**Dynamic Programming Method for   
Optimal Control Systems**

Oybek Qosimov1, a), Shakhzodbek Sayfiyev2, Gulrukh Sultonova3

*1Andijan State Technical Institute, Andijan, Uzbekistan*

*2Navoi State University of Mining and Technology, Navoiy, Uzbekistan*

*3Tashkent State Technical University named after Islam Karimov, Tashkent, Uzbekistan*

*a)Corresponding author: oybek910066575@gmail.com*

**Abstract.** An optimal control system is a system that, by one method or another, is given the best quality in some particular aspect. In general, any scientifically based system is considered optimal, because, choosing one or another system, we thereby prefer it to another. The assessment of the system's performance is carried out using a sufficiently precise and always available optimization criterion, without which a reasonable choice of the system is impossible. The task of the theory of optimal control systems is to determine the general control laws of the object, which make it possible to think about what is possible and what is impossible to achieve in real conditions. Such a perfect formulation of the task is the task of determining a control algorithm that is optimal from some point of view, given complete a priori information about the control object (its mathematical definition taking into account the constraints imposed on various coordinates of the system), as well as a sufficiently precise mathematical expression of the control goal.

**Keywords:** optimal, programming, discrete, adaptability, dynamic, control.

**INTRODUCTION**

Bellman's Optimality Principle Dynamic programming is based on the optimality principle, which was formulated by Bellman for a wide range of problems. According to this principle, the optimal control strategy does not depend on the "previous state" of the system, but is determined by its current state and the control goal. We will explain the dynamic programming method with a simple example of controlling an object whose behavior is characterized by a first-order equation

(1)

where u and u are the only control and coordinate of the system, where u is bounded by the form u ϵΩ (u). It is assumed that the coordinate u also does not exceed the allowed values. Given the boundary conditions u(0) = u0; u(T) = ut, it is required to minimize the functional of this form

(2)

**METHODS**

Dynamic programming equation in discrete form First, we discretize the problem, that is, we convert the approximately continuous system into a discrete one. This step is necessary when preparing the problem for computer solution, and at the same time greatly simplifies the procedure for finding the optimal control. We divide equation (1) into N equal sections with duration Δ = T/N, and write them in the last derivatives:

Or

(3)

In this

We change the integral to a sum

(4)

In expression (2), when k = N, the last term of the sum is denoted by φ(uN), since it does not depend on the control, that is, at t = T the control process ends and uN = 0.

Now the problem is to determine the successive values of the control effects uk(k = 0, N–1) that minimize the sum (4) under the condition (3) and the constraints uϵΩ(u). Thus, the variational problem is reduced to the problem of finding the minimum of a multivariable function.

The dynamic programming method allows this operation to be reduced to the successive minimization of a single-variable function. For this, the path from the end of the process (t = T) to its beginning (t = 0) is used [1-4].

First, we assume that the time instant tN-1 = ∆(N–l) is considered. All values of uk, except the last one, are realized in some view and in this case, the time instant tN-1 has certain values uN-l corresponding to it. According to the principle of plausibility, the uN-1 effects do not depend on the “previous state” of the system and are determined only by the state of the system characterized by the value yN-1 and the control goal. In the last section of the path of motion (from tN-1 to tN), the quantity tN-1 affects only those members of the sum (4) that belong to this section, i.e.

Using (3) for k = N–1, we write the resulting expression in the form below

(5)

If the control objective is to minimize Q, then this condition must also hold for the sections under consideration. We define min QN-1 = SN-1. As can be seen from (5), the value of SN-1 depends on the state of the system at time tN-1, i.e., yN-1. Then

In this case, to determine SN-1, it is necessary to minimize QN-1 with respect to only one variable uN-1. By performing this operation, we obtain SN-1 as a function of yN-1. Before proceeding to the next stages of solving this function, it is necessary to remember it in the computer's memory. Moving from the end of the path to the previous section (from tN-2 to tN-1), we obtain

