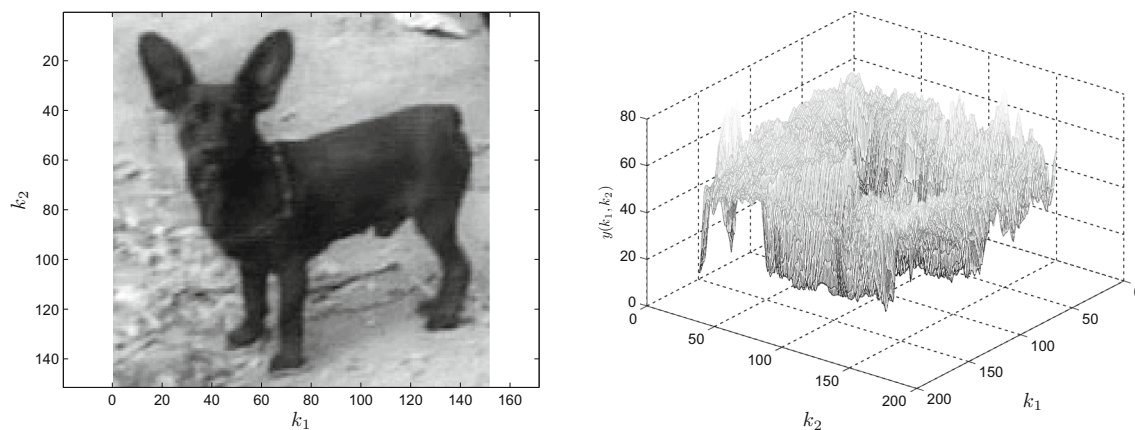


Fig. 10 The “Small Dog” image (left) and 3D plot (right)

belong to the discrete circle of radius r . Four discrete circles of consecutive radii $r = 1, \dots, 5$ are presented in Figs. 8 and 9. Inside the black discrete circles there are white continuous circles of appropriate radii.

The sum of all discrete circles of consecutive radii $r = 1, 2, \dots, r_{max}$ with the same center $c(k_1, k_2)$ forms a discrete disc. If the same discrete point is related to two different radii the lower has the priority. Hence, one defines a matrix with

elements being the radii (defining the discrete distance from the center). Such an exemplary distance matrix is presented below for $r_{max} = 5$ (on the left). Here, in all distance matrix corners radii $r = 6, 7 > r_{max}$ appeared. They are extracted by putting ∞ . A simplified distance matrix is given in (16) as a right matrix.

containing Dirac pulse response values related to the distance from the disk center. The matrix in question, denoted as $\mathbf{M}[g(r_{max})]$ for $r_{max} = 5$ is given below. This matrix is a mask [4,6]

Now, one assumes that a 2D signal (a discrete image) is represented by a matrix \mathbf{U} of dimensions $m \times n$ where $m, n \gg r_{max}$.

$$\mathbf{M}[g(r_{max})] = \{m(k_1, k_2)\}_{\substack{k_1 = 1, 2, \dots, m \\ k_2 = 1, 2, \dots, n}}$$

$$= \begin{bmatrix} 0 & 0 & 0 & g(5) & g(5) & g(5) & g(5) & g(5) & 0 & 0 & 0 \\ 0 & g(5) & g(5) & g(4) & g(4) & g(4) & g(4) & g(4) & g(5) & g(5) & 0 \\ 0 & g(5) & g(4) & g(3) & g(3) & g(3) & g(3) & g(3) & g(4) & g(5) & 0 \\ g(5) & g(4) & g(3) & g(3) & g(2) & g(2) & g(2) & g(3) & g(3) & g(4) & g(5) \\ g(5) & g(4) & g(3) & g(2) & g(1) & g(1) & g(1) & g(2) & g(3) & g(4) & g(5) \\ g(5) & g(4) & g(3) & g(2) & g(1) & g(0) & g(1) & g(2) & g(3) & g(4) & g(5) \\ g(5) & g(4) & g(3) & g(2) & g(1) & g(1) & g(1) & g(2) & g(3) & g(4) & g(5) \\ g(5) & g(4) & g(3) & g(3) & g(2) & g(2) & g(2) & g(3) & g(3) & g(4) & g(5) \\ 0 & g(5) & g(4) & g(3) & g(3) & g(3) & g(3) & g(3) & g(4) & g(5) & 0 \\ 0 & g(5) & g(5) & g(4) & g(4) & g(4) & g(4) & g(4) & g(5) & g(5) & 0 \\ 0 & 0 & 0 & g(5) & g(5) & g(5) & g(5) & g(5) & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

