INVESTIGATION OF FIXED-POINT COMPUTATION INFLUENCE ON NUMERICAL SOLUTIONS OF FRACTIONAL DIFFERENTIAL EQUATIONS

Paweł PIĄTEK*, Jerzy BARANOWSKI*

*AGH University of Science and Technology, The Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, Department of Automatics, Al. Mickiewicza 30/B1, 30-059 Kraków

ppi@agh.edu.pl, jb@agh.edu.pl

Abstract: In this paper the problem of the influence of fixed point computation on numerical solutions of linear differential equations of fractional order is considered. It is a practically important problem, because of potential possibilities of using dynamical systems of fractional order in the tasks of control and filtering. Discussion includes numerical method is based on the Grünwald-Letnikov fractional derivative and how the application of fixed-point architecture influences its operation. Conclusions are illustrated with results of floating-point arithmetic with double precision and fixed point arithmetic with different word lengths.

1. INTRODUCTION

Dynamical system described by fractional differential equations take an increasing role in technical sciences. The initial concept dating to private correspondence of Leibnitz and L'Hospital from 1695, was systematically developed however outside the main stream. Currently we can say, that mathematical side of the problem is well rounded, what can be observed by presence of multiple monographs such as Miller and Ross (1993); Oldham and Spanier (1974); Podlubny (1999); Samko et al. (1993).

In recent years especially interesting is the aspect of applications. They are found in modelling of supercapacitors, distributed parameter systems, problems of variational calculus or modelling of very complicated phenomena such as flame spreading Lederman et al. (2002); Weilbeer (2005). Besides modelling also fractional systems are used to influence reality as controllers Ortigueira (2008); Ruszewski (2008) or filters Magin et al. (2011). In the context of fractional order systems also problems such as state estimation (Dzieliński and Sierociuk (2008)), controllability (Klamka (2009)) or stability (Kaczorek (2008a); Busłowicz (2008); Kalinowski and Busłowicz (2011)) are considered. A comprehensive survey of theory and applications of fractional calculus in control engineering can be found in Ostalczyk (2008).

In this paper authors focus on the problem of actual implementation of fractional order systems. Many works are devoted to the concept of approximation of fractional order systems with integer order systems (see for example Djouambi et al. (2007); Sobolewski and Ruszewski (2011)). This paper analyses the application of numerical methods for solving fractional order differential equations. Because the focus of this research is the implementation of fractional controllers and filters on commercially available hardware platforms special emphasis is placed on influence of fixed point computation. In the following parts of the paper considered class of systems is described, solution of differential equations on dedicated hardware platforms with individual section on problems quantisation. Then discretisation of fractional differential equations is analysed. Finally numerical experiments are conducted in both floating and fixed point arithmetic.

2. CONSIDERED SYSTEMS

In this paper linear fractional order dynamical system described by a following system of fractional order equations

$$\begin{aligned} &\frac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad 0 < \alpha \le 1 \\ &\mathbf{x}(0) = \mathbf{x}_0 \end{aligned} \tag{1}$$

where: $\mathbf{x}(t) \in R^n$, $\mathbf{u}(t) \in R^r$ and $\mathbf{A}$, $\mathbf{B}$ are constant matrices of appropriate dimensions. Fractional differentiation operation of order α is given by Caputo definition (see for example Kaczorek (2008b)).

$$\frac{\mathrm{d}^\alpha}{\mathrm{d}t^\alpha}x(t) = \frac{1}{\Gamma(n-\alpha)}\int_0^t \frac{x^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}}\mathrm{d}\tau, \quad n = \lceil \alpha \rceil \tag{2}$$

where: Γ function is given by

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,\mathrm{d}t$$

Important fact is that in analogue to integer order equations one can express solution of (1) by variation of constants, that is

$$\mathbf{x}(t) = \mathbf{\Phi}_0(t)\mathbf{x}_0 + \int_0^t \mathbf{\Phi}(t-\tau)\mathbf{B}\mathbf{u}(\tau)\mathrm{d}\tau \tag{3}$$

$$\mathbf{\Phi}_0(t) = \mathbf{E}_\alpha(\mathbf{A}t^\alpha) \tag{4}$$

$$\mathbf{\Phi}(t) = t^{\alpha-1}\mathbf{E}_{\alpha,\alpha}(\mathbf{A}t^\alpha) \tag{5}$$

where: $\mathbf{E}$ is the Mittag-Leffler function (see for example Weilbeer (2006)) given by:

$$\mathbf{E}_{\alpha,\beta}(\mathbf{z}) = \sum_{k=0}^{\infty} \frac{\mathbf{z}^k}{\Gamma(\alpha k + \beta)}, \quad \alpha > 0, \beta > 0 \tag{6}$$

$$\mathbf{E}_{\alpha}(\mathbf{z}) = \mathbf{E}_{\alpha,1}(\mathbf{z}), \quad \alpha > 0 \tag{7}$$

