Methodological support for control systems in the analysis of dynamic objects during their modeling and management*

Olexandr Fomin[1,*,†], Segii Polozhaenko[2,†], Andrii Prokofiev[3,†] and Oleksiy Tataryn[4,†]

[1,2,3,4] *Odesa Polytechnic National University, 1, Shevchenko ave., Odesa, 65044, Ukraine*

Abstract

A characteristic feature of modern control systems is the widespread introduction of various types of computing systems that implement methods and algorithms of control and modeling. Therefore, the quality of functioning of control systems is largely determined by the characteristics of the used computational tools. The problem of ensuring the quality of functioning of control systems with a computer in the control loop is very complex both technically and mathematically. In solving this problem, practice has put forward one of the first places the scientific direction, associated with the development of methods and means to ensure the reliability of computational processes and information processing.

In real control systems or modeling processes of calculation are inevitably accompanied by various kinds of noises and errors. Thus, in the case of numerical analysis of the equations of dynamics on a digital machine, the solution errors, in addition to all, can be caused by failures and failures in the work of the hardware.

Under failures we understand distortions of processed and controlling information under the influence of primary self-eliminating reasons (incorrect operation of binary hardware elements, interferences, power supply voltage fluctuations and so on). Moreover, most failures have an accidental character. Therefore, an effective current control is needed, which allows detecting and eliminating an error before it spreads in the control circuit, which determines the operability as one of the main characteristics of error control means.

Keywords

Control system, control and simulation processes, calculation failures.

1. Introduction

Currently, both software and hardware methods of control of simulation processes are used, as well as their combinations [1 – 12]. Software control methods, in turn, can be divided into test and software-logic [7]. The test control is intended to check the hardware or software failures [13] at the moment, when the work task is not solved on the computing and controlling device or the test signals are allowed during the problem solving [14, 15].

In the case under consideration, it is necessary to control the computational process in the working mode of operation of the control system or simulation. And the main object of control are hardware failures. More rational are program-logical methods of control, allowing you to control in the process of solving the problem. Sufficiently complete review of existing methods of program-logic control is given in a number of works [16 – 19]. The main limitation of application of existing methods of control of numerical solution of complex non-linear differential equations (as the main mathematical apparatus describing dynamic behavior of systems) in control and modeling systems is the time taken for control. Below we propose methods to control errors caused by failures in numerical solution of differential equations (in terms of control theory – equations of dynamics) in real time, i.e. at the rate with the realization of the control action in the control loop.

Proceedings of the Workshop on Scientific and Practical Issues of Cybersecurity and Information Technology at the V international scientific and practical conference Information security and information technology (ISecIT 2025), June 09–11, 2025, Lutsk, Ukraine

* Corresponding author.
† These authors contributed equally.
fomin@op.edu.ua (O. Fomin); polozhaenko@op.edu.ua (S. Polozhaenko); fallbrick1985@gmail.com (A. Prokofiev); otataryn@stud.op.edu.ua (O. Tataryn)
0000-0002-8816-0652 (O. Fomin); 0000-0002-4082-8270 (S. Polozhaenko); 0000-0002-4520-4248 (A. Prokofiev); 0009-0005-3888-6569 (O. Tataryn)

The aim of the work is to develop constructive algorithms to control the solution of the equations of dynamics of objects in the problems of synthesis of control actions.

2. Self-assessment of numerical solution algorithms

Numerical methods for solving the dynamics equations are step by step procedures for obtaining the desired function. In this case, the computational process is naturally divided into relatively constant in time parts corresponding to a certain desired function at the integration step. Besides, the computational process is divided inside the integration step into relatively constant in time parts corresponding to determination of values of the right part of the solved equation system

$$\dot{\mathbf{Y}} = \mathbf{f}(\mathbf{Y}, \mathbf{U}, t), \mathbf{Y}(t_0) = \mathbf{Y}_0 \tag{1}$$

where $\mathbf{Y} = (y_1, y_2, ..., y_n)^{\mathrm{T}}$ – vector of state variables, $\mathbf{U} = (u_1, u_2, ..., u_m)^{\mathrm{T}}$ – vector of control signals, $\mathbf{f} = (f_1, f_2, ..., f_n)^{\mathrm{T}}$ – vector-function, t – independent time parameter.

