MACHINE SCIENCE
ISSN: 2227-6912
E-ISSN: 2790-0479
2024, № 1 Pages 46-57

THEORETICAL AND PRACTICAL ASPECTS OF THE APPLICATION OF THE DYNAMIC PROGRAMMING METHOD IN OPTIMAL CONTROL PROBLEMS

Viktor ARTEMYEV [1,a], Natalia MOKROVA[1,b], Anar HAJIYEV[2,c*]

[1]Federal State Budgetary Educational Institution of Higher Education Russian Biotechnological University, Moscow, Russia

[2]Department of Machine design and industrial technologies, Azerbaijan Technical University, Baku, Azerbaijan

E-mail: [a]artemyevvs@mgupp.ru, [b]mokrovanv@mgupp.ru, [c*]anar_hajiyev_1991@mail.ru

https://doi.org/10.61413/GIPV6858

Abstract: The article examines the dynamic programming method based on the principle of optimality, analyzes the theoretical aspects of the method, as well as its use for analyzing a wide range of systems whose behavior in the future can be fully or statistically predicted based on their current state. The research results suggest that dynamic programming is used to solve a variety of tasks, including, but not limited to, the development of algorithms in the fields of machine learning, automated management and the definition of a management strategy for production systems. The paper presents aspects of the application of the dynamic programming method to solve practical problems of optimal process control, demonstrating its effectiveness and versatility in conditions of real operational constraints.

Keywords: *dynamic programming, mathematical model, object control, system behavior.*

Introduction.

The mathematical description of the technological object determines the formulation and methods of solving the optimal control problem. The mathematical model of the control object is usually presented in the form of differential equations or systems of equations describing the dynamics of changes in the state of the system under the influence of external and internal factors. Object management involves the search for such a strategy of influencing the system, which will lead to optimal modes of conducting technological operations in accordance with the specified criteria [1, 2]. The purpose of building a management algorithm is to optimize the operation of an object in accordance with specified efficiency criteria [3], which includes minimizing costs, maximizing productivity or achieving a certain quality of management [4].

The increasing volumes of production systems and their complexity lead to the need to formulate tasks for adapting management facilities to changing operating conditions of the facility [5], expanding the requirements for its efficiency and reliability. The management of complex systems using traditional methods becomes ineffective [6, 7]. Which leads to the need to develop new approaches and management methods [8] capable of ensuring high adaptability and optimal behavior of the object in real time [9]. To solve this problem, a dynamic programming method is used, which allows us to effectively find optimal management strategies based on the Bellman optimality principle [10]. To implement effective management, it is necessary to integrate a system with various levels of control, from monitoring and analytics to active process management [11, 12]. A systematic approach combining data from all levels and subsystems is the basis for managerial decision-making

Formulation of the problem.

The dynamic programming method is used for a wide class of problems in the theory of optimal automatic control systems [13, 14, 15].

Consider the problem of controlling an object with the equation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x,u) \tag{1}$$

where x is an n-dimensional vector with coordinates $x_{1,\dots,}x_n$, and u is an r–dimensional vector with coordinates $u_{1,\dots,}u_r$. Let

$$u \in \Phi(u), \tag{2}$$

and it is required to minimize the functional

$$Q = \int_0^T G\left[x(t),u(t)\right]dt, \tag{3}$$

where, for example, we will consider T fixed for now.

The dynamic programming method is based on the optimality principle formulated by R. Bellman for a wide range of systems whose future behavior is completely or statistically determined by their state in the present. Therefore, it does not depend on the nature of their "prehistory", i.e. the behavior of the system in the past, as long as the system is currently in this state.

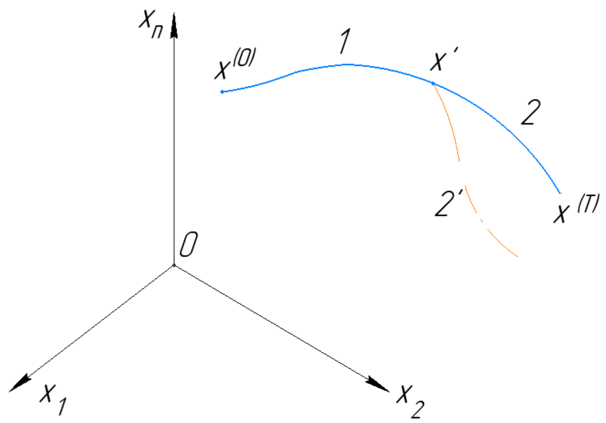

Figure 1. Illustration of the optimality principle

To illustrate, consider the optimal trajectory in the n-dimensional phase space of Fig. 1 with the initial and final values of the vector x equal to x^0 at $t = t^0$ (usually $t_0 = 0$) and $x^{(\mathrm{T})}$ at $t = T > t_0$. Let the initial conditions $x^{(0)}$ be given; the value of $x^{(\mathrm{T})}$, generally speaking, is unknown. Note some intermediate point x' of the trajectory corresponding to $t = t'$, where $t_0 < t' < T$, and call the section of the trajectory from $x^{(0)}$ to x' the first (1) in Fig. 1, and from x' to $x^{(\mathrm{T})}$ is the second (2). For the second section, as an independent trajectory from (3), we get $\int_{t'}^{T} G[x,u]\,dt$. The trajectory is optimal with the minimum value of the integral. The integral is minimal. The principle of optimality can be formulated as follows: the second section of the optimal trajectory is, in turn, the optimal trajectory.

