Survivability of series–parallel systems with multilevel protection

Edward Korczak\textsuperscript{a}, Gregory Levitin\textsuperscript{b,}\textsuperscript{*}, Hanoch Ben Haim\textsuperscript{b}

\textsuperscript{a}Telecommunications Research Institute, Poligonowa St., 30, 04-051 Warsaw, Poland
\textsuperscript{b}Development and Technology Division, Reliability Department, Planning, Bait Amir, The Israel Electric Corporation Ltd, P.O. Box 10, Haifa 31000, Israel

Received 31 August 2004; accepted 13 October 2004

Abstract

In this paper, we consider vulnerable systems which can have different states corresponding to different combinations of available elements composing the system. Each state can be characterized by a performance rate, which is the quantitative measure of a system’s ability to perform its task. Both the impact of external factors (attack) and internal causes (failures) affect system survivability, which is determined as the probability of meeting a given demand.

In order to increase the system’s survivability a multilevel protection can be applied to its subsystems. In such systems, the protected subsystems are destroyed by external impacts only if all of the levels of their protection are destroyed.

The paper describes an algorithm for evaluating the survivability of series–parallel systems with arbitrary configuration of multilevel protection. The algorithm is based on a composition of Boolean and the Universal Generating Function techniques. The adaptation of the algorithm for numerical implementation is suggested.

Illustrative examples are presented.

© 2004 Published by Elsevier Ltd.

Keywords: Survivability; Multilevel protection; Multi-state system; Universal generating function

1. Introduction

Survivability, the ability of a system to tolerate intentional attacks or accidental failures or errors, is becoming especially important when a system operates in battle conditions or is affected by a corrosive medium or another hostile environment.

When applied to multi-state systems, the survivability depends on a system’s ability to meet the demand (the required performance level). In this case, the outage effect will be essentially different for units with different nominal performances and will also depend on demand. Therefore, the performance rates (productivity or capacity) of system elements should be taken into account as well as the level of demand when the entire system’s survivability is estimated.

Numerous studies have been devoted to estimating the impact of external factors on the system’s survivability based on the common cause failure approach [1–10]. These studies consider systems with identical elements (\(k\)-out-of-\(n\) formulation) and do not take into account varying element performance rates. In recent works [11–16], the effect of the external impact on the survivability of different types of multi-state systems was studied.

In real systems built according to the defense-in-depth methodology [17], a multilevel protection is often used. The multilevel protection means that a subsystem and its inner level protection are in their turn protected by the protection of the outer level. This double-protected subsystem has its outer protection and so forth. In such systems, the protected subsystems can be destroyed by external impacts only if all of the levels of their protection are destroyed.

In [16], a recursive algorithm for evaluating the survivability of series–parallel multi-state systems with multilevel protection was suggested. This algorithm presumed that any group of all elements having the same protection (protection group) composes a series–parallel structure. It also presumes that different protection
groups cannot overlap (if element from PG A belongs also to PG B then either all of the elements from PG A belong to PG B or all of the elements from PG B belong to PG A). These assumptions do not hold in many real systems (for example different types of relay protections in electrical systems can be implemented in a way that contradicts the assumptions).

In this paper, we present an algorithm that evaluates the survivability of series–parallel systems with arbitrary structure of multilevel protection. This algorithm is based on technique different from one used in [16].

2. Model

2.1. Acronyms

MSS multi-state system
PG protection group

2.2. Notation

\( n \) number of elements composing MSS
\( M \) number of protections in MSS
\( x_m \) Boolean variable indicating state of protection \( m \) (\( x_m = 1 \) if protection is destroyed)
\( x = (x_1, \ldots, x_M) \) vector of protection states
\( v_m \) vulnerability of \( m \)th protection given it is exposed to an impact: \( v_m = \Pr\{x_m = 1\} \)
\( v = (v_1, \ldots, v_M) \) vector of protection vulnerabilities
\( \Theta_j \) set of numbers of protections that protect element \( j \)
\( G_j \) random performance rate of MSS element \( j \)
\( g_{jk} \) performance rate of MSS element \( j \) at state \( k \)
\( p_{jk} \) probability that MSS element \( j \) is in state \( k \) given it is not affected by external impacts
\( K_j \) number of different states of basic element \( j \)
\( G^* \) random output performance rate of the entire MSS
\( S \) survivability of the entire MSS
\( w \) minimal allowable level of MSS performance
\( u(z) \) \( u \)-function representing performance distribution of \( j \)th basic element given it is not affected by external impacts
\( U_j(z, x) \) \( u \)-function representing performance distribution of \( j \)th basic element as function of the protection states \( x \)
\( \delta(U(z, w)) \) operator over MSS \( u \)-function which determines probability \( \Pr\{G^* \geq w\} \)
\( \otimes \) composition operator over \( u \)-functions
\( \phi_{\text{ser}} \) structure function corresponding to series connection of elements
\( \phi_{\text{par}} \) structure function corresponding to parallel connection of elements

2.3. Assumptions

The multi-state system (MSS) consists of \( n \) basic elements (the lowest-level parts of the system) composing a series–parallel structure in a reliability logic-diagram sense.

