Jarosław FORENC
POLITECHNIKA BIAŁOSTOCKA, WYDZIAŁ ELEKTRYCZNY

Parallel analysis of transient states in electric motor

Ph.D. Jarosław FORENC

Received M.Sc. degree in computer science and Ph.D. degree in electrical engineering from Białystok Technical University in 1997 and 2006, respectively. His research interests include parallel computations in electrical engineering and parallel analysis of transient states in physical systems.

e-mail: j.forenc@pb.edu.pl

Abstract

The analysis of transient states in asynchronous slip-ring motor with the application of the parallel method is presented in the paper. Transient states are described by a system of non-linear ordinary differential equations. Solving systems of such equations is a sequential process. The proposed parallel method converts sequential computations into intensively parallel ones. The general idea of this method is based on decomposition of the integration interval into sub-intervals. Computations in sub-intervals are done based on initial conditions determined on the basis of an approximation of the convergence graph by the exponential function.

Keywords: ordinary differential equations, parallel computing, transient states.

Równoległa analiza stanów nieustalonych w silniku elektrycznym

Streszczenie

W artykule przedstawiono zastosowanie oryginalnej metody równoległej analizy stanów nieustalonych do badania dynamiki modelu silnika asynchronicznego pierścieniowego. Metoda ta przeznaczona jest do analizy stanów nieustalonych występujących w obwodach elektrycznych w przypadku, gdy stan nieustalony opisany jest układem równań różniczkowych zwyczajnych, liniowych lub nieliniowych (równaniem stanu). Ogólna idea metody opiera się na dekompozycji przedziału całkowania (t_0, t_N) na podprzedziały (rys. 2). Obliczenia zmiennych stanu w poszczególnych podprzedziałach wykonywane są równolegle przy zastosowaniu jednej ze znanych sekwencyjnych, jednokrokowych metod numerycznych rozwiązywania układów równań różniczkowych zwyczajnych. Wykonanie równolegle obliczeń wymaga znajomości wartości zmiennych stanu na początku każdego podprzedziału (warunków początkowych). W chwili t_0 wartości te znane są z założenia. W pozostałych podprzedziałach wartości zmiennych stanu wyznaczane są na podstawie przybliżenia wykresu zbieżności rozwiązania sekwencyjnego funkcją wykładniczą (3). Algorytm metody zaimplementowany został w strategii „Master-Slave" (rys. 1). Proces master wyznacza sekwencyjnie wartości zmiennych stanu na początku podprzedziałów i przesyła je do procesów slave. Wszystkie procesy (master i slave) wykonują równolegle obliczenia wartości zmiennych stanu w odpowiednich podprzedziałach przedziału całkowania. Po zakończeniu obliczeń proces master odbiera wyniki obliczeń od procesów slave i zapisuje rozwiązanie końcowe. Jako przykład zastosowania powyższej metody przedstawiona została analiza dynamiki modelu silnika asynchronicznego pierścieniowego. Stan nieustalony w silniku opisany jest układem pięciu nieliniowych równań różniczkowych zwyczajnych (5). Obliczenia przeprowadzone zostały przy zastosowaniu systemu klaster składającego się z 6 stacji roboczych. Podczas obliczeń otrzymano dobre przybliżenie wartości zmiennych stanu na początku każdego podprzedziału, co zapewniło dobrą dokładność rozwiązania końcowego.

Słowa kluczowe: równania różniczkowe zwyczajne, obliczenia równoległe, stany nieustalone.

1. Introduction

Transient states in electrical circuits are described by systems of ordinary and partial differential equations (continuous systems) or systems of difference equations (discrete systems) [1-3]. In this paper the transient states described by a system of ordinary differential equations (the state equation), linear or non-linear, will be considered. In most cases such systems of equations are solved using numerical methods [4-6], which are typically sequential methods. In such methods knowledge of the values of the variables from the previous step is necessary in order to determine values of the variables in the next step. There are several reasons the application of parallel methods for solving systems of ODEs can be helpful [7]: solving of the functions of the right-hand side of the differential equations is too time consuming, the number of equations in the system is large, the integration interval is long, the system must be solved repeatedly, achieving high accuracy of computation results in a short time requires a very small integration step size. Several approaches towards the parallel solution of ODEs have been developed. A good overview of those methods can be found in papers [8-10] and monographs [7, 11].

