SYSTEM SIGNATURES AND RELIABILITY FUNCTIONS

JEAN-LUC MARICHAL

Abstract. The concept of signature is a useful tool in the analysis of semicoherent systems with continuous and i.i.d. component lifetimes, especially for the comparison of different system designs and the computation of the system reliability. We provide various conversion formulas between the signature and the reliability function of the system through the corresponding vector of dominations. We also show how the signature can be computed from the reliability function via basic manipulations such as differentiation, coefficient extraction, and integration.

1. Introduction

Consider an \( n \)-component system \((C, \phi)\), where \( C \) is the set \([n] = \{1, \ldots, n\}\) of its components and \( \phi: \{0,1\}^n \to \{0,1\} \) is its structure function which expresses the state of the system in terms of the states of its components. We assume that the system is semicoherent, which means that the structure function \( \phi \) is nondecreasing in each variable and satisfies the conditions \( \phi(0, \ldots, 0) = 0 \) and \( \phi(1, \ldots, 1) = 1 \). We also assume that the components have continuous and i.i.d. lifetimes \( T_1, \ldots, T_n \).

Samaniego \[9\] introduced the signature of such a system as the \( n \)-vector \( s = (s_1, \ldots, s_n) \) whose \( k \)-th coordinate \( s_k \) is the probability that the \( k \)-th component failure causes the system to fail. That is,

\[
 s_k = \Pr(T_S = T_{k:n}), \quad k = 1, \ldots, n,
\]

where \( T_S \) denotes the system lifetime and \( T_{k:n} \) denotes the \( k \)-th smallest lifetime, i.e., the \( k \)-th order statistic obtained by rearranging the variables \( T_1, \ldots, T_n \) in ascending order of magnitude. From this definition we immediately derive the identity \( \sum_{k=1}^n s_k = 1 \).

Boland \[2\] showed that \( s_k \) can be explicitly written in the form

\[
 s_k = \sum_{A \subseteq C, |A| = n-k+1} \frac{1}{\binom{n}{|A|}} \phi(A) - \sum_{A \subseteq C, |A| = n-k} \frac{1}{\binom{n}{|A|}} \phi(A).
\]

Here and throughout we identify Boolean \( n \)-vectors \( x \in \{0,1\}^n \) and subsets \( A \subseteq [n] \) in the usual way, that is, by setting \( x_i = 1 \) if and only if \( i \in A \). Thus we use the same symbol to denote both a function \( f: \{0,1\}^n \to \mathbb{R} \) and the corresponding set function \( f:2^{[n]} \to \mathbb{R} \) interchangeably. For instance, we write \( \phi(0, \ldots, 0) = \phi(\emptyset) \) and \( \phi(1, \ldots, 1) = \phi(C) \).

It is very often convenient to express the signature vector \( s \) in terms of the tail signature vector, a concept introduced by Boland \[2\] and named so by Gertsbakh et
The tail signature vector of the system is the \((n+1)\)-vector \(S = (S_0, \ldots, S_n)\) defined from \(s\) by
\[
S_k = \sum_{i=k+1}^{n} s_i, \quad k = 0, \ldots, n.
\]
In particular, we have \(S_0 = 1\) and \(S_n = 0\). Moreover, it is clear that the signature \(s\) can be retrieved from the tail signature \(S\) through the formula
\[
s_k = S_{k-1} - S_k, \quad k = 1, \ldots, n.
\]
Recall also that the reliability function associated with the structure function \(\phi\) is the multilinear function \(h: [0,1]^n \to \mathbb{R}\) defined by
\[
h(x) = h(x_1, \ldots, x_n) = \sum_{A \subseteq C} \phi(A) \prod_{i \in A} x_i \prod_{i \in C \setminus A} (1-x_i)
\]
(see [1, Chap. 2] for general background on reliability functions and [8, Section 3.2] for a more recent reference). It is easy to see that this function can always be put in the unique standard multilinear form
\[
h(x) = \sum_{A \subseteq C} d(A) \prod_{i \in A} x_i,
\]
where, for every \(A \subseteq C\), the coefficient \(d(A)\) is an integer.