It should be noted that Mittag-Leffler function is the generalisation of e^{z} and for $\alpha = 1$ the following equality occurs

$$\mathbf{E}_1(\mathbf{z}) = e^{\mathbf{z}} \tag{8}$$

In this paper only initial conditions equal to zero will be considered. It is justified by the fact that the main goal is to devise methods of effective filter and controller implementation. Moreover one can transform a fractional system into one with zero initial conditions through addition of additional inhomogeneity (see for example Podlubny (2000)).

3. SOLVING DIFFERENTIAL EQUATIONS WITH DEDICATED CONTROL SYSTEMS

In classical control systems that is those, which model of controller or system is described by integer order differential equations the following hardware platforms are used:

- universal platforms:
  - classical computer systems,
  - industrial PLC controllers,
  - universal microprocessor controllers,
- dedicated platforms:
  - using general purpose processors,
  - using digital signal processors (DSP),
  - using FPGA circuits.

In case of fractional order differential equations this division stays correct. Because of possibility of obtaining very short computation times - dedicated systems are very promising. Among those especially systems using FPGA circuits raise interest.

Using a dedicated control system for computation of both ordinary and fractional differential equations carries many consequences. Substantial benefits are that one can achieve substantial increase in computation speed and keep the regimes of real time processing. On the other hand use of dedicated systems introduces multiple constraints associated with their construction and type of operation. The most serious limit introduced by dedicated control systems is lack of support for floating-point arithmetic. Most microcontrollers designed for control systems do not have an integrated floating-point coprocessor. Similar situation occurs for DSPs. One can of course show solutions supporting floating-point formats but that is not the norm. Different case is for implementation of such formats in FPGA circuits. These circuits are rather freely configurable. One can also implement the support for writing of the floating-point data format. However because of needed amount of circuit's hardware resources it is not always possible or economically feasible.

In this paper control systems with fixed-point data formats are considered. In case of the FPGA circuits these formats are supported by hardware description languages (e.g. VHDL) or are relatively easy to implement. The most substantial merit of using the fixed-point arithmetic is the possibility of construction of parallel data processing structures, which can significantly accelerate computation (see Wiatr (2003)). Other important merit is the possibility of using computation words of desired length (see Piątek (2007)). When programming microcontrollers or DSP the programmer can use the data types available in the microprocessor architecture. Using of non standard data types is associated with need for additional operations, which can increase the computation time.

When solving systems of differential equations in computer systems, so also in the control systems we have to deal with quantisation of signals and parameters in time (discretisation) and in values (quantisation) caused by digital character of computation. Both these operations have their properties and can disrupt the results of computation - that is the solution of the system of differential equations.

4. QUANTISATION

Application of digital systems, especially those which use a fixed-point data format causes introduction of numerical errors to the computation. Sources of these errors are (see Gevers and Li (1993); Świder (2003, 2002)):

- quantisation of analogue signals – for example by A/D converters in control systems;
- computation result overflow errors caused by too short data word length;
- round-off errors of arithmetic operations – multiplication, addition;
- quantisation errors of model coefficients resulting from writing them on words with finite length.

Converter quantisation errors are determined by resolution of used A/D converter system. In the case of modelling the converter model by stochastic methods it is assumed that the converter model consists of a sampling system and a quantiser. Quantiser is modelled as a summation node introducing a random error to the signal. It is assumed that this error is a discrete white noise not correlated with the sampled signal and its variance is dependent on the number of converter bits (see Świder (2003)). Quantisation noise created in the process of analogue-digital conversion can be filtered in the control system by the usage of appropriate digital filters.

Overflow errors are practically present only in the systems performing computations using fixed point arithmetic. They occur in the situations, when the result of arithmetic operation requires writing in the registry of larger number of bits than it is available in the computation system. In some situations (e.g. using notation in the two's complement code) it causes large relative errors (see Gevers and Li (1993)). Elimination of overflow errors relies on appropriate scaling of signals and coefficients of the model. Such operations unfortunately introduce additional round-off errors associated with changing the signals and model coefficients ranges. In computation systems using floating-point arithmetic overflow errors are not present or occur rarely, because of the large ranges of such data storage.

