Automated Linearization of a System of Nonlinear Ordinary Differential Equations

Aigerim Mazakova, Sholpan Jomartova, Waldemar Wójcik, Talgat Mazakov, and Gulzat Ziyatbekova

Abstract—This paper investigates the possibility of automatically linearizing nonlinear models. Constructing a linearised model for a nonlinear system is quite labor-intensive and practically unrealistic when the dimension is greater than 3. Therefore, it is important to automate the process of linearisation of the original nonlinear model. Based on the application of computer algebra, a constructive algorithm for the linearisation of a system of non-linear ordinary differential equations was developed. A software was developed on MatLab. The effectiveness of the proposed algorithm has been demonstrated on applied problems: an unmanned aerial vehicle dynamics model and a two-link robot model. The obtained linearized models were then used to test the stability of the original models. In order to account for possible inaccuracies in the measurements of the technical parameters of the model, an interval linearized model is adopted. For such a model, the procedure for constructing the corresponding interval characteristic polynomial and the corresponding Hurwitz matrix is automated. On the basis of the analysis of the properties of the main minors of the Hurwitz matrix, the stability of the studied system was analyzed.

Keywords—Ordinary differential equation; Computer algebra; Stability; Controllability; MatLab

I. INTRODUCTION

THE study of many robotic, economic systems leads to the construction of mathematical models described by nonlinear ordinary differential equations. Until now, there have been no universal approaches for nonlinear systems. The study of such mathematical models requires mandatory consideration of the type of nonlinearities [1-4].

In general, a nonlinear model can be represented as a system of ordinary differential equations [5-6]:

$$\frac{dq}{dt} = f(q, \theta, t) + B(t)u \qquad (1)$$

where θ is a vector of parameters of dimension l, $q(t)$ is a vector of model variables of dimension n, $u(t)$ are model inputs that specify control methods, time $t \in [0, T]$. T – set. It is assumed that the vector function $f(q, \theta, t)$ is defined and continuous together with its partial derivatives against q.

The initial conditions are added to the system of equations (1):

$$q(0) = q_0. \qquad (2)$$

There are the following restrictions on control:

$$u(t) \in U = \left\{ \begin{array}{c} u(t): u_i(t) \in C[[0,T]; \\ -L_i \le u_i(t) \le L_i, i = \overline{1,m}, t \in [0,T] \end{array} \right\}. \quad (3)$$

When studying the following properties of the mathematical model (1)-(3) as controllability and stability, the linearization method is often used [7-11].

The nonlinear model (1) corresponds to the linearized system of differential equations:

$$\dot{q} = A(\theta, t)q + B(t)u, \qquad (4)$$

$$q(0) = q_0 \qquad (5)$$

where $A(\theta, t) - n * n$ – matrix whose elements depend on the vector of parameters and time $t \in [0, T]$.

The matrix $A(\theta, t)$ is determined from (1) as follows:

$$A(\theta, t) = \frac{\partial f(q^s, \theta, t)}{\partial q} \qquad (6)$$

In (6), the vector function $q^s(t)$ (of dimension n) $t \in [0, T]$, is assumed to be given based on the requirements for the problem.

For the controllability problem $q^s(t)$ can be given as follows:

$$q^s(t) = const = q_T, t \in [0, T]. \qquad (7)$$

where q_T is the desired final state of a system (1).

Controllability problem: is there a control $u(t)$ that satisfies condition (3) and transfers system (4) from the initial state (5) to the desired final state (7) in a given time T.

Definition [12]. System (1) is called locally null-controlled on the interval $t \in [0, T]$ if there exists a δ-neighbourhood of the point q_T such that for any vector $q_0 \in R^n$: $\|q_0 - q_T\| < \delta$ there exists such admissible control $u_0(t) \in U$, that there is a solution (1) that satisfies the conditions

$$q(0) = q_0 \text{ and } q(T) = q_T.$$

For the stability problem $q^s(t)$ can be given in the form (7), where q_T is the equilibrium position of the original system (1) investigated for its stability property.