Thus, the initial state of the system is x' at the initial moment of time $t = t'$, then regardless of how the system came to this state, its optimal subsequent movement will be trajectory 2. Let's assume the opposite, then criterion (3), considered for the time interval from *t* to *T*, will be the smallest is not for trajectory 2, but for some other trajectory $2'$, starting from point x', and shown by a dotted line in Fig. 1. But in this case, it would be possible to build a "better" trajectory than trajectory 1-2, and for the initial task it is only necessary to choose the control u so that trajectory 1 and then 2 are described. We proceeded from the optimality of trajectory 1-2. The contradiction proves the impossibility of the existence of trajectory 2, providing a lower value of *Q* than trajectory $2'$. So, trajectory 2 is optimal.

The optimality principle formulated above is a very general prerequisite for an optimal process, valid for both continuous and discrete systems. Only in the case when the endpoint is set c' from the first section at $t = t'$, the first section is also the optimal trajectory in itself. In this case, the state of the system at the time under consideration is understood to be the state corresponding to the point x' at time $t = t'$.

Let's say the motion of a controlled object is characterized by a first-order equation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f_1(x,u), \tag{4}$$

where *x* is the only coordinate of the system, and *u* is the only control action limited to some area (2). Let the initial condition $x(0) = x^{(0)}$ be given. Let's assume that we need to find a control law *u(t)* that minimizes the integral

$$\mathrm{Q} = \int_{\mathrm{t}_0}^{\mathrm{T}} \mathrm{G}_1(\mathrm{x},\mathrm{u})\mathrm{dt} + \varphi_1\left[x(T)\right], \tag{5}$$

where t_0 will usually be considered equal to zero, and the value of $T = const$. Let's replace the continuous system with a discrete-continuous one from the point of view of convenience of machine calculations, as well as determining the class of functions under consideration. The main scope of the dynamic programming method lies precisely in the field of discrete-continuous or purely discrete systems, or systems approximated to them.

We divide the interval *(0, T)* into *N* equal sections of small length Δ and consider only the discrete values $x = x(k)$ and $u = u(k)(k = 0,1,\dots,N)$ at time points $t = 0, 1\Delta, 2\Delta, \dots, k\Delta \dots, (N-1)\Delta, N\Delta = T$. Then the differential equation (4) of the object can be approximately replaced by an equation in finite differences

$$\frac{x(k+1) - x(k)}{\Delta} = f_1\left[x(k), u(k)\right], \tag{6}$$

or

$$x(k+1) = x(k) + f\left[x(k), u(k)\right], \tag{7}$$

were

$$f\left[x(k), u(k)\right] = \Delta f_1\left[x(k), u(k)\right]. \tag{8}$$

The initial condition remains the same:

$$x(0) = [x]_{t=0} = x^0. \tag{9}$$

The integral (5) is approximately replaced by the sum

$$Q=\sum_{n=0}^{N-1}G\left[x(k),u(k)\right]+\varphi\left[x(N)\right] \tag{10}$$

were

$$\begin{cases} G\left[x(k),\ u(k)\right]=G_1\left[x(k),\ u(k)\right]\Delta, \\ \varphi\left[x(N)\right]=\varphi_1\left[x(N\Delta)\right]=\varphi_1\left[x(T)\right] \end{cases} \tag{11}$$

The task is to determine the sequence of discrete values of the control action and, i.e., the values *u (0)* and *u (1) ..., u (N – 1)* minimizing the sum (10) under conditions (2), (7) and (9). Thus, it is required to find the minimum of the function of many variables and the method of dynamic programming makes it possible to reduce this operation to a sequence of minimizations of a function of one variable.

