The method of inverse problems of dynamics for the synthesis of a system of stabilization of the movement of a dynamic object on operatively programmable trajectories

Mashkov O. A.[1], Chumakevich V. A.[2], Mamchur Yu. V.[1], Kosenko V. R.[1]

[1] *National Center for Control and Testing of Space Facilities of the State Space Agency of Ukraine,
8 Moscow Str., 01010, Kiev, Ukraine*
[2] *Lviv Polytechnic National University,
12 S. Bandera Str., 79013, Lviv, Ukraine*

(Received 1 April 2019; Revised 1 October 2019; Accepted 2 October 2019)

The article is devoted to the analysis of the possibility of applying the method of inverse problems of dynamics for the synthesis of a system of spatial stabilization of the motion of a dynamic object on an operatively programmable trajectory. The article proposes to apply the method of inverse problems of dynamics for the synthesis of a system for stabilizing the motion of a dynamic object on an operatively programmable trajectory. It is concluded that the procedure for applying the method of inverse problems of dynamics provides for the sequential execution of two procedures. The first procedure involves setting the desired trajectory of movement of a dynamic object and determining the vector of necessary control forces for the implementation of this trajectory of movement. The second procedure involves determining the control function (control deviations) to create such forces. In the development of the concepts of the algorithmic approach (inverse problems of dynamics), an analytical expression for the governing force is obtained. The proposed block diagram of the control algorithm can be used to synthesize control systems for complex dynamic objects, for example, remotely piloted aircraft.

Keywords: *control algorithm, dynamic object, control force, control function, inverse dynamics problem, operatively programmable trajectory, control system synthesis, motion stabilization.*

2010 MSC: 49J27, 90C48, 93C25 **DOI:** 10.23939/mmc2020.01.029

1. Introduction

Currently, inverse problems occupy an important place in the study of theoretical and mechanical models. It is known that a mathematically rigorous formulation of the concept of inverse problems of dynamics was given by Galiullin A. C., Petrov B. N., Popov E. P., Krutko P. D. The relevant subject has been intensively developed in the works of followers, first of all, Yemelyanov S. V., Ermoshina O. V., Zhevnina A. A., Kanatnikova A. N., Kolesnikova K. S., Krishchenka A. P., Mukhametzyanova I. A., Mukharlyamova R., Toloknova V. Moreover, starting with the work of Galiullin I. A., it became possible to investigate similar problems not only in Euclidean spaces but also on arbitrary differentiable manifolds [1–7].

Many inverse problems of dynamics are initially connected with the conditions of the program motion of aerohydrodynamic or spacecraft. These works are devoted to the selection of control functions or parameters of the apparatus, which ensure its movement along a trajectory with specified properties [2–8].

Of great importance is the problem of analytic approximation of the programmed motion and the estimation of its error, which is solved with the help of various methods of analysis of ordinary differential equations. One of the most effective procedures for solving such problems is the version of the harmonic balance method, described in the works of Delamotte B. and Polanda D. A very important problem is the choice of control, which ensures a steady periodic motion of a mechanical object in the phase space. In the mathematical formulation, this means the existence of a stable limit cycle for the corresponding system of differential equations [1,8,9].

Thus, obtaining conditions that ensure the movement of a dynamic object along an operatively programmable desired trajectory is relevant from the point of view of theory and practice.

Until recently, in the theory of automatic control, the tasks were mainly considered in which it was necessary to maintain the output signal (mode) of the control object at the same constant level (stabilization task), or to ensure tracking of an unknown predetermined influence (tracking task). However, due to the increasing complexity of management objects, many tasks arose in which you first need to calculate the desired law of change of the controlled process (program motion), and then build the control law that provides accurate or approximate implementation of this process in the event of possible emergency situations.

The complexity of solving this problem in practice is determined by the following three factors [1, 2, 9–14]:

1. The real movement at the initial moment may differ significantly from the program. This is due to the existing initial disturbances, as well as the occurrence of an emergency situation.
2. The real values of the parameters of the control object due to parametric perturbations differ from their estimates, which are used in the construction of the equation.
3. There are permanent disturbances.

Therefore, if the control system is not able to counter the influence of the listed internal factors (on-board equipment failures) and external factors (disturbances), then the real movement may differ significantly from the software one. At the same time, the goal of control (movement along the program trajectory) will obviously not be achieved. Thus, the problem arises of building a software motion control system, which ensures the implementation of a program trajectory with a given accuracy with the presence of various kinds of disturbances.

The application of classical control methods to the construction of control systems for programmed movement encounters certain difficulties. These difficulties are associated with the uncertainty of the properties of the controlled object and the conditions of operation of the control object [6,8,9,11–14].