Consider the following nonlinear system

$$\frac{dq}{dt} = f(q), \qquad (8)$$

where the function

$$f(q_T) \equiv 0, A = \frac{\partial f(q_T)}{\partial q}$$

Computational Technologies CS MES RK, Al-Farabi Kazakh National University, Kazakhstan (e-mail: tmazakov@mail.ru, ziyatbekova@mail.ru).

W. Wójcik is with Lublin Technical University, Poland (e-mail: waldemar.wojcik@pollub.pl).

S. Jomartova and A. Mazakova are with Al-Farabi Kazakh National University, Kazakhstan (e-mail: jomartova@mail.ru, aigerym97@mail.ru).

Lyapunov's theorem on stability in the first approximation states that if all eigenvalues of the matrix A have negative real parts, then the equilibrium position q_T of the system (8) is exponentially stable [13]. Manual construction of the matrix $A(\theta, t)$ causes great difficulties already with dimension n greater than 3, and in the case of dependence of the vector function $f(q, \theta, t)$ on the parameters θ with dimension n greater than 3, it is practically impossible.

II. METHODOLOGY MAIN RESULTS

As was noted in the introduction, the process of constructing a linearized model (4) for the original nonlinear system of the form (1) is quite laborious and, if the dimension n is greater than 3, it is practically unrealizable. Also, when building a linearized model, there is a «human factor» that does not guarantee the correctness of the calculation of arithmetic expressions. In this regard, the task of automating the process of linearization of the original nonlinear model is relevant. To construct a linearized model (5), it is proposed to use computer algebra systems. In the proposed work, the MatLab system [14-16] was used.

The emergence in the 20th century of a new direction of mathematical science, computer algebra, made it possible to automate the process of obtaining the equations of a mathematical model of dynamic systems based on analytical computing systems (ACS) [17]. ACS first provided a basic set of procedures for processing algebraic expressions: calculation, substitution, simplification, and differentiation. Then, on their basis, more complex operations were implemented: expansion into series, canonicalization of expressions, integration in special cases. With a variety of packages, they are used in many areas – from solving problems of celestial mechanics to theoretical studies of algorithms [18,19].

III. RESULT AND DISCUSSION

Further research will be carried out on two examples a UAV and a robot.

A. Unmanned aerial vehicle (UAV) research

Unmanned aircraft are developing all over the world because of the need for light, relatively cheap aircraft with high maneuverability characteristics and capable of performing a wide range of tasks. Unmanned aerial vehicles (UAVs) are successfully used in military operations around the world, and at the same time, they are also successfully performing civilian tasks.

Consider the following mathematical model of UAV dynamics

$$\begin{cases} \dot{V} = g(n_{xa} - \sin\Theta) \\ \dot{\Theta} = g(n_{ya}\cos\gamma - \cos\Theta)/V \\ \dot{\Psi} = -gn_{ya}\sin\gamma/(V\cos\Theta) \\ \dot{x} = V\cos\Theta\cos\Psi \\ \dot{y} = V\sin\Theta \\ \dot{z} = -V\cos\Theta\sin\Psi \end{cases} \qquad (9)$$

$$n_{xa} = \frac{P\cos\alpha - X_a}{mg}, n_{ya} = \frac{P\sin\alpha + Y_a}{mg} \qquad (10)$$

Here x, y, z are the coordinates of the center of mass of the aircraft in the normal Earth coordinate system, V is the flight speed, Θ is the angle of the trajectory, Ψ is the heading angle, α is the angle of attack, γ is the roll angle, P is the engine thrust, X_a – aerodynamic drag, Y_a – aerodynamic lift, m – aircraft mass,

g – gravitational acceleration, n_{xa} – longitudinal g-load and n_{ya} – lateral g-load (in flow axes).

The overloads n_{xa}, n_{ya} and roll angle γ (10) [20] are taken as control variables in (9).