We realize the movement from the end of the process, from the moment $t = T$, to its beginning. We consider the moment $t=(N-1)\Delta$. The values $u(i)$ $(i=\ 0,\ 1\ldots,N-2)$, except for the last $u(N-1)$, have already been implemented in some way, and some value $x(N-1)$ corresponding to the moment $t=(N-1)\Delta$ has been obtained. According to the principle of optimality, the impact of $u(N-1)$ does not depend on the "background" of the system and is determined only by the state of $x(N-1)$ and the purpose of management. Consider the last section of the trajectory from $t=(N-1)\Delta$ to $t=N\Delta$. The value of $u(N-1)$ affects only those terms of the sum (10) that relate to this section. Denote the sum of these terms by Q_{N-1}:

$$Q_{N-1}=G\left[x(N-1),u(N-1)\right]+\varphi\left[x(N)\right] \tag{12}$$

From (7) we get

$$x(N)=x(N-1)+f\left[x(N-1)\right] \tag{13}$$

Therefore, $x(N)$ also depends on $u(N-1)$. Let's find an acceptable value $u(N-1)$ satisfying (13) and minimizing the value Q_{N-1}. Denote the found minimum value Q_{N-1} by S_{N-1}. This value obviously depends on the state of the system at $t=\ (N-\ 1)$, i.e. on the value of $x(N-1)$ included in (12) and (13). So, $S_{N-1}=S_{N-1}\left[x(N-1)\right]$. Let's write an expression for S_{N-1}:

$$\begin{aligned} \mathrm{S}_{\mathrm{N}-1}\left[\mathrm{x}(\mathrm{N}-1)\right]&=\min_{u(n-1)\in\omega(u)}Q_{N-1}=\min_{u(n-1)\in\omega(u)}\left\{G\left[x(N-1),u(N-1)\right]+\varphi\left[x(N)\right]\right\}= \\ &=\min_{u(n-1)\in\omega(u)}\left\{G\left[x(N-1)\right]+\varphi\left[x(N-1)+f\left[x(N-1),u(N-1)\right]\right]\right\} \end{aligned} \tag{14}$$

To define S_{N-1} is required to minimize only the variable $u(N-1)$, S_{N-1} we get a function from $x(N-1)$, then fix the resulting value. Let's move on to the penultimate section, considering two sections – the last and the penultimate, note that the choice of $u(N-2)$ and $u(N-1)$ affects only the summands (10) included in the expression

$$Q_{N-2}=G\left[x(N-2),u(N-2)\right]+\{G[x(N-1),u(N-1\}]+\varphi\left[x(N)\right]\} \tag{15}$$

The value $x(N-2)$ at the initial moment of the penultimate interval, obtained as a result of the prehistory of the process, will be considered set. It follows from the principle of optimality that only the value of $x(N-2)$ and the goal of control – minimizing Q_{N-2} – determine optimal control in the area under consideration. Let's find the value S_{N-1} – the minimum of Q_{N-2} by $u(N-2)$ and $u(N-1)$. But the minimum of $u(N-1)$ in (15) has already been found for each value of $x(N-1)$, and the latter depends on $u(N-2)$. In addition, when minimizing Q_{N-1}, the corresponding optimal value $u(N-1)$ is found along the way; we denote it by $u^*(N-1)$. If you also consider that the rst term in (15) does not depend on $u(N-2)$, we can write:

$$S_{N-2}\left[x(N-2)\right]=\min_{\substack{u(N-2)\in\omega(u)\\ u(N-1)\in\omega(u)}} Q_{N-2} = \min_{u(N-2)\in\omega(u)}\left\{G\left[x(N-2),u(N-2)\right]+S_{N-1}\left[x(N-1)\right]\right\}=$$

$$=\min_{u(N-2)\in\omega(u)}\{G\left[x(N-2),u(N-2)\right]+S_{N-1}-f\left[x(N-2),u(N-2)\right]]\}$$

because of (7) implies $x(N-1)=x(N-2)+f\left[x(N-2),u(N-2)\right]$.

Note that the minimization is performed using one variable $u(N-2)$. In this case, we find $u^*(N-2)$ – the optimal value of u $u(N-2)$ – and the value S_{N-2} – the minimum of the function Q_{N-2}. Both $u^*(N-2)$ and S_{N-2} are functions of $x(N-2)$. Now we fix the value of S_{N-2}. It is important to note that the found optimal value $u^*(N-2)$ minimizes the entire expression in the curly bracket of the formula S_{N-2}, and not specifically the summand $G\left[x(N-2),y(N-2)\right]$. Therefore, a strategy in which each value of $u(N-j)$ is chosen by minimizing only a specific term $G\left[x(N-j),y(N-j)\right]$ in the sum (10) is not at all optimal. The optimal strategy takes into account the ultimate goal, i.e. minimizing the entire expression in the curly bracket, depending on $u(N-j)$.