The means of achieving parallelism in solving systems of ODEs is classified into three categories: parallelism across the system, parallelism across the method and parallelism across time [7, 11]. In parallelism across the system and parallelism across the method each single process has to communicate with all other processes during each integration step. It may have a negative effect on the performance of computations, when they are executed in parallel systems with a slow communication. In this paper the parallel method for solving systems of ODEs, which belongs to the parallelism across time category, and does not require communication in each integration step, is proposed. This method originates from the approach towards the parallel analysis of transient states in electrical circuits published in [12]. The general idea of the method is based on decomposition of the time domain (the integration interval). Parallel computations in subsequent sub-intervals are conducted with the use of sequential numerical Runge-Kutta method. To start parallel computations it is necessary to know the values at the beginning of each sub-interval. In the proposed method those values are determined on the basis of an approximation of the convergence graph by the exponential function. As an example of the application of the parallel method, the analysis of dynamics of the asynchronous slip-ring motor will be shown.

2. The Parallel Analysis of Transient States

The proposed method is intended to conduct the analysis of transient states in electrical circuits in which the transient state is described by a state equation (a system of first-order ordinary differential equations), linear:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(t_0) = x_0 , \tag{1}$$

or non-linear:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{u}(t), t), \quad \mathbf{x}(t_0) = x_0 , \tag{2}$$

where: $\mathbf{x}(t) \in \mathbf{R}^n$ - the vector of state variables, $\mathbf{u}(t) \in \mathbf{R}^m$ - the vector of the input functions, $\mathbf{A}$ - the state matrix, $\mathbf{B}$ - the input matrix, $\mathbf{x_0}$ - the vector of initial conditions, $f(\mathbf{x}(t),\mathbf{u}(t),t)$ - the vector of nonlinear functions.

The main aim of the application of this method is to reduce the time of the state equation solving and thereby to reduce the time of the transient state analysis.

The parallel algorithm has been implemented using the "Master-Slave" principle. (Fig. 1).

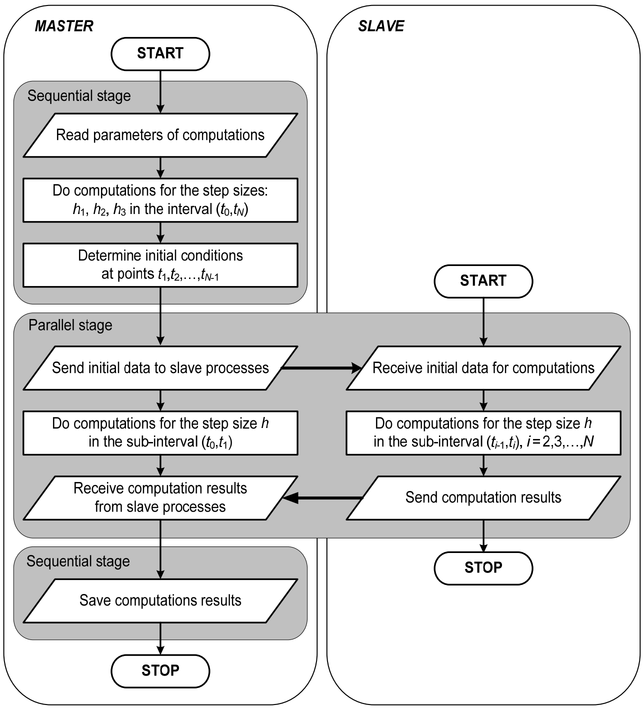

Fig. 1. The parallel algorithm flow chart
Rys. 1. Schemat blokowy algorytmu równoległego

The left part of the flow chart shows the operation executed by master process while the right part - by the slave processes. The algorithm is composed of three main stages. The first and the third stage are executed sequentially by the master process, while the second stage is executed in parallel by the master and all slave processes.