Studies show that the application of control laws based on solving the inverse problem of dynamics allows us to reduce the level of random components in state coordinates. Such control laws are effective, for example, when building control systems for unmanned aerial vehicles [1,6,8,15].

2. Basic relations

Consider a managed object whose state at time t is described by a vector function $X(t) = [X_1(t), \ldots, X_n(t)]$. Components $X_i(t)$, $i = \overline{1,n}$ are phase coordinates in n-dimensional Euclidean space $\mathbb{R}^n$.

The dynamics of the motion of a controlled object over time is described by a matrix differential equation

$$\begin{aligned} \dot{X}(t) &= F[X(t), U(t), Z(t)] + P(t); \\ X(t_0) &= X_0, \quad t \in [t_0, t_T], \end{aligned} \tag{1}$$

where $U(t)$ is m-dimensional vector of control; $Z(t)$ is r-dimensional vector of object parameters measured in flig

$$\Xi(t) = [\xi_1(t), \xi_2(t), \ldots, \xi_r(t)],$$

$P(t)$ is n-dimensional vector of perturbations; X_0 is initial state; $T = t_T - t_0$ is time of movement of the object.

There are restrictions on the management and state of the object

$$X(t) \in Q_X; \quad U(t) \in Q_U.$$

On the time interval $[t_0, t_T]$, the control function $U(t)$ is piecewise continuous and can have discontinuities of the first kind at isolated points.

We assume that the value of the parameter ξ_i is unknown. External perturbations $P(t)$ and their probabilistic properties are not known. At the same time $P(t) \in Q_p$, with all $t \in [t_0, t_T]$, Q_p is defined physical set.

The purpose of management is to ensure the movement of an object along a program phase trajectory of movement by means of an appropriate choice $U(t) \in Q_u$.

Under the program of the movement of an object we understand the desired law of a state change over time $X(t)$. If this law is a solution of equation (1) with some admissible $U(t) \in Q_u$, then we will call it program motion and denote $X_n(t)$, $t \in [t_0, t_T]$. An admissible control that corresponds to $X_n(t)$ is called programmed control $U_n(t)$ [1, 6, 8, 9].

This task is formulated by Barbashin E. A. [1, 6, 9] and called the task of the implementation of a given trajectory. Software control $U_n(t)$ and the generated by it $X_p(t)$ will be called optimal and denoted accordingly $U_n^0(t)$, $X_n^0(t)$ if they deliver at least the specified quality functionality

$$J\left[U_n^0(\cdot), X_n^0(\cdot)\right] = \min J\left[U_n(\cdot), X_n(\cdot)\right].$$

The paper discusses the method of synthesis of program control systems based on the ideas of the method of inverse problems of dynamics for multidimensional systems [1, 16–20]. The implementation time of the algorithm is up to 30 minutes.

However, it should be noted the difficulties of practical application of existing classical methods. So, for example, the problem of analytic design of regulators for a fifth-order linear system, solved on the basis of the matrix Riccati equation, takes up to several minutes time (depending on the degree of integration and the parameters of the object model). Therefore, this approach can not be applied in real-time systems, such as unmanned aerial vehicles.

In the direct formulation, the task of synthesizing algorithms for software control systems is reduced to defining the laws for the formation of control functions U_n^0 (movement of system control mechanisms, steering system deviations), subject to some indicators that indirectly characterize the quality of the control process J (Fig. 1).

In the reverse formulation (Fig. 2), the definition problem U_n^0 is considered in two stages. At the first stage, the controlling forces F are determined, which give the system the necessary movement. These forces can correspond to real physical forces and moments that need to be applied to the control object to ensure optimal programmatic movement $X_n^0(t)$. At the second stage, questions of the practical implementation of the necessary forces are considered and the control functions are determined $U_n^0(t)$.

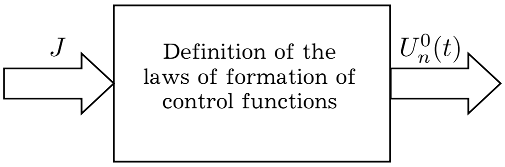

Fig. 1. The direct task of synthesis.

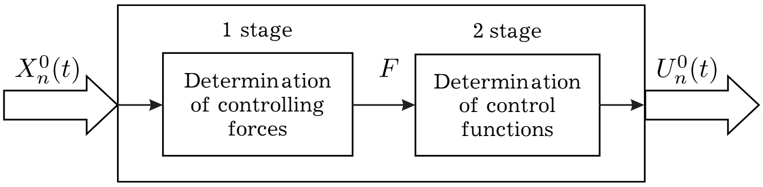
$X_n^0(t)$
1 stage
Determination of controlling forces
F
2 stage
Determination of control functions
$U_n^0(t)$

Fig. 2. The inverse problem of synthesis.