Let's continue the procedure of moving from the end to the beginning of the interval *(T, 0)*. Taking into account the third section from the end requires consideration of that part of the sum *Q*, which depends on $u(N-3)$. Let's denote this part by Q_{N-3}:

$$Q_{N-3}=G\left[x(N-3),u(N-3)\right]+$$
$$+\left\{G\left[x(N-2),u(N-2)\right]+G\left[x(N-1),u(N-1)\right]+\varphi\left[x(N)\right]\right\}$$

Based on expression (13), we write $x(N-2)=x(N-3)+f\left[x(N-3),u(N-3)\right]$. Next, the minimum of the expression in the curly bracket in the expression Q_{N-3} is $S_{N-2}\left[x(N-2)\right]$. Therefore, the minimum S_{N-3} of the expression Q_{N-3} is equal to

$$S_{N-3}\left[x(N-3)\right]=\min_{u(N-3)\in\omega(u)}\left\{G\left[x(N-3),u(N-3)\right]+S_{N-2}\left[x(N-3)\right]\right\}=$$

$$=\min_{u(N-3)\in\omega(u)}\{G\left[x(N-3),u(N-3)\right]+S_{N-2}-f\left[x(N-3),\ u(N-3)\right]]\}$$

Passing in a similar way to $S_{N-4},\ldots,S_{N-k}$, we obtain the recurrent formula

$$S_{N-k}\left[x(N-k)\right]=\min_{u(N-k)\in\omega(u)}\{G\left[x(N-k),u(N-k)\right]+ \\ +S_{N-k+1}\left[x(N-k)\right]+f\left[x(N-k),\,u(N-k)\right]\} \tag{16}$$

In parallel, we determine the optimal value of $u*$, depending on $x(N-k)$

$$u^{*}(N-k)=u^{*}\left[x(N-k)\right] \tag{17}$$

And the minimizing expression in curly bracket (16). Calculating by formula (16) sequentially S_{N-k} for $k=$ 1, 2, $\ldots,N$, we come to determine the optimal value of $u*(0)$, i.e. the value of the control action required at the initial moment of time. Simultaneously with determining the value of $u(0)$, we obtain S_0, i.e. the minimum value of the criterion *Q* under optimal control. The analytical expression of the minimization results turns out to be impossible; therefore, this procedure is performed numerically. The solution process is transferred to an object of any order *n* with equation (1) and any number of control actions $u_l\,(l=1,\ldots,r)$. It is only necessary to replace the scalars *x*, *u*, *f* in the above formulas with vectors *x*, *u* and *f*. In this case, vectors should be introduced for the kth instant of time $t=k\Delta$

$$\begin{cases} x(k)=\{x_1(k),\ldots,x_n(k)\}, \\ u(k)=\{u_1(k),\ldots,u_r(k)\}. \end{cases} \tag{18}$$

Here $u_j\,(N-k)$ is the *j*-th control action, and $x_j\,(N-k)$ is the *j*-th coordinate at the moment $t=(N-k)\Delta$.

Let's replace the differential equations (1) with equations in finite differences, and the integral (3) with the sum. Then the reasoning, which is completely similar to the above, shows that formula (16) is replaced by the expression

$$S_{N-k}\left[x(N-k)\right]=\min_{u(N-k)\in\omega(u)}\{G\left[x(N-k),u(N-k)\right]+ \\ +S_{N-k+1}\left[x(N-k)\right]+f\left[x(N-k),\,u(N-k)\right]\} \tag{19}$$

The calculation procedure is similar if in *f* it clearly depends on time.

Next, at each stage, we find the minimum of the function *r* of the variables $u_1(N-k),\ldots,u_r(N-k)$, the optimal values are the scalar S_{N-k} and the vector $u^{*}(N-k)$ - the essence of the function of the vector $x(N-k)$, i.e. the function *n* variables $x_1(n-k)_{,}\ldots,x_n(n-k)$.

Dynamic programming is not a solution to any problem, at one time the method was not used because of the so-called curse of dimensionality. With the development of computer technology, instead of analytical patterns, it became possible to search for solutions in the form of graphs or tables, i.e. calculation procedures until the desired result is obtained. The simpler the calculation procedure, the better the method. Dynamic programming is characterized by a radical simplification of the calculation procedure in comparison with the direct method of solving the problem. Indeed, the problem of minimizing the sum (10) can be considered as the problem of minimizing the function of *N* variables $u(0),u(1),\ldots,u(N-1)$.

To solve the minimization problem, it is necessary to express each *x(k)* as a functional dependence on all previous control actions $u(0)$, $u(k-1)$ (and initial conditions) using formula (7), i.e. to find a solution for *x(k)* in general form. As a result of such a replacement, expression (10) will become more complicated, and finding the smallest of possibly several minima with a large number of variables is also difficult.

Meanwhile, dynamic programming allows you to replace the minimization of a complex function of many variables with a sequence of minimizations. At the same time, in each of the minimization processes, the minimum of a much less complex function of one or more variables (*n* variables for an object of the nth order) is determined. Therefore, using dynamic programming, it is possible to solve a number of problems that are unsolvable by direct minimization. It does not follow from the above that the direct method is always unacceptable, it is applicable with a limited number of variables. In general, dynamic programming provides a significant rationalization of calculations compared to the direct method. In this case, the solution can be extremely cumbersome. Indeed, at each stage of calculations, it is necessary to find and memorize the functions $S_{N-k}(x)$ and $S_{N-k+1}(x)$, i.e., in general, two functions of *n* variables. Memorizing such functions for large values of n requires a significant amount of memory and in difficult cases is practically achievable only with the help of any approximations.