By identifying the variables \(x_1, \ldots, x_n\) in function \(h(x)\), we define its diagonal section \(h(x, \ldots, x)\), which will be simply denoted by \(h(x)\). From Eqs. (4) and (5), we immediately obtain
\[
h(x) = \sum_{A \subseteq C} \phi(A) x^{|A|} (1 - x)^{n - |A|} = \sum_{A \subseteq C} d(A) x^{|A|},
\]
or equivalently,
\[
h(x) = \sum_{k=0}^{n} \phi_k x^k (1 - x)^{n-k} = \sum_{k=0}^{n} d_k x^k,
\]
where
\[
\phi_k = \sum_{|A| = k} \phi(A) \quad \text{and} \quad d_k = \sum_{|A| = k} d(A), \quad k = 0, \ldots, n.
\]

Clearly, we have \(\phi_0 = \phi(\emptyset) = 0\) and \(d_0 = d(\emptyset) = h(0) = 0\). The \(n\)-vector \(d = (d_1, \ldots, d_n)\) is called the vector of dominations of the system (see, e.g., [10, Sect. 6.2]).

**Example 1.** Consider the bridge structure as indicated in Figure 1. The corresponding structure function is given by
\[
\phi(x_1, \ldots, x_5) = x_1 x_4 \sqcup x_2 x_5 \sqcup x_1 x_3 x_5 \sqcup x_2 x_3 x_4,
\]
where \(\sqcup\) is the (associative) coproduct operation defined by \(x \sqcup y = 1 - (1-x)(1-y)\). The corresponding reliability function, given in Eq. (4), can be computed by expanding the coproducts in \(\phi\) and simplifying the resulting algebraic expression using \(x_i^2 = x_i\). We have
\[
h(x_1, \ldots, x_5) = x_1 x_4 + x_2 x_5 + x_1 x_3 x_5 + x_2 x_3 x_4 - x_1 x_2 x_3 x_4 - x_1 x_2 x_3 x_5 - x_1 x_2 x_4 x_5 - x_1 x_3 x_4 x_5 - x_2 x_3 x_4 x_5 + 2 x_1 x_2 x_3 x_4 x_5.
\]
We then obtain its diagonal section

\[ h(x) = 2x^2 + 2x^3 - 5x^4 + 2x^5 \]

and finally the domination vector \( d = (0, 2, 2, -5, 2) \). □

![Figure 1. Bridge structure](image)

The computation of the signature of a large system by means of Eq. (1) may be cumbersome and tedious since it requires the evaluation of \( \phi(A) \) for every \( A \subseteq C \). However, Boland et al. [3] observed that the \( n \)-vectors \( s \) and \( d \) can always be computed from each other through simple bijective linear transformations (see also [10, Sect. 6.3]). Thus, the signature vector \( s \) can be efficiently computed from the domination vector \( d \), or equivalently, from the diagonal section \( h(x) \) of the reliability function. Since Eqs. (2) and (3) provide linear conversion formulas between vectors \( s \) and \( S \), we observe that any of the vectors \( s, S, \) and \( d \) can be computed from any other by means of a bijective linear transformation (see Figure 2).

![Figure 2. Bijective linear transformations](image)

In Section 2 of this paper we yield these linear transformations explicitly and present them as linear conversion formulas. From these conversion formulas we derive efficient algorithms for the computation of any of these vectors from any other.

We also show how the computation of the vectors \( s \) and \( S \) can be performed from basic manipulations of function \( h(x) \) such as differentiation, reflection, coefficient extraction, and integration. For instance, we establish the polynomial identity (see Eq. (26))

\[
\left( \sum_{k=1}^{n} \binom{n}{k} s_k x^k \right) = \int_0^x (R^{n-1} h')(t + 1) dt,
\]

where \( h'(x) \) is the derivative of \( h(x) \) and \( (R^{n-1} h')(x) \) is the polynomial function obtained from \( h'(x) \) by switching the coefficients of \( x^k \) and \( x^{n-1-k} \) for \( k = 0, \ldots, n-1 \). Applying this result to the bridge system described in Example 1, we can easily see that Eq. (3) reduces to

\[
5s_1 x + 10s_2 x^2 + 10s_3 x^3 + 5s_4 x^4 + s_5 x^5 = 2x^2 + 6x^3 + x^4.
\]
By equating the corresponding coefficients we immediately derive the signature vector 
\( s = \left(0, \frac{1}{\gamma}, \frac{2}{\gamma}, \ldots, \frac{5}{\gamma}, 0\right) \).