It is significant that such a formulation of the problem opens up new possibilities for building control systems based on structural schemes that are fundamentally different from the usual ones [16–20]. This is especially evident in those cases when it is necessary to simultaneously control all or several phase coordinates of an n-dimensional object.

3. Formalization of the task of implementing a given motion path of a dynamic object

Consider the features of the application of the inverse problem method for a multidimensional object, which is described by a matrix equation in operator form [1, 17–19]

$$A(p)X = B(p)U, \tag{2}$$

where

$$A(p) = \{a_{ij}(p)\}, \quad i,j = \overline{1,n}, \tag{3}$$

$$X^{\mathrm{T}} = [X_1 \dots X_n], \tag{4}$$

$$B(p) = \{b_{ij}(p)\}, \quad i = \overline{1,n}, \quad j = \overline{1,r}, \tag{5}$$

$$U^{\mathrm{T}} = [U_1 \dots U_r]. \tag{6}$$

It is known that such a control can be represented in the form of a Cauchy. We construct a control algorithm in which the object moves from a point $X(0)$ to the beginning of the phase coordinates along a trajectory

$$X_n(t) = \begin{bmatrix} C_{11}e^{\lambda_1 t} + \ldots + C_{1k}e^{\lambda_k t} \\ C_{21}e^{\lambda_1 t} + \ldots + C_{2k}e^{\lambda_k t} \\ \dotfill \\ C_{n1}e^{\lambda_1 t} + \ldots + C_{nk}e^{\lambda_k t} \end{bmatrix}, \tag{7}$$

or in matrix form

$$X_n(t) = Ce^{\Lambda t}, \tag{8}$$

where

$$C = \{C_{ij}\}, \quad i = \overline{1,n}, \quad i = \overline{1,k}, \tag{9}$$

$$\left[e^{\Lambda t}\right]^{\mathrm{T}} = \left[e^{\lambda_1 t}e^{\lambda_2 t}\dots e^{\lambda_k t}\right], \tag{10}$$

where λ_l, $l = \overline{1,k}$ are various real or complex conjugate numbers that satisfy the condition $\operatorname{Re}\lambda_1 < 0$.

The constant coefficients C_{ij}, $i = \overline{1,n}$, $i = \overline{1,k}$ are uniquely determined by the initial values of the phase coordinates and their $(k-1)$ derivative.

Choosing the appropriate values of the parameters C_{ij}, $i = \overline{1,n}$, λ_l, $l = \overline{1,k}$, you can get a lot of different software movements. This choice can be used to obtain those program motions (2) that satisfy various constraints (for example, a constructive constraint or a constraint on traffic safety, angle of list, angle of attack, angle of stall)

$$X(0) = \begin{bmatrix} C_{11} + C_{12} + \ldots + C_{1k} \\ C_{21} + C_{22} + \ldots + C_{2k} \\ \dotfill \\ C_{n1} + C_{n2} + \ldots + C_{nk} \end{bmatrix},$$

$$\dot{X}(0) = \begin{bmatrix} C_{11}\lambda_1 + C_{12}\lambda_2 + \ldots + C_{1k}\lambda_k \\ C_{21}\lambda_1 + C_{22}\lambda_2 + \ldots + C_{2k}\lambda_k \\ \dotfill \\ C_{n1}\lambda_1 + C_{n2}\lambda_2 + \ldots + C_{nk}\lambda_k \end{bmatrix},$$

$$X^{(k-1)}(0) = \begin{bmatrix} C_{11}\lambda_1^{k-1} + C_{12}\lambda_2^{k-1} + \ldots + C_{1k}\lambda_k^{k-1} \\ C_{21}\lambda_1^{k-1} + C_{22}\lambda_2^{k-1} + \ldots + C_{2k}\lambda_k^{k-1} \\ \dotfill \\ C_{n1}\lambda_1^{k-1} + C_{n2}\lambda_2^{k-1} + \ldots + C_{nk}\lambda_k^{k-1} \end{bmatrix},$$