The described technique is transferred without fundamental changes to optimal systems with random processes. To illustrate, let's consider an example in which, in addition to *u*, a random disturbance *z* acts on an object of the first order. Then equation (7) will be replaced by equality

$$x(k+1) = x(k) + f\left[x(k), u(k), z(k)\right] \tag{20}$$

where *z(k)* are the discrete values of the disturbance. Now *x(k)* and criterion (10) become random variables. Therefore, as a new criterion *Q*, the value of which needs to be minimized, we choose the mathematical expectation of expression (10), and we also introduce *z* into the number of arguments *G* for generality:

$$Q = M\left\{\sum_{n=0}^{N-1} G\left[x(k), u(k), z(l)\right] + \varphi\left[x(N)\right]\right\} \tag{21}$$

Here *M* is the mathematical expectation. In this example, we will consider the values *z(i)* and *z(j)* for *j* to be independent and assume that the densities of distributions $P\left[z(0)\right], P\left[z(1)\right), \dots \dots, P\left[z(N)\right]$ are known. Using the proposed method, we first find a function for each fixed $x(N-1)$

$$\begin{aligned} S_{N-1}\left[x(N-1)\right] &= \min_{u(N-1)\in\omega(u)} Q_{N-1} = \\ &= \min_{u(N-1)\in\omega(u)} M\{G\left[x(N-1), u(N-1), z(N-1)\right] + \\ &+ \varphi\left[x(N-1) + f\left[x(N-1), u(N-1), z(N-1)\right]\right]\} = \\ &= \min_{u(N-1)\in\omega(u)} \int_{-\infty}^{\infty} P\left[z(N-1)\right]\{G\left[x(N-1), u(N-1), z(N-1)\right] + \\ &+ \varphi\left[x(N-1) + f\left[x(N-1), u(N-1), z(N-1)\right]\right]\} dz(N-1). \end{aligned} \tag{22}$$

When minimizing, the optimal value of $u^*\left[x(N-k)\right]$ is determined simultaneously. Having memorized $S_{N-1}\left[x(N-1)\right]$, we find the following function

$$S_{N-2}\left[x(N-2)\right]=\min_{u(N-2)\in\omega(u)} M\{G\left[x(N-2),u(N-2),z(N-2)\right]+$$

$$+S_{N-1}\left[x(N-1)\right]\}=\min_{u(N-2)\in\omega(u)}\int_{-\infty}^{\infty}P\left[z(N-2)\right]\{G\left[x(N-2),u(N-2),z(N-12)\right]+ \tag{23}$$

$$+S_{N-1}\left[x(N-2)+f[x(n-2),u(N-2),z(N-2)\right]\}dz(N-2).$$

The solution methodology turned out to be essentially the same as for regular systems. A similar technique is applicable to an object of any order. We can also consider more general problems in which $\mathrm{P}\left[z(i)\right]$ are unknown in advance, and some optimal procedure for processing observations allows us to accumulate information about the densities of distributions.

Solution of the problem

The dynamic programming method, with some additional assumptions, can be applied to the study of continuous systems. Let the motion of an object be characterized by the equations

$$\frac{\mathrm{d}x}{\mathrm{d}t}=f(x,u,t). \tag{24}$$

At the initial moment of time t_0, the vector *x* is equal to $x^{(0)}$, and the optimality criterion has the form, $T = const$

$$Q=\int_{t_0}^{T}G(x,u,t)\mathrm{d}t \tag{25}$$

Let's assume that an optimal trajectory has been found leading from the starting point $x^{(0)}$ to the end point $x^{(T)}$. The minimum value of the criterion *Q* corresponding to the optimal trajectory is denoted by $S\left(x^{(0)},t^{(0)}\right)$. According to the principle of optimality, the section of the trajectory from the point x corresponding to the moment $t>t_0$ to the endpoint $x^{(T)}$ Fig. 2 is also the optimal trajectory, and the part of the criterion *Q* corresponding to this section and the time interval from *t* to *T* has the minimum possible value.