The functioning of each element \( j \) is characterized by its random performance \( G_j \). The element can have \( K_j \) different states (from total failure up to perfect functioning) with performance rates \( g_{jk} (1 \leq k \leq K_j) \). In a state of total failure, the element performance rate is equal to zero (\( g_{j1} = 0 \)). The performance distribution of each element when it is not affected by external impacts is given as:

\[
\Pr\{G_j = g_{jk}\} = p_{jk}, \quad \sum_{k=1}^{K_j} p_{jk} = 1 \quad \text{for } 1 \leq j \leq n. \quad (1)
\]

Single elements or groups of elements can be protected. All the elements having the same protection compose a protection group (PG).

Any protection group or its part can belong to another protection group.

Each protection can be destroyed by external impacts with a given probability further referred to as protection vulnerability.

Any element is destroyed by external impact if and only if all of the protections of the protection groups that it belongs to are destroyed.

The performance of any destroyed element is equal to zero.

The element destruction caused by an external impact and its transitions in the space of states caused by failures and repairs are independent events.

The random performance rate of the entire MSS \( G^* \) depends on the nature of the elements’ interaction in the system and on the distribution of the elements’ performance.

The system survives if its performance rate is not less than the minimal allowable level \( w \). The MSS survivability is the probability that the system survives:

\[
S = \Pr\{G^* \geq w\}. \quad (2)
\]

2.4. Model interpretation

The model is based on probabilities of element and protection states. These probabilities can be usually elicited from statistics. The interpretation of the model depends on definitions of these probabilities and definition of the entire system survivability.

If the system survivability is defined as the probability that the system tolerates a single impact, \( p_j \) should be interpreted as probability that element \( j \) is in state \( k \) at the moment when the external impact occurs. The protection vulnerability is the probability that the protection is destroyed by the impact.
If the system survivability is defined as the probability that the repairable system tolerates multiple impacts during its mission time, the steady state probability \( p_{jk} \) can be interpreted as the time-average fraction of time when element \( j \) operates at performance level \( k \) given the element is not affected by external impacts. The protection vulnerability is the probability that the protection fails to prevent impacts (either because it is destroyed by previous impacts and not restored yet or because it is destroyed by a given impact). The situation when the system is subject to external impacts for a fraction \( f_{\text{imp}} \) of its mission time can be modeled by introducing additional protection that protects the entire system and has vulnerability \( f_{\text{imp}} \).

According to the assumptions any element is destroyed if all its protections are destroyed. The situation when element can survive (with a given probability \( p_{\text{surv}} \)) destruction of all its protections can be modeled by introducing additional protection that directly protects this element and has vulnerability \( 1 - p_{\text{surv}} \).

### 3. MSS survivability evaluation based on the universal generating function method

The procedure used in this paper for the system survivability evaluation is based on the universal generating function (\( u \)-function) technique, which was introduced in [18] and which proved to be very effective for the reliability evaluation of different types of multi-state systems [11–16, 19,20].

#### 3.1. \( u \)-functions of individual elements and their compositions

The \( u \)-function of a discrete random variable \( Y \) is defined as a polynomial

\[
u(z) = \sum_{k=1}^{K} q_k z^{y_k},
\]

where the variable \( Y \) has \( K \) possible values and \( q_k \) is the probability that \( Y \) is equal to \( y_k \). To evaluate the probability that the random variable \( Y \) is not less than the value \( w \) the coefficients of polynomial \( u(z) \) should be summed for every term with \( y_k \geq w \):

\[
\Pr\{Y \geq w\} = \sum_{y_k \geq w} q_k.
\]

This can be done using the following \( \delta \) operator over \( u(z) \)

\[
\delta(u(z), w) = \delta \left( \sum_{k=1}^{K} q_k z^{y_k}, w \right) = \sum_{k=1}^{K} \delta(q_k z^{y_k}, w),
\]

where for the individual term \( q_k z^{y_k} \):

\[
\delta(q_k z^{y_k}, w) = \begin{cases} q_k, & y_k \geq w \\ 0, & y_k < w. \end{cases}
\]

In our case, the polynomial \( u(z) \) can define performance distributions, i.e. it represents all of the possible mutually exclusive states of the element (or system) by relating the probabilities of each state to the performance of the element in that state. Note that the performance distribution of the basic element \( j \) defined by the vectors \( \{ g_{jk}, 1 \leq k \leq K_j \} \) and \( \{ p_{jk}, 1 \leq k \leq K_j \} \) can now be represented as

\[
u_j(z) = \sum_{k=1}^{K_j} p_{jk} z^{g_{jk}}.
\]