$$\mathbf{M}[D(r_{max})] = \begin{bmatrix} 7 & 6 & 6 & 5 & 5 & 5 & 5 & 5 & 6 & 6 & 7 \\ 6 & 5 & 5 & 4 & 4 & 4 & 4 & 4 & 5 & 5 & 6 \\ 6 & 5 & 4 & 3 & 3 & 3 & 3 & 3 & 4 & 5 & 6 \\ 5 & 4 & 3 & 3 & 2 & 2 & 2 & 3 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 & 1 & 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 & 1 & 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 3 & 2 & 2 & 2 & 3 & 3 & 4 & 5 \\ 6 & 5 & 4 & 3 & 3 & 3 & 3 & 4 & 5 & 6 \\ 6 & 5 & 5 & 4 & 4 & 4 & 4 & 5 & 5 & 6 \\ 7 & 6 & 6 & 5 & 5 & 5 & 5 & 6 & 6 & 7 \end{bmatrix}$$

$$\mathbf{M}[D(r_{max})] = \begin{bmatrix} \infty & \infty & \infty & 5 & 5 & 5 & 5 & \infty & \infty & \infty \\ \infty & 5 & 5 & 4 & 4 & 4 & 4 & 5 & 5 & \infty \\ \infty & 5 & 4 & 3 & 3 & 3 & 3 & 4 & 5 & \infty \\ 5 & 4 & 3 & 3 & 2 & 2 & 2 & 3 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 & 1 & 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 & 1 & 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 3 & 2 & 2 & 2 & 3 & 3 & 4 & 5 \\ \infty & 5 & 4 & 3 & 3 & 3 & 3 & 4 & 5 & \infty \\ \infty & 5 & 5 & 4 & 4 & 4 & 4 & 5 & 5 & \infty \\ \infty & \infty & \infty & 5 & 5 & 5 & 5 & \infty & \infty & \infty \end{bmatrix} \tag{16}$$

Then the processed discrete point (pixel) is calculated according to the fomula

$$y(i, j) = \frac{\sum_{k_1=i-r_{max}}^{i+r_{max}} \sum_{k_2=j-r_{max}}^{j+r_{max}} m(k_1, k_2)u(k_1, k_2)}{8r_{max}^2} \tag{18}$$

Introducing a notation suitable for Matlab [1], $r_{max} = r_{max}$, $mm = \mathbf{M}[g(r_{max})]$

$$mu = \begin{bmatrix} u(i - r_{max}, j - r_{max}) \cdots u(i - r_{max}, j + r_{max}) \\ \vdots \\ u(i + r_{max}, j - r_{max}) \cdots u(i + r_{max}, j + r_{max}) \end{bmatrix} \tag{19}$$

formula (20) is calculated in one command

$$y(i, j) = \text{ones}(1, 2 * r_{max} + 1) * mm . * mu * \text{ones}(2 * r_{max} + 1, 1) / (8 * r_{max} * r_{max}) \tag{20}$$

A particularly simple filtration effect calculation is obtained for (6) with $p = 1, q = 0, A_0 = 0, B_0 = 1$ and $\mu = -\nu$ where $0 < \nu \leq 2$

5.2 2D-FOF definition

For a given radius r_{max} and 1D-FOF Dirac pulse response values $g(0), g(1), g(2), \dots, g(r_{max})$ one defines a matrix

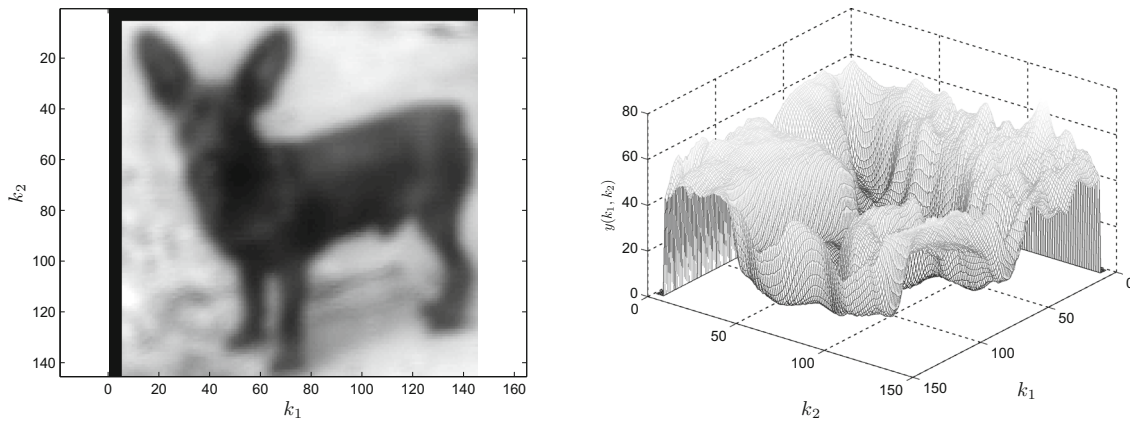


Fig. 11 The “Small Dog” filtered image (*left*) and 3D plot (*right*)