Other two kinds of errors - round off errors of arithmetic operations and parameter quantisation errors are always present during digital realisation of control algorithms and it is not possible to completely eliminate their influence on the result of computation (see Gevers and Li (1993)). Arithmetic operation round off errors are introduced during the computations connected to determination of system response and their level is dependent on the structure of algorithm and the data word length. Model coefficient quantisation errors are introduced by the finite data word length. Ideal values of parameters are rounded to the values that can be stored. Similar to the arithmetic operation errors, coefficient quantisation errors are dependent on the structure of algorithm and the data word length. Effects connected with these two kinds of errors are called FWL (Finite Word Length) effects (see Gevers and Li (1993)). They can be limited by increasing the length of data words and by changing model structure. Length modification is not always possible. Usually in computer systems only two or three word lengths are available, and in simple microprocessor system even only one. Relatively simple increase of precision is possible only in the range of data types supported by the architecture and additional improvements (above the machine command precision) has a cost of a substantial increase in the number of commands required for determination of system response. In case of realisation of control system with dedicated architecture for example with FPGA circuits, word length can be adjusted at will. Too long word lengths however cause substantial increase in the hardware resources usage, which can be interpreted as the increase in the computation cost.

5. DISCRETISATION OF FRACTIONAL ORDER DIFFERENTIAL EQUATIONS

There are different classes of numerical methods for solving fractional differential equations (see Weilbeer (2005)). One of them are linear multistep methods. Their construction relies on transformation of fractional differential equation to the equivalent Volterra integral equation and solving it through quadratures. It is similar to Adams methods for ODE (see Hairer et al. (2000)). Another group considers equivalent Abel-Volterra equation and solves it via power series - these are generalised Taylor expansion and Adomian decomposition method. One more group are collocation methods also popular for integral equations. For applications in the context of filter and controller implementations the most practical seem to be backward difference methods. This class includes Diethelm method and quadrature based Lubich method.

In this paper third backward difference method is considered – that is the method based on the Grünwald-Letnikov fractional derivative. By this definition the fractional derivative takes form of a limit of fractional difference quotients

$$\frac{\mathrm{d}^{\alpha}}{\mathrm{d}t^{\alpha}}x(t)=\lim_{h\to 0}\frac{(\Delta_h^{\alpha}x)(t)}{h^{\alpha}} \tag{9}$$

where:

$$(\Delta_h^{\alpha}x)(t)=\sum_{k=0}^{m}(-1)^k\binom{\alpha}{k}x(t-kh)\Bigg|_{h=t/m} \tag{10}$$

Generalised Newton symbol is given by

$$\binom{\alpha}{k}=\frac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}= \tag{11}$$

$$=\begin{cases}\dfrac{\alpha(\alpha-1)\cdot\ldots\cdot(\alpha-j+1)}{j!} & \text{for} \quad j\in N\\ 1 & \text{for} \quad j=0\end{cases} \tag{12}$$

Fractional derivative takes form

$$\frac{\mathrm{d}^{\alpha}}{\mathrm{d}t^{\alpha}}x(t)=\lim_{\substack{h\to 0\\ h=t/m}}\frac{1}{h^{\alpha}}\sum_{k=0}^{m}(-1)^k\binom{\alpha}{k}x(t-kh) \tag{13}$$

It should be noted that definitions of Grünwald-Letnikov and Caputo are not fully equivalent. It is especially important in the context of fractional differential equations, where initial conditions influence the solution in different way (see Weilbeer (2005)). If initial conditions are zero, as in the considered case the solutions are however equal.

As it can be seen in the fractional difference when h decreases m increases, so in the limit sum is infinite. The idea of numerical solution on the interval $t\in[0,T]$ relies on determining finite m and omitting the limit. In that way differential equation (1) becomes

$$\frac{1}{h^{\alpha}}\sum_{k=0}^{m}(-1)^k\binom{\alpha}{k}\mathbf{x}(t-kh)=\mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t), \quad t\in[0,T],\ h=T/m \tag{14}$$

or equivalently

$$\frac{\mathbf{x}(t)}{h^{\alpha}}+\frac{1}{h^{\alpha}}\sum_{k=1}^{p}(-1)^k\binom{\alpha}{k}\mathbf{x}(t-kh)=\mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t) \quad h=T/m,\ t=ph,\ p=0,1,\ldots,m \tag{15}$$

It should be noted that $\mathbf{x}(t)$ is present on both sides of equality. In case of nonlinear systems it would require iterational procedures, however because the considered system is linear so

$$\mathbf{x}(t)=(\mathbf{I}-h^{\alpha}\mathbf{A})^{-1}\left(h^{\alpha}\mathbf{B}\mathbf{u}(t)-\sum_{k=1}^{p}c_k\,\mathbf{x}(t-kh)\right) \tag{16}$$

$$h=T/m,\ t=ph,\ p=0,1,\ldots,m \tag{17}$$

$$c_k=(-1)^k\binom{\alpha}{k},\quad k=1,2,\ldots,m \tag{18}$$

Different approach can be seen in the work of Podlubny (2000). Method presented there formulates the problem of numerical solution as a system of linear algebraic equations solving the fractional differential equations in all points of the interval simultaneously. That approach has many benefits, but is not adequate for series signal processing.