Let us consider the possibilities of numerical solution procedures (1) on organization of their control (self-control). To organize control (self-control) it is necessary to have reference values to compare the results of calculations. Controlled results of previous calculations or results with low probability of failure can be used as reference values. For any method of numerical solution of equations of dynamics of a type (1) failure control can be organized by comparing the results y_i and y_{i+1}, obtained at the current and subsequent steps of integration, respectively. The presence or absence of failures is determined by checking the condition

$$\xi = \rho(y_i, y_{i+1}) \in D \tag{2}$$

where ξ – is a measure of proximity of values y_i and y_{i+1}, D – is the region of admissible values of this measure.

The sizes of the region D depend on solution changes from step to step and determine the accuracy of control. If the boundaries of the area D correspond to the maximum possible difference between the values of y_i and y_{i+1} at the whole step of integration in the absence of failures, the error that can be missed in the control is equal to twice the value of the area D, which follows from (2), since in the latter the value y_{i+1} should be replaced with y_i. In this case, the accuracy of control may be unsatisfactory.

To control errors caused by failures and malfunctions, methods of controlling the methodological error of numerical solution on the integration step or peculiarities of construction of computational algorithms can be used.

In prediction and correction methods [20], the difference $\xi = |y_{\mathrm{f}} - y_{\mathrm{c}}|$ between the predicted y_f and the corrected y_c values of the desired function can be used to judge about the presence of a hardware failure. However, the existing correlation between the values y_f and y_c removes the reliability of control.

Let us show it on the example of the improved Euler-Cauchy method. Let the prediction be carried out by the formula:

$$y_{i+1} = y_i + hf_i, \tag{3}$$

where h – is a step of integration, $f_i = f(\mathbf{Y}, \mathbf{U}, t_i)$, and the correction is carried out according to expression

$$y_{i+1} = y_i + \frac{h}{2}\left(f_i + \widetilde{f}_{i+1}\right) \tag{4}$$

where $\widetilde{f}_{i+1} = f\left(\widetilde{\mathbf{Y}}, \mathbf{U}, t_{i+1}\right)$.

As can be seen from (3) and (4), at one step of integration the function $f(\mathbf{Y}, \mathbf{U}, t_i)$ is calculated twice at the points t_i and t_{i+1}. Since this process, as a rule, takes the main part of computational work, let us assume that a failure has occurred at the moment of the calculation $f(\mathbf{Y}, \mathbf{U}, t_i)$ and the function value f_i has been obtained with error Δf_i. Then the function value calculated by formula (3) will be

$$\widetilde{y}_{i+1} = y_i + h\left(f_i + \Delta f_i\right) = y_i + h\,f_i + h\Delta f_i \tag{5}$$

Here the value is the $\Delta f_{i+1}=h\Delta f_i$ – error in the prediction value. Next, we will propose methods of algorithmic control, suitable for control of calculation process at any methods of numerical solution of equations of dynamics. In this case we will consider the control, based on application of additional (control) algorithm on each step of numerical solution of the equations of dynamics of the form (1).

3. Overview of results and sources

Let us consider the problem of obtaining algorithm A_k, which we will formulate as follows: for a certain class of problems Q to obtain an algorithm of solution A_k, the result of which $\mathbf{Y}$ in relation to the problem solution $q \in Q$ coincides with the accuracy ε of the result obtained by the main algorithm A_k. In this case the time and memory costs for the algorithm's realization must be within the specified limits.

When computing by the basic algorithm A_b for the class of problems under consideration for modeling the dynamics of dynamic objects, most of the time is spent on computing the right side of the system of equations (1). The computation time by the algorithm A_k can be significantly reduced in comparison with the algorithm if it uses the values of the function $f(\mathbf{Y}, \mathbf{U}, t)$ obtained by the algorithm A_b. Since the computing process according to the algorithm A_k is also subject to failures, the refusal to calculate the function $f(\mathbf{Y}, \mathbf{U}, t)$ will greatly reduce the probability of a first-order error α (i.e., the correct initial hypothesis will be rejected). In this case the accuracy of control at the controlled step is improved, if the controlled algorithm uses the information controlled at previous steps, in other words: use the idea of extrapolation for control. Then control of each component of vector equation (1) using extrapolation method can be done separately. Therefore, the construction of the control algorithm A_k can be considered for some component of the vector $\mathbf{Y}$ and, at the same time, not to lose generality of the statement.