The controlling force $f_n[t]$ that realizes the trajectory is denoted $X_n(t)$. We will look for such a trajectory in the form

$$f_n[t] = A(p)\,X_n(t). \tag{11}$$

To obtain the program law of change in the control force, we substitute the expression (7) into (11)

$$f_n(t)=\begin{bmatrix} a_{11}(p)(c_{11}e^{\lambda_1 t}+\ldots+c_{1k}e^{\lambda_k t})+\ldots+a_{1n}(p)(c_{11}e^{\lambda_1 t}+\ldots+c_{1k}e^{\lambda_k t}) \\ a_{21}(p)(c_{11}e^{\lambda_1 t}+\ldots+c_{1k}e^{\lambda_k t})+\ldots+a_{2n}(p)(c_{11}e^{\lambda_1 t}+\ldots+c_{1k}e^{\lambda_k t}) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}(p)(c_{11}e^{\lambda_1 t}+\ldots+c_{1k}e^{\lambda_k t})+\ldots+a_{nn}(p)(c_{11}e^{\lambda_1 t}+\ldots+c_{1k}e^{\lambda_k t}) \end{bmatrix}. \qquad (12)$$

Believe that [1, 2, 9, 15, 16]

$$A(p)=\{a_{ij}(p)\}=\left\{m_{ij}p^2+r_{ij}p+h_{ij}\right\}.$$

Substituting this expression into expression (12), we can get the following expression in matrix form for the control force

$$f_n[t]=(MC\Lambda_d^2+RC\Lambda_d+HC)e^{\lambda t}, \qquad (13)$$

where

$$M=\{m_{ij}\}, \quad i,j=\overline{1,n}; \qquad C=\{c_{ij}\}, \quad i=\overline{1,n}, \quad j=\overline{1,k};$$

$$\Lambda_d=\operatorname{diag}\{\lambda_i\}, \quad i=\overline{1,k}; \qquad R=\{r_{ij}\}, \quad i,j=\overline{1,n}; \qquad \left[e^{\lambda t}\right]^{\mathrm{T}}=\left\{e^{\lambda_i t}\right\}, \quad i=\overline{1,k}.$$

To start the trajectory $X_n(t)$ at

$$X_n(0)=\begin{bmatrix} X_{10} & \dot X_{10} & \ddot X_{10} & \ldots & X_{10}^{(k-1)} \\ X_{20} & \dot X_{20} & \ddot X_{20} & \ldots & X_{20}^{(k-1)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ X_{n0} & \dot X_{n0} & \ddot X_{n0} & \ldots & X_{n0}^{(k-1)} \end{bmatrix},$$

elements of the matrix $C=\{c_{ij}\}$, $i=\overline{1,n}$, $j=\overline{1,k}$ should be determined as a result of solving n systems of k algebraic equations

$$\begin{cases} C_{11}+C_{12}+\ldots+C_{1k}=X_{10}, \\ C_{11}\lambda_1+C_{12}\lambda_2+\ldots+C_{1k}\lambda_k=\dot X_{10}, \\ \cdots\cdots\cdots\cdots\cdots\cdots, \\ C_{11}\lambda_1^{(k-1)}+C_{12}\lambda_2^{(k-1)}+\ldots+C_{1k}\lambda_k^{(k-1)}=X_{10}^{(k-1)}, \end{cases}$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots,$$
$$\begin{cases} C_{n1}+C_{n2}+\ldots+C_{nk}=X_{n0}, \\ C_{n1}\lambda_1+C_{n2}\lambda_2+\ldots+C_{nk}\lambda_k=\dot X_{n0}, \\ \cdots\cdots\cdots\cdots\cdots\cdots, \\ C_{n1}\lambda_1^{(k-1)}+C_{n2}\lambda_2^{(k-1)}+\ldots+C_{nk}\lambda_k^{(k-1)}=X_{n0}^{(k-1)}. \end{cases}$$

Or in matrix form

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \ldots & \lambda_k \\ \ldots & \ldots & \ldots & \ldots \\ \lambda_1^{k-1} & \lambda_2^{k-1} & \ldots & \lambda_k^{k-1} \end{bmatrix} \times \begin{bmatrix} C_{11} \\ C_{12} \\ \ldots \\ C_{1k} \end{bmatrix} = \begin{bmatrix} X_{10} \\ \dot X_{10} \\ \ldots \\ X_{10}^{(k-1)} \end{bmatrix},$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots,$$
$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \ldots & \lambda_k \\ \ldots & \ldots & \ldots & \ldots \\ \lambda_1^{k-1} & \lambda_2^{k-1} & \ldots & \lambda_k^{k-1} \end{bmatrix} \times \begin{bmatrix} C_{n1} \\ C_{n2} \\ \ldots \\ C_{nk} \end{bmatrix} = \begin{bmatrix} X_{n0} \\ \dot X_{n0} \\ \ldots \\ X_{n0}^{(k-1)} \end{bmatrix}.$$