In the first stage of the algorithm, the master process divides the integration interval (t_0, t_N) into N equal sub-intervals of time (t_0,t_1), (t_1,t_2),…,(t_{N-1},t_N) (Fig. 2). The number of sub-intervals should be equal to the number of processes used in the second stage of the parallel algorithm.

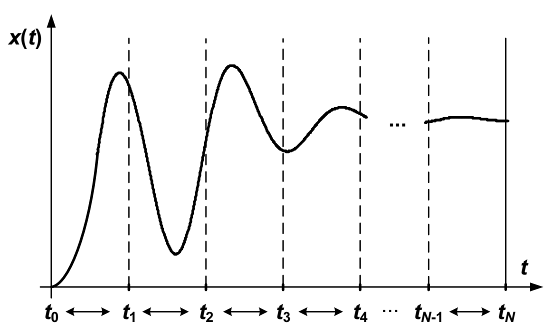

Fig. 2. Division of the integration interval (t_0, t_N) into sub-intervals
Rys. 2. Podział przedziału całkowania (t_0, t_N) na podprzedziały

In order to execute parallel computations in the second stage of the algorithm it is necessary to know the values of the state variables at the beginning of each sub-interval, i.e. at time points t_0, t_1, …, t_{N-1}. These values are also called the initial conditions. At time point t_0 the initial conditions are known from the assumptions, whereas at remaining time points the initial conditions must be determined. During examinations of the convergence of the solutions obtained with the use of sequential numerical method for solving systems of ODEs, it was observed that the convergence graph presented in logarithmic scale (Fig. 3) could be approximated by exponential function:

$$x(t) = A(1 - e^{-\alpha t}) \,. \tag{3}$$

The observation mentioned above was the basis for working out the method for determination of the initial conditions at the beginning of each sub-interval. Therefore, if the state variables values (x_{h1}, x_{h2}, x_{h3}) computed for large integration step sizes (h_1, h_2, h_3) are known, it is possible to compute A and α parameters of exponential function (3) and then to determine an approximate value (x_{hs}) for a small h_s step size (Fig. 3). Detailed description of the method for determination of the initial conditions was presented in papers [13] and [14].

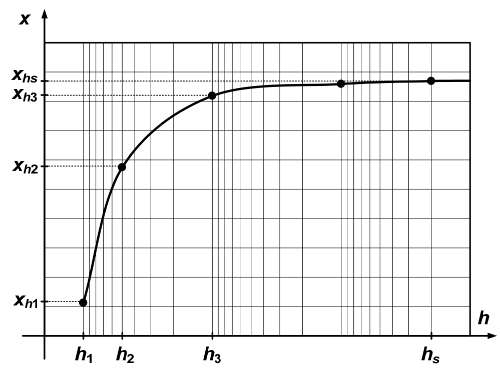

Fig. 3. The convergence graph ($h_1 > h_2 > h_3 > h_s$)
Rys. 3. Wykres zbieżności ($h_1 > h_2 > h_3 > h_s$)

In the practical implementation of the algorithm the master process solves state equation sequentially three times in the whole integration interval (t_0, t_N) with three integration step sizes h_1, h_2 and h_3, respectively. Next, on the basis of the obtained solutions, separately for each state variable and for each time point t_1, t_2, …, t_{N-1}, the master process computes parameters of equation (3) and then the initial conditions.

After determination of initial conditions, the second, parallel, stage of the algorithm is started. In this stage the master process sends the initial data to slave processes. After receiving of data, all processes (also the master process) execute computations with a small integration step size h. The master process computes the values of the state variables in the first sub-interval (t_0, t_1), the first slave process - in the second sub-interval (t_1, t_2), the second slave process - in the third sub-interval (t_2, t_3), etc. When the computations are finished, the slave processes send computed solutions to the master process.

