**Briš Radim**
*Technical University of Ostrava, Ostrava, Czech Republic*

# Direct unavailability calculations of highly reliable systems

### Keywords
highly reliable system, unavailability, acyclic graph

### Abstract
The paper presents a new analytical algorithm which is able to carry out direct and exact reliability quantification of highly reliable systems with maintenance (both preventive and corrective). A directed acyclic graph is used as a system representation. The algorithm allows take into account highly reliable and maintained input components. All considered models are implemented into the new algorithm. The algorithm is based on a special new procedure which permits only summarization between two or more non-negative numbers that can be very different. If the summarization of very small positive numbers transformed into the machine code is performed effectively no error is committed at the operation. Reliability quantification is demonstrated on a real system from practice and on its highly reliable modifications. The selected system is frequently used system - high pressure injection system occurring in many late references.

## 1. Introduction

It is a Monte Carlo simulation method [1] which is used for the quantification of reliability when accurate analytic or numerical procedures do not lead to satisfactory computations of system reliability. Since highly reliable systems require many Monte Carlo trials to obtain reasonably precise estimates of the reliability, various variance-reducing techniques [2], eventually techniques based on reduction of prior information [3] have been developed. A direct simulation technique has been improved by the application of a parallel algorithm [4] to such extent that it can be used for real complex systems which can be then modelled and quantitatively estimated from the point of view of the reliability without unreal simplified conditions which analytic methods usually expect. If a complex system undergoes a fault tree (FT) analysis, new and fast FT quantification method with high degree of accuracy for large FTs, based also on a Monte Carlo simulation technique and truncation errors was introduced in [5]. However, if it is necessary to work and quantitatively estimate highly reliable systems, for which unreliability indicators (i.e. system non-functions) move in the order $10^{-5}$ and higher (i.e.$10^{-6}$ etc.), the simulation technique, whatever improved, can meet the problems of prolonged and inaccurate computations.

Quantitative analysis of FT can be performed by the binary decision diagram method (BDD). The BDD algorithm provides an exact top event probability where no truncation or approximation is employed. The only error which can occur is due to numerical operations. The results in [6] show that the BDD algorithm is best method with minimal relative error in comparisons with the classical Sum-of-Product algorithm, i.e. the classical minimal cut sets/rare event approach applied for selected unreliability approximations (Murchland, Barlow-Proschan lower bound and Vesely approximations). The relative error curve of the classical Sum-Of-Products algorithm is always above the error curve of the BDD algorithm. The authors conclude that this is due to the rare-event approximation, which is optimistic for given unreliability approximations. The distance between the two curves decreases as the probabilities of basic events decrease. The authors further supposed that the errors can be strongly impacted by rounding errors. This paper is oriented just on removing of the rounding errors when a highly reliable system has to be quantified from reliability point of view.

Highly reliable systems appear more often in research practice and they are closely connected with a penetrative increase of progress. We can observe the systems for example in space applications where complex device is often designed for an operation lasting very long time without a possibility of help of human hand. Computer hardware and software have become an integral part of many sophisticated and complex systems, such as systems for space exploration. This trend has been the motivation for research efforts to improve software reliability and performance by introducing redundancy in computing hardware and/or software [7]. Safety systems of nuclear power stations represent other example of highly reliable systems. They have to be reliable enough to comply with still increasing internationally agreed safety criteria and moreover they are mostly so called sleeping systems which start and operate only in the case of big accidents. Their hypothetical failures are then not apparent (hidden failures) and thus reparable only at optimally selected inspective times. NEC producer [8] has offered system components for use in supercomputers in order to support the need for an extremely high performance as well as in car-mount systems in which reliability over a wide temperature range is a critical feature. The improving reliability will be an essential feature of the products of the next generation.

The question solved in the paper is how to model the reliability behaviour of these systems and how to find an efficient procedure for computation of reliability indicators. The rest of the paper is organized in the following way: Section 2 brings basic problem formulation and all admissible models of input components. Section 3 describes basic principles of the new algorithm which is based on error-free summation of different non-negative numbers. Details of the algorithm are explained in a few steps demonstrated by figures. Application of the algorithm is showed in context with a system, assigned by the help of acyclic graph. Section 4 brings results with tested system computed by the use of the new algorithm. Highly reliable modifications of the system from practice have been calculated to emphasize merits of the algorithm. Section 5 summarizes the most important findings.