In this case, the control value of the function at the $(i+1)$-th step of integration can be defined by the formula

$$\bar{y}_{i+1} = \sum_{p=0}^{p_1} a_p y_{p-1} + h\sum_{s=1}^{s_1} b_s f\left(\mathbf{Y}, U, t_{i-s}\right) \tag{6}$$

$h=t_{i+1}-t_i$ – step of integration, a_p, b_s, – coefficients.

In (6), only values of function and its derivative, controlled during previous steps of integration, are used. Upper limits of sums p_1 and s_1 are selected from the conditions of required accuracy of the controlled algorithm. Since the controlled algorithm is not always required to ensure stability, the coefficients a_p, b_s, can be selected from the condition of minimal local error (i.e., equality to zero of the residual term r_i=0 between the exact and numerical solution (6)).

Many algorithms for the numerical solution of ordinary differential equations determine estimates of the derivative function between integration nodes. Since these estimates provide information on the behavior of the solution at a point closer to the controlled value of the function than the value $f(\mathbf{Y}, \mathbf{U}, t_{i-s})$ in (6) at s_1=1, it is advisable to take this information into account when extrapolating. If, in extrapolation, we take into account one estimate of the derivative obtained by the basic algorithm A_b, then one term will be added to expression (6)

$$\bar{y}_{i+1} = \sum_{p=0}^{p_1} a_p y_{p-1} + h\left[\sum_{s=1}^{s_1} b_s f(\mathbf{Y}, U, t_{i-s}) + b_0 f(\mathbf{Y}, U, t_{i+l})\right] \tag{7}$$

where $f(\mathbf{Y}, \mathbf{U}, t_{i-s})$ – is the estimate of the derivative, and b_0 – is the coefficient.

For example, a fourth-order precision control formula in which the derivative estimate is used is $\bar{y}_{i+1} = a_1 y_1 + a_2 y_2 + h(b_0 \dot{y}_{i-1+\alpha} + b_1 \dot{y}_{i-1} + b_2 y_{i-2})$.

Obviously, in the general case, any information about the solution obtained by the basic algorithm A_b and checked at the previous steps of integration can be used for extrapolation.

It should be noted that extrapolation uses finite differences of the necessary order. For example, in the linear method of extrapolation, the control value of the function at a point t_{i+1} is defined as

$$\bar{y}_{i+1} = y_i + \frac{y_i - y_{i-1}}{t_i - t_{i-1}}(t_{i+1} - t_i) \tag{8}$$

Application of expression (8) for organization of control of numerical solution of equation of dynamics of a kind (1) does not allow taking into account all available information about controlled solution. Indeed, in numerical solution of equation of dynamics (1), as a rule, not only the values of function in integration nodes are known, but also the values of derivatives and their estimates which are used in (8). In addition, the use of finite differences to determine derivatives increases the error of the control value of the function $\bar{y}_{i+1}$.

Thus, expression (6), in contrast to formulas of the type (8), allows taking into account the results of calculations using the main algorithm more completely A_b, which leads to more accurate results in extrapolations.

As an example, Fig. 1 and Fig. 2 show the dependencies of the mathematical expectation and the standard deviation of the value ξ from the integration step h in the equation of the form (1), which describe the dynamics of an airplane flight.

4. Interpolation control method

When using the extrapolation method for control, as already noted, it is possible to correct the solution for errors not by the repeated method, but by replacing y_{i+1} by $\bar{y}_{i+1}$, which is essential in real-time control and modeling. However, the control accuracy depends on the accuracy of extrapolation of the solution using expression (6) and may not be satisfactory.

From the point of view of increasing control accuracy, a method based on the idea of interpolation may be considered preferable to the extrapolation method. The interpolation control formulas, in contrast to extrapolation formulas, take into account changes in the solution at the controlled calculation step, which leads to a decrease in the difference of values and, obtained by algorithms A_b and A_k. Increasing the accuracy of interpolation can be obtained by attracting the results of calculations by the main algorithm A_b before the controlled calculation step. This leads to multi-step interpolation formulas. To increase the control sensitivity to errors that have been classified as failures, use the control interpolation formula to determine the function value obtained in the calculation step closest to the controlled calculation step.