To obtain the \( u \)-function of a subsystem containing two elements, composition operators are introduced. These operators determine the \( u \)-function for two elements connected in parallel and in series, respectively, using simple algebraic operations on the individual \( u \)-functions of basic elements. All the composition operators take the form

\[
u_u(z) \otimes u_j(z) = \sum_{k=1}^{K_p} \sum_{h=1}^{K_j} p_{jk} z^{g_{jk}} \otimes \sum_{k=1}^{K_h} \sum_{h=1}^{K_j} p_{jh} z^{g_{jh}} = \sum_{k=1}^{K_p} \sum_{h=1}^{K_j} p_{jk} p_{jh} z^{g_{jk}+g_{jh}}
\]

The obtained \( u \)-function relates the probability of each state of a subsystem (equal to the product of the probabilities of states of its independent elements) to the performance rate of the subsystem in this state. The function \( \phi(\cdot) \) in composition operators expresses the entire performance rate of the subsystem consisting of two elements connected in parallel or in a series in terms of the individual performance rates of the elements. The definition of the function \( \phi(\cdot) \) strictly depends on the physical nature of the system performance measure and on the nature of the interaction of the system elements. In [19] two types of MSS are considered. For the sake of simplicity we consider here only those MSS in which the performance measure is defined as productivity or capacity (continuous materials or energy transmission systems, manufacturing systems, power supply systems). To apply the suggested method to other types of MSS one has only to choose the corresponding functions \( \phi(\cdot) \) [19,20].

In MSS of the considered type, the total performance rate of a pair of elements connected in parallel is equal to the sum of the performance rates of the individual elements. When the elements are connected in series, the element with the lowest performance rate becomes the bottleneck of the subsystem. Therefore, for a pair of elements connected in series the performance rate of the subsystem is equal to
the minimum of the performance rates of the individual elements.

Therefore, the composition operators $\otimes_{\varphi_{pr}}$ and $\otimes_{\varphi_{sr}}$ defined for the parallel and series connections of a pair of elements, respectively, take the form

$$u_j(z) \otimes u_j(z) = \sum_{k=1}^{K} p_{j} \otimes_{\varphi_{pr}} \otimes_{\varphi_{sr}} p_{j} \otimes_{\varphi_{pr}} p_{j} \otimes_{\varphi_{sr}} (9)$$

and

$$u_j(z) \otimes u_j(z) = \sum_{k=1}^{K} p_{j} \otimes_{\varphi_{pr}} \otimes_{\varphi_{sr}} p_{j} \otimes_{\varphi_{pr}} p_{j} \otimes_{\varphi_{sr}} (10)$$

Note that any subsystem consisting of two elements can be considered as a single equivalent element with a performance distribution equal to the performance distribution of the subsystem (represented by $u$-function obtained by the corresponding composition operator over $u$-functions of the two elements).

3.2. $u$-Functions of protected elements and their compositions

Let $\Theta_j$ be the set of numbers of protections that protect element $j$ and $x_m$ be the state of protection $m$ ($x_m=1$ if protection $m$ is destroyed and $x_m=0$ if it survives). According to our assumptions element $j$ is not destroyed by external impacts if at least one of protections belonging to $\Theta_j$ survives. The performance distribution of element $j$ in this case is $u_j(z)$. If all of the protections belonging to $\Theta_j$ are destroyed, element $j$ is also destroyed and its performance distribution in this case can be represented by the $u$-function $z^0$. Using these considerations we can represent the element performance distribution as a function of the states of the protections as:

$$U_j(z, x) = \left[1 - \prod_{m\in\Theta_j} x_m\right] u_j(z) + \left(\prod_{m\in\Theta_j} x_m\right) z^0. (11)$$

In order to obtain the representation of the performance distribution of a pair of elements $i$ and $j$ as a function of the protection states one can apply the same composition operators (9) and (10).

$$U_j(z, x) \otimes U_j(z, x) = \left[1 - \prod_{m\in\Theta_j} x_m\right] u_j(z) + \left(\prod_{m\in\Theta_j} x_m\right) z^0$$

$$\times \otimes_{\varphi} \left[\left(1 - \prod_{m\in\Theta_j} x_m\right) u_j(z) + \left(\prod_{m\in\Theta_j} x_m\right) z^0\right]$$

$$= \left[1 - \prod_{m\in\Theta_j} x_m\right] u_j(z) + \left(\prod_{m\in\Theta_j} x_m\right) z^0$$

$$+ \left(\prod_{m\in\Theta_j} x_m\right) z^0 \otimes u_j(z)$$

$$+ \left(\prod_{m\in\Theta_j} x_m\right) z^0 \otimes u_j(z). (12)$$

Indeed, the Boolean expressions in each term of $U_j(z, x) \otimes U_j(z, x)$ represent condition of existence of the given combination of not destroyed elements and the composition operators over the corresponding $u$-functions represent the performance distributions of these combinations.