Let us introduce the following notation:

$$q = \begin{bmatrix} V \\ \Theta \\ \Psi \\ x \\ y \\ z \end{bmatrix}, q_0 = \begin{bmatrix} V_0 \\ \Theta_0 \\ \Psi_0 \\ x_0 \\ y_0 \\ z_0 \end{bmatrix}, q_1 = \begin{bmatrix} V_1 \\ \Theta_1 \\ \Psi_1 \\ x_1 \\ y_1 \\ z_1 \end{bmatrix}. \qquad (11)$$

With the introduced notation, the system of equations (9)-(10) can be represented in the form (12):

$$\frac{dq}{dt} = f(q, t) + B(t)u \qquad (12)$$

Using the program written in MatLab and presented below, we obtain the following results of system linearization (9).

```
syms x1 x2 x3 x4 x5 x6 x A B
x0 = [2 5 4 4 5 6];
x = [x1 x2 x3 x4 x5 x6];
t = [0 10];  y0 = x0;
g = 9.8; P = 2000; m = 3; alfa = 30;
gamma = 45; Xa = 0.32; Ya = 0.4;
nxa = (P*cos(alfa) - Xa)/(m*g);
nya = (P*sin(alfa) + Ya)/(m*g);
F = [g*(nxa-sin⁗(x2));
g*(nya*cos(gamma)-cos(x2))/x1;
-g*nya*sin(gamma)/(x1*cos(x2));
x1*cos(x2)*cos(x3); x1*sin(x2);
-x1*cos(x2)*sin(x3)];
for i=1: length(x0)
  for j=1: length(x0)
   A (i, j)=diff(F(i), x(j));
  end
end
B=subs (A, x, x0);
disp(vpa(A, 3));
disp(vpa(B, 3));
```

We represent the result of the program execution in the following form:

$$A(x) = \frac{\partial f(x,t)}{\partial x} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{16} \\ a_{21} & a_{22} & \dots & a_{26} \\ \dots & \dots & \dots & \dots \\ a_{61} & a_{62} & \dots & a_{66} \end{pmatrix} \qquad (13)$$

We write out only nonzero elements of the matrix A(x):
$a_{12} = -9.80 * \cos(x2)$,
$a_{21} = -1.*(-346. -9.80 * \cos(x2))/x1^2$,
$a_{22} = 9.80 * \sin(x2)/x1$,
$a_{31} = -560./x1^2/\cos(x2)$,
$a_{32} = 60./x1/\cos(x2)^2 * \sin(x2)$,
$a_{41} = \cos(x2) * \cos(x3)$,
$a_{42} = -x1 * \sin(x2) * \cos(x3)$,
$a_{43} = -x1 * \cos(x2) * \sin(x3)$,
$a_{51} = \sin(x2), a_{52} = x1 * \cos(x2)$,
$a_{61} = -\cos(x2) * \sin(x3)$,
$a_{62} = x1 * \sin(x2) * \sin(x3)$,
$a_{63} = -x1 * \cos(x2) * \cos(x3)$.

Let us calculate the value of the matrix $A(x)$ at the point $x_0 = [2\ 5\ 4\ 4\ 5\ 6]$:

$$A(x_0) = \begin{pmatrix} 0 & -2.78 & 0 & 0 & 0 & 0 \\ 87.2 & -4.70 & 0 & 0 & 0 & 0 \\ -493. & -3330. & 0 & 0 & 0 & 0 \\ -0.186 & -1.25 & 0.430 & 0 & 0 & 0 \\ -0.959 & 0.568 & 0 & 0 & 0 & 0 \\ 0.215 & 1.45 & 0.372 & 0 & 0 & 0 \end{pmatrix} \quad (14)$$

As a result of the program execution, (13) the analytical form of the matrix A and its value at the point $x_0 = [2\ 5\ 4\ 4\ 5\ 6]$ were obtained.

In the future, matrix A, presented in the form (14), can be used to analyze the stability or controllability of the UAV model at a given point.

B. Two-link robot research

A mathematical model of a flat two-link articulated manipulator, shown in figure 1 [21], is considered.
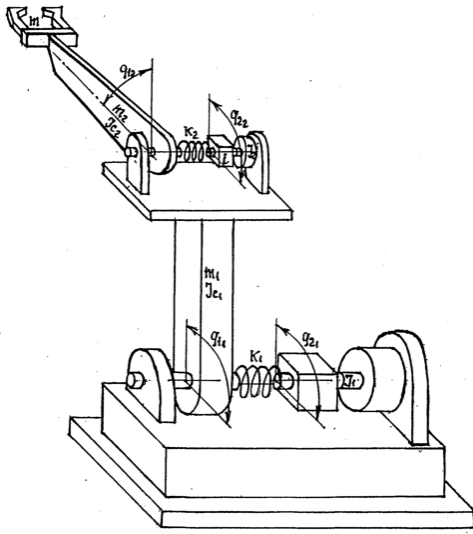