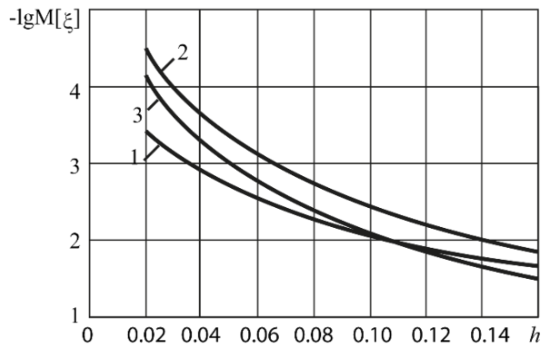

Figure 1: dependencies of the mathematical expectation of the value ξ from the integration step h in the equation of the form (1).

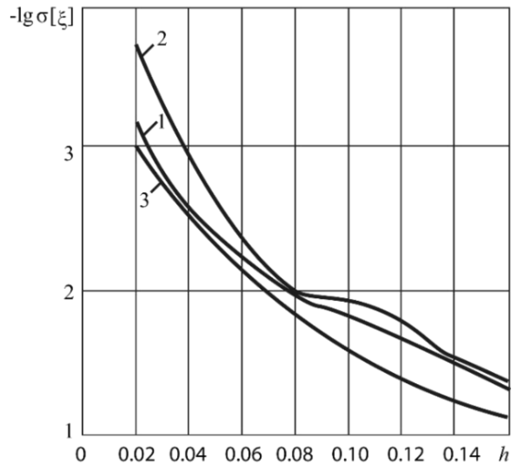

Figure 2: dependencies the standard deviation of the value ξ from the integration step h in the equation of the form (1) $\delta\sigma$.

Considering the above, the control multi-step interpolation formula by means of which the process of calculations at the (i+1)-th step is controlled, obviously, can be represented as follows:

$$\bar{y}_i = \sum_{p=1}^{p_1} a_p y_{i-p} + h\sum_{s=1}^{s_1} b_s f\left(\mathbf{Y}, U, t_{i-s}\right) \tag{9}$$

where a_p, b_s – are real coefficients; $p \neq 0$ (otherwise a_0=1, and all other coefficients are zero).

As in the case of the extrapolation formula (6) in (9), the upper limits of summation have to be chosen depending on the required order of control accuracy, and the coefficients a_p and b_s – from the condition of the accuracy of representation of polynomials of appropriate degree.

The principal difference between the interpolation expression (8) and the extrapolation expression (6) as control algorithms is as follows. In the case of extrapolation formulas, the directly controlled value is the value of the function obtained by the basic algorithm A_b at the controlled step of integration, and the control value $\bar{y}_i$ is obtained by expression (6). In case of interpolation formulas the controlled value is obtained by expression (8), and the reference value is the value of the function obtained by the main algorithm A_b on the previous (i-p) step of the calculation, i.e., in fact, indirect control is carried out. The error of calculating the value $\bar{y}_{i+1}$ according to the basic algorithm A_b is transformed into the error $\Delta\bar{y}_i$ of determination $\bar{y}_i$ by the formula (8). The deviation $\Delta\bar{y}_i$ in this case, in the framework of the linear theory of accuracy, will be defined as

$$\Delta\bar{y}_i = L\eta$$

where

$$L = \partial \bar{y}_i / \partial y_{i+1}$$

As applied to expression (8), L is determined by

$$L = a_* - b_* h \frac{\partial f(\mathbf{Y}, U, t_{i+1})}{\partial y_{i+1}}$$

where a_* and b_* are the coefficients, at y_{i+1} in (9).

Fig. 3 and Fig. 4 shows the dependence of the mathematical expectation and the standard deviation of the value

$$\xi_i = \left[\sum_{j=1}^{p} \left(\frac{y_{i,j} - \bar{y}_{i,j}}{\max\limits_i |y_j|} \right)^2 \right]^{1/2}$$

on the integration step h for equation (1) describing the aircraft flight dynamics. The curves 1 and 2 were obtained by using interpolation formulas of the 1st and 2nd orders on the $(i+1)$-th step of integration according to the basic algorithm:

$$y_i = y_{i+1} - h\dot{y}_{i+1} \,,$$

$$\bar{y}_i = y_{i+1} - 0{,}5[h(\dot{y}_{i+1} + \dot{y}_i)]$$

For comparison, curve 3 corresponding to the application of the extrapolation formula for control

$$\bar{y}_{i+1} = y_i + h(2{,}25\dot{y}_{i-1} - 1{,}25\dot{y}_{i-2})$$

It can be seen that by increasing the integration step, the accuracy of the interpolation control method, as compared to the extrapolation method, increases.