In the third stage the master process saves the final solution, which consists of parts computed by particular processes, on his local hard disk.

3. Analysis of Transient State in Electric Motor

As a practical example of the application of the proposed method, the analysis of transient states in asynchronous slip-ring motor will be presented. The motor has the following rating data:

$$\begin{array}{lll} P_N = 10\,\mathrm{kW}, & n_N = 1445 \text{ r.p.m.}, & U_N = 220\,\mathrm{V} \\ I_{1N} = 21\,\mathrm{A}, & I_{2N} = 28\,\mathrm{A}, & p = 2, \\ \eta_N = 88.5\,\%, & \cos\varphi_N = 0.82, & M_{KN} / M_N = 3.5 \end{array} \,. \tag{4}$$

The transient state in this motor is described by a system of five non-linear ordinary differential equations [15]:

$$\begin{aligned}
\dot{x}_1 &= -a_1x_1 - a_5x_2 + a_4x_3 - b_4x_2x_5 - b_3x_4x_5 + e_1 \\
\dot{x}_2 &= a_5x_1 - a_1x_2 + a_4x_4 + b_4x_1x_5 + b_3x_3x_5 + e_2 \\
\dot{x}_3 &= a_2x_1 - a_3x_3 - a_5x_4 + b_2x_2x_5 + b_1x_4x_5 - e_3 \\
\dot{x}_4 &= a_2x_2 + a_5x_3 - a_3x_4 - b_2x_1x_5 - b_1x_3x_5 - e_4 \\
\dot{x}_5 &= -c_2x_5 + c_1x_1x_4 - c_1x_2x_3 - M \\
x_i(0) &= 0, \quad i = 1,2,\ldots,5
\end{aligned} \quad . \qquad (5)$$

where x_1, x_2 - standard form of the stator current, x_3, x_4 - standard form of the rotor current and x_5 - the angular velocity.

This system is presented in standard form, which was obtained by a transformation of the stator and rotor current equations with the use of orthogonal matrix [15]. The coefficients on the right hand side of equation (5) have the values resulting from the rating data of the motor and its electrical parameters. For the analyzed electric motor, the coefficients have the following values:

$$\begin{array}{llll}
a_1 = 82.826, & a_2 = 129.201, & a_3 = 64.19, & a_4 = 38.619 \\
a_5 = 314.159, & b_1 = 32.515, & b_2 = 50.722, & b_3 = 19.562 \\
b_4 = 30.515, & c_1 = 0.0885, & c_2 = 0.01, & e_1 = 72945 \\
e_2 = 72945, & e_3 = 113788, & e_4 = 113788, & M = 5
\end{array} \quad . \qquad (6)$$

The system of equations (5) was solved using the parallel method presented in previous paragraph. The computations were carried out with the use of a homogeneous cluster, which consists of six workstations. The cluster nodes are based on Intel SE7505VB2 motherboard, each equipped with an Intel Xeon 2.66 GHz processor, 1 GB RAM and an 80 GB hard disk drive. Individual nodes are connected by a Gigabit Ethernet with the Allied Telesyn AT-9410GB switch and the Intel 82540EM integrated network interface card. The software environment is Ubuntu Linux and Open MPI as the message passing library.

During computations the integration interval (t_0, t_N): t_0 = 0 s, t_N = 1.2 s was divided into six equal sub-intervals (t_0, t_1), …, (t_4, t_5). To determine the values of the state variables at the beginning of each sub-interval (the initial conditions), $h_1 = 2.5 \cdot 10^{-3}$ s, $h_2 = 10^{-3}$ s and $h_3 = 5 \cdot 10^{-4}$ s step sizes were assumed. The main computations were carried out with the $h = 10^{-8}$ s step size. The fourth-order Runge-Kutta method with a fixed step size was used as a sequential numerical method for solving system of ODEs.

Fig. 4-8 present the obtained parallel solution of all state variables. In these figures the limits of division of the integration interval into sub-intervals are also marked.