In Section 3 we examine the general non-i.i.d. setting where the component lifetimes \( T_1, \ldots, T_n \) may be dependent. We show how a modification of the structure function enables us to formally extend the conversion formulas and algorithms obtained in Section 2 to the general dependent setting. We also show that, even though this formal extension is mathematically elegant, it is of little interest in practice.

2. Conversion Formulas

Recall that Eq. (9) gives the standard multilinear form of the reliability function 
\( h(x) \). As mentioned for instance in [3], p. 31, the link between the coefficients 
\( d(A) \) and the values \( \phi(A) \) is given through the following linear conversion formulas 
(obtained from the Möbius inversion theorem)

\[
\phi(A) = \sum_{B \subseteq A} d(B) \quad \text{and} \quad d(A) = \sum_{B \subseteq A} (-1)^{|A| - |B|} \phi(B).
\] (9)

The following proposition yields the linear conversion formulas between the 
\( n \)-vectors \( d = (d_1, \ldots, d_n) \) and \( (\phi_1, \ldots, \phi_n) \). Note that an alternative form of Eq. (11) 
was previously found by Samaniego [10, Sect. 6.3].

**Proposition 1.** We have

\[
\phi_k = \sum_{j=0}^{k} \binom{n-j}{k-j} d_j, \quad k = 1, \ldots, n,
\] (10)

and

\[
d_k = \sum_{j=0}^{k} (-1)^{k-j} \binom{n-j}{k-j} \phi_j, \quad k = 1, \ldots, n.
\] (11)

**Proof.** By Eqs. (7) and (9) we have

\[\phi_k = \sum_{A \subseteq C \atop |A|=k} \phi(A) = \sum_{A \subseteq C \atop |A|=k} \sum_{B \subseteq A} d(B).\]

Permuting the sums and then setting \( j = |B| \), we obtain

\[
\phi_k = \sum_{B \subseteq C \atop |B|=k} d(B) \sum_{A \subseteq B \atop |A|=k} 1 = \sum_{B \subseteq C \atop |B|=k} \binom{n-|B|}{k-|B|} d(B) = \sum_{j=0}^{k} \binom{n-j}{k-j} \sum_{B \subseteq C \atop |B|=j} d(B),
\]

which proves Eq. (10). Formula (11) can be established similarly. \( \square \)

We are now ready to establish conversion formulas and algorithms as announced in the introduction.

2.1. Conversions between \( s \) and \( \mathbf{S} \). We already know that the linear conversion 
formulas between the vectors \( s \) and \( \mathbf{S} \) are given by Eqs. (2) and (3). This conversion 
can also be explicitly expressed by means of a polynomial identity. Let \( \sum_{k=1}^{n} s_k x^k \) 
and \( \sum_{k=0}^{n} \mathbf{s}_k x^k \) be the generating functions of vectors \( s \) and \( \mathbf{S} \), respectively. Then 
we have the polynomial identity

\[
\sum_{k=1}^{n} s_k x^k = 1 + (x-1) \sum_{k=0}^{n} \mathbf{s}_k x^k.
\] (12)
Indeed, using Eq. (3) and summation by parts, we obtain
\[
\sum_{k=1}^{n} s_k x^k = \sum_{k=1}^{n} (\overline{S}_{k-1} - \overline{S}_k) x^k = x + \sum_{k=1}^{n} \overline{S}_k (x^{k+1} - x^k),
\]
which proves Eq. (12).

For the bridge system described in Example 1, the generating functions of vectors \( s \) and \( \overline{S} \) are given by
\[
\frac{1}{3} x^2 + \frac{4}{3} x^3 + \frac{1}{3} x^4 \quad \text{and} \quad 1 + x + \frac{4}{3} x^2 + \frac{1}{3} x^3,
\]
respectively. We can easily verify that Eq. (12) holds for these functions.