## 2. A problem formulation and component models

Let us have a system assigned with a help of a directed acyclic graph (AG) [4]. Terminal nodes of the AG that represent functionality of input system components are established by the definition of deterministic or stochastic process, to which they are subordinate. From them we can compute a time course of the availability coefficient, possibly unavailability of individual terminal nodes, using methodology of basic renewal theory, as for example in [9]. The aim is then to find a correspondent time course of the unavailability coefficient for the highest SS node which represents reliability behaviour of the whole system.

### 2.1. Models of components – terminal nodes

In the first phase of research, an exponential distribution for the time to a failure will be supposed, possibly for the time to a restoration. Under this condition, all frequently used models with both preventive and corrective maintenance may be described by three of the following models:

Model with elements (terminal nodes in AG) that can not be repaired

Model with repairable elements (CM – Corrective Maintenance) for apparent failures, i.e. a model when a possible failure is identified at the occurrence and immediately afterwards it starts a process leading to its restoration.

Model with repairable elements with hidden failures, i.e. a model when a failure is identified only at special deterministically assigned times, appearing with a given period (moments of periodical inspections). In the case of its occurrence at these times an analogical restoration process starts, as in the previous case.

An analytical accurate computation of time dependence of the (un)availability coefficient was for the first two situations explained enough and derived in [10]. Let us remind that in the first case of the element that can not be repaired a final course of unavailability coefficient *P(t)* is presented by a distribution function of time to failure of the element:

$$P(t) = 1 - e^{-\lambda t}, \qquad (1)$$

where $\lambda$ is the failure rate.

In the second case we can derive a relation on the basis of Laplace´s transformation for a similar coefficient

$$P(t) = 1 - \left[ \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)t} \right] = \frac{\lambda}{\lambda + \mu}\left[ 1 - e^{-(\lambda+\mu)t} \right], \quad t > 0 \qquad (2)$$

where $\mu$ is the repair rate.

The third model is rather more general than the earlier stated in the model with periodical exchanges [10] where we assumed a deterministic time to repair. If we hereby presume that a time to the end of repair is exponential random variable, it is necessary to derive an analytical computation of time course of the function of unavailability coefficient.

## 2.2. Unavailability coefficient for a model with repairable elements and hidden failures

With the same indication of failure and restoration intensities as given above we can describe the unavailability coefficient with the following function:

$$P(\tau) = (1 - P_C).(1 - e^{-\lambda\tau}) + P_C\left[1 + \frac{\mu}{\mu - \lambda}(e^{-\mu\tau} - e^{-\lambda\tau})\right], \quad \tau > 0 \tag{3}$$

where $\tau$ is a time which has passed since the last planned inspection, *Pc* is the probability of a non-functional state of an element at the moment of inspection at the beginning of the interval to the next inspection.
Proof of the relationship (3) brings ref. [12].

Note:

1. For the purposes of an effective computer calculation (see Section 3) the expression in the brackets can be converted into the formation:

$$\left[1 + \frac{\mu}{\mu - \lambda}(e^{-\mu\tau} - e^{-\lambda\tau})\right] = 1 - \frac{\mu}{\mu - \lambda}e^{-\lambda\tau}\left[1 - e^{-(\mu - \lambda)\tau}\right], \quad \tau > 0 \tag{4}$$

2. In other hypotheses we will need this expression to be always positive, what is also easy to proof.

## 3. The new algorithm

## 3.1. Probabilities of functional and non-functional state

It is evident that probabilities of a functional *p* and non-functional state *q* comply with a relation

$$q + p = 1.$$

Taking into consideration the final accuracy of real numbers in a computer it is important which one from *p* or *q* we count. If we want to fully use the machine accuracy, we have to compute the smaller one from both probabilities.

Example:

We take into account the following sum:

0.000 002 7816 + 0.999 997 218

If we counted hypothetically on a computer with three-digit decimal numbers, then for the value of $q$ = 0.00000278, we would instead of a correct value $p$ = 0.999997218 have only $p$ = 1**.**

In return for $q$ = 1- $p$, we would get: $q$ = 1- $p$ = 0, keeping at disposal $p$ = 1.