Fig. 1. View of the two-link articulated manipulator "PUMA 550 / 560"

When compiling the equations of motion, the following system parameters were taken into account: l_1 and l_2 – the lengths of the links, l_{c_1} and l_{c_2} – the coordinates of the centers of inertia, m_1 and m_2 – the masses of the links, m – the mass of the tong with load, J_{c_1} and J_{c_2} – the moments of inertia of the links relative to the axes passing through the centers of mass of the links, J_1 and J_2 – the moments of inertia of the rotors of the engines, reduced to the axes of the hinges (it is assumed that the first engine is installed on the base, and the total mass of the second engine installed on the first link, taken into account in the corresponding parameters of the first link), i is the gear ratio of the second engine [22].

The equations of motion are described by the following model

$$\begin{cases} D(q_1)\ddot{q}_1 + B(q_1)\ddot{q}_2 + C_1(q_1,\dot{q}_1,\dot{q}_2)\dot{q}_1 + \\ \quad + C_2(q_1,\dot{q}_1)\dot{q}_2 + K(q_1 - q_2) + g(q_1) = 0 \\ J\ddot{q}_2 + B^T(q_1)\ddot{q}_1 + C_3(q_1,\dot{q}_1)\dot{q}_1 + K(q_2 - q_1) = u \end{cases} \quad (15)$$

where $q_1 = \begin{bmatrix} q_{1_1} \\ q_{1_2} \end{bmatrix}$, $q_2 = \begin{bmatrix} q_{2_1} \\ q_{2_2} \end{bmatrix}$ are vectors of rotation angles of links and rotors, respectively;

$$D(q_1) = \begin{bmatrix} a_1 + 2a_2\cos(q_{1_2}) & a_3 + a_2\cos(q_{1_2}) \\ a_3 + a_2\cos(q_{1_2}) & a_3 \end{bmatrix};$$

$$B(q_1) = \begin{bmatrix} 0 & \frac{J_2}{i} \\ 0 & 0 \end{bmatrix};$$

$$\begin{bmatrix} C_1(q_1,\dot{q}_1,\dot{q}_2) & C_2(q_1,\dot{q}_1) \\ C_3(q_1,\dot{q}_1) & 0 \end{bmatrix}\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} -a_2\dot{q}_{1_2}(2\dot{q}_{1_1} + \dot{q}_{1_2})\sin(q_{1_2}) \\ a_2\dot{q}_{1_1}\sin(q_{1_2}) \end{bmatrix};$$
$$(16)$$

$$g(q_1) = \begin{bmatrix} (ml_2 + m_2l_{c_2})g\cos(q_{1_1} + q_{1_2}) + (m_1l_{c_1} + m_2l_1 + ml_1)g\cos(q_{1_1}) \\ (ml_2 + m_2l_{c_2})g\cos(q_{1_1} + q_{1_2}) \end{bmatrix};$$

$$J = \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix};$$
$$K = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix};$$

$a_1 = m(l_1^2 + l_2^2) + m_1l_{c_1}^2 + m_2(l_1^2 + l_{c_2}^2) + J_{c_1} + J_{c_2} + \frac{J_2}{i^2};$
$a_2 = l_1(ml_2 + m_2l_{c_2});$
$a_3 = ml_2^2 + m_2l_{c_2} + J_{c_2}.$

Let us introduce the notation:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} q_{1_1} \\ q_{1_2} \\ \dot{q}_{1_1} \\ \dot{q}_{1_2} \\ q_{2_1} \\ q_{2_2} \\ \dot{q}_{2_1} \\ \dot{q}_{2_2} \end{pmatrix}, \quad (17)$$

With the introduced notations (17) in works [23, 24] the process of analytical derivation of equation (15) in the form of the following system of ordinary differential equations is automated

$$\frac{dx}{dt} = f(x, u, t) \quad (18)$$

where $f_1 = x_3, f_2 = x_4, f_5 = x_7, f_6 = x_8,$

The functions f_3, f_4, f_7, f_8 depend on the state variables of the system x and linearly on the control u. Below is a view of just one of these functions.