The subsystem consisting of elements $i$ and $j$ can be further treated as a single element $f$ having the performance distribution represented by the $u$-function $U_j(z, x) = U_j(z, x) \otimes_{\varphi} U_j(z, x)$. In general $u$-function of any subsystem obtained by the recursive application of the composition operators takes the form:

$$U_j(z) = b_{f1}(x)u_{f1}(z) + \ldots + b_{fF}(x)u_{fF}(z) (13)$$

where $b_{f}(x)$ is the Boolean function representing the condition of existence of $f$th combination of not destroyed elements, $u_{f}(z)$ is the $u$-function representing the performance distribution of this combination, $F$ is the total number of the possible combinations.

Observe that events when different combinations of not destroyed elements exist are mutually exclusive. It can be seen that for any realization of the random binary vector $x$ only one of functions $b_{f}(x)$ is not equal to zero.

The $u$-function of a system consisting of two subsystems $U_j(z, x)$ and $U_j(z, x)$ can be obtained as

$$U_j(z, x) \otimes U_j(z, x) = \left[b_{f1}(x)u_{f1}(z) + \ldots + b_{fF}(x)u_{fF}(z)\right] \otimes_{\varphi} \left[b_{f1}(x)u_{f1}(z) + \ldots + b_{fF}(x)u_{fF}(z)\right]$$

$$= \sum_{i=1}^{F} \sum_{k=1}^{E} \left[b_{f}(x)b_{f}(x)u_{f}(z)u_{f}(z)\right] (14).$$
Recursively applying composition operator (14) one finally obtains the \( u \)-function of the entire system \( U(z, x) \) in the form:

\[
U(z, x) = b_1(x)u_1(z) + \cdots + b_H(x)u_H(z). \tag{15}
\]

This \( u \)-function takes the form of sum of conditional system performance distributions corresponding to different combinations of not destroyed elements multiplied by Boolean functions that represent conditions of existence of the corresponding combinations.

Having the vulnerability of each protection \( m \nu_m = \Pr[x_m = 1] \) one can obtain probability of each combination of not destroyed elements as \( \Pr[b_j(x) = 1] \). Since the Boolean functions \( b_j(x) \) take a multi-linear form (any Boolean function can be transformed into this form) this probability can be obtained by replacing Boolean variables \( x_m \) with probabilities \( \nu_m \) in the functions, i.e. \( \Pr[b_j(x) = 1] = b_j(\nu(x)) \) [21].

Now we have probability of each combination \( k \) of not destroyed elements \( b_k(\nu) \) and conditional distribution of system performance for each combination \( u_k(z) \). The performance distribution of the entire system can be obtained as

\[
U(z) = U(z, \nu) = b_1(\nu)u_1(z) + \cdots + b_H(\nu)u_H(z). \tag{16}
\]

Applying the operator \( \delta \) (5) with the given demand over \( u \)-function \( U(z) \) one obtains the system survivability:

\[
S(w) = \delta(U(z), w). \tag{17}
\]

3.3. Algorithm for MSS survivability evaluation

Consecutively applying the composition operators and replacing pairs of elements by equivalent elements one can obtain the \( u \)-function representing the performance distribution of the entire system. The following recursive algorithm obtains the system survivability:

1. Obtain the sets of protection numbers \( \Theta_j \) for each element \( j \).
2. Obtain \( u \)-functions \( U_j(z, x) \) of all of the system elements using Eqs. (7) and (11).
3. If the system contains a pair elements connected in parallel or in a series replace this pair with an equivalent element with \( u \)-function obtained by \( \otimes \) or \( \circ \) operator using Eqs. (14) and (9) or (10), respectively.
4. If the system contains more than one element return to step 3.
5. Determine the \( u \)-function of the entire series–parallel system applying the procedure (16).
6. Obtain the system survivability for the given demand \( w \) by applying the operator (17) over the \( u \)-function representing the system performance distribution.

![Fig. 1. Example of MSS with four elements and three protections.](image)

3.4. Example of determining the system performance distribution

Consider a series–parallel system (Fig. 1) consisting of four multi-state elements and three different protections.

Following the step 1 of the algorithm obtain:

- \( \Theta_1 = \{1, 2, 3\} \), \( \Theta_2 = \{1\} \), \( \Theta_3 = \{2\} \), \( \Theta_4 = \{3\} \).