As it can be seen, when changing m also h is changed which can cause FWL effects. In the next section the behaviour of numerical solution of fractional differential equation obtained with (16) behaves when changing parameters.

6. FLOATING-POINT ARCHITECTURE SIMULATIONS

In order to perform simulational analysis of the solution of fractional differential equation the following example needs to be considered.

Example 1. (Kaczorek (2008b)) The unit step response of the following system is considered

$$\begin{aligned} &\frac{\mathrm{d}^{\alpha}}{\mathrm{d}t^{\alpha}} x(t) = -x(t) + u(t) \\ &x(0) = 0 \in R \\ &u(t) = 1(t) \end{aligned} \tag{19}$$

From (3) the solution is

$$\begin{aligned} x(t) = {} & \int_0^t (t-\tau)^{\alpha-1} E_{\alpha,\alpha}(-(t-\tau)^{\alpha}) \mathrm{d}\tau = \\ = {} & \int_0^t s^{\alpha-1} E_{\alpha,\alpha}(-s^{\alpha}) \mathrm{d}s = \\ = {} & \int_0^t \sum_{k=0}^{\infty} \frac{(-1)^k s^{\alpha k+\alpha-1}}{\Gamma(\alpha(k+1))} \mathrm{d}s = \\ = {} & \sum_{k=0}^{\infty} \frac{(-1)^k}{\Gamma(\alpha(k+1))} \int_0^t s^{\alpha k+\alpha-1} \mathrm{d}s = \\ = {} & \sum_{k=0}^{\infty} \frac{(-1)^k}{\Gamma(\alpha(k+1))} \frac{t^{\alpha(k+1)}}{\alpha(k+1)} = t^{\alpha} E_{\alpha,\alpha+1}(-t^{\alpha}) \end{aligned} \tag{20}$$

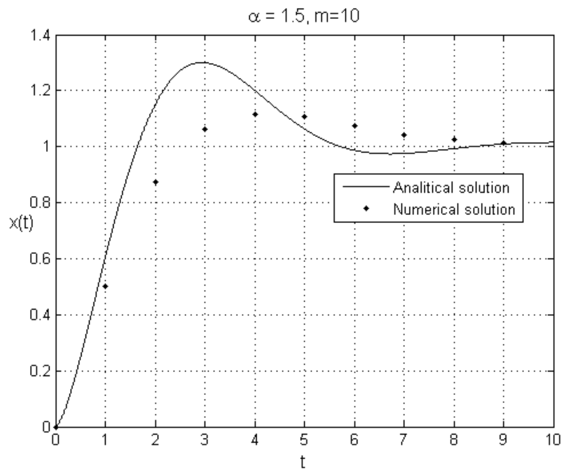

Fig. 1. Comparison of analytical and numerical solution of fractional differential equation (19) with α=3/2 for m=10

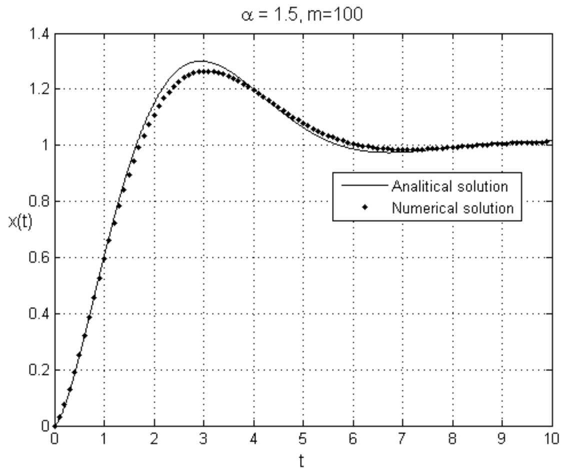

Fig. 2. Comparison of analytical and numerical solution of fractional differential equation (19) with α=3/2 for m=100

