Maron A.I.

THE METHOD OF ALGORITHMS' CONSTRUCTION FOR FAULT FINDING IN RAILWAY AUTOMATION SYSTEMS

The paper offers the method of optimal algorithms construction for fault finding in systems where losses from failures grow exponentially with recovery time increasing. Such a situation is typical for railway automation systems.

Keywords: *fault finding algorithms, railway automation systems, loss due to failures.*

1. Introduction

Maintenance of railway automation systems includes efficient recovery in case of their failures. The mandatory step in the process of efficient recovery is fault finding, the duration of which to a large extent determines the size of losses due to a failure. Search time and the influence of the "human factor" can be significantly reduced by applying beforehand developed optimal algorithms of fault finding for critical systems, which are usually drawn up in the form of instructions. The currently known methods for constructing such algorithms assume that the optimality criterion is the average time of fault finding [1]. At the same time, in many critical automation systems the losses due to a failure are not linearly related to the recovery time, but grow much more rapidly. Thus, in case of an automatic block system failure on a railway section with heavy traffic the total time delay of trains increases exponentially with increasing recovery time. [2] In this paper, for the first time the method of constructing an optimal fault finding algorithm for such systems.

2. Statement of the problem and the solution method

Without diminishing the generality of the results, for the sake of description simplicity we formulate the problem as follows. The system, which is the subject of fault finding, is presented as a functional model of n blocks, numbered from 1 to n. One of the blocks is the output. The number n is assigned to it. Check of P^k, the signal at the output of the block k (k = 1, 2..., n) has a positive result $\pi^{\kappa}{}_1$, only if it is in good order and all the blocks prior to it are operational. Otherwise, the check of P^k will have a negative result $\pi^{\kappa}{}_0$, which means that either the block k is faulty, or one of the blocks prior to it is out of order. The check of P^n is performed. The result is a negative result, indicating failure of the system due to a malfunction of one of the blocks. The probability of a block i fault condition is equal to $p_{i.}$ Check time of P^k is equal to $t_{k.}$.

Fault finding will be carried out according to the developed conditional algorithm A, where the next check is selected depending on the result of the previous one. In this algorithm, its own time of fault finding T_i (A) corresponds to each block *i*. It is equal to the sum of the checks, which will have to be performed in accordance to the algorithm A. Losses due to system downtime are connected exponentially with time of system recovery. We shall assume that the time to replace the faulty unit is much less of the time to find fault. Then we can write that the loss due to a failure of the block *i* in adopted algorithm A for fault finding will be equal to

$$L_i(A) = \exp(r^*T_i(A)),$$

where r is the numerical factor the value of which for an automatic block system depends on the intensity of train operation on a railway section.

Average losses due to the system failure L (A) for a given algorithm A is the *i* sum from 1 to *n*, of products of probabilities p_i *by* L_i (A).

It is necessary to construct an algorithm A^*, where the average losses due to the system failure **are minimal.**

Even at *n* = 10-20 the number of different search algorithms for fault findings that can be constructed for a simple topology of the system under consideration is enormous. It is practically impossible to find among them the optimal algorithm by exhaustive method. Below a method is suggested that allows us to overcome the barrier of dimension and actually construct the required algorithm.

We shall consider fault findings as the process of system motion control. It should be transferred from the initial state, when all *n* faults are possible, into one of the final states when the faulty block is precisely defined. The number of such final states is *n*. All other states are intermediate ones. A subset of system blocks corresponds to each state, up to which accuracy the faulty block is localized in the course of checks' performance. For example, if for the system shown in figure 1, to perform the check P^1, then at its negative results, the system will transfer into a state S(2,3,4). A certain number of system blocks ||S|| corresponds to each state, up to which accuracy the faulty block is localized in the course of checks' performance. In what follows, we shall name as the level of fault localization. So, for S(2,3,4), we have ||S|| = 3. A certain number of checks P(S), which can be performed to further localize the fault, correspond to a non-final state S. So, for S(2,3,4) P^2 these are checks P^2 and P^3. As a result of check performance P^k in a state S included in P(S), the system will pass into a state $S(\pi^\kappa_1)$, if the result is π^κ_1 or $S(\pi^\kappa_0)$ if the result is.

The author has proved that the minimum average losses due to the system failure can be found using the recurrence relation

$$L^*(S) = \min L(S,\Pi^k) = \min\{\exp(r^*t_k)^*[L^*(S(\pi^\kappa_0)) + L^*(S(\pi^\kappa_1))]\}. \qquad (1)$$

The minimum in (1) is taken over all checks P^k, included in P(S). Calculations should be carried out consistently for all possible states S, starting with those, which has ||S|| = 2, and ending with the initial state, where ||S|| = *n*. At the same time, we assume that $L^*(S) = p_i$ for the final states in which ||S|| = 1.