The \( u \)-functions of the elements according to (11) are:

\[
U_1(z, x) = (1 - x_1x_2x_3)u_1(z) + x_1x_2x_3z_c^0,
\]

\[
U_2(z, x) = (1 - x_1)u_2(z) + x_1z_c^0,
\]

\[
U_3(z, x) = (1 - x_1)u_3(z) + x_2z_c^0,
\]

\[
U_4(z, x) = (1 - x_3)u_4(z) + x_3z_c^0.
\]

Following step 3 of the algorithm replace elements 2 and 3 with equivalent element 5 (note that for the composition operators considered in this paper \( u(z) \otimes z_c^0 = z_c^0 \) and \( u(z) \circ z_c^0 = u(z) \) for any \( u(z) \)):

\[
U_5(z, x) = U_2(z, x) \otimes U_3(z, x)
\]

\[
= [(1 - x_1)u_2(z) + x_1z_c^0] \otimes [(1 - x_2)u_3(z) + x_2z_c^0]
\]

Replace elements 5 and 4 with equivalent element 6:

\[
U_6(z, x) = U_5(z, x) \otimes U_4(z, x) \circ [(1 - x_1)(1 - x_2)u_2(z)
\]

\[
\otimes u_3(z) + [(1 - x_1)x_2 + x_1(1 - x_2) + x_2z_c^0] \]

\[
\otimes [(1 - x_1)u_2(z) + x_2z_c^0] = [(1 - x_1)(1 - x_2)u_2(z)
\]

\[
\otimes u_3(z) + (1 - x_1)(1 - x_2)z_c^0]
\]

\[
\otimes [(1 - x_1)u_2(z) + x_2z_c^0]
\]

\[
= (1 - x_1)(1 - x_2)(1 - x_3)u_2(z) \otimes u_3(z) \otimes u_4(z)
\]

\[
+ [1 - (1 - x_1)(1 - x_2)(1 - x_3)]z_c^0.
\]
Replace elements 1 and 6 with equivalent element 7:

\[
U_f(z, x) = u_f(z, x) \otimes \bar{u}_f(z, x)
\]

\[
= [(1 - x_z x_2 x_3) u_t(1) + x_1 x_2 x_3 x^0] \otimes [(1 - x_1)(1 - x_2)]
\]

\[
\times (1 - x_3) u_t(2) \otimes u_t(3) \otimes u_t(4) + [1 - (1 - x_1)]
\]

\[
\times (1 - x_3)(1 - x_2) (1 - x_1) = (1 - x_1)(1 - x_2)(1 - x_3)
\]

\[
\times u_t(1) \otimes [u_t(2) \otimes u_t(3) \otimes u_t(4)]
\]

\[
+ (x_1 + x_2 + x_3 - x_1 x_2 - x_1 x_3 - x_2 x_3)
\]

\[
\times (1 - x_1) + x_1 x_2 x_3 x^0.
\]

Finally, we obtain the system \(u\)-function as:

\[
U(z) = U_f(z, v)
\]

\[
= (1 - v_1)(1 - v_2)(1 - v_3) u_t(1)
\]

\[
\otimes [u_t(2) \otimes u_t(3) \otimes u_t(4)]
\]

\[
+ [v_1 + v_2 + v_3 - v_1 v_2 - v_1 v_3]
\]

\[
- v_2 v_3 u_t(1) + v_3 v_2 v_1 x^0.
\]

4. Numeric technique for system survivability evaluation

4.1. Boolean representation of multi-linear forms

The technique presented in Section 3 presumes analytical derivation of Boolean functions \(b_f(x)\) in the multi-linear form. In order to implement the algorithm numerically we use a \(M\)-length binary string \(e = \{e(1), \ldots, e(M)\}\) to define any product of Boolean variables such that \(e(i) = 1\) if \(x_i\) is included into the product. In analogy with \(u\)-function (3) any product of non-negated binary variables \(\prod_{m \in \Theta_j} x_m\) can be represented by the \(s\)-function \(s^c\) in which the binary string \(c_j\) is defined as follows:

\[
c_j(i) = \begin{cases} 1, & i \in \Theta_j \\ 0, & i \notin \Theta_j \end{cases}
\]

such that

\[
\prod_{m \in \Theta_j} x_m = \prod_{i=1}^M x_{c_j(i)}.
\]

The essential property of such representation is that the product

\[
\prod_{m \in \Theta_j} x_m \prod_{k \in \Theta_j} x_k = \prod_{m \in \Theta_j \cup \Theta_j} x_m
\]

corresponds to \(s\)-function \(s^{c_j \cup c_j}\) where the binary string \(c_j \cup c_j\) is obtained by Boolean operator OR over binary strings \(c_j\) and \(c_j\).

Any algebraic sum of products of non-negated Boolean variables

\[
\sum_{k=1}^K q_k \prod_{m \in \Theta_j} x_m
\]

can be represented by \(s\)-function \(e(s)\) having the form

\[
e(s) = \sum_{k=1}^K q_k s^{e_k}.
\]

For example, expression \(1 - \prod_{m \in \Theta_j} x_m\) can be represented by the following \(e\)-function

\[
e(s) = s^0 - s^{e_1},
\]

where \(0\) is the \(M\)-length string of zeros.