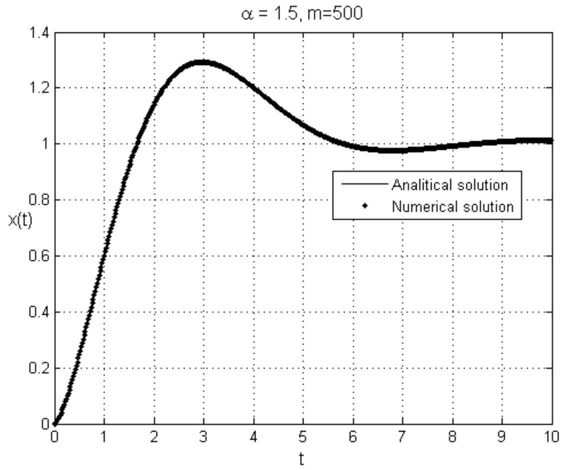

Fig. 3. Comparison of analytical and numerical solution of fractional differential equation (19) with α=3/2 for m=500

The step response was expressed by Mittag-Leffler function (6) It should be noted that for $\alpha > 1$ initial conditions for all $n < \alpha$ need to be specified.

Obtained analytical solution can be used for verification of correctness of (16). System with $\alpha = 3/2$ is considered. Comparisons are made for different m. Computations were performed in Matlab in double precision. Analytical solution consisted of 100 first expression of power series form of Mittag-Leffler function (6). The analysis was performed on interval $t \in [0, 10]$.

It should be noted that for $\alpha > 1$ solutions have oscillatory character. Solution consisting of 10 points (Fig. 1) represents the oscillations but it happens in different moment and with much smaller amplitude. Increasing precision to 100 points the solution improves (Fig. 2), and for 500 points (Fig. 3) numerical solution becomes truly close to an analytical one. It should be noted that increasing number of points in the interval the requirements toward solutions increase, as in every step of computation all the earlier ones are necessary.

7. FIXED POINT ARCHITECTURE SIMULATIONS

For numerical experiments Matlab environment was used with the Fixed-Point Toolbox. With this software one can create and use variables with desired word lengths in bits. These simulations were performed for step response of system (19). Numerical method (16) was used and number of steps per interval was set to $m = 100$. Compared are:

- analytical solution;
- numerical solution using method (16) operating with floating-point arithmetic;
- numerical solution using method (16) operating with fixed-point arithmetic.

In the last case a fixed point notation allowing operation on numbers with nonzero fractional part. These numbers are coded with use of two's complement code (see Biernat(2001); Pochopień(2004)). Thanks to using it scaling could be avoided. Figure 4 presents the format of this fixed-point notation.

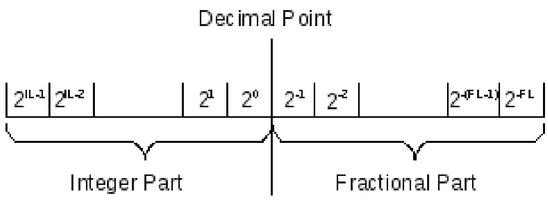

Fig. 4. Fixed point notation during the experiments

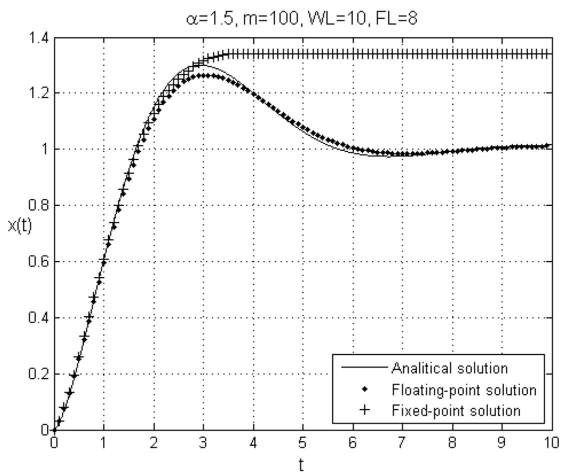

Fig. 5. System unit step response (FL=8)

Corresponding to the Fig. 4 following quantities were introduced:

- FL denotes number of bits devoted to the fractional part,
- IL denotes number of bits devoted to the integer part,
- total number of bits in the data word was
  WL = IL + FL .

It was decided to use a single word length for all elements of the algorithm. That means that both system coefficients, constants associated with α and number of steps and system state were denoted in variables with the same word length and the same lengths of fractional and integer parts. Nine numerical experiments were performed, in which step response of system (19) was computed. In every experiment the word length for the fractional part was increased by one from 8 to 16 bits. The most representative were the results obtained for fractional parts of 8, 9, 10, 12 and 16 bits. For all the experiments IL=2 was set.