It is apparent that it gets to a great loss of accuracy if we counted *p* instead of *q*. Our result will be maximally precise saving accuracy of *q*.

Seeing that probabilities of a non-function state of a highly reliable system is very small, we have to concentrate on numerical expression of these probabilities. For these purposes it is necessary to reorganize the computer calculation and set certain rules which do not have the influence on accuracy of the computation at the numeration process.

## 3.2. Probability calculation of non-functional states of terminal nodes

The probability calculation of non-functioning state (unavailability coefficient) of the simplest possible not repaired element (or terminal node) can be done by the use of relation (1).

Similarly, for other models of system elements the computation of an expression

$$1 - e^{-x}, \tag{5}$$

for $x \geq 0$ is a crucial moment at probability numerical expression of a non-function (unavailability coefficient).

For values x<<1, i.e. near 0, direct numerical expression written by the formula would lead to great errors! At subtraction of two near numbers it gets to a considerable loss of accuracy. On personal computer the smallest number ε, for which it is numerically evident that

$$1 + \varepsilon \neq 1,$$

is approximately $10^{-18}$. If

$$x \approx 10^{-25},$$

the real value of the expression (5) will be near $10^{-25}$. A direct numerical calculation of the expression gets a zero!

As the algorithm was created in a programming environment Matlab, for the need of this paper was used the Matlab function “exmp1” which enables exact calculation of the expression (5) based on Taylor's decomposition.

### 3.3. The numeration substance of probability of a non-functional state of a node

The probability of a non-functional state of a node of an AG, for which the individual input edges are independent, is in fact given by going over all possible combinations of probabilities of the input edges. For 20 input edges we have regularly a million combinations.

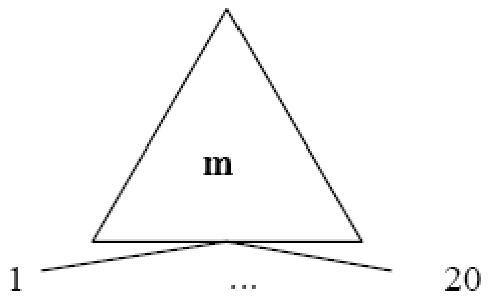



*Figure 1.* One node of the acyclic graph with 20 edges.

One partial contribution to the probability of a non-functional state of the node in *Figure1* has a form:

$$q_1 . q_2 \cdots q_{i-1} \, p_i . q_{i+1} \cdots q_{j-1} . p_j . q_{j+1} \cdots q_{20} \, ,$$

where a number of occurring probabilities *p* (here the number equals to 2) can not reach "m". The probability of a non-functional state of the node is generally given by a sum of a big quantity of very small numbers. These numbers are generally very different!
If the sum will be carried out in the way that the addition runs in the order from the biggest one to the smallest ones, certainly a lost stems from rounding off, more than the addition runs in the order from the smallest ones to the biggest values. And even in this second case there is not possible to determine effectively how much accuracy "has been lost".
Note: In the case of dependence of the input edges (terminal nodes) we cannot express the behaviour of an individual node numerically. There is necessary to work with the whole relevant sub-graph. Combinatorial character for the quantification will stay nevertheless unchanged.

### 3.4. The error-free sum of different non-negative numbers

The first step to the solution of this problem is to find a method for the "accurate" sum of many non-negative numbers.
The arithmetic unit of a computer (PC) works in a binary scale. A positive real number of today's PC contains 53 valid binary numbers, see *Figure 2.* A possible order ranges from approximately -1000 to 1000.

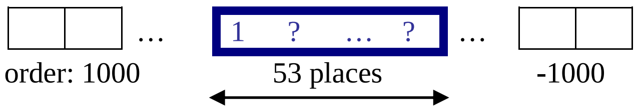



*Figure 2.* A positive real number in binary scale

The line indicated as "order" means an order of a binary number.
The algorithm for the "accurate" quantification of sums of many non-negative numbers consists from a few steps:

1. The whole possible machine range of binary positions (bites) is partitioned into segments of 32 positions for orders, according to the following scheme in *Figure 3.*

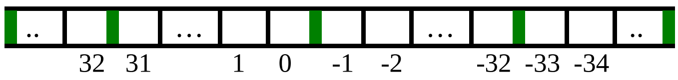



*Figure 3.* Segments composed from 32 binary positions

The number of these segments will be approx.:

$$\frac{2000}{32} \cong 63$$

2. Total sum is memorized as one real number, which is composed from 32 bite segments. Each from these segments has additional 21 bites used as transmission.