It can be seen that for two Boolean functions \(b_i(x)\) and \(b_j(x)\) that take the multi-linear form (21) and are represented by \(e\)-functions \(e_i(s)\) and \(e_j(s)\) respectively, the Boolean function

\[
b_i(x) b_j(x) = \left( \sum_{k=1}^K q_k e_k(s) \right) \otimes \left( \sum_{h=1}^H \sum_{l \in \Theta_j} q_{hl} s_{hl}^{e_l} \right)
\]

\[
= \sum_{k=1}^K \sum_{h=1}^H \sum_{l \in \Theta_j} q_k q_{hl} s_{hl}^{e_k(e_l)}.
\]

(25)

For example if \(M=4\) and \(b_i(x) = (1 - x_1 x_2 x_3)\), \(b_j(x) = (1 - x_1 x_2 x_3)\), we obtain

\[
\epsilon_i(s) = s^{(0,0,0,0)} - s^{(1,1,1,0)}, \quad \epsilon_j(s) = s^{(0,0,0,0)} - s^{(1,1,1,0)}
\]

and

\[
\epsilon_i(s) \otimes \epsilon_j(s) = s^{(0,0,0,0)} - s^{(1,1,1,0)} - s^{(1,1,1,0)} + s^{(0,0,0,0)}
\]

\[
= s^{(0,0,0,0)} - s^{(1,1,1,0)}.
\]

which corresponds to Boolean function \((1 - x_2 x_3)\) which is equal to \(b_k(x) b_j(x)\).

Using the \(\epsilon\)-functions we can represent the general form (13) of element \(u\)-function \(U_f(z, x)\) as

\[
U_f(z, s) = \epsilon_{f_1}(s) \bar{u}_{f_1}(z) + \cdots + \epsilon_{f_P}(s) \bar{u}_{f_P}(z)
\]

(26)
and define the composition operator as follows

\[ U_f(z, s) \otimes U_f(z, s) = \prod_{i=1}^{F} \prod_{k=1}^{E} \left[ \epsilon_F(s) \tilde\epsilon_E(z) \right] \]

\[ = \left[ \epsilon_F(s) \tilde\epsilon_E(z) + \cdots + \epsilon_F(s) \tilde\epsilon_E(z) \right] \otimes \left[ \epsilon_F(s) \tilde\epsilon_E(z) \right] \]

\[ + \cdots + \epsilon_F(s) \tilde\epsilon_E(z) \]

\[ = \sum_{i=1}^{F} \sum_{k=1}^{E} \left[ \epsilon_F(s) \otimes \epsilon_F(s) \right] [\tilde\epsilon_E(z) \otimes \tilde\epsilon_E(z)]. \quad (27) \]

Having the system performance distribution function in the form

\[ U(z, s) = \sum_{h=1}^{H} \epsilon_h(s) \tilde\epsilon_h(z) \quad (28) \]

where

\[ \epsilon_h(s) = \sum_{k=1}^{K_h} q_k^{(h)} s^{x_k}, \quad (29) \]

and taking into account Eq. (19) one can obtain the \( \eta \)-function of the system in the form (16) using the following equation:

\[ U(z) = \sum_{h=1}^{H} \left( \sum_{k=1}^{K_h} q_k^{(h)} \prod_{m=1}^{M} v_m^{(h)n} \right) \tilde\epsilon_h(z). \quad (30) \]

In order to obtain the numeric algorithm for evaluating MSS survivability one can use the algorithm presented in Section 3.3 in which \( U_f(z, x) \) is replaced with \( U_f(z, s) \) and Eq. (30) is applied instead of Eq. (16).

4.2. Example of the numeric algorithm

Consider the system presented in Section 3.4 and apply the suggested technique.

The \( \eta \)-functions of the elements according to (26) are:

\[ U_1(z, s) = (s^{(0,0,0)} - s^{(1,1,1)}, 0) \]

\[ U_2(z, s) = (s^{(0,0,0)} - s^{(1,0,0)}, 0) \]

\[ U_3(z, s) = (s^{(0,0,0)} - s^{(1,0,0)}, 0) \]

\[ U_4(z, s) = (s^{(0,0,0)} - s^{(0,0,1)}, 0) \]