$f_3 = $-12300/(-202581-20100*cos(x2) +22445*cos(x2) ^2) *(67/20*x4*(2*x3+x4) *sin(x2)-25000*x1-67/10*cos(x1+x2) -55/2*cos(x1)) +100*(63+67*cos(x2))/(-202581- 20100* cos(x2) +22445*cos(x2) ^2) *(67/20*x4* (2*x3+x4) *sin(x2)-6500*x6 +67/10*cos(x1+x2)) +12300/(-2к02581-0100* cos(x2) +22445*cos(x2)^2)*(-67/20*x3*sin(x2)-6500*x2+u),

With the help of a program written in MatLab and the initial fragment of which is presented below, we obtain the following results of system linearization (18).

```
syms x1 x2 x3 x4 x5 x6 x7 x8 x A b c
x0 = [0 0 4 2 0 0 1 1];
x = [x1 x2 x3 x4 x5 x6 x7 x8];
t = [0 10];
y0 = x0; u = 0;
F = [...x3;...x4;...-369./(-6070.-603.*cos(x2)+673.*cos(x2)^2)*
(3.35*x4* (2.*x3+x4)*sin(x2)-.25000.*x1-6.70*cos(x1+x2)-
27.5*cos(x1)) +(189.+201.*cos(x2))/(-6070.-603.*cos(x2)+
673.*cos(x2)^2) *(3.35*x4*(2.*x3+x4)*sin(x2)-6500.*x6+
6.70*cos(x1+x2))+369./(-6070.-603.*cos(x2)+673.*cos(x2)^2)*
(-3.35*x3*sin(x2)-6500.*x2+u);
```

...(369.+201.*cos(x2))/(-6070.-603.*cos(x2)+
673.*cos(x2)^2)*(3.35*x4*(2.*x3+x4) *sin(x2)-25000.* x1-
6.70*cos(x1+x2) -27.5*cos(x1))-(1180.+402.*cos(x2))/(-6070.-
603.*cos(x2)+673.*cos(x2)^2)*(3.35*x4*(2.*x3+x4)*sin(x2)-
6500.*x6+6.70*cos(x1+x2))-1.*(369.+201.*cos(x2))/(-6070.-
603.*cos(x2)+673.*cos(x2)^2)*(-3.35*x3*sin(x2)-6500.*x2+u);
x7; x8; 168*x3*sin(x2)-1250.*x5+.05*u;
...-1.*(369.+201.*cos(x2))/(-6070.-603.*cos(x2)+673.*
cos(x2)^2)*(3.35*x4*(2.*x3+x4)*sin(x2)-25000.*x1-6.70*
cos(x1+x2)-27.5*cos(x1)) +(1180.+402.*cos(x2))/(-6070.-
603.*cos(x2)+673.*cos(x2)^2)*(3.35*x4*(2.*x3+x4)* sin(x2)-
6500.*x6+6.70*cos(x1+x2))+(-1650.+224.*cos(x2)^2)/(-6070.-
603.*cos(x2)+673.* cos(x2)^2)*(-3.35*x3*sin(x2)-
6500.*x2+u)];

Let us represent the result of the program execution in the following form:

$$A(x) = \frac{\partial f(x,t)}{\partial x} = \begin{pmatrix} a_{11} & a_{12} & ... & a_{18} \\ a_{21} & a_{22} & ... & a_{28} \\ ... & ... & ... & ... \\ a_{81} & a_{82} & ... & a_{68} \end{pmatrix} \qquad (19)$$

We write out only nonzero elements of the matrix $A(x)$:

$a_{13} = 1, \qquad a_{24} = 1,$

$a_{31} = (369.0 * (6.7 * \sin(x1 + x2) + 27.5 * \sin(x1) - 0.25))/(603.0 * \cos(x2) - 673.0 * \cos(x2)^2 + 6070.0) + (6.7 * \sin(x1 + x2) * (201.0 * \cos(x2) + 189.0))/(603.0 * \cos(x2) - 673.0 * \cos(x2)^2 + 6070.0),$