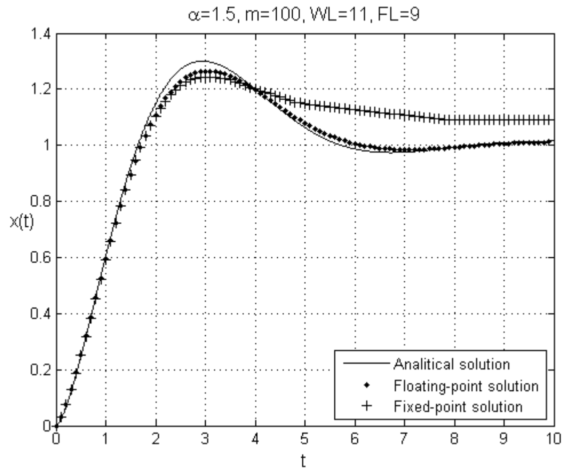

Fig. 6. System unit step response (FL=9)

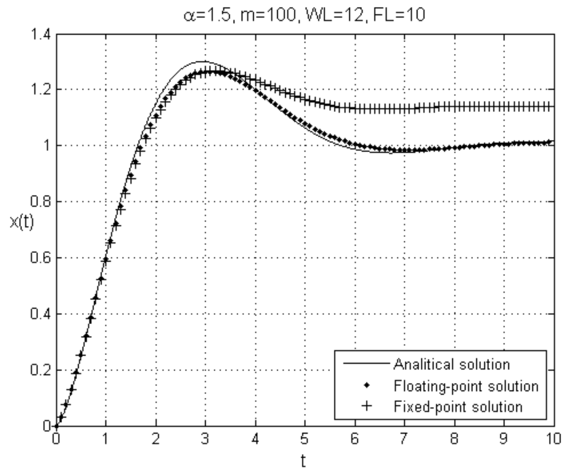

Fig. 7. System unit step response (FL=10)

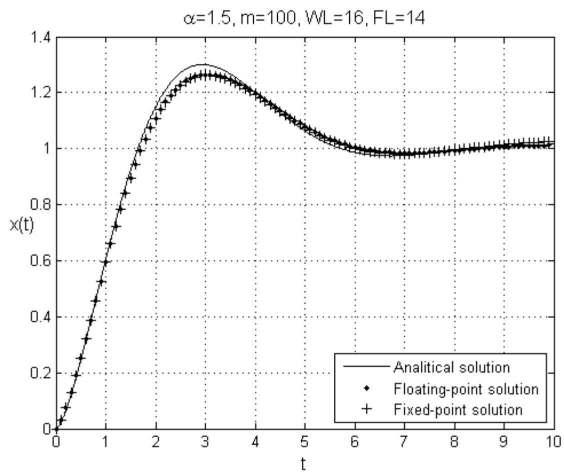

Fig. 8. System unit step response (FL=14)

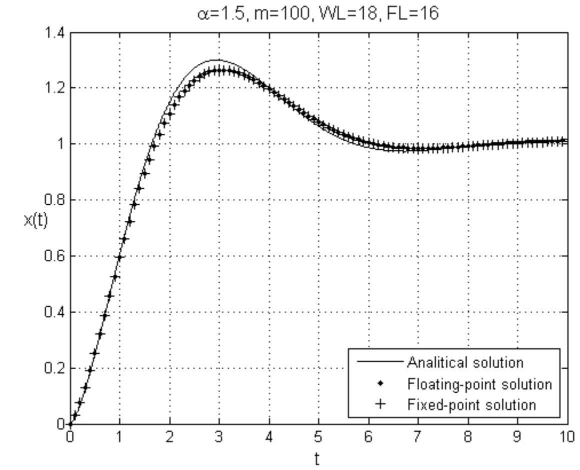

Fig. 9. System unit step response (FL=16)

Results of simulations are presented in Figs. 5, 6, 7, 8 and 9. In the figures three responses are presented: analytical, computed numerically with floating-point and computed numerically with fixed point.