Following step 3 of the algorithm and using operator (27) we obtain:

\[ U_5(z, s) = U_5(z, s) \otimes U_5(z, s) \]

\[ = \left[ (s^{(0,0,0)} - s^{(1,0,0)}, 0) \right] \otimes \left[ (s^{(0,0,0)} - s^{(1,1,1)}, 0) \right] \]

\[ = \left[ (s^{(0,0,0)} - s^{(1,0,0)}, 0) \right] \otimes \left[ (s^{(0,0,0)} - s^{(1,1,1)}, 0) \right] \]

\[ = \left[ (s^{(0,0,0)} - s^{(1,0,0)}, 0) \right] \otimes \left[ (s^{(0,0,0)} - s^{(1,1,1)}, 0) \right] \]

\[ + \left[ (s^{(0,0,0)} - s^{(0,0,1)}, 0) \right] \otimes \left[ (s^{(1,0,0)} - s^{(1,1,0)}, 0) \right] \]

\[ + \left[ (s^{(0,0,0)} - s^{(0,0,1)}, 0) \right] \otimes \left[ (s^{(1,1,0)} - s^{(1,1,0)}, 0) \right] \]

\[ = \left[ (s^{(0,0,0)} - s^{(1,0,0)}, 0) \right] \otimes \left[ (s^{(0,0,0)} - s^{(1,1,1)}, 0) \right] \]

\[ + \left[ (s^{(0,0,0)} - s^{(0,0,1)}, 0) \right] \otimes \left[ (s^{(1,0,0)} - s^{(1,1,0)}, 0) \right] \]

\[ + \left[ (s^{(0,0,0)} - s^{(0,0,1)}, 0) \right] \otimes \left[ (s^{(1,0,0)} - s^{(1,1,0)}, 0) \right] \]

\[ = \left[ (s^{(0,0,0)} - s^{(1,0,0)}, 0) \right] \otimes \left[ (s^{(0,0,0)} - s^{(1,1,1)}, 0) \right] \]

\[ + \left[ (s^{(0,0,0)} - s^{(0,0,1)}, 0) \right] \otimes \left[ (s^{(1,0,0)} - s^{(1,1,0)}, 0) \right] \]

\[ + \left[ (s^{(0,0,0)} - s^{(0,0,1)}, 0) \right] \otimes \left[ (s^{(1,0,0)} - s^{(1,1,0)}, 0) \right] \]
Using Eq. (30) one obtains
\[ U(z) = (1 - v_1 - v_2 + v_1 v_2 - v_3 + v_1 v_3 - v_1 v_2 + v_1 v_3) \]

\[ - v_1 v_2 v_3 u_1(z) \otimes (u_2(z) \otimes u_3(z) \otimes u_4(z)) + [v_1 + v_2] \]

\[ + v_3 - v_1 v_2 - v_1 v_3 - v_2 v_3 u_1(z) + v_3 v_2 v_1 \]

which is equal to the result obtained in Section 3.4.

5. Simplification technique

Collecting the like terms in the \( e \)- and \( u \)-functions obtained by applying the composition operators (23) one reduces the length of these functions and truncates the computations. However, additional computational complexity reduction can be obtained by implementing simple rules that can be easily derived using the Boolean nature of variables \( x_i \) (the fact that \( x_i x_j = x_j \)). These rules presented here in the form of Boolean expressions can be easily implemented in the composition operator over \( e \)-functions.

1. \( x_i (1 - x_j) = 0 \);
2. \( b(x) b(f(x)) = b(x) \) and \( b(x)(1 - b(x)) = 0 \) for arbitrary

   Boolean function \( b(x) \)

   example: \( x_1 x_3 - x_5 = (1 - x_1 x_3) + x_5 = 0 \);

3. \( \begin{cases} (1 - \prod_{m \in \Theta} x_m) (1 - \prod_{m \in \Theta} x_m) = (1 - \prod_{m \in \Theta} x_m) \ 	ext{if} \ \Theta \subseteq \Theta \end{cases} \)

   if \( x_i \)

   example: \( (1 - x_1 x_3 x_5) (1 - x_1 x_3 x_5) = (1 - x_1 x_3) \);

4. \( x_i (1 - \prod_{m \in \Theta} x_m) = x_i (1 - \prod_{m \in \Theta, x_i \notin \Theta} x_m) \)

   example: \( x_3 (1 - x_1 x_3 x_5) x_3 (1 - x_1 x_3) \);

5. \( (1 - \prod_{m \in \Theta} x_m) (1 - \prod_{m \notin \Theta} x_m) = (1 - \prod_{m \notin \Theta} x_m) \ 	ext{if} \ \Theta \subseteq \Theta \)

   example: \( \begin{cases} (1 - (1 - x_1) (1 - x_3)) (1 - x_3) \end{cases} \)

   \[ [1 - (1 - x_1) (1 - x_3)] = [1 - (1 - x_1) (1 - x_3)] \]

6. \( (1 - x_i) (1 - \prod_{m \in \Theta} x_m) = (1 - x_i) \)

   \[ (1 - \prod_{m \in \Theta, x_i \notin \Theta} x_m) \]

Example: \( (1 - x_3) [1 - (1 - x_1) (1 - x_3)] = (1 - x_3) [1 - (1 - x_1) (1 - x_3)] \).