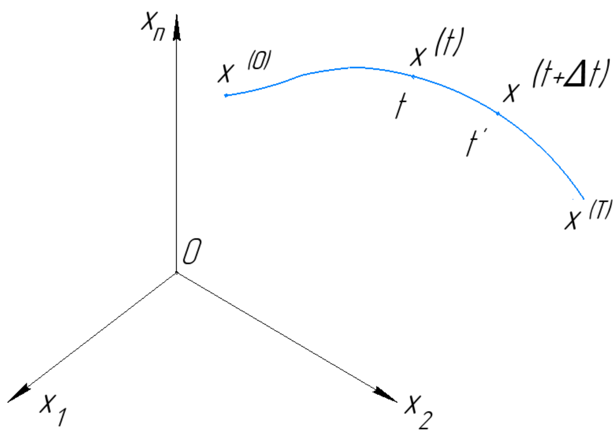

Figure 2. Illustration of the optimality principle, continuous case

Let's denote this value by $S\left[x(t),t\right]$. Let $\Delta\mathrm{t}$ be a small period of time, and $S\left[x(t+\Delta\mathrm{t}),t+\Delta\mathrm{t}\right]=S\left[x',t'\right]$ be the minimum value of the part of the integral *Q* that corresponds to the optimal area trajectories from point $x(t+\Delta t)=x'$ to the end point $x^{(T)}$ and, consequently, the time interval from $t+\Delta t=t'$ to *T*. The ratio between $S[x',t']$ and $S[x,t]$ is completely analogous to formula (19); you just need to write $S[x,t]$ instead of $S_{N-k}[x(N-k)]$, $S[x',t']$ instead of $S_{N-k+1}[x(N-k+1)]$, finally, $G[x(t),u(t),t]\Delta t$ instead of $G[x(N-k),u(N-k)]$. The last substitution was made in the first of the equations (11). Since Δt is a small but finite period of time and replacing the differential equation with an expression in finite differences is inaccurate, it is necessary to add the expression $0_1(\Delta t)$ to some equality, i.e. the magnitude of the order of smallness is higher than Δt:

$$\lim_{\Delta t\to 0}\frac{0_1(\Delta t)}{\Delta t}=0 \tag{26}$$

Instead of (19) can be written:

$$S[x,t]=\min_{u(t)\in\omega(u)}\{G[x,u,t]\Delta t+S[x',t']\}+0_1(\Delta t) \tag{27}$$

The dependence of (27) is possible to obtain and regardless of the discrete case discussed above. Indeed, according to the definition

$$S[x,t]=\min_{u(t)\in\omega(u)}\int_t^T G(x,u,\tau)\,d\tau \quad (t\le\tau\le T) \tag{28}$$

Here *S* is the minimum value of the integral obtained on the set of all permissible controls $u(\tau)$ in the range from *t* to *T*. The integral (28) can be represented as the sum of two terms corresponding to the intervals from *t* to $t+\Delta t$ and from $t+\Delta t$ to *T*. Since Δt is small, you can write

$$S[x,t]=\min_{u(t)\in\omega(u)}[G(x,u,t)\Delta t+\int_{t'}^T G(x,u,v)\,dv]+0_1(\Delta t) \tag{29}$$

where Δt is considered small, and $0_1(\Delta t)$ is of the order of smallness higher than Δt. Since the first term in the square bracket (29) depends only on the value of *u(t)* at time *t*, and only the integral in the square bracket also depends on the values of *u(v)* in the interval of change of *v* from $t'\neq t+\Delta t$ to *T*, then you can write

$$\begin{aligned}S[x,t]&=\min_{u(t)\in\omega(u)}[G(x,u,t)\Delta t+\min_{u(t)\in\omega(u)}\int_{t'}^T G(x,u,t)\,dv]+0_1(\Delta t)=\\&=\min_{u(t)\in\omega(u)}\{G(x,u,t)\Delta t+S[x',t']\}+0_1(\Delta t)\end{aligned} \tag{30}$$

Here, under the sign of the minimum is the value *u(t)* at time *t*, formulas (30) and (27) coincide. Just as in (19), it should be noted that $x'=x(t+\Delta t)$ depends on *u(t)*. From (24) we find for small Δt

$$x'=x(t+\Delta t)=x(t)+\frac{dx}{dt}\Delta t+0_2(\Delta t)=x(t)+f[x(t),u(t),t]\Delta t+0_2(\Delta t), \tag{31}$$

where $0_2(\Delta t)$ is a value of the highest order of smallness compared to Δt. Formula (31) is similar to expression (17).

Now suppose that the function $S(\cdot)$ really exists, is continuous and has partial derivatives with respect to the variables $x_i\,(i=1,\dots,n)$, and with respect to *t*, i.e. all $\partial S/\partial x_i\ \ (i=1,\dots,n)$ and $\partial S/\partial t$ exist, the validity of the subsequent conclusion depends on the validity of the given assumption. If the latter is not justified, then the reasoning is only heuristic in nature. However, there are cases when the assumption is unfair, and the application of dynamic programming to continuous systems needs, as shown in a number of works, in general, additional justification.