The solution of these matrix equations is

$$\begin{aligned}\left[C_{11}\ldots C_{1k}\right]^{\mathrm{T}} &= \Phi^{-1}\left[X_{10}\,\dot{X}_{10}\ldots X_{10}^{(k-1)}\right]^{\mathrm{T}},\\ \left[C_{n1}\ldots C_{nk}\right]^{\mathrm{T}} &= \Phi^{-1}\left[X_{n0}\,\dot{X}_{n0}\ldots X_{n0}^{(k-1)}\right]^{\mathrm{T}}\end{aligned} \tag{14}$$

where

$$\Phi = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_1 & \lambda_2 & \ldots & \lambda_k \\ \ldots & \ldots & \ldots & \ldots \\ \lambda_1^{k-1} & \lambda_2^{k-1} & \ldots & \lambda_k^{k-1} \end{bmatrix} \tag{15}$$

Since the values λ_i, $i = \overline{1,k}$ chosen are different, there is an inverse matrix Φ^{-1}.

Substituting expression (15) into expression (14), we get

$$C = \begin{bmatrix} X_{10} & \dot{X}_{10} & \ddot{X}_{10} & \ldots & X_{10}^{(k-1)} \\ X_{20} & \dot{X}_{20} & \ddot{X}_{20} & \ldots & X_{20}^{(k-1)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ X_{n0} & \dot{X}_{n0} & \ddot{X}_{n0} & \ldots & X_{n0}^{(k-1)} \end{bmatrix} \cdot \Phi^{-1\mathrm{T}} = X(0)\Phi^{-1\mathrm{T}} \tag{16}$$

The expression for the control force vector can be found by substituting expression (16) into expression (13)

$$f_n[t] = \left\{M\left(X(0)[\Phi^{-1}]^{\mathrm{T}}\right)\Lambda_d^2 + R\left(X(0)[\Phi^{-1}]^{\mathrm{T}}\right)\Lambda_d + H\left(X(0)[\Phi^{-1}]^{\mathrm{T}}\right)\right\}e^{\Lambda t} \tag{17}$$

Analyzing the obtained expression (17) it can be concluded that to obtain the necessary control force it is necessary in advance:

— describe the control object by defining the elements of the matrices M, R, H;
— set initial conditions for each controlled coordinate and their first $(k-1)$ derivatives;
— set the programmed motion path, by appropriate selection of elements λ_i, $i = \overline{1,k}$.

Obtaining the expression for the control force ends the first stage of solving the problem.

Let us now consider another stage associated with the implementation of the force change program f_n. The task of the practical implementation of the force f_n control algorithm is to determine the vector of control functions U_n (the magnitudes of deviations of the controls) with the help of which the force is created $f_n[X_n, U_n]$. It is obvious that the solution of this task is determined by the structure and parameters of the control object, and the information being measured.

The calculated ratios with which are calculated $f_n[X_n(t)]$ and $U_n(t)$ make up the content of the motion control algorithm.

We consider, at the same time, that the control object (1) has controllability properties, the force f_n is uniquely determined by the magnitude of the control vector $U^T = (U_1 \ldots U_r)$ [1,2,4,8–10,15,16,21]. Therefore, for each time t and its corresponding state $X(t)$, you can specify such $U_n(t)$ at which

$$f(X,\, U_n) = f_n[t]. \tag{18}$$

Mathematical expression (18) can be solved relatively U_n analytically (a special case), or algorithmically (the most general case).

Consider the second way to solve equation (18), which is of the greatest interest from the point of view of practical implementation.

Let $f(X, U)$such that

$$U_i \cdot f(X, U_i, U_j) > 0,$$

$$U_j = \mathrm{const}; \quad i = \overline{1,n}, \quad j = \overline{1,n}, \quad i = j.$$

We will seek a solution to equation (18) in the form:

$$\dot{U}_n = \rho \Delta f(t), \tag{19}$$

where

$$\rho^{\mathrm{T}} = \operatorname{diag}(\rho_1, \ldots, \rho_r),$$

$$\Delta f(t) = f_n(t) - f(X, U).$$

This approach to defining control functions is based on the results of studies conducted in [11,16–20].

The choice of algorithm (19) is determined by the properties of the function $f(X, U)$

$$\lim_{t \to \infty} U(t) = U_n(t).$$

Structurally, the algorithm (19) can be represented as a closed system, the scheme of which is shown in Fig. 3.