The interpolation method of control allows you to increase the accuracy of obtaining reference values. However, when organizing control of the computational process it should be borne in mind that when using the interpolation method of control there is no possibility to replace the erroneous value obtained by the basic algorithm A_b with the value obtained by the control algorithm A_k (which is possible by the extrapolation method).

5. Adaptive control method

The coefficients of control algorithms of the form (6) and (8) are determined from the condition of obtaining the exact result by the control algorithm A_k for the solution of $y_j(t)$, $j = \overline{1, n}$, representing a polynomial of the highest possible degree. Taking into account the above said, the choice of coefficients a_p and b_s allows to get the given order of control error at relatively small volume of calculations by the control algorithm A_k in the course of calculation. In fact, the control accuracy defined by the residual of the Taylor series for control formulas varies from step to step, depending on the smoothness of the solution $y_j(t)$, $j = \overline{1, n}$.

Let us propose a method for organizing the control of the solution of the system of equations (1) with the help of the control algorithm A_k, which reconstructs itself depending on the behavior of the controlled solution $y_j(t)$, $j = \overline{1, n}$. The peculiarity of this approach consists in the fact that the results controlled at the next step are used to determine the parameters of the control algorithm A_k. In this case, the parameters are determined from the condition of a minimum of some measure of proximity of results obtained by the control algorithm A_k and the basic algorithm A_b at the section of the solution including the last step of calculations under control. The obtained algorithm, denoted as A_a, is used to control the next calculation step. After that the parameters are refined again. Obviously, the algorithm A_a, in view of the above, can be regarded as adaptive.

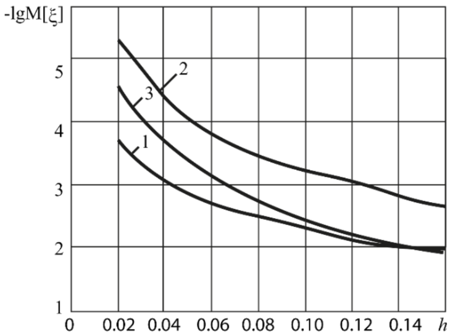

Figure 3: Dependence of the math ematical expectation of the value ξ on the integration step *h* for equation (1)

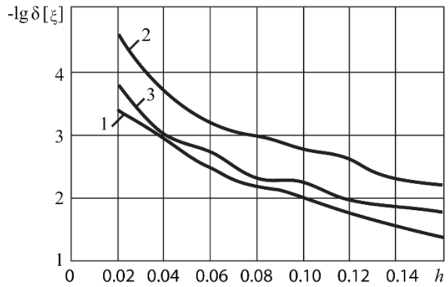

Figure 4: Dependence of standard deviation of the value ξ on the integration step *h* for equation (1)

The method under consideration allows the following interpretation: on the basis of the controlled information about a physical process (i.e. its dynamics described by system (1) and the assumption that behavior of the solution $y_j(t)$, $j=\overline{1,n}$, from step to step changes slowly enough) the problem of parametric identification is solved for a given structure of the mathematical process description (i.e. system (1)). The resulting model in the form of an algorithm A_a is used for subsequent control. Otherwise, as control information becomes available, the model used for control (algorithm A_a) is specified (or – adapted). Further, we will use methods of parametric identification of inertia-free objects [3, 21] to determine the parameters of the control algorithm, which is already considered as the algorithm A_a.