Let us calculate the value of the matrix $A(x)$ at the point x0 = [0 0 4 2 0 0 1 1];

$A(x_0) =$

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -0.015375 & 400.34 & 0 & 0 & 0 & 422.5 & 0 & 0 \\ 2375.0 & -607.47 & 0 & 0 & 0 & ,-1713.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 672.0 & 0 & 0 & -1250.0 & 0 & 0 & 0 \\ -2375.0 & -1559.3 & 0 & 0 & 0 & 1713.8 & 0 & 0 \end{pmatrix}$$

(20)

In the future, matrix *A*, presented in the form (20), can be used to analyze the stability or controllability of a two-link robot model at a given point.

C. Stability of the linearized system

Next, consider the problem of stability of the linearized system described by the following ordinary differential equations.

$$\dot{x} = \mathbf{A}x. \qquad (21)$$

Here the elements of matrix *A* are determined by the technical characteristics of the model under study.

Due to possible inaccuracies in measurements, wear and temperature conditions during operation, it is assumed that the elements of the matrix *A* are given as intervals.

The following definition is given in [10].

A system (21) with interval matrix *A*, the elements of which have a normal distribution, is called interval asymptotically stable according to Lyapunov, if for any interval solution

$$x(t) = [\overline{x}(t) - \varepsilon_x(t), \overline{x}(t) + \varepsilon_x(t)] = \{\overline{x}(t), \varepsilon_x(t)\}, \qquad t \in [0, \infty)$$

the statement is true:

- for any ε>0 and $t_0 \in [0, \infty)$ there exists $\delta = \delta(\varepsilon, t_0)$ such that for all solutions $x = x(t)$, satisfying the condition $\|x(t_0)\| < \delta$, the inequality $\|x(t)\| < \varepsilon$, at t, is valid $t \in [t_0, \infty)$;

- for any $t_0 \in [0, \infty)$ there exists $\lambda = \lambda(t_0)$ such that all solutions $x = x(t)$, satisfying the condition $\|x(t_0)\| < \lambda$, has the property: $\lim_{t->\infty} \|\overline{x}(t)\| = 0$

To determine the stability of the interval matrix on the basis of the system of analytical calculations, the procedure of constructing the interval characteristic polynomial is automated.

$$\psi(\lambda) = \det(\lambda E - \mathbf{A}) = b_n \lambda^n + b_{n-1} \lambda^{n-1} + ... + b_0. \qquad (22)$$

The following procedures are included in the developed package:

1) procedure XarPolFadInt(a:MatIntr; n:integer; var b:VecIntr) – procedure for calculating the coefficients of the characteristic polynomial of the interval matrix a by the Fadeev method; n is the dimensionality of the matrix; the result in the interval vector c;

2) procedure XarPolLevInt(a:MatIntr; n:integer; var b:VecIntr) – the procedure for calculating the coefficients of the characteristic polynomial of the interval matrix a by the Leverier method; n – matrix dimension; the result in the interval vector c;

3) procedure XarPolDanInt(a:MatIntr; n:integer; var b:VecIntr) – procedure for calculating the coefficients of the characteristic polynomial of the interval matrix a by the Danilevsky method; n – the dimensionality of the matrix; the result in the interval vector c;

4) procedure XarPolKriInt(a:MatIntr; n:integer; var b:VecIntr) – procedure for calculating the coefficients of the characteristic polynomial of the interval matrix a by the Krylov method; n – matrix dimension; the result in the interval vector c [23, 24].

To determine the stability of the linearized system (21) based on the coefficients of the interval characteristic polynomial (22), the Hurwitz matrix is constructed.

$$G = \begin{bmatrix} b_1 & b_0 & 0 & 0 & ... & 0 \\ b_3 & b_2 & b_1 & b_0 & ... & 0 \\ ... & ... & ... & ... & ... & ... \\ b_{2n-1} & b_{2n-2} & b_{2n-3} & b_{2n-4} & ... & b_n \end{bmatrix}. \qquad (23)$$

Let's introduce the notation $\Delta_i, i = \overline{1, n}$ – are the principal minors of the matrix G.