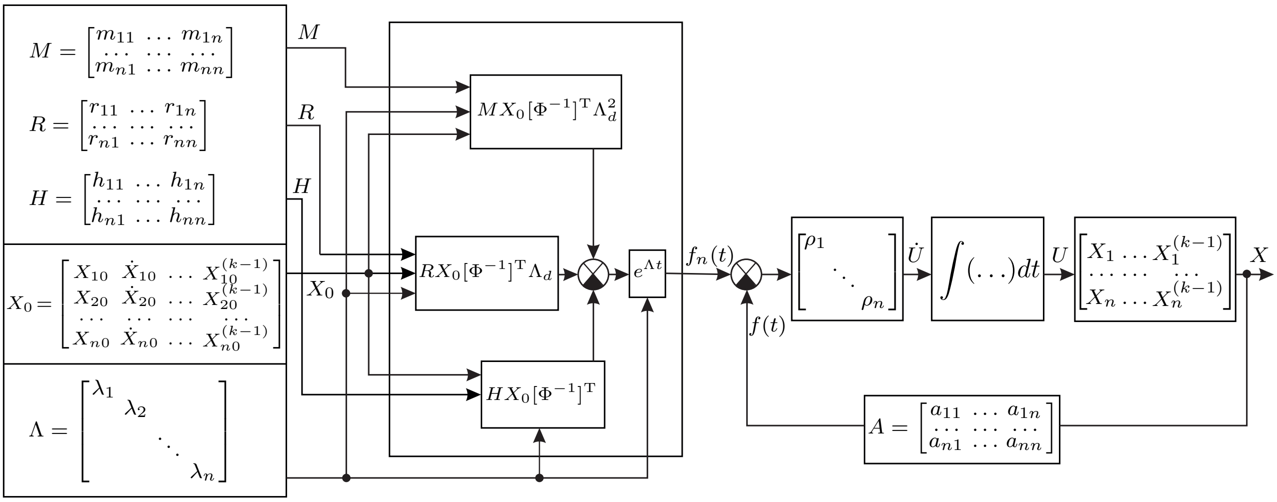
$M = \begin{bmatrix} m_{11} & \ldots & m_{1n} \\ \ldots & \ldots & \ldots \\ m_{n1} & \ldots & m_{nn} \end{bmatrix}$
$R = \begin{bmatrix} r_{11} & \ldots & r_{1n} \\ \ldots & \ldots & \ldots \\ r_{n1} & \ldots & r_{nn} \end{bmatrix}$
$H = \begin{bmatrix} h_{11} & \ldots & h_{1n} \\ \ldots & \ldots & \ldots \\ h_{n1} & \ldots & h_{nn} \end{bmatrix}$
$X_0 = \begin{bmatrix} X_{10} & \dot{X}_{10} & \ldots & X_{10}^{(k-1)} \\ X_{20} & \dot{X}_{20} & \ldots & X_{20}^{(k-1)} \\ \ldots & \ldots & \ldots & \ldots \\ X_{n0} & \dot{X}_{n0} & \ldots & X_{n0}^{(k-1)} \end{bmatrix}$
$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$
M
R
H
X_0
$MX_0[\Phi^{-1}]^{\mathrm{T}}\Lambda_d^2$
$RX_0[\Phi^{-1}]^{\mathrm{T}}\Lambda_d$
$HX_0[\Phi^{-1}]^{\mathrm{T}}$
$e^{\Lambda t}$
$f_n(t)$
$f(t)$
$\begin{bmatrix} \rho_1 & & \\ & \ddots & \\ & & \rho_n \end{bmatrix}$
$\dot{U}$
$\int (\ldots) dt$
U
$\begin{bmatrix} X_1 \ldots X_1^{(k-1)} \\ \ldots\ldots & \ldots \\ X_n \ldots X_n^{(k-1)} \end{bmatrix}$
X
$A = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix}$

Fig. 3. Algorithm block diagram.

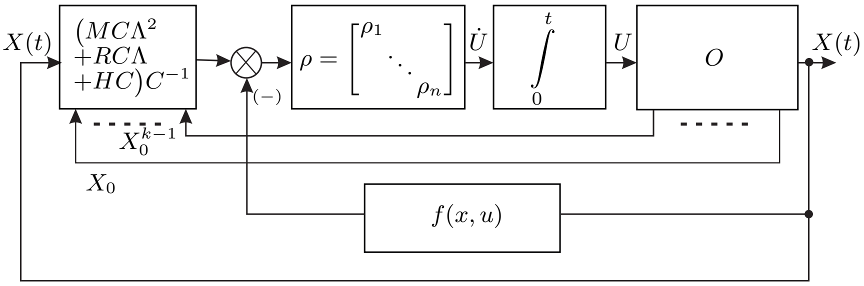
$X(t)$
$(MC\Lambda^2 + RC\Lambda + HC)C^{-1}$
$(-)$
$\rho = \begin{bmatrix} \rho_1 & & \\ & \ddots & \\ & & \rho_n \end{bmatrix}$
$\dot{U}$
$\int_0^t$
U
O
$X(t)$
X_0^{k-1}
X_0
$f(x, u)$

Fig. 4. Structural scheme of the control algorithm.