Using results of calculations on the basic algorithm A_b obtained to the (i+1)-th step of integration, we can determine parameters of the control algorithm A_a on the (i+1)-th step of the system of algebraic equations

$$\mathbf{CQ} = \mathbf{B} \tag{10}$$

For the ectrapolation control method which uses evaluation of the derivative at the point $t_{i-1+\alpha}$, the values included in expression (10) are determined on the (i+1)-th step of integration as follows

$$\mathbf{Q} = \left(a_{-1}, a_1, \ldots, a_{p_1}, \ldots, b_0, b_1, \ldots, b_{s_1}\right)^{\mathrm{T}},$$

$$\mathbf{B} = \left(y_i, y_{i-1}, \ldots, y_{i-k+1}\right)^{\mathrm{T}},$$

$$\mathbf{C} = \begin{bmatrix} y_{i-1} \cdots y_{i-p_1-1} & \dot{y}_{i-2+\alpha} & \dot{y}_{i-2} \cdots \dot{y}_{i-s_1-1} \\ y_{i-2} \cdots y_{i-p_1-2} & \dot{y}_{i-3+\alpha} & \dot{y}_{i-3} \cdots \dot{y}_{i-s_1-2} \\ \cdots \quad \cdots . \;\; .. & \cdots & \cdots \quad \cdots \quad \cdots \quad \cdots \\ y_{i-k} \cdots y_{i-p_1-k} & \dot{y}_{i-k-1+\alpha} & \dot{y}_{i-k-1} \cdots \dot{y}_{i-s_1-k} \end{bmatrix}$$

For the interpolation method of control: control method

$$\mathbf{Q} = \left(a_{-1}, a_1, \dots, a_{p_1}, \dots, b_{-1}, b_0, b_1, \dots, b_{s-1_1}\right)^{\mathrm{T}},$$

$$\mathbf{B} = \left(y_{i-1}, y_{i-2}, \dots, y_{i-k}\right)^{\mathrm{T}},$$

$$\mathbf{C} = \begin{bmatrix} y_i & y_{i-2} & \cdots & \dot{y}_{i-p_1-1} & \dot{y}_i & \dot{y}_{i-1} & \cdots & \dot{y}_{i-s_1-1} \\ y_{i-1} & y_{i-3} & \cdots & \dot{y}_{i-p_1-2} & \dot{y}_{i-1} & \dot{y}_{i-2} & \cdots & \dot{y}_{i-s_1-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{i-k+1} & y_{i-k-1} & \cdots & \dot{y}_{i-p_1-k} & \dot{y}_{i-k+1} & \dot{y}_{i-k} & \cdots & \dot{y}_{i-s_1-k} \end{bmatrix}$$

If the matrix **C** is not degenerate and the number of its rows is equal to the dimension of the parameter vector (for the extrapolation method $\mathbf{\Pi}_{\text{ers}} = \left(a_{-1}, a_1, a_2, \dots, a_p, b_0, b_1, \dots, b_s\right)$ and interpolation method $\mathbf{\Pi}_{\text{int}} = \left(a_1, a_2, \dots, a_p,\ b_1, \dots, b_s\right)$, respectively) defined by the value of *k*, then the evaluation $\hat{\mathbf{Q}}$ of the parameter matrix *Q* (for both extrapolation and interpolation methods) is the solution of the system of algebraic equations

$$\hat{\mathbf{Q}} = \mathbf{C}^{-1}\mathbf{B}$$

and the results obtained by the main and control algorithms on the sequence $y_i, y_{i-1}, \dots, y_{i-k+1}$ coincide.

If the matrix **C** is rectangular, we can use the method of least squares to determine the estimate of the parameter vector.

Example. As an example, consider the Runge-Kutta algorithm of second order precision [22, 23] with control when solving the equation of dynamics

$$\dot{y} = f(\mathbf{Y}, U, t) \tag{11}$$

The solution of problem (11) boils down to the following:

1. Do the calculations by Runge-Kutta algorithm

$$\begin{gathered} k_1 = hf(y_i, u_i, t_i), \\ k_2 = hf\left[y_i + \beta k_1, u(t_i + \alpha h), t_i + \alpha h\right], \\ y_{i+1} = y_i + \sum_{r=1}^{i} \gamma_r k_r, \end{gathered} \tag{12}$$

where α, β, γ – parameters of the algorithm. 2.

2. Determine the control value of the function by the first-order extrapolation formula of the form (6)

$$\bar{y}_{i+1} = ay_i - hb\dot{y}_{i-1+\alpha} \tag{13}$$

where $\dot{y}_{i-1+\alpha} = f\left[y_i + \beta k_1, u(t_i + \alpha h), t_i + \alpha h\right]$ – is the value of estimation of derivative, obtained by algorithm A_b on the *i*-th step of integration. At the first step of integration coefficients *a*, *b* are determined from the condition of exact result according to the control algorithm A_k for the polynomial of the first degree of accuracy. 3.