It follows from rule 2 that

\[ [b(x) u_1(z) + (1 - b(x)) z_0] \otimes [b(x) u_2(z) + (1 - b(x)) z_0] \]

\[ = b(x) u_1(z) \otimes u_2(z) + (1 - b(x)) z_0. \]

This gives the following simplification rule that can be used in the algorithm presented in Section 3.3.

If several elements have the same protections sets \( \Theta \) and compose a series–parallel structure, they can be replaced by an equivalent element that has \( u \)-function \( u_{eq}(z) \) obtained by applying the operators (9) and (10) over \( u \)-functions \( u(z) \) of these elements. The function \( U_{eq}(z, s) \) representing the performance distribution of this equivalent element takes the form

\[ U_{eq}(z, s) = (s^\Theta - s^\Theta u_{eq}(z) + s^\Theta z_0. \]

6. Computational example

A power plant lubrication oil supply system consists of seven multi-state elements. The performance distributions of the elements are presented in Table 1. Six different fire

<table>
<thead>
<tr>
<th>No. of elements</th>
<th>Description</th>
<th>Protection sets ( \Theta_i )</th>
<th>Performance distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pump</td>
<td>{1,2,4} {1,2} {1,4}</td>
<td>( g ) 7 6 3 0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>Power converter</td>
<td>{3,4} {3,4} {3,4}</td>
<td>( p ) 0.75 0.15 0.05 0.05</td>
</tr>
<tr>
<td>3</td>
<td>Pump</td>
<td>{1,4} {1,2} {2,4}</td>
<td>( g ) 8 6 0 0 0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>Pump</td>
<td>{3,4} {1,4} {2,4}</td>
<td>( p ) 0.85 0.05 0.10 0.10</td>
</tr>
<tr>
<td>5</td>
<td>Power source</td>
<td>{2,5} {2,5,6} {5}</td>
<td>( g ) 5 3 0 0 0 0 0</td>
</tr>
<tr>
<td>6</td>
<td>Power source</td>
<td>{5,6} {2,5,6} {6}</td>
<td>( p ) 0.75 0.15 0.10 0.05</td>
</tr>
<tr>
<td>7</td>
<td>Power source</td>
<td>{6} {6} {6}</td>
<td>( p ) 0.80 0.15 0.05 0.05</td>
</tr>
</tbody>
</table>

Table 2

Vulnerability of protections

<table>
<thead>
<tr>
<th>No. of protection ( j )</th>
<th>Description</th>
<th>Vulnerability ( v_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Water and foam spray</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>Fire barriers</td>
<td>0.6</td>
</tr>
<tr>
<td>3</td>
<td>Fire barriers</td>
<td>0.4</td>
</tr>
<tr>
<td>4</td>
<td>Water and foam spray</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>Extinguishing agents release</td>
<td>0.6</td>
</tr>
<tr>
<td>6</td>
<td>Extinguishing agents release</td>
<td>0.1</td>
</tr>
</tbody>
</table>
protections are used to protect different groups of the elements. Each protection prevents the destruction of the corresponding group of the elements from fire, but each protection itself can be destroyed by the fire. The state transitions of system elements caused by their internal failures and repairs are mutually independent and do not depend on states of protections. The system vulnerability is defined as the probability that it tolerates single fire event.

The vulnerabilities of the protections (interpreted as probabilities of their destruction during a single fire) are given in Table 2. Consider three different configurations of system multilevel protection presented in Fig. 2. The element protection sets $\Theta_i$ corresponding to different configurations are presented in Table 1.

For example in configuration A element 1 (pump) can be protected from fire either by one of two spray systems (protections 1 and 4) or by fire barrier (protection 2). The pump is destroyed only if both spray systems and the fire barrier are destroyed by fire.

Fig. 3 presents the MSS survivability as a function of the demand for each configuration. One can see that while both configurations A and B provide greater system survivability than configuration C, the comparison of configurations A and B shows that A outperforms B in the range of small demands and B outperforms A in the range of greater demands. This shows that when different protection configurations are compared the expected system demand should be taken into account.

7. Conclusions

This paper presents a numeric algorithm for solving the problem of evaluating the survivability of series-parallel systems with multilevel protection, formulated in [16].

Unlike the method presented in [16] this algorithm (based on a composition of Boolean and the Universal Generating Function techniques) can be applied to systems with arbitrary configuration of protections (protections can overlap; the protected subsystems must not be series-parallel ones). The system elements can have different states characterized by different performance levels. The system survivability is defined as its ability to function at a performance level that satisfies a given demand. Depending on the interpretation of state probabilities of elements and protections, the system survivability refers to single impact occurrence or to entire mission time with possible multiple impacts.
References