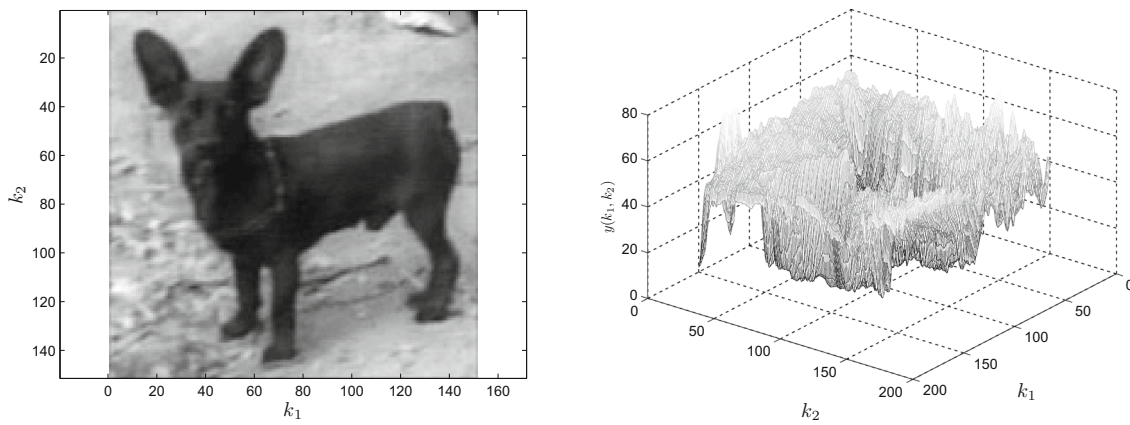


Fig. 12 The “Small Dog” filtered by Matlab gaussian filter image (*left*) and related 3D plot (*right*)

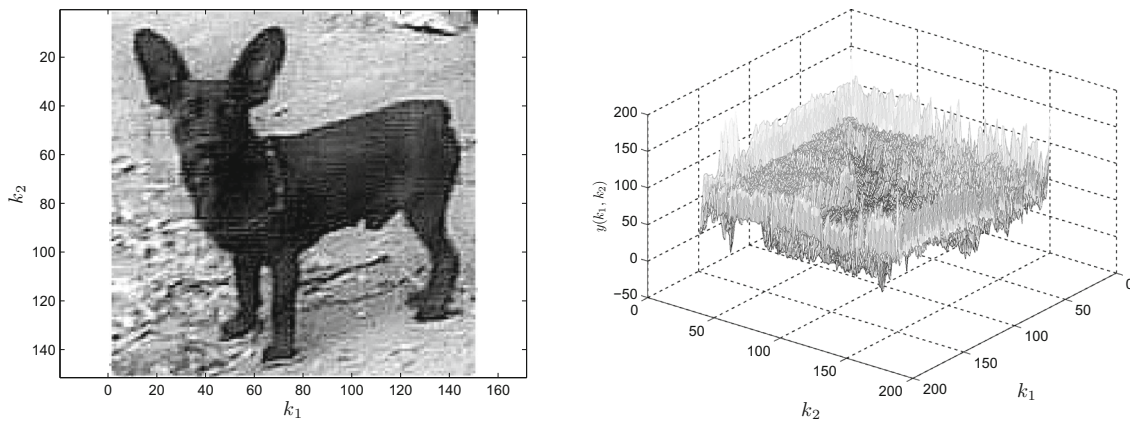


Fig. 13 The “Small Dog” filtered by Matlab “unsharp” filter image (*left*) and related 3D plot (*right*)