3. Determine the value of

$$\xi = \left|y_{i+1} - \bar{y}_{i+1}\right|$$

4. Check condition

$$\xi \le \xi_{\sup}, \tag{14}$$

where $\xi_{\sup}$ – is a permissible value of ξ. If condition (14) is not satisfied, then correct the solution (for example, by replacing with y_{i+1} на $\bar{y}_{i+1}$). If $t < t_k$, where t_k – integration time, then proceed to item 5, otherwise carry out the end of the calculation process.

5. Form the matrix

$$\mathbf{C} = \begin{bmatrix} y_{i+1} & \dot{y}_{i-2+\alpha} \\ \dots & \dots & \dots \\ y_{i-k} & \dot{y}_{i-k-1+\alpha} \end{bmatrix}$$

where $\dot{y}_{i-k-1+\alpha}$ – value of evaluation of derivative, obtained by algorithm (12) on (i-k-1)-step of integration. The value defines the number of matrix rows.

6. Form the vector

$$\mathbf{B} = (y_i, \dots, y_{i-k-1})^{\mathrm{T}}$$

7. Determine the coefficients of the control algorithm (13) from the system of algebraic equations

$$\mathbf{CQ} = \mathbf{B}$$

where $\mathbf{Q} = (a, b)^{\mathrm{T}}$.

To continue the solution go to step 1.

As can be seen from the solution procedure above, additional calculations are required to determine the current value of coefficients a, b. It is possible to simplify the control procedure by determining not the whole vector Q, but its part, assuming that the remaining vector components have equal values obtained from the exact result of the control formula for a polynomial of a possibly high degree. Thus, in the considered procedure, it was possible to determine the first coefficient a by assuming the other b=1. This corresponds to the transformation of the model (truncation of the model by parameters).

6. Conclusion

The main purpose of the considered algorithms is to detect failures during the numerical solution of the equations of dynamics in real-time simulation and control systems.

Since both the basic Ab and the control Ak algorithms are executed by different programs, the proposed control methods also allow detection of hardware failures during computations.

Spending time on control depend on the method of control, the order of accuracy of the control formula, the accuracy measure of calculations and are determined by the number of multiplication and addition operations on the control formula, the time of determination of the accuracy measure of calculations and the operation of comparing the obtained value of the accuracy measure of calculations with tolerances.

The main advantages of the proposed methods of control should be the simplicity and low cost of time for control at a relatively high control accuracy, which can be regulated by both changing the method of control and changing the order of accuracy of control formulas.

The disadvantages of the proposed methods include "inherent noise" (i.e., mismatch of results at the integration step obtained by the basic Ab and control Ak algorithms), which reduces the control accuracy. The characteristics of "intrinsic noise" of the control algorithms Ak are the initial data for choosing the control method and defining the control quality indices.

The fact, that the model of the controlled process (i.e. equation of dynamics of the form (1)) is known, is essentially used in suggested methods of control. This allows to refuse from application of finite differences for determination of derivatives and to organize iterative control. When finite differences are used, it is difficult to organize iterative control because the controlled variable and the control variable can be used together to determine the derivative estimate.

The proposed control methods may be used to control dynamic processes, the mathematical model of which is known. For example, with their help it is possible to organize control of sensors of the state of a dynamic object.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Crnkovic I., Hinch B., Jonsson T., Kiziltan Z. Specification, Implementation and Deployment of Components // Communications of the ACM.–2002.–vol.45.–№ 10.–P.35– 40.

[2] R. Murray-Smith, T. Johansen, Multiple model approaches to nonlinear modelling and control, CRC Press, 2020.

[3] J. Schoukens, L. Ljung, Nonlinear system identification: a user-oriented roadmap, Preprint submitted to IEEE Control Systems Magazine, 2019.

[4] L. Wenyuan, L. Zhu, F. Feng, W. Zhang, Q.-J. Zhang, L. Qian, G. Liu. A time delay neural network based technique for nonlinear microwave device modelling, in: Micromachines, Basel, volume 11, no. 9: 831, 2020. doi: 10.3390/mi11090831.

[5] H. Eivazi, H. Veisi, M. H. Naderi, V. Esfahanian, Deep neural networks for nonlinear model order reduction of unsteady flows, Physics of Fluids, volume 32, no. 105104 (2020). doi: 10.1063/5.0020526.