(6)

Using the acceptability principle again, we can say that only the value of yN-2 and the control objective (minimizing QN-2) determine the optimal control uN-2 and uN-1 on the considered section of the road. However, the minimum with respect to uN-1, and therefore the optimal value uN-1 = u\*N-1 itself, was found for any possible value of yN-1. This allows us to write the following, taking into account that the first term of (6) does not depend on uN-1, and the other two are equal to QN-1

Taking into account (3) when k = N–2, we get

(7)

Here, the minimization is also performed with respect to one variable uN-2. In this case, the minimum of the function QN-2 is determined, the optimal value of which is equal to u\*N-2 and SN-2. The value of SN-2 is entered into the computer memory, and SN-1 is deleted. Like u\*N-2, SN-2 is also considered a function of yN-2. Continuing in this way, we obtain the expression in the recurrent form

(8)

Expression (8) is called a discrete-form dynamic programming equation.

It is important to note that the optimal control u\*N-k minimizes not only the first integral FN-k, but all the expressions in the form of (8). The strategy of choosing each value of uN-k by optimizing only the corresponding integral FN-k is not optimal, because it does not take into account the ultimate goal of control.

By successively calculating the values of SN–k and, accordingly, u\*N–k according to expression (8), we finally obtain the value of the required control effect u\*(0) at the initial moment of time and the minimum value of the efficiency criterion S(0). This completes the search for optimal control.

The entire solution process is easily transferred to a control object represented by a differential equation of any level with any number of output coordinates of the system and control effects. It is only necessary to replace the scalars u, u in expression (8) with vectors u, u, and the function f with a vector-function f. Dynamic programming in continuous form As A.M. Letov showed, the dynamic programming equation can also be written in continuous form [5-6].

Considering that the left-hand side of expression (8) is independent of the control, we can move the value of SN–k to the right under the minimum sign. Then expression (8) can be written in the following form

(9)

Assuming that ∆u → 0 and ∆y = f(y, i), we perform a limit transition and write (9) in continuous form.

(10)

To obtain the minimum of the expression in square brackets in equation (10), it is necessary to differentiate it with respect to the control. Then, the condition for the minimum of (10) can be replaced by the following system of equations

(11)

Solving system (11) allows us to find the connection i = φ(u). In this case, the optimal control in the sense of minimizing criterion (2) is implemented. To do this, first we determine the value of dS(y)/dy from the second equation of system (11), and then we determine the desired connection i =φ(u).

If the system has p output coordinates u = |u1,...,up| and r controls i = |i1,...,ir|, we obtain

(12)

Or

(13)

The system of equations (13) is the most common written form of the dynamic programming equation in continuous form. Recall that the function S(u) must be continuous and differentiable with respect to ui. Here, dS(y)/dyi acts as the uncertain multiplier λi in the variational problem for the conditional extremum, and the equation dyi/dt = f(u,y) is similar to the coupling equations.

**RESULTS AND DISCUSSION**

Power-carrying tubular conductor should be laid between points O and T in such a way that the cost of its laying is minimal. This is a static variational problem. Since the laying is carried out on a section of a territory that is limited for various reasons and there is no mathematical description of this location, it is advisable to use the recurrent connection (8) to solve such a problem.

Solution We divide the road between O and T into several horizontal and vertical sections with an interval Δ, that is, we discretize the problem. The construction cost FN–k of each section can be calculated in advance using a topographic map and other a priori information.

The solution of the problem starts from the last point T (point N). Point N can be reached with one final step, either from points N1–1 or N2–1.

Note that from each of these points to point N there is only one way, that is, in the last step, there must be a unique control uN-1. We assume that the costs associated with the completion of construction at point N are minimal and equal to zero at this point, that is, φ(uN) = 0. Then, according to (5), we obtain

i=1.2.