$$\begin{aligned}
 \mathbf{M} \left[D \left(\mathbf{a}^{(-\nu)}(r_{max}) \right) \right] &= \mathbf{M} \left[D \left(\mathbf{a}^{(\mu)}(r_{max}) \right) \right] \\
 &= \begin{bmatrix}
 0 & 0 & 0 & a^{(\mu)}(5) & a^{(\mu)}(5) & a^{(\mu)}(5) & a^{(\mu)}(5) & a^{(\mu)}(5) & 0 & 0 & 0 \\
 0 & a^{(\mu)}(5) & a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(4) & a^{(\mu)}(4) & a^{(\mu)}(4) & a^{(\mu)}(4) & a^{(\mu)}(5) & a^{(\mu)}(5) & 0 \\
 0 & a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(4) & a^{(\mu)}(5) & 0 \\
 a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(2) & a^{(\mu)}(2) & a^{(\mu)}(2) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(4) & a^{(\mu)}(5) \\
 a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(3) & a^{(\mu)}(2) & a^{(\mu)}(1) & a^{(\mu)}(1) & a^{(\mu)}(1) & a^{(\mu)}(2) & a^{(\mu)}(3) & a^{(\mu)}(4) & a^{(\mu)}(5) \\
 a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(3) & a^{(\mu)}(2) & a^{(\mu)}(1) & a^{(\mu)}(0) & a^{(\mu)}(1) & a^{(\mu)}(2) & a^{(\mu)}(3) & a^{(\mu)}(4) & a^{(\mu)}(5) \\
 a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(3) & a^{(\mu)}(2) & a^{(\mu)}(1) & a^{(\mu)}(1) & a^{(\mu)}(1) & a^{(\mu)}(2) & a^{(\mu)}(3) & a^{(\mu)}(4) & a^{(\mu)}(5) \\
 a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(2) & a^{(\mu)}(2) & a^{(\mu)}(2) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(4) & a^{(\mu)}(5) \\
 0 & a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(3) & a^{(\mu)}(4) & a^{(\mu)}(5) & 0 \\
 0 & a^{(\mu)}(5) & a^{(\mu)}(5) & a^{(\mu)}(4) & a^{(\mu)}(4) & a^{(\mu)}(4) & a^{(\mu)}(4) & a^{(\mu)}(4) & a^{(\mu)}(5) & a^{(\mu)}(5) & 0 \\
 0 & 0 & 0 & a^{(\mu)}(5) & a^{(\mu)}(5) & a^{(\mu)}(5) & a^{(\mu)}(5) & a^{(\mu)}(5) & 0 & 0 & 0
 \end{bmatrix}
 \end{aligned}
 \tag{21}$$

The image of “The Small Dog” and its 3D plot are presented in Fig. 10 left and right, respectively. For 1D-FOF (13) with $\mu = 0.5$, $L = 5$, $A_0 = 0.5$, $B_0 = 0.7$ the filtration effects are given in Fig. 11 left and right, respectively. The filtered image is clearly blurred. The noise is largely removed.

To compare the 2D-FODF proposed in this paper the effects of the following filtering results are presented. In Fig. 12 the “gaussian” filter application (left) and its 3D plot is given.

In Fig. 13 the “unsharp.m” Matlab function is applied. The results are presented in Fig. 13.

The filtering results are usually subjected to subjective human evaluation, rather than to the objective criteria, but one should emphasise that the FODF parameters are not optimized due to the assumed performance criterion.

6 Conclusions

In classical IOF (15) there are two parameters to select: FO μ and the coefficient A_0 . The FODF demands an additional parameter to select FO $0 < \nu < 1$. The parameter and order selection can be performed due to ISE criterion. The FODF response calculation is more complicated but such filters offer much greater opportunities to shape the dynamics of the 2D-FOF output. The main competitive advantage of the FOF over the IOF is steeper slope and sharper slump of the magnitude plots.

Acknowledgements The research was supported by the Polish National Science Centre in 2013–2015 as a research Project (DEC-2012/05/B/ST 6/03647).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Weeks M (2011) Digital signal processing using Matlab and Wavelets. Jones and Bartlett Publishers, Sudbury
2. Ifeachor EC, Jervis BW (1993) Digital signal processing: a practical approach. Addison-Wesley, Harlow
3. Proakis JG, Manolakis DG (2007) Digital signal processing: principles, algorithms, and applications. Prentice Hall, Upper Saddle River
4. Bose T (2004) Digital signal and image processing. Wiley, New York
5. Pitas I (2000) Digital image processing. Algorithms and applications. John Wiley & Sons, Inc., New York
6. Pratt WK (1991) Digital image processing. Wiley, New York
7. Das S, Pan I (2012) Fractional order signal processing: introductory concepts and applications. Springer, Heidelberg
8. Miller KS, Ross B (1993) An introduction to the the fractional calculus and fractional differential equations. A Wiley-Interscience Publications, New York
9. Ortigueira MD (2011) Fractional calculus for scientists and engineers. Springer Science & Business Media, Berlin
10. Ortigueira MD, Machado JAT (2015) What is a fractional derivative? J Comput Phys 293:4–13

11. Podlubny I (1999) Fractional differential equations. Academic Press, London
12. Ostalczyk P (2016) Discrete fractional calculus: selected applications in control and image processing, vol 4. World Scientific, Singapore
13. Machado JAT (2015) Generalized convolution. Appl Math Comput 257:34–39
14. Ionescu C, Machado JAT, de Keyser R (2011) Fractional-order impulse response of the respiratory system. Comput Math Appl 62(3):845–854
15. Bresenham J (1965) Algorithm for computer control of a digital plotter. IBM Syst J 4(1):25–30