This scheme corresponds to the algorithmic solution of the equation $f(X, U) = f_n(t)$. The programmatic value of the control force is determined by the relation (17).

It should be borne in mind that even if program control $U_n(t)$ as a function of time is built, the practical benefits of its use are small: various kinds of disturbances, as well as the uncertainty of the properties and conditions of the object, including in emergency situations, will prevent the implementation of program motion. Therefore, the control should be formed taking into account additional information coming into the control system in the process of movement. This requirement is met by the feedback control.

To obtain a feedback control algorithm, we present the programmed motion in the form

$$X_n(t) = Ce^{\Lambda t}, \tag{20}$$

where $C = \{c_{ij}\}$, $i, j = \overline{1, n}$ is the square matrix of dimension $n \times n$ [1–7, 10, 15–18, 21].

It should be noted that in this case the matrix C is modified in comparison with the expression (16) — reduced to square. In many practical cases, such a representation of program motion is possible and permissible [1, 17–20]. Then from the condition of equality $X(t) = X_n(t)$ we get

$$X(t) = Ce^{\Lambda t}.$$

Therefore

$$e^{\Delta t} = C^{-1}X(t), \quad \det C \neq 0. \tag{21}$$

Then the force control law (17) with regard to expression (21) can be represented as

$$f_n[X(t)] = \left(MC\Lambda^2 + RC\Lambda + HC\right)\left[C^{-1}X(t)\right]. \tag{22}$$

At each moment of time $f_n(t)$ it is formed by measuring the current state of the system, that is, based on feedback. It is characteristic that the coefficients of the force control algorithm are determined by the initial conditions. The block diagram of the control algorithm constructed in accordance with (22) is shown in Fig. 4.

4. Conclusions

The article proposes to apply the method of inverse problems of dynamics for the synthesis of a system for stabilizing the motion of a dynamic object on an operational programmable trajectory. The procedure for applying the method of inverse problems of dynamics provides for the sequential execution of the following procedures. At the first stage, the desired trajectory of the dynamic object $X_n^0(t)$ is specified and the vector of necessary control forces F is determined for the realization of this trajectory of movement. At the second stage, the control functions $U_n^0(t)$ (control deviations) for creating such forces F are determined.

In the development of the concepts of the algorithmic approach — inverse problems of dynamics, an analytical expression is obtained for the governing force. The proposed block diagram of the control algorithm can be used to synthesize control systems for complex dynamic objects, for example, remotely piloted aircraft.

[1] Mashkov O. A. Synthesis of multidimensional automatic systems based on the solution of inverse problems of dynamics. KVVAIU, Kiev (1989), (in Russian).

[2] Galiullin A. C. Methods for solving inverse problems of dynamics. Science, Moscow (1986), (in Russian).

[3] Ermoshina O. V., Krishchenko A. P. Synthesis of programmed control of the orientation of a spacecraft by the method of inverse problems of dynamics. Journal of Computer and Systems Sciences International. **39** (2), 313–320 (2000).

[4] Kondratieva L. A. Approksimacionnaja obratnaja zadacha dlja predel'nyh ciklov. Kachestvennoe i chislennoe issledovanie matematicheskih modelej dinamicheskih sistem. Mezhvuz. sb. nauchn. trudov. 72–75 (2006), (in Russian).

[5] Kondratyeva L. A. The Inverse Boundary-Values Problems of Dynamics on Manifolds. Vestnik Rossijskogo universiteta druzhby narodov: Serija Matematika, informatika, fizika. **1**, 34–38 (2010), (in Russian).

[6] Krutko P. D. Inverse problems of the dynamics of controlled systems. Mashinostroenie, Moscow (2004), (in Russian).

[7] Lavrov N. G., Strashinin E. E., Shalimov L. N. Back dynamic concept applied to re-entry vehicle attitude control problem. Vestnik Juzhno-Ural'skogo gosudarstvennogo universiteta. **26**, 4–9 (2009), (in Russian).

[8] Artyushin L. M., Mashkov O. A., Durniak B. V., Sivov N. S. The theory of automatic control. UAD, Lviv (2004), (in Ukrainian).

[9] Artyushin L. M., Mashkov O. A., Sivov N. S. The theory of automatic control. Air Force CI, Kiev (1996), (in Russian).

[10] Velishchanskii M. A., Krishchenko A. P., Tkachev S. B. Synthesis of spacecraft reorientation algorithms based on the concept of the inverse problem of dynamics. Journal of Computer and Systems Sciences International. **42** (5), 811–818 (2003).