3. At first a given non-zero number of the sum that must be added is decomposed according to before assigned firm borders (step 1) mostly into three parts containing 32 binary numbers of the number at most, according to the scheme in *Figure 4.* The individual segments are indexed by numbers 1-63.

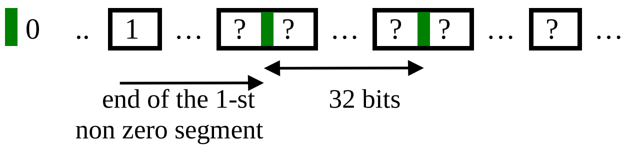



*Figure 4.* Decomposition of a given non-zero number

4. Then the individual parts of this decomposed number are added to the corresponding members of the sum number, as in *Figure 5.*

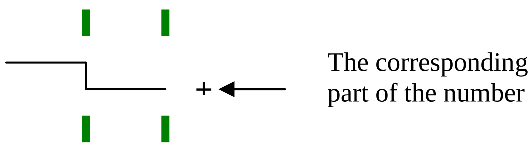



*Figure 5.* Adding a number to the sum number

5. Always after the processing of $2^{20}$ numbers (the limit is chosen so that it could not lead to overflowing of the sum number at any circumstances) a modification of the sum number is carried out which is indicated as the "clearance" process. Upwards a transmission is separated (in the following *Figure 6* it is identified by a symbol β) which is added to the upper sum.

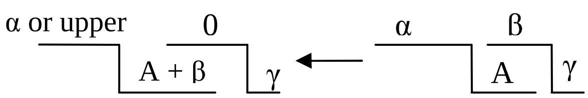



*Figure 6.* Clearance process

6. If a final sum is required, at first the clearance process has to be carried out. Then the group of three sums is elaborated, from which the upper is the highest non-zero one (identified by a symbol α in *Figure 7*). We make a sum of these three sums as usual in a binary scale, when *p* in the following expression is given by an index of the highest non-zero segment:

$$sum = \alpha.2^{p} + \beta.2^{p-32} + \gamma.2^{p-64}$$

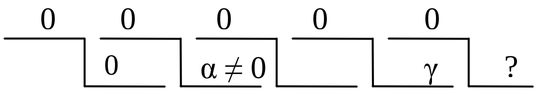



*Figure 7.* Demonstration of the final summarization

So numbers in their full machine accuracy (53 binary numbers beginning with 1) are the input for this process of adding. The output is the only number in the same machine accuracy (53 binary numbers beginning with 1). The number is mechanically the nearest number to the real accurate error-free sum which contains in principle up to 2000 binary numbers.

### 3.5. Permissible context of the usage not leading to the loss of accuracy

The probability of a non-functional state of a repairable component (repairable component with hidden failures) is given by the formula (3), which can be simplified as

$$P(\tau) = (1 - P_C).\alpha(\tau) + P_C.\beta(\tau),$$

where $P_C$ is the probability of a non-functional state of an element at the moment of the inspection at the beginning of the interval till the next inspection; $\alpha$, $\beta$ are non-negative mechanically accurate and numerically expressed functions.

One contribution to the computation of a non-functional state of a node generally has a form

$$q_1.q_2...(1 - q_k)....$$

In both cases occurs (*1- q*). It has already been explained that when we use $p \equiv 1- q$, it can come to the catastrophic loss of accuracy. A basic question then comes out: Do we have to modify further the stated patterns for the purpose of removing the subtraction? Fortunately not. In the introduced context of the product *(1- q).α*, where *α* is expressed numerically in a full machine accuracy there is no loss in machine accuracy! Thanks to rounding off the final product to 53 binary numbers, lower orders of the expression *(1-q)*, i.e. a binary numbers on 54th place and other places behind the first valid number, can not practically influence the result.