The method of constructing an optimal algorithm will consist of the following stages.

By using the relation (1) calculate the value of L*(S) for all possible states S, starting with those, in which the level of fault localization is equal to 2, and ending with the final state, calculated The value found for the final state are the minimal losses of system failure. Remember these values and numbers of checks k*, in which they are obtained.

Successively perform the following actions, starting from the initial state. The check fixed at the first stage, take as a first check of the desired optimal algorithm for fault finding. Determine the states originated at its negative and positive results. The checks found in the first stage for these states should be taken as a result of the following checks at the corresponding result of the previous check. Continue the process until all the final states are reached. As a result the desired algorithm A* will be obtained.

For illustration, we shall present a simple example.

Figure 1 illustrates a system of four blocks. The probabilities of failures recorded over the blocks, and duration of checks in minutes are indicated over the corresponding to the check outputs of the blocks. Factor r = 0.25. Table 1 shows the possible states, the corresponding values of $L^*(S)$ and numbers of checks k *. Let us explain the computational scheme of the first stage using the example of the state of S(2,3,4). After performing the check P^2 the system transfers into the state S(3,4) or into the final state S(2). Accordingly $L(S,P^2)$ = exp(0,25*8)*(0,25 + 1,12) = 10,13.

For P^3 checking we obtain $L(S,\Pi^3)$ = 13,69. Since $L(S,\Pi^2) < L(S,P^3)$, then in accordance with (1) we accept $L^*(S)$ = 13,69 and select the check $P^{2.}$.Then we memorize the values in Table 1. Let us explain the procedure for the second stage. Let us consider the initial state. We adopt its corresponding check P^3 as the first check. The checks found in the first stage for the states S(1,3) and S 2,4), we take as the next checks, at the negative and positive results.

The optimal algorithm is designed and shown in Figure 1 in the form of a binary tree. Vertices correspond to the checks, and arcs correspond to their results, and leaves to malfunctions. It corresponds to the minimum average losses L $(A^{*)}$ = 46.01. Note that replacing in (1) the minimum by the maximum, we can find the maximum losses and the worst corresponding algorithm. For this example, max L (A) = 172.92 (!).

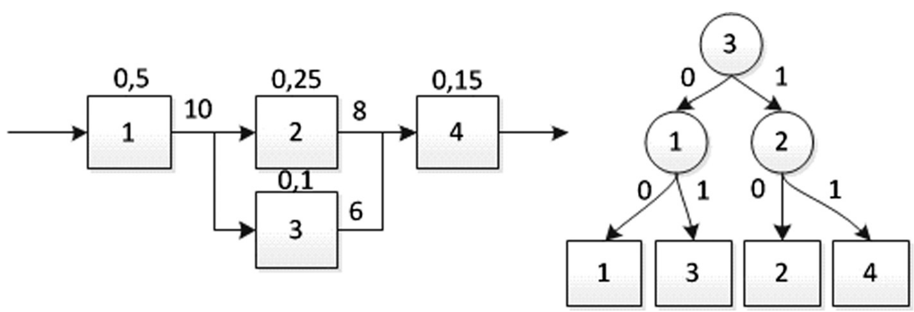

Fig. 1. Example: the system and the algorithm of fault finding

Table 1. An example of calculations

‖S‖	S	k	$S(\pi^{\kappa}_{0})$	$S(\pi^{\kappa}_{1})$	$L(S,P^k)$	L*(S)	k*
2	1.2	1	1	2	9.14	9.14	1
	1.3	1	1	3	7.31	7.31	**1**
	2.4	2	2	4	2.96	2.96	**2**
	3.4	3	3	4	**1.12**	**1.12**	3
3	2,3,4	2	2	3.4	10.13	10.13	2
		3	3	2.4	13.69		
4	1,2,3,4	1	1	2,3,4	129.45	**46.01**	**3**
		2	1.2	3.4	75.79		
		3	1.3	2.4	46.01		

3. The conclusion

For the first time the method of constructing optimal algorithms of fault finding for systems, whose failures lead to losses grow exponentially with increasing recovery time. Such a situation is typical for railway automation systems. To apply the method, special software has been developed. The result can also be used for diagnostic of responsible projects. [3]

References

1. **Sapozhnikov Vl. V., Sapozhnikov V.V.** Fundamentals of technical diagnostics – M.: Marshrut, 2004.

2. **Pernikis B.D., Yagudin R.Sh.** Prevention and elimination of faults in signaling systems. – M.: Transport, 1994.

3. **Maron A.I., Maron M.A.** Information approach to the organization of project control // Business – InformaticsIT – 2012. – # 4. – pp. 54 – 60.