Hurwitz interval stability criterion [10]: for $\text{Re}\,\lambda_j(\mathbf{A}) < 0, j = \overline{1, n}$ is necessary and sufficient for the main diagonal minors $\Delta_i, i = \overline{1, n}$ of the matrix G to satisfy the condition $\Delta_i \in (0, \infty), i = \overline{1, n}$.

The following procedures have been developed for automated testing of interval stability:

1) *procedure RausGurvizInt(a:VecIntr; n:integer; var b:VecIntr)* – procedure for calculating the determinants

of the main minors of the interval Raus-Hurwitz matrix constructed on the basis of the coefficients of the characteristic interval polynomial a; n – matrix dimension; the result in the interval vector b.

2) *procedure GurvizInt(a:VecIntr; n:integer; var Igr:Integer)* – procedure for determining the stability of a characteristic interval polynomial based on the Hurwitz criterion; n – polynomial degree; result Igr =0 in the case of stability and Igr =1 in the case of instability.

3) *procedure RausInt (a:VecIntr; n:integer; var Igr:Integer);*– procedure for determining the stability of a characteristic interval polynomial based on the Rauss criterion; n – polynomial degree; result Igr=0 in the case of stability and Igr > 0 in the case of instability.

As additional service procedures and functions have been developed:

1) *function VgrPolInt(a:VecIntr; n:Integer):real*; – function for determining the upper bound of the moduli of a polynomial with interval real coefficients (n – polynomial degree).

2) *function VgrPolPolInt(a:VecIntr;n: Integer):real*; – function for determining the upper bound of positive real roots of a polynomial with interval real coefficients (n – polynomial degree).

3) *procedure KrGerhInt(a:MatIntr; n:Integer; var Cmin,Cmax:real*); – procedure for localizing the eigenvalues of the interval matrix a by the Gershgorin method}; n – matrix dimensionality; the result in variables of real type Cmin and Cmax;

4) *function PolSchInt(a:VecIntr; n:Integer):integer*; – determining the positivity of an interval polynomial (whose coefficients are given in the interval vector a; n – polynomial degree) by the D. Shilyak method.

5) *function PolPolInt(a:VecIntr; n:Integer):integer*; – for determination of positivity of an interval polynomial (whose coefficients are given in the interval vector a ; n – polynomial degree) by S. A. Aysagaliev's method.

6) *function RauthMInt(a, b:VecIntr; n:Integer):integer*; – determining the number of sign changes in the Rauss matrix constructed by interval polynomials (whose coefficients are given in the interval vector a and its derivative b; n – polynomial degree); the result is the number of sign changes [23, 24].

IV. CONCLUSION

On the basis of MatLab analytical computing systems, the linearization procedure for a nonlinear system described by ordinary differential equations is automated.

The effectiveness of the proposed procedure is demonstrated by two examples.

For a mathematical model of the dynamics of an unmanned aerial vehicle, described by a system of ordinary differential equations of the 6th order, a linearized system is automatically constructed.

Two procedures have been developed for the mathematical model of a two-link robot. The first procedure allows to normalize the mathematical model obtained on the basis of the Lagrange equations of the second kind. The second procedure uses the normalized mathematical model to build its linearized copy.

To investigate the stability of linearized models, we have automated a procedure for constructing interval characteristic polynomials, the Hurwitz matrix, and analyze its stability properties.

The use of interval analysis in solving the problem of stability of the dynamics of mechanical systems allows us to obtain a criterion of guaranteed stability.

In the future, the results obtained in the article can be used to study the dynamics of various socio-economic, technical systems.

REFERENCES

[1] O.I. Borisov, "Methods of control of robotics applications" St. Petersburg: ITMO University, 2016.

[2] A.G. Bulgakov, V.A. Vorobyov, "Industrial robots. Kinematics, dynamics, control and management", Moscow: Solon-Press, 2011.

[3] V.A. Kolemaev, "Mathematical Economics", Moscow: Unity-Dana, 2002.

[4] V.V. Khristianovsky, V.P. Shcherbina, "Economic and mathematical methods and models: theory and practice", Donetsk: DonNU, 2010.