The accuracy of obtained parallel solution is mainly dependant on the accuracy of determination of the initial conditions. In order to estimate errors introduced by parallel method, the system of non-linear equations was solved with the application of a sequential algorithm of the Runge-Kutta method with step size $h = 10^{-8}$ s (the same step size as in the main computations in the parallel method).

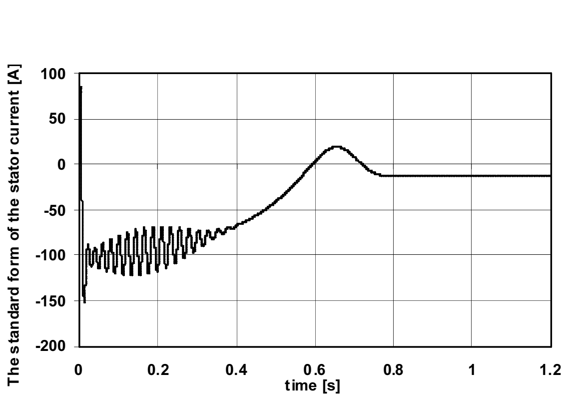

Fig. 4. The solution of the state variable x_1 - the standard form of the stator current
Rys. 4. Rozwiązanie dla zmiennej stanu x_1 - standardowa postać prądu stojana

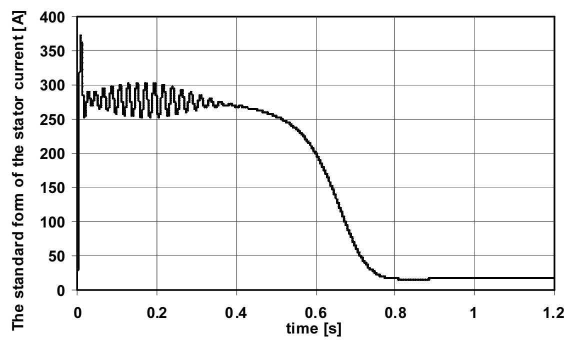

Fig. 5. The solution of the state variable x_2 - the standard form of the stator current
Rys. 5. Rozwiązanie dla zmiennej stanu x_2 - standardowa postać prądu stojana

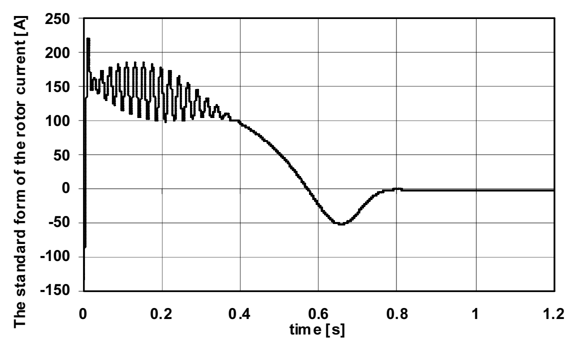

Fig. 6. The solution of the state variable x_3 - the standard form of the rotor current
Rys. 6. Rozwiązanie dla zmiennej stanu x_3 - standardowa postać prądu wirnika

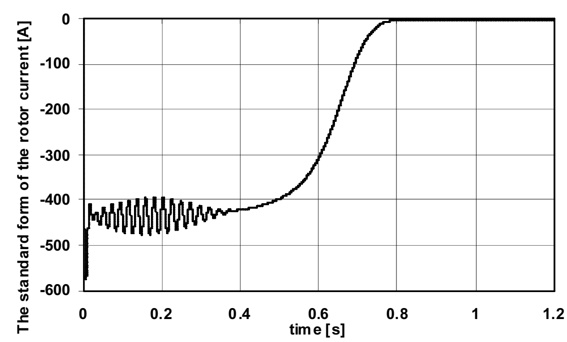

Fig. 7. The solution of the state variable x_4 - the standard form of the rotor current
Rys. 7. Rozwiązanie dla zmiennej stanu x_4 - standardowa postać prądu wirnika