Substitute the expression x' from (31) into formula (27) and decompose $S[x',t']$ into a Taylor series in the vicinity of point *(x, t)* we get:

$$\begin{gathered} S\ [x',t'] = S\left[x(t+\Delta t),t+\Delta t\right] = \\ = S[x(t\{+f\left[x(t),u(t),t\right]\Delta t + 0_2(\Delta t);t+\Delta t] = \\ = s\left[x,t\right] + \sum_{i=1}^{n}\frac{\partial S[x,t]}{\partial x_i} f_i\left[x,u,t\right]\Delta t + \frac{\partial S[x,t]}{\partial t}\Delta t + 0_3(\Delta t), \end{gathered} \tag{32}$$

where $0_3(\Delta t)$ is the value of higher order of smallness compared to Δt. We can write the formula in a more compact, introducing the gradient of the function *S(x, t)* is a vector with the coordinates of $\partial S/\partial x_i\ \ (i=1,\dots,n)$

$$grad\ S = \left(\frac{\partial S}{\partial x_1},\dots,\frac{\partial S}{\partial x_n}\right) \tag{33}$$

Then (32) takes the form

$$\begin{gathered} S\left[x',t'\right] = S\left[x(t+\Delta t),t+\Delta t\right] = \\ = S\left[x,t\right] + grad\,S\left[x,t\right], f\left[x(t),u(t),t\right]\Delta t + \frac{\partial S[x,t]}{\partial t}\Delta t + 0_3(\Delta t) \end{gathered} \tag{34}$$

Substituting (34) and (27) and taking out the values $S[x,t]$ and $\partial S/\partial t,$ independent of *u(t)*, the formula takes the form:

$$-\frac{\partial S[x,t]}{\partial t} = \min_{u(t)\in\omega(u)}\left\{G\left[x(t),u(t),t\right] + grad\,S\left[x,t\right], f\left[x(t),u(t),t\right]\right\} + \frac{0_4(\Delta t)}{\Delta t} \tag{35}$$

where $0_4(\Delta t)$ is a value of the highest order of smallness compared to Δt. Now let's aim Δt to zero, since $0_4(\Delta t)$ obeys condition (26), then the last term in the right part (35) disappears at $\Delta t \to 0$. Therefore, in the limit we get

$$-\frac{\partial S[x,t]}{\partial t} = \min_{u(t)\in\omega(u)}\left\{G\left[x(t),u(t),t\right] + grad\,S\left[x,t\right], f\left[x(t),u(t),t\right]\right\} \tag{36}$$

Expression (36) is called the Bellman equation and is a partial differential equation.

To concretize the theoretical calculations, let's look at an example. Let $r=1$ and $n=\ 2$ be in the special case, and $G=G(x_1,x_2)$ and the only control action is denoted by *i*. The equations of the object are:

$$\frac{\partial x_1}{\partial t} = f_1 = ux_1 + x_2, \frac{\partial x_2}{\partial t} = f_2 = u^2 \tag{37}$$

Then equation (36) takes the form ($S[x,t]$ replaced by *S*)

$$-\frac{\partial S}{\partial t}=\min_{u}\{G(x_1,x_2)+\frac{\partial S}{\partial x_1}\left(ux_1+x_2\right)+\frac{\partial S}{\partial x_2}u^2 \quad (38)$$

Assuming that $\partial S/\partial x_2>0$, we find the minimum of the curly bracket in *u*, equating the derivative to zero. The optimal value $u*$ that minimizes the square bracket we write in the form

$$u^*=-\frac{1}{2}x_1\frac{\partial S}{\partial x_1}\frac{1}{\partial S/\partial x_1} \quad (39)$$

Substituting (39) the expression in equation (38), we obtain the partial differential equation:

$$-\frac{\partial S}{\partial t}=G\left(x_1,x_2\right)+\frac{\partial S}{\partial x_1}x_2-x_1^2\frac{(\partial S/\partial x_1)^2}{4\partial S/\partial x_2} \quad (40)$$

The partial differential equation (40) can be solved, since the boundary conditions are known for it, *S[x,t]* is a known function. For example, for criterion (5) it is $\varphi_1\left[x,(T)\right]$, since for $t_0=T$ the integral in (5) is zero. For criterion (25), the function *S[x,t]* is zero. Knowing the boundary function *S[x,t]*, it is possible to integrate equation (40) by some known method. One of the usual methods of approximate integration consists in discretizing the problem and solving the resulting recurrence relations of the form (19). In some cases, it is possible to find an approximate solution in another way or even obtain an exact solution in a closed form. The resulting value of $u*$ represents optimal control.

Results and conclusions.

Based on equation (36), dependence (38) is obtained, which describes the change in the function *S* [*x, t*] over time depending on the parameters x_1 and x_2, as well as the control parameter *u*. This equation minimizes an expression involving the function $G(x_1,x_2)$, linear and quadratic terms by *u*. The derivative of *u* is zero, which makes it possible to find the optimal value of $u*$, which minimizes this expression, when substituted in (38), we obtain dependence (40), solved under given boundary conditions for *S* [*x, t*]. The boundary conditions for *S* [*x, t*] can be different, depending on the context of the problem, as indicated in studies with criteria (5) and (25). Using these conditions, equation (40) can be integrated by various methods. One of the most common approaches is the discretization of space and time and the subsequent solution of recurrence relations. It is also possible to find an approximate or exact solution to the equation in a closed form.