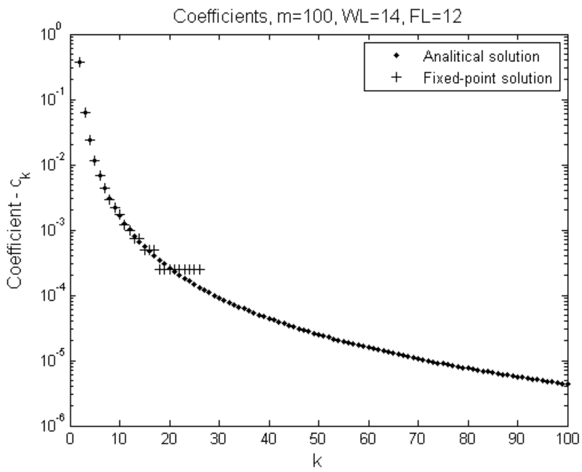

Fig. 10. Coefficient values for WL=14

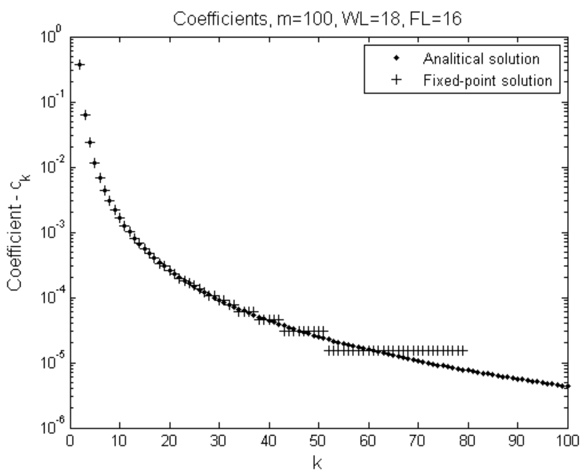

Fig. 11. Coefficient values for WL=18

Analysis of the figures, allows to observe, that reduction of fractional part word length increases the numerical error of such computed fractional part. For FL=8 (Fig. 5) the response differs so much that it loses its original character.

Further study allowed to find one of the reasons for substantial differences between analytical, floating-point and fixed-point solutions. It appears that it has a strong connection to the coefficients c_k (18). In Fig. 10 and 11 values of coefficients c_k computed analytically and numerically with application of fixed point arithmetic with different word lengths. Vertical axes in those figures are in the logarithmic scale for easier observation of the effects.

For word length WL=14 the effect of quantisation is evidently visible for coefficients with index greater than 13. Moreover coefficients with index greater than 27 they become equivalent to zero, regardless that analytically they are different from zero. For word length WL=18 the similar effect is visible, however quantisation is visible for indices greater than 34 and they become zero for indices greater than 80. Coefficients equal to zero are not visible in the plot, as 0 does not belong to the domain of algorithm.

It should be noted, that this effect causes qualitative change in the system character. From the system with potentially infinite memory it becomes a system with finite memory. It should be compared with practically stable discrete fractional systems (see for example Kaczorek(2011)).

8. CONCLUSIONS

After analysis of results of numerical experiments it can be concluded, that main reasons for errors occurring when using fixed-point arithmetic are the quantisation and rounding of coefficients (18). In figures it can be observed, that for analysed systems these coefficients are reduced along with index. For small values this effect is especially visible. Below certain value (certain index) quantisation reduces them to zero. Simulations illustrated, that the errors caused by using fixed-point arithmetic can significantly change the response of analysed system. Word length should be then chosen very carefully. In further works the possibility of using different word lengths for coefficients and state. Additional modification of numerical method should be considered in order to increase robustness to these errors.

It should be also noted, that zeroing of coefficients due to fixed-point computation leads to system with finite memory. It is very similar to practically stable discrete fractional systems. It is interesting how other properties of these systems transfer to analysed systems.

REFERENCES

1. **Biernat J.** (2001), *Metody i układy arytmetyki komputerowej*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław.
2. **Busłowicz M.** (2008), *Frequency domain method for stability analysis of linear continuous-time fractional systems*, In Malinowski K. and Rutkowski L. editors, Recent advances in control and automation, chapter 2, pages 83-92, Akademicka Oficyna Wydawnicza EXIT, Warszawa.