### 3.6. Determination of system probability behaviour according to a graph structure

Let all elements appearing in the system are independent. The probability of a non-functional state of a system, assigned by the help of AG, is thus simply gained on the basis of estimation of all nodes upwards. For instance for AG in *Figure 8* the following steps have to be made:

- numerical expression of the probability of a non-functional state of terminal nodes, i.e. elements 8,9,10 and 5,6,7.
- numerical expression of the probability of a non-functional state of an internal node 4 which is given by the following sum:

$$q_8.q_9.q_{10} + (1-q_8).q_9.q_{10}$$
$$+q_8.(1-q_9).q_{10} + q_8.q_9.(1-q_{10})$$
$$+q_8.(1-q_9).(1-q_{10})$$
$$+(1-q_8).q_9.(1-q_{10})$$
$$+(1-q_8).(1-q_9).q_{10}$$

- numerical expression of the probability of a non-functional state of an internal node 3 which is given by the only item

$$q_5.q_6.q_7$$

- numerical expression of the probability of a non-functional state of a terminal node 2
- numerical expression of the probability of a non-functional state of the highest SS node 1 which is given:

$$q_2.q_3.q_4 + (1-q_2).q_3.q_4$$
$$+q_2.(1-q_3).q_4 + q_2.q_3.(1-q_4)$$

In the case of AG with dependent elements, where every multiple used node causes dependence, the situation is much more complex. We have to decompose a set of nodes to a disjunctive system of mutually independent subsets. The process has been also implemented to the new algorithm.

*Note:* Internal numeration of nodes is such that the node with a less number can not be inferior to the node with greater number. Nodes are numbered in the decreasing order of numbers.

## 4. Results with tested systems

### 4.1. Tested system from reference

As a tested system on which the algorithm was applied, the HPIS (High Pressure Injection System) of a NPP was selected. A simplified HPIS of a Pressurized Water Reactor is shown in *Figure 9*, which has been adapted from the literature [13]. This system is normally in stand-by and consists of three pumps and seven valves organized as shown in *Figure 9*. Under accidental conditions the HPIS can be used to remove heat from the reactor in those events in which steam generators are unavailable. For example, in case of a Small-Break Loss-Of-Coolant Accident the HPIS safety function draws water from the Refueling Water Storage Tank (RWST) and must discharge it into the cold legs of the Reactor Cooling System through any of the two injection paths.

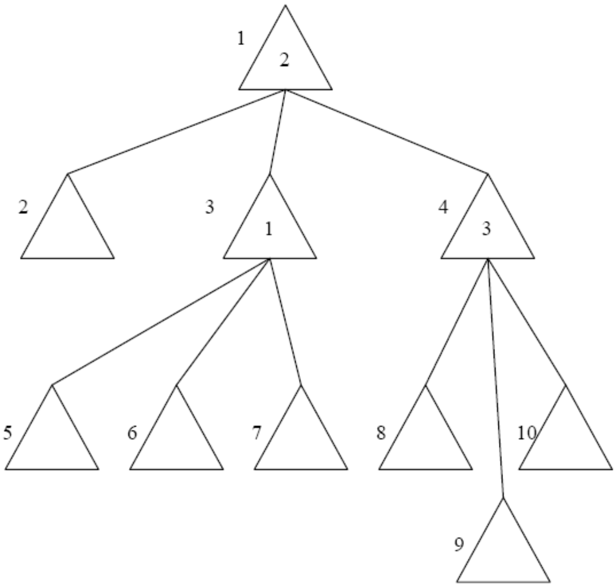



*Figure 8.* The system assigned by the help of a structure AG

Normally, pumps discharge into the injection paths A and B through valves 3 and 5, although crossover valves 4, 6 and 7 provide alternative flow paths in case of failure of the normal feed.

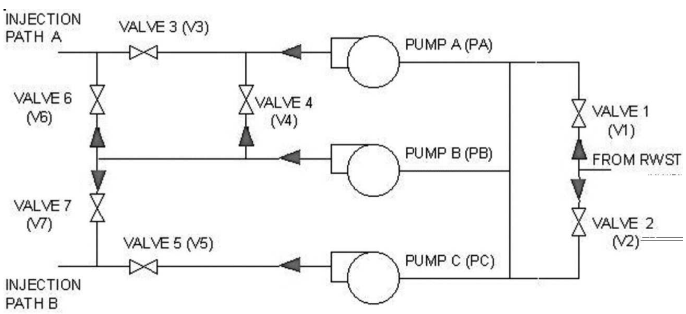



*Figure 9.* HPIS.