Finding the optimal control is an important result, since it allows not only to solve the partial differential equation, but also to optimize the control process depending on the given conditions and parameters of the system [16]. This emphasizes the importance of accurately determining the function *S* [*x, t*] and its initial or boundary values for the successful application of optimal control methods.

REFERENCES

[1]. Moiseev N.N. *Mathematical problems of system analysis.* – M.: Nauka, 1981, – 488 p.
[2]. Negojce K. *Application of systems theory to control problems*. –M: Mir, 1981, – 179 p.
[3]. V.A. Besekersky. E.P. Popov. *Theory of automatic control systems*. – St. Petersburg: Profession, 2003. – 752 p.
[4]. Avvakumov, S.N., Kiselev, Y.N. *Some algorithms of optimal control.* Proc. Steklov Inst. Math. 255 (Suppl 2), S1–S15 (2006). https://doi.org/10.1134/S0081543806060010.
[5]. Boris S. Mordukhovich, M. Ebrahim Sarabi. *Variational analysis and full stability of optimal solutions to constrained and minimax problems, Nonlinear Analysis: Theory, Methods &*

Applications. Volume 121, 2015, Pages 36-53, ISSN 0362-546X, doi.org/10.1016/j.na.2014.10.013

[6]. Kosheleva I.V., Danilov A.I., Mokrova N.V. *Decomposition control of a cascade of catalytic reforming reactors. / Advances in chemistry and chemical technology.* Volume XVI. 2002. No. 1 (18). p.32-33.

[7]. Mokrova N.V., Volodin V.M. Justification for the choice of methods for solving the problem of optimal control of complex processes. Bulletin of TSTU, 2006. Volume 12. 22-28 p. ISSN 0136-5835.

[8]. Mokrova N.V., Volodin V.M. *Modeling of decomposition control of multi-stage processes*. / Chemical and oil and gas engineering. 2007. No. 2. p.17-19. ISSN 0023-11-26. Mokrova N.V., Volodin V.M. *Analysis of high-end complex chemical technological system check-out*. Chemical and Petroleum Engineering. 2008. № 2. C. 3 – 5. ISSN 0023-11-26.

[9]. Bellman R. *Dynamic programming* . Transl. from English – M.: Izdatinlit, 1960. – 400 p.

[10]. Hu Wen-Tsen., Umbetov U. *Decentralized control of multidimensional objects with decomposition by situations.* News of the National Academy of Sciences of the Republic of Kazakhstan, Physics and Mathematics Series, 2007, No. 1, pp. 82 – 85.

[11]. Volodin V.M., Guseva A.Yu. *Optimal control of multi-stage processes with complex flow structures.* Chemical and petroleum engineering. No. 3. 1997. pp. 20 – 21.

[12]. Wilco van Harselaar, Niels Schreuders, Theo Hofman, Stephan Rinderknecht. *Improved Implementation of Dynamic Programming on the Example of Hybrid Electric Vehicle Control*, IFAC-PapersOnLine, Volume 52, Issue 5, 2019, Pages 147-152, ISSN 2405-8963, doi.org/10.1016/j.ifacol.2019.09.024.

[13]. A. S. Maksimov, S. D. Savostin, V. S. Artemyev. *SCADA systems* . – Kursk: Closed Joint Stock Company "University Book", 2023. – 127 p. – ISBN 978-5-907776-95-1.

[14]. V. Artemyev, A. Medvedev, V. Yaroshevich. *Investigation of optimal control of variable systems in the dynamic spectrum.* Machine Science. – 2023. – Vol. 12, No. 1. – P. 68-75.

[15]. V. Artemyev, S. Mokrushin, S. Savostin [et al.].*Processing of time signals in a discrete time domain.* Machine Science. – 2023. – Vol. 12, No. 1. – P. 46-54.

[16]. V. S. Artemyev, M. N. Makhiboroda, S. L. Yablochnikov [et al.]. *Implementation of Adaptive Control with Parametric Uncertainty*. Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED), Moscow, 10–11 november 2022. – Moscow: IEEE, 2022. – P. 9965505. – DOI 10.1109/TIRVED56496.2022.9965505.

[17]. A. Haag, M. Bargende, P. Antony, F. Panik. *Iterative refinement of the discretization of the dynamic programming state grid*. In 16. Int. Stuttgarter Symposium, Springer (2016), pp. 145-154.

[18]. Wei Zhou, Lin Yang, Yishan Cai, Tianxing Ying. *Dynamic programming for New Energy Vehicles based on their work modes part I: Electric Vehicles and Hybrid Electric Vehicles.* Journal of Power Sources, Volume 406, 2018, Pages 151-166, ISSN 0378-7753, doi.org/10.1016/j.jpowsour.2018.10.047

[19]. Mokrova N.V. *Problem of optimum control of manufacture of the activated coals.* Ugol. 2007. №7. c.72-74. ISSN: 0041-5790.

Received: 06.02.2024
Accepted: 22.05.2024