2.2. Conversions between \( \overline{S} \) and \( d \). Combining Eq. (2) with Eqs. (1) and (7), we observe that
\[
(13) \quad \overline{S}_k = \frac{1}{(n)} \sum_{|A|=n-k} \phi(A) = \frac{1}{(n)} \phi_{n-k}, \quad k = 0, \ldots, n.
\]

Recall that a path set of the system is a component subset \( A \) such that \( \phi(A) = 1 \). It follows from Eqs. (7) and (13) that \( \phi_k \) is precisely the number of path sets of size \( k \) and that \( \overline{S}_{n-k} \) is the proportion of component subsets of size \( k \) which are path sets.

Combining Eqs. (11) and (11) with Eq. (13), we immediately obtain the following conversion formulas between the vectors \( \overline{S} \) and \( d \).

Proposition 2. We have
\[
(14) \quad \overline{S}_k = \sum_{j=0}^{n-k} \binom{n-j}{k} d_j = \sum_{j=0}^{n-k} \binom{n-j}{j} d_j, \quad k = 0, \ldots, n,
\]
\[
(15) \quad d_k = \binom{n}{k} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} \overline{S}_{n-j}, \quad k = 0, \ldots, n.
\]

Equation (15) can be rewritten in a simpler form by using the classical difference operator \( \Delta_i \) which maps a sequence \( z_i \) to the sequence \( \Delta_i z_i = z_{i+1} - z_i \). Defining the \( k \)-th difference \( \Delta_i^k z_i \) of a sequence \( z_i \) recursively as \( \Delta_i^0 z_i = z_i \) and \( \Delta_i^k z_i = \Delta_i \Delta_i^{k-1} z_i \), we can show by induction on \( k \) that
\[
(16) \quad \Delta_i^k z_i = \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} z_{i+j}.
\]
Comparing Eq. (15) with Eq. (16) immediately shows that Eq. (15) can be rewritten as
\[
(17) \quad d_k = \binom{n}{k} \left( \Delta_i^k \overline{S}_{n-i} \right)_{i=0}^{k}, \quad k = 1, \ldots, n,
\]
and the vector \( d \) can then be computed efficiently from a classical difference table (see Table 1).

Setting \( D_{j,k} = \binom{n}{k} \left( \Delta_i^k \overline{S}_{n-i} \right)_{i=j}^{k} \), from Eq. (17) we can easily derive the following algorithm for the computation of \( d \). This algorithm requires only \( \frac{1}{2} n (n+1) \) additions and multiplications.

Algorithm 1. The following algorithm inputs vector \( \overline{S} \) and outputs vector \( d \). It uses the variables \( D_{j,k} \) for \( k = 0, \ldots, n \) and \( j = 0, \ldots, n-k \).

Step 1. For \( j = 0, \ldots, n \), set \( D_{j,0} := \overline{S}_{n-j} \).
Table 1. Computation of $d$ from $\mathcal{S}$

| $\mathcal{S}_n$ | $\binom{n}{1}(\Delta_i \mathcal{S}_{n-i})|_{i=0}$ | $\binom{n}{2}(\Delta_i^2 \mathcal{S}_{n-i})|_{i=0}$ |
|-----------------|---------------------------------|---------------------------------|
| $\mathcal{S}_{n-1}$ | $\binom{n}{1}(\Delta_i \mathcal{S}_{n-i})|_{i=1}$ | $\binom{n}{2}(\Delta_i^2 \mathcal{S}_{n-i})|_{i=1}$ |
| $\mathcal{S}_{n-2}$ | $\binom{n}{1}(\Delta_i \mathcal{S}_{n-i})|_{i=2}$ | $\binom{n}{2}(\Delta_i^2 \mathcal{S}_{n-i})|_{i=2}$ |
| $\mathcal{S}_{n-3}$ | $\binom{n}{1}(\Delta_i \mathcal{S}_{n-i})|_{i=3}$ | $\binom{n}{2}(\Delta_i^2 \mathcal{S}_{n-i})|_{i=3}$ |
| $\vdots$ | $\binom{n}{1}(\Delta_i \mathcal{S}_{n-i})|_{i=m}$ | $\binom{n}{2}(\Delta_i^2 \mathcal{S}_{n-i})|_{i=m}$ |

Table 2. Computation of $d$ from $\mathcal{S}$ (Example 2)

| $j$ | $0$ | $1/5$ | $4/5$ |
|-----|-----|------|------|
| $d_j$ | $2$ | $-5$ | $5$ |

The converse transformation (14) can then be computed efficiently by the following algorithm, in which we compute the quantities

$$S_{j,k} = \sum_{i=0}^{k} \binom{k}{i} \binom{n-j}{i} d_{i+j}.$$ 

Algorithm 2. The following algorithm inputs vector $d$ and outputs vector $\mathcal{S}$. It uses the variables $S_{j,k}$ for $k = 0, \ldots, n$ and $j = 0, \ldots, n - k$.