*Table 1* shows typical Test Intervals (TIs) requirements included within the HPIS Technical Specifications. In addition, the relevant component unavailability data for this case of application are adopted from [14].
*Table 2* shows a possible and initial distribution of TI and the time to first test, i.e. TP {TA, TB, TC, TD, TE, TF}, for the components of the HPIS grouped according to the applicable TI {T1, T2, T3}. For example, decision variables associated to pump PA correspond to the set {T1, TA}.

*Table 1.* Typical TI for the HPIS.

| Setting | Pumps (P) | Valves (V) |
|---|---|---|
| TI | 2184 | 2184 |

*Table 2.* Initial values for TI and TP.

| TI vs TP | T1 | T2 | T3 |
|---|---|---|---|
| PA | TA | | |
| PB | TB | | |
| PC | TC | | |
| V1 | TA | | |
| V2 | TC | | |
| V3 | TA | | |
| V4 | | TD | |
| V5 | TC | | |
| V6 | | | TE |
| V7 | | | TF |

In addition, the following relationships also apply, what concerns TI

T1=2184 hr, T2=3*T1, T3=3*T1,

and TP:

TA=24, TB=48, TC=72, TD=TB, TE=TB+T1,

TF=TB+2*T1.

It follows from *Table 2* and the several relationships shown above that the preventive maintenance policy of the system can be based on the following vector of decision variables

$$\mathbf{x} = \{\mathrm{TI, TA, TB, TC}\}$$

## 4.2. Assessment of initial conditions

First, it is worthy to assess the departure point in order to establish the basis for comparing results. Herein, initial conditions are represented by the particularization of the decision vector for the initial set of TI and TP as follows:

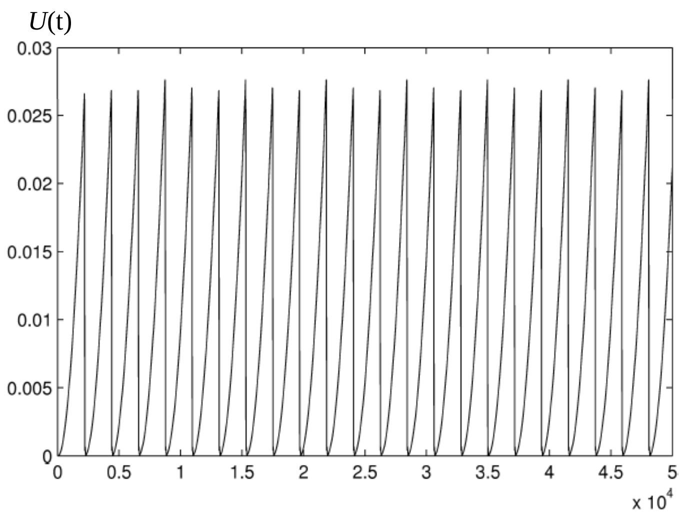
$$\mathbf{x} = \{2184, 24, 48, 72\}$$

*Figure 10* represent the evolution of the time dependent unavailability of the HPIS in the initial case. Basic reliability data have been a bit simplified in comparison with the reference [13], see *Table 3.*

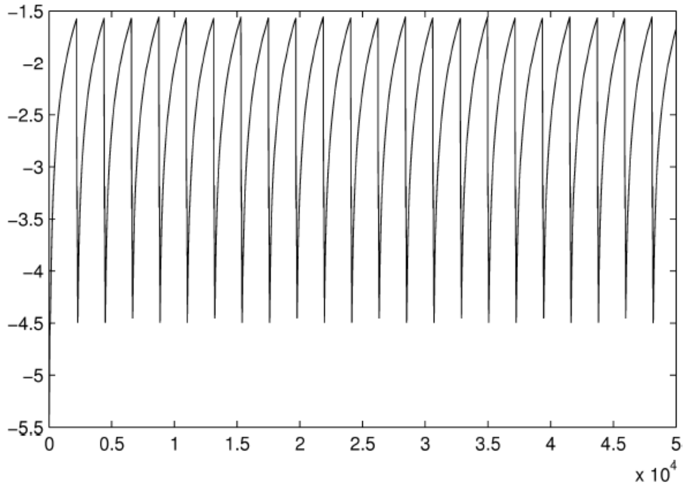
*Table 3.* HPIS-reliability data for Pumps (P) and Val ves (V).