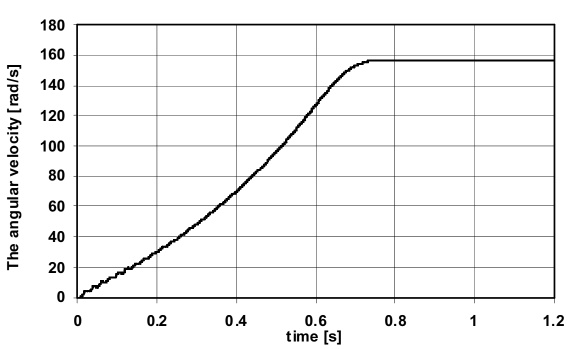

Fig. 8. The solution of the state variable x_5 - the angular velocity
Rys. 8. Rozwiązanie dla zmiennej stanu x_5 - prędkość kątowa

The obtained values of the sequential and parallel solutions at the beginning of each sub-interval are presented in Table 1 and Table 2, respectively.

Tab. 1. The values of the sequential solution at beginnings of sub-intervals
Tab. 1. Wartości rozwiązania sekwencyjnego na początku podprzedziałów

	$t_1 = 0.2$ s	$t_2 = 0.4$ s	$t_3 = 0.6$ s	$t_4 = 0.08$ s	$t_5 = 1.0$ s
x_1	-114.5762	-67.4113	3.6549	-13.0463	-12.5053
x_2	266.3318	268.4583	199.2937	16.2878	16.1026
x_3	170.1519	95.4645	-24.4774	-1.6258	-2.5287
x_4	-411.8975	-420.6091	-307.4145	-2.9664	-2.6768
x_5	29.8401	70.2515	127.1571	156.8273	156.7735

Tab. 2. The values of the parallel solution at beginnings of sub-intervals
Tab. 2. Wartości rozwiązania równoległego na początku podprzedziałów

	$t_1 = 0.2$ s	$t_2 = 0.4$ s	$t_3 = 0.6$ s	$t_4 = 0.08$ s	$t_5 = 1.0$ s
x_1	-114.5759	-67.4112	3.6553	-13.0463	-12.5053
x_2	266.3373	268.4586	199.3095	16.2884	16.1026
x_3	170.1573	95.4645	-24.4649	-1.6257	-2.5287
x_4	-411.8619	-420.6089	-307.4135	-2.9664	-2.6768
x_5	29.8402	70.2516	127.1573	156.8273	156.7735

The relative errors of the determination of the initial conditions were calculated with the use of the values presented in above-mentioned tables, as well as, with the use of formula:

$$\delta x = \frac{\left| x_{SEQ} - x_{PAR} \right|}{x_{SEQ}} \cdot 100\% \,. \qquad (7)$$

where x_{SEQ} - the values of sequential solution, x_{PAR} - the values of parallel solution. The obtained relative errors are presented in Table 3.

Tab. 3. The relative errors of the determination of state variables values at the beginning of each sub-interval
Tab. 3. Błędy względne wyznaczania wartości zmiennych stanu na początku podprzedziałów

	$t_1 = 0.2$ s	$t_2 = 0.4$ s	$t_3 = 0.6$ s	$t_4 = 0.08$ s	$t_5 = 1.0$ s
δx_1	0.0003 %	< 0.0001 %	0.0079 %	< 0.0001 %	< 0.0001 %
δx_2	0.0021 %	0.0001 %	0.0079 %	0.0034 %	< 0.0001 %
δx_3	0.0031 %	< 0.0001 %	0.051 %	0.0033 %	< 0.0001 %
δx_4	0.0086 %	< 0.0001 %	0.0003 %	0.0013 %	< 0.0001 %
δx_5	0.0004 %	0.0002 %	0.0002 %	< 0.0001 %	< 0.0001 %

As we can see, the relative errors are very small. It should ensure a high accuracy of the parallel solution in the whole integration interval.