[5] B.L. Teush, "Ordinary differential equations: A course for future engineers", Moscow: Lenand, 2022.

[6] [6] V.G. Borisov Applied Problems of the Theory of Ordinary Differential Equations. Mechanical motion. - Kemerovo: Kemerovo State University, 2015. – 130 p.

[7] F. Boarotto, M. Sigalott, "Dwell-time control sets and applications to the stability analysis of linear switched systems," Journal of Differential Equations, vol. 268, no. 4, pp. 1345-1378, 2020. https://doi.org/10.48550/arXiv.1902.03757

[8] V. Ayala, A. Da Silva, P. Jouan, G. Zsigmond, "Control sets of linear systems on semi-simple Lie groups," Journal of Differential Equations, vol. 269. no. 1, pp. 449-466. 2020. https://doi.org/10.1016/j.jde.2019.12.010

[9] W. Zhang, Q-L. Han, Y. Tang, Y. Liu, "Sampled-data control for a class of linear time-varying systems," Automatica, vol. 103, pp. 126-134. 2019. https://doi.org/10.1016/j.automatica.2019.01.027

[10] T. Mazakov, W. Wójcik, Sh. Jomartova, N. Karymsakova, G. Ziyatbekova, A. Tursynbai, "The Stability Interval of the Set of Linear System," International Journal of Electronics and Telecommunications, vol. 67, no. 2, pp. 155-161, 2021. https://doi.org/10.24425/ijet.2021.135957

[11] A. Mazakova, S. Jomartova, T. Mazakov, T. Shormanov, B. Amirkhanov, "Controllability of an unmanned aerial vehicle," IEEE 7th International Energy Conference (ENERGYCON), pp. 1-5. 2022. https://doi.org/10.1109/ENERGYCON53164.2022.9830244

[12] A.A. Shcheglova, "On controllability of nonlinear differential-algebraic equations," Proc. of International Conference "Dynamic Systems: Stability, Control, Optimization", Minsk, pp.166-168, 2008.

[13] A.M. Molchanov, "On the stability of nonlinear systems," Pushchino: Institute of Mathematical Problems of Biology RAS, 2013.

[14] V.Z. Aladyev, V.K. Boyko, E.A. Rovba, "Programming and Development of Applications in Maple", Grodno: GrSU., Tallinn: International Noosphere Academy, 2007.

[15] Y. Lazarev, "Modeling Processes and Systems in Matlab," St. Petersburg: Peter: VNV Publishing Group, 2005.

[16] V.P. Dyakonov, "Systems of computer algebra Derive," Moscow: Solon-Press, 2002.

[17] M.V. Grosheva, G.B. Efimov, V.A. Samsonov, "History of using analytical calculations in mechanics problems," Moscow: The Keldysh Institute for Applied Mechanics of the Russian Academy of Sciences, 2005.

[18] V.P. Dyakonov, "Encyclopedia of computer algebra", Moscow: DMK Press, 2010.

[19] R.N. Kvyetnyy, O.Y. Sofina, A.V. Lozun, A. Smolarz, O. Zhirnova, "Modification of fractal coding algorithm by a combination of modern technologies and parallel computations," Proc. SPIE, vol. 9816, art. no. 98161R, 2015. https://doi.org/10.1117/12.2229009

[20] A.N. Kvitko, "Methods for solving boundary value problems for controllable systems of ordinary differential equations and their

application in solving problems of controlling the motion of the center of mass of an aircraft," PhD. thesis, 2000.

[21] S.F. Burdakov, "Mathematical models and identification of robots with elastic elements," Leningrad: LSTU. 1990.

[22] L.B. Freidovich, "Stability and control of manipulating robots" PhD. thesis in Physics and Mathematics, St. Petersburg, 1999.

[23] Sh.A. Jomartova, G.N. Karymsakova, A.S. Tursynbai, "Application of analytical calculations system for the derivation of equations of dynamics of robotic systems," Bulletin of KazATK, no.2, pp. 207-213, Almaty 2020.

[24] N.T. Karymsakova, "Development of controllability criteria for dynamic systems with limited control," PhD thesis, Almaty 2022.