| | $\lambda_j$(h$^{-1}$) | $\rho_j$ | $d_j$(h) |
|---|---|---|---|
| P | $3.89\times10^{-6}$ | $5.3\times10^{-4}$ | 24 |
| V | $5.83\times10^{-6}$ | $1.82\times10^{-3}$ | 2.6 |

$\lambda_j$ = failure rate of the *j*th component
$\rho_j$ = probability of failure on demand of the *j*th component
$d_j$ = mean down time of the *j*th component due to corrective maintenance.
The downtime of the *j*th component due to testing has been neglected as well as probability of human error. The purpose of the simplification is in that the algorithm is first of all intended for the highly reliable systems.
In the *Figure 5* one can see that maximal unavailability during mission time $T_M$ = 50.000 hours is a value between 0.025 and 0.03.




Log *U*(t)

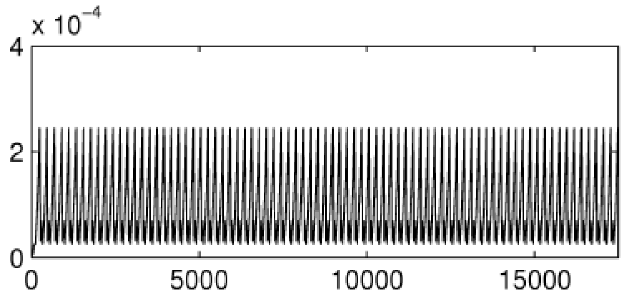



Chronological time [hours]

*Figure 10.* HPIS Time-dependent unavailability *U*(t)and decimal logarithm of the unavailability. Initial case.

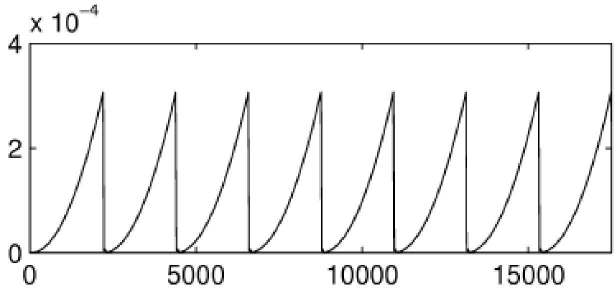
### 4.3. Highly reliable system modifications

As a first modification the same system has been taken into account with 10 times shorter TIs. This modification results in unavailability improvement to about two orders what is demonstrated in *Figure 11*. Maximal unavailability is 2.4 $\times 10^{-4}$. Similar results can be reached by 10 times improved $\lambda_j$ and $\rho_j$ what is demonstrated in *Figure 12*, where maximal unavailability does not overstep the value of 3.1 $\times 10^{-4}$. Final modification with 10× improvements of both TIs and $\lambda_j$ , $\rho_j$ we can see on *Figure 13*, where maximal unavailability is bellow the value of 2.5 $\times 10^{-6}$.



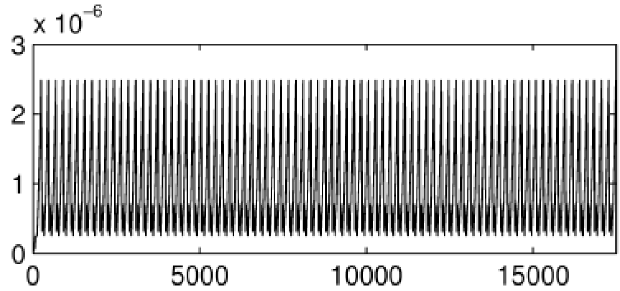

Chronological time [hours]

*Figure 11.* HPIS Time-dependent unavailability *U*(t) within first 16 600 hours. Modification with 10 times shorter TIs.




Chronological time [hours]

*Figure 12.* HPIS Time-dependent unavailability *U*(t) within first 16 600 hours. Modification with 10 times improved $\lambda_j$ and $\rho_j$ .




Chronological time [hours]

*Figure 13.* HPIS Time-dependent unavailability *U*(t) within first 16 600 hours. Modification with modification with 10× improvements of both TIs and $\lambda_j$ , $\rho_j$.