3. **Djouambi A. B., Charef A. F., Besançon A. V.** (2007), Optimal approximation, simulation and analog realization of the fundamental fractional order transfer function, *Int. J. Appl. Math. Comput. Sci.*, 17(4):455-462.
4. **Dzieliński A., Sierociuk D.** (2008), Obserwator zredukowany dla dyskretnych układów ułamkowego rzędu, In Malinowski K. and Rutkowski L., editors, Sterowanie i Automatyzacja: Aktualne problemy i ich rozwiązania, chapter 2, pages 66-75. Akademicka Oficyna Wydawnicza EXIT.
5. **Gevers M., Li G.** (1993), *Parametrizations in Control, Estimation, and Filtering Problems: Accuracy Aspects*, Springer-Verlag, London.
6. **Hairer E., Nørsett S. P., Wanner G.** (2000), *Solving Ordinary Differential Equations: I Nonstiff problems*, Springer..
7. **Kaczorek T.** (2008a), Practical stability of positive fractional discrete-time linear systems, *Bulletin of the Polish Academy of Sciences Technical Sciences*, 56(4):313-317.
8. **Kaczorek T.** (2008b), Fractional positive linear systems and electrical circuits, *Materiały XXXI Międzynarodowej konferencji z podstaw elektrotechniki i teorii obwodów IC-SPETO*, 3-4, Ustroń , 28-31.05.2008.
9. **Kaczorek T.** (2011), *Positive fractional linear systems*, Pomiary Automatyka Robotyka, 2:91--112.
10. **Kalinowski T., Busłowicz M.** (2011), Odporna stabilność rodziny wielomianów niecałkowitego stopnia o współczynnikach wieloliniowo zależnych od niepewnych parametrów, *Pomiary Automatyka Robotyka*, 2:576-585.
11. **Klamka J.** (2009), Controllability and minimum energy control problem of infinite dimensional fractional discrete-time systems with delays, *Proceedings of 2009 First Asian Conference on Intelligent Information and Database Systems*, 398-403.
12. **Lederman C., Roquejoffre J. M., Wolanski N.** (2002), Mathematical justification of a nonlinear integro-differential equation for the propagation of spherical flames, *C. R. Math. Acad. Sci. Paris*, 334(7):569—574.
13. **Magin R., Ortigueira M. D., Podlubny I., Trujillo J.** (2011), On the fractional signals and systems. Signal Processing, *Advances in Fractional Signals and Systems*, 91(3):350 - 371.
14. **Miller K. S., Ross B.** (1993), *An introduction to the fractional calculus and fractional differential equations*, John Wiley & Sons Inc., New York.
15. **Oldham K. B., Spanier J.** (1974), *The fractional calculus*, Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London.
16. **Ortigueira M. D.** (2008), An Introduction to the Fractional Continuous-Time Linear Systems: The 21st Century Systems, *IEEE Circuits and Systems Magazine*, 8(3):19-26.
17. **Ostalczyk P.** (2008), *Zarys rachunku różniczkowo-całkowego ułamkowych rzędów. Teoria i zastosowania w automatyce*, Wydawnictwo Politechniki Łódzkiej.
18. **Piątek P.** (2007), *Wykorzystanie specjalizowanych architektur sprzętowych do realizacji krytycznych czasowo zadań sterowania*, *Application of Specialized Hardware Architectures for Realization of Time-critical Control Tasks*, Rozprawa doktorska, Promotor: W. Grega, Akademia Górniczo-Hutnicza im. Stanisława Staszica, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Kraków.
19. **Pochopień B.** (2004), *Arytmetyka w systemach cyfrowych*, Akademicka Oficyna Wydawnicza EXIT, Warszawa.
20. **Podlubny I.** (1999), Fractional differential equations. An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications, volume 198 of *Mathematics in Science and Engineering*, Academic Press Inc., San Diego, California.
21. **Podlubny I.** (2000), *Matrix approach to discrete fractional calculus*, Fractional Calculus and Applied Analysis, 3(4):359--386.
22. **Ruszewski A.** (2008), *Stabilization of Fractional-order Strejc's process model with time delay using fractional-order PI controller*, In Malinowski, K. and Rutkowski, L., editors, Recent advances in control and automation, chapter 2, pages 103-113. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2008.
23. **Samko S., Kilbas A., Marichev O.** (1993), *Fractional Integrals and derivatives: Theory and Applications*, Gordon and Breach Science Publishers.
24. **Sobolewski A., Ruszewski A.** (2011), Realizacja praktyczna regulatora niecałkowitego rzędu, *Pomiary Automatyka Robotyka*, 2:586-594.
25. **Świder Z.** (2002), Redukcja błędów zaokrągleń w układzie sterowania niewrażliwym na zakłócenie periodyczne, *Materiały XIV Krajowej Konferencji Automatyki*, Zielona Góra, 537-542.
26. **Świder Z.** (2003), *Realizacje cyfrowe algorytmów sterowania i filtracji*, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów, 2003.
27. **Weilbeer M.** (2005), *Efficient Numerical Methods for Fractional Differential Equations and their Analytical Background*, PhD thesis, Technischen Universität Braunschweig.
28. **Wiatr K.** (2003), *Akceleracja obliczeń w systemach wizyjnych*, WNT, Warszawa.

Work partially financed by National Centre of Science funds for 2011-2013 as a research project. Contract no. N N514 644440.