In parallel computing the most commonly used performance metric is speedup. In order to determine the speedup, the integration interval was divided into 2, 3, 4, 5 and 6 sub-intervals and computations were carried out with the use of 2, 3, 4, 5 and 6 computing nodes, respectively. The obtained time of the parallel method solution was compared with the time of the sequential solution (Runge-Kutta method with fixed step size: $h = 10^{-8}$ s). The achieved values of the speedup are presented in Fig. 9.

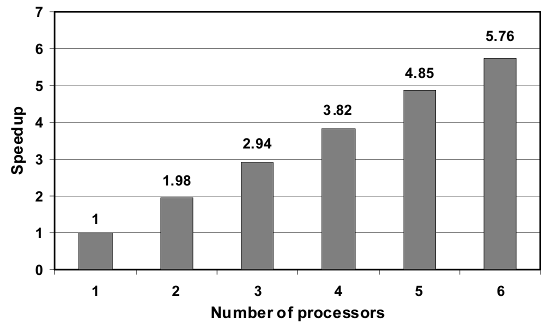

Fig. 9. The achieved values of the speedup
Rys. 9. Otrzymane wartości przyspieszenia obliczeń

4. Conclusions

In this paper the analysis of dynamics of the asynchronous slip-ring motor was presented. The applied parallel method for transient state analysis is composed of three main stages. Despite the fact that two stages are executed sequentially, good speedup was obtained in the presented example. The initial conditions at the beginning of each sub-interval determined on the basis of an approximation of the convergence graph by exponential function were close to the values of the sequential solution. It ensured a high accuracy of the parallel solution.

This work was carried out within the frame of Białystok Technical University Grant No: S/WE/3/08.

5. References

[1] Kaczorek T., Dzieliński A., Dąbrowski W., Łopatka R.: Podstawy teorii sterowania. WNT, Warszawa, 2005.
[2] Nise N.S.: Control Systems Engineering. 4th Edition. John Wiley & Sons, 2003.
[3] Isidori A.: Nonlinear Control Systems. Springer-Verlag, Berlin Heidelberg New York, 1995.
[4] Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.: Numerical Recipes. The Art of Scientific Computing. Third Edition. Cambridge University Press, 2007.
[5] Butcher J. C.: Numerical Methods for Ordinary Differential Equations. John Wiley & Sons, 2004.
[6] Zarowski Ch. J.: An Introduction to Numerical Analysis for Electrical and Computer Engineers. John Wiley & Sons, 2003.
[7] Burrage K.: Parallel and Sequential Methods for Ordinary Differential Equations. Clarendon Press, Oxford, 1995.
[8] Gear C. W., Xuhai X.: Parallelism Across Time in ODEs. Applied Numerical Mathematics 11, pp. 45-68, 1993.
[9] Gear C. W.: Massive Parallelism Across Space in ODEs. Applied Numerical Mathematics 11, pp. 27-43, 1993.
[10] Burrage K.: Parallel Methods for Initial Value Problems. Applied Numerical Mathematics 11, pp. 5-25, 1993.
[11] Petcu D.: Parallelism in Solving Ordinary Differential Equations. Mathematical Monographs 64, Timisoara University Press, 1998.
[12] Forenc J.: Analiza spekulacyjna stanów nieustalonych w układach elektrycznych. Praca dokt. Politechnika Białostocka, Białystok, 2006.
[13] Forenc J.: The Parallel Method for the State Equation Solving. Computer Applications in Electrical Engineering, ed. by R. Nawrowski, Poznań University of Technology, Poznań, pp. 87-96, 2007.
[14] Forenc J.: Determination of the initial conditions in the parallel method for state equation solving. Electronics and Telecommunications Quarterly, Vol. 54, No 3, pp. 377-388, 2008.
[15] Baron B., Marcol A., Pawlikowski S.: Metody numeryczne w Delphi 4. Helion, Gliwice, 1999.

otrzymano / received: 02.11.2009
przyjęto do druku / accepted: 04.01.2010 *artykuł recenzowany*