### 5. Conclusion

Maintaining the full machine accuracy requires mainly not to carry out subtraction of near values. All required outputs are therefore necessary to express in the form of the sum of numbers with consistent sign (in our case non-negative).
A problem of a sum of many non-negative numbers can be solved by decomposing a sum into more partial sums which can be carried out without a loss! The process has been numerically realized within a programming environment Matlab.
Numerical expression of probabilities of a non-functional state of one node of an AG has a combinatorial character. We have to go over all combinations of input edges behaviour leading to a non-functional state of the node. The astronomic increase of combinations with the increasing number of elements causes that the program will be usable only up to a certain size of a system. Already at moderate exceeding the critical size of the system it comes to enormous increase of machine time. All computations above run below 1s, on Pentium (R) 4 CPU 3.40GHz, 2.00 GB RAM.
The algorithm enables to carry out exact unavailability analysis of real maintained systems with both preventive and corrective maintenance.

The future research will continue with the aim to use the algorithm for maintenance optimization, i.e. to find such a maintenance strategy to minimize the maintenance cost at a prescribed maximal unavailability level.

## Acknowledgements

This work is supported by The Ministry of Education, Youth and Sports of the Czech Republic - project CEZ MSM6198910007.

## References

[1] Marseguerra, M, & Zio, E. (2001). *Principles of Monte Carlo simulation for application to reliability and availability analysis*. In: Zio E, Demichela M, Piccinini N, editors. Safety and reliability towards a safer world, Torino, Italy, September 16–20, 2001. Tutorial notes. p. 37–62.

[2] Tanaka, T, Kumamoto, H. & Inoue, K. (1989). Evaluation of a dynamic reliability problem based on order of component failure. *IEEE Trans Reliab* 38,573–6.

[3] Baca, A. (1993). Examples of Monte Carlo methods in reliability estimation based on reduction of prior information. *IEEE Trans Reliab* 42(4), 645–9.

[4] Briš, R. (2008). Parallel simulation algorithm for maintenance optimization based on directed Acyclic Graph. *Reliab Eng Syst Saf* 93, 852-62.

[5] Choi, J.S. & Cho, N.Z. (2007). A practical method for accurate quantification of large fault trees. *Reliab Eng Syst Saf* 92, 971-82.

[6] Dutuit, Y. & Rauzy, A. (2005). Approximate estimation of system reliability via fault trees. *Reliab Eng Syst Saf* 87, 163-72.

[7] Rahnamai, K.R., Caglayan, A.K. & Stubbs, C.L. (1989). *Enforced software diversity for a tracking application*. Final Report No. R8908 for NASA, Contract No. NAS1-17705.

[8] Mochizuki, Y., Hayashi, Y., Oda, N., Takeuchi, K. & Takeda, K. (2006). Technology for High Reliability System LSIs. *NEC Technical Journal* 1, No3.

[9] Briš, R. (2007). *Stochastic Ageing Models – Extensions of the Classic Renewal Theory. Reliability: Theory & Applications*, ISSN 1932-2321, 2(3-4), 19-27.

[10] Briš, R. & Drábek, V. (2007). *Mathematical Modeling of both Monitored and Dormant Failures*. In: Lisa Bartlett, editor. Advances in Risk and Reliability Technology Symposium. Published by Loughborough University , 376-393.

[11] Dutuit, Y. & Chatelet, E. (1997). *TEST CASE No. 1, Periodically tested parallel system. Test-case activity of European Safety and Reliability Association.* ISdF-ESRA 1997. In: Workshop within the European conference on safety and reliability ESREL 1997, Lisbon.

[12] Bris, R. (2008). Exact reliability quantification of highly reliable systems with maintenance. *Safety, Reliability and Risk Analysis: Theory, Methods and Applications* – Martorell et al. (eds), Taylor & Francis Group, pg. 489-496, ISBN 978-0-415-48513-5.

[13] Harunuzzaman, M. & Aldemir, T. (1996). Optimization of standby safety system maintenance schedules in nuclear power plants. *Nuclear Technology* 113, 354–67.

[14] Martorell, S. Carlos, S., Villanueva J.F., Sanchez, A.I., Galvan, B., Salazar, D. & Cepin, M., (2006). Use of multiple objective evolutionary algorithms in optimizing surveillance requirements. *Reliability Engineering and System Safety* 91, 1027–1038.