[6] C. A. Mitrea, C. K. M. Lee, Z. Wu, A comparison between neural networks and traditional forecasting methods: case study, in: International journal of engineering business management, volume 1, no. 2, 2009, pp. 19–24. doi: 10.5772/6777.

[7] Kaminski, G., Williams, G. and Ammann, P. (2008), Reconciling perspectives of software logic testing. Softw. Test. Verif. Reliab., 18: 149-188.

[8] O. O. Fomin, O. D. Ruban, O. V. Rudkovskyi, Method for construction the diagnostic features space of switched reluctance motors based on integral dynamic models, in: Problemele energeticii regionale, no. 4, 2020, pp. 35–44.

[9] A. R. Rao, M. Reimherr, Non-linear functional modelling using neural networks, 2021. URL: https://arxiv.org/abs/2104.09371.

[10] R. Pintelon, E. Louarroudi, J. Lataire, Nonparametric estimation of time-variant dynamics, in: 17th IFAC Symposium on System Identification, Beijing, China, 2015.

[11] B. A. Khudayarov, F. Zh. Turaev, Mathematical Simulation of Nonlinear Oscillations of Viscoelastic Pipelines Conveying Fluid, Applied Mathematical Modelling 66 (2019) 662-679, https://doi.org/10.1016/j.apm.2018.10.008.

[12] S. Polozhaenko, O. Fomin, A. Orlov, V. Krykun and D. Lys, "Interpretation Method for Dynamic States Neural Network Models," 2022 IEEE 3rd International Conference on System Analysis & Intelligent Computing (SAIC), 2022, pp. 1-5, doi: 10.1109/SAIC57818.2022.9923008.

[13] R. Bakhshi, S. Kunche, M. Pecht. Intermittent Failures in Hardware and Software / Journal of Electronic Packaging, Transactions of the ASME. 2014. Vol. 136(1). 011014

[14] Kuravsky L.S., Marmalyuk P.A., Baranov S.N., Alkhimov V.I., Yuryev G.A. and Artyukhina S.V. A New Technique for Testing Professional Skills and Competencies and Examples of its Practical Applications. – Applied Mathematical Sciences. Vol. 9. 2015. No 21. Pp. 1003-1026.

[15] Kuravsky L.S., Marmalyuk P.A., Yuryev G.A., Dumin P.N., Panfilova A.S. Probabilistic modeling of a testing procedure. // Applied Mathematical Sciences. 2015. T. 9. № 81-84. C. 4053-4066.

[16] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, in: ACM Computer Surveys, volume 51, no. 5, 2018, pp. 1–42. doi: 10.1145/3236009.

[17] R. Melendy, M. Foster, M. Foster. A Review Of Programmable Logic Controllers In Control Systems Education / American Society for Engineering Education. Chad. 2010.

[18] Seth S.C., Pan L., Agrawal V.D. PREDICT-Probabilistic Estimation of Digital Circuits Testability // 15th Annual Fault-Tolerant Computing Symposium. Ann Arbor, MI. June, 2005. P. 220-225.

[19] Savir J., Ditlow G.S., Bardell P.H. Random Pattern Testability // IEEE Trans. On Comp. 2014. - Vol. C-33, № 1. - P. 79-90.

[20] S. Paternain, M. Morari and A. Ribeiro, "A Prediction-Correction Method for Model Predictive Control," 2018 Annual American Control Conference (ACC), 2018, pp. 4189-4194, doi: 10.23919/ACC.2018.8431785.

[21] A. Agresti, Foundations of linear and generalized linear models, Wiley series in probability and statistics, 2017.

[22] N. Ahmad, S. Charan, V. P. Singh. Study of Numerical Accuracy of Runge-Kutta Second, Third and Fourth Order Method / International Journal of Computer & Mathematical Sciences. 2015. Vol. 4, Issue 6. Pp. 111-118.

[23] Fomin O., Polozhaenko S., Bidyuk P., Tataryn O. et al. Improving measurements accuracy in weight-in-motion systems using dynamic neural networks. ICST-2024: Information Control Systems & Technologies. Odesa, Ukraine. CEUR Workshop Proceedings. 2024, 3790. P. 483–493. URL: https://ceur-ws.org/Vol-3790/paper42.pdf.