[11] Shabatura U. V., Paranchuk Ya. S., Chumakevich V. O. Problemy stvorennia funktsionalno-stiikykh elektromekhanichnykh kompleksiv. Bulletin of Lviv Polytechnic National University, Series of Electrical Power and Electromechanical Systems. **707**, 114–119 (2011), (in Ukrainian).

[12] Chumakevych V. Singularity of the functional-stable electromechanical complex of agricultural purpose. Bulletin of Lviv National Agrarian University. **20**, 115–128 (2016), (in Ukrainian).

[13] Mashkov V. New approach to system level self-diagnosis. 2011 IEEE 11th International Conference on Computer and Information Technology. 579–584 (2011).

[14] Mashkov O. A., Chumakevych V. O., Sokulsky O. E., Chyrun L. B. Features of determining controlling effects in functionally-stable systems with the recovery of a control. Mathematical Modeling and Computing. **6** (1), 85–91 (2019).

[15] Improving the efficiency of on-board information management complexes on the basis of detecting and parrying failures in the management process. Report on R & D No. 09026. KVVIAU, 226–264 (1990), (in Russian).

[16] Mashkov O. A. Ocenka vremeni perehodnogo procesa v dinamicheskoj sisteme s algoritmom upravlenija na osnove reshenija obratnyh zadach dinamiki. Otdel'nyj tematicheskij nauchno-tehnicheskij sbornik. KVVAIU, Kiev. 65–67 (1985), (in Russian).

[17] Mashkov O. A. Ocenka ustojchivosti dinamicheskoj sistemy s algoritmom upravlenija, poluchennym na osnove reshenija obratnyh zadach dinamiki. Otdel'nyj tematicheskij nauchno-tehnicheskij sbornik. KVVAIU, Kiev. 62–64 (1985), (in Russian).

[18] Mashkov O. A., Durniak B. V., Mamchur Yu. V., Timchenko O. V. Syntez alhorytmu prohramnoho keruvannia na trenazheri dystantsiino pilotovanoho litalnoho aparata na osnovi alhorytmichnoi protsedury rishennia obernenoi zadachi dynamiky (determinovana postanovka). Collection of scientific works "Modeling and information technologies". **82**, 166–176 (2018), (in Ukrainian).

[19] Mashkov O. A., Durniak B. V., Mamchur Yu. V., Timchenko O. V. Syntez alhorytmu prohramnoho keruvannia na trenazheri dystantsiino pilotovanoho litalnoho aparata na osnovi alhorytmichnoi protsedury rishennia zvorotnoi zadachi dynamiky (stokhastychna postanovka). Collection of scientific works "Modeling and information technologies". **83**, 146–153 (2018), (in Ukrainian).

[20] Mashkov O., Mamchur Yu. Formalization of ecological monitoring unmanned aerial vehicle control simulator pilot training problem based on inverse dynamics problem solution. Emerging Technologies. **2** (6), 24–30 (2018), (in Ukrainian).

[21] Artyushin L. M., Panov V. I., Shamov G. V. Synthesis of flight control algorithms based on solving inverse problems of dynamics. KVVAIU, Kiev (1982), (in Russian).

Використання методу обернених задач динаміки для синтезу системи стабілізації руху динамічного об'єкта на оперативно програмованих траєкторіях

Машков О. А.[1], Чумакевич В. О.[2], Мамчур Ю. В.[1], Косенко В. Р.[1]

[1] *Національний центр управління та випробувань космічних засобів*
Державного космічного агентства України,
вул. Московська, 8, Київ, 01010, Україна
[2] *Національний університет "Львівська політехніка",*
вул. С. Бандери, 12, Львів, 79013, Україна

Стаття присвячена аналізу можливості застосування методу обернених задач динаміки для синтезу системи просторової стабілізації руху динамічного об'єкта на оперативно програмованих траєкторіях. У статті запропоновано застосувати метод обернених задач динаміки для синтезу системи стабілізації руху динамічного об'єкта на оперативно програмованій траєкторії. Визначено, що процедура застосування методу обернених задач динаміки передбачає послідовне виконання двох процедур. Перша процедура передбачає задання бажаної траєкторія руху динамічного об'єкта і визначення вектора необхідних керуючих сил для реалізації цієї траєкторії руху. Друга процедура передбачає визначення керуючої функції (відхилення органів управління) для створення таких сил. У розвитку концепцій алгоритмічного підходу — обернених задач динаміки отримано аналітичний вираз для керуючої сили. Запропонована структурна схема алгоритму управління може бути використана для синтезу систем керування складних динамічних об'єктів, наприклад дистанційно пілотованих літальних апаратів.

Ключові слова: *алгоритм управління, динамічний об'єкт, керуюча сила, керуюча функція, обернена задача динаміки, оперативно програмована траєкторія, синтез системи керування, стабілізація руху.*