Thus, it is possible to determine the costs at the last stage and write them to the computer memory (in Figure 1, the numerical expressions of the costs are shown in circles). Let us move on to the penultimate stage of the construction (from point Ni–2 to point Ni–1). From points N1–2 and N3–2, as before, there is only one path to point Ni–1, and for point N2–2 there are such paths, therefore, the control uN2-1 is now also equal to two. Let us calculate the costs at the penultimate stage, taking into account the last one:

The results are entered into the computer memory. We note that the path from point N2–2 to point N, which minimizes the costs QN-2, is suitable for controlling the transfer of the pipeline to point N2–1, while the path through point N1–1 is excluded from further consideration.

In this way, moving to the remaining points, we minimize the value of QN-k, eliminating the infeasible trajectories at each step. Finally, reaching point O and moving from it to point N, we find the optimal trajectory for the entire pipeline.

Eliminating the infeasible trajectories during the movement from T to O significantly simplifies the calculations compared to the method of simple trajectory cancellation (in our case, the number of arithmetic operations is reduced by more than one order of magnitude.)

**CONCLUSION**

In conclusion, it can be said that Bellman's optimality principle is the basis of dynamic programming, the optimality principle explained by for a wide range of problems. According to this principle, the optimal control strategy does not depend on the "previous state" of the system, but is determined by its current state and the control goal. The dynamic programming method allows this operation to be reduced to the successive minimization of a function of one variable. For this, the path of movement from the end of the process (t = T) to its beginning (t = 0) was used. From the optimality principle, we can say that only the value of yN-2 and the control goal (minimizing QN-2) determine the optimal control uN-2 and uN-1 on the considered section of the path of movement. However, the minimum in uN-1, and therefore the optimal value uN-1 = u\*N-1 itself, was found for any possible value of yN-1. This allowed us to write the following, taking into account that the first term of (6) does not depend on uN-1 and that the other two terms are equal to QN-1. In this way, moving to the remaining points, at each step, the unacceptable trajectories are eliminated. Excluding unacceptable trajectories during the movement from T to O significantly simplifies the calculations compared to the simple method of eliminating trajectories; in our case, the number of arithmetic operations is reduced by more than one order of magnitude.

**REFERENCES**

1. Nikolos, D. (1998). Optimal self-testing embedded parity checkers. IEEE Transactions on Computers, 47(3), 313–321. <https://doi.org/10.1109/12.660167>
2. Efanov, D. V., Pogodina, T. S., Aripov, N. M., Boltayev, S. T., Azizov, A. R., Ametova, E. K., & Toshboyev, Z. B. (2025). A Method for Synthesizing Self-Checking Discrete Systems with Calculations Testing Based on Parity and Self-Duality of Calculated Functions. Computation, 13(9), 220. <https://doi.org/10.3390/computation13090220>
3. Efanov, D., & Pogodina, T. (2023). Properties Investigation of Self-Dual Combinational Devices with Calculation Control Based on Hamming Codes. Informatics and Automation, 22(2), 349–392. <https://doi.org/10.15622/ia.22.2.5>
4. Sapozhnikov, V., Sapozhnikov, V., & Efanov, D. (2021). Synthesis of self-checking built-in control circuits based on logical complement to the equilibrium two-out-of-five code. Izvestiâ Vysših Učebnyh Zavedenij Priborostroenie, 64(3), 163–175. <https://doi.org/10.17586/0021-3454-2021-64-3-163-175>
5. Rungger, M., & Stursberg, O. (2009). Optimal Control for Deterministic Hybrid Systems using Dynamic Programming. IFAC Proceedings Volumes, 42(17), 316–321. <https://doi.org/10.3182/20090916-3-es-3003.00055>
6. Sundström, O., Ambühl, D., & Guzzella, L. (2009). On Implementation of Dynamic Programming for Optimal Control Problems with Final State Constraints. Oil & Gas Science and Technology – Revue D’IFP Energies Nouvelles, 65(1), 91–102. <https://doi.org/10.2516/ogst/2009020>