Step 1. For $j = 0, \ldots, n$, set $S_{j,0} := d_j$.

Step 2. For $k = 1, \ldots, n$

For $j = 0, \ldots, n - k$

$$S_{j,k} := \frac{j+1}{n-j} S_{j+1,k-1} + S_{j,k-1}.$$
Step 3. For }k = 0, \ldots, n\text{, set } S_{n-k} := S_{0,k}.

2.3. Conversions between }s\text{ and }d\text{. The following proposition yields the conversion formulas between the vectors }s\text{ and }d\text{. Note that a non-explicit version of Eq. (18) was previously found in Boland et al. [3] (see also Theorem 6.1 in [10]).

Proposition 3. We have

\begin{align}
(18) \quad s_k &= \sum_{j=0}^{n-k} \binom{n-j}{k} \frac{j+1}{n-j} d_{j+1} = \sum_{j=1}^{n-k+1} \binom{n-k}{j-1} d_j, \quad k = 1, \ldots, n, \\
(19) \quad d_k &= \binom{n}{k} \sum_{j=0}^{k-1} (-1)^{k-1-j} \binom{k-1}{j} s_{n-j}, \quad k = 1, \ldots, n, \\
(20) \quad d_k &= \binom{n}{k} (\Delta_i^1 s_{n-i})_{i=0}, \quad k = 1, \ldots, n,
\end{align}

Proof. Combining Eq. (14) with Eq. (3), we obtain

\[ s_k = S_{k-1} - S_k = \sum_{j=1}^{n-k} \binom{n-k+1}{j} d_j - \sum_{j=1}^{n-k} \binom{n-k}{j} d_j = \sum_{j=1}^{n-k} \binom{n-k}{j-1} d_j + \frac{1}{n-k} d_{n-k+1}, \]

which proves Eq. (18). By Eq. (3) we have \( \Delta_i^1 \bar{s}_{n-i} = s_{n-i} \) for \( i = 0, \ldots, n-1 \). Equation (20) then follows from Eq. (17). Equation (19) then follows immediately from Eq. (20).

Equation (20) shows that }d\text{ can be efficiently computed directly from }s\text{ by means of a difference table (see Table 3).

\begin{table}[h]
\centering
\begin{tabular}{c|c}
\hline
\binom{n}{1} s_n & \binom{n}{2} (\Delta_i^1 s_{n-i})_{i=0} \\
\binom{n}{1} s_{n-1} & \binom{n}{2} (\Delta_i^2 s_{n-i})_{i=1} \\
\binom{n}{1} s_{n-2} & \binom{n}{2} (\Delta_i^3 s_{n-i})_{i=2} \\
\vdots & \vdots \\
\hline
\end{tabular}
\caption{Computation of }d\text{ from }s\text{.}
\end{table}

Setting }d_{j,k} = \binom{n}{k} (\Delta_i^{k-1} s_{n-i})_{i=j-1}, \text{ we can also derive the following algorithm for the computation of vector }d\text{. This algorithm requires only } \frac{1}{2} n(n-1) \text{ additions and multiplications.}

Algorithm 3. The following algorithm inputs vector }s\text{ and outputs vector }d\text{. It uses the variables }d_{j,k}\text{ for }k = 1, \ldots, n\text{ and }j = 1, \ldots, n-k+1.

\begin{enumerate}
\item For }j = 1, \ldots, n\text{, set } d_{j,1} := n s_{n-j+1}.
\item For }k = 2, \ldots, n\text{, set } d_{j,k} := \frac{n-k+1}{k} \left( d_{j+1,k-1} - d_{j,k-1} \right)
\item For }k = 1, \ldots, n\text{, set } d_k := d_{1,k}.
\end{enumerate}
Example 3. Consider again the bridge system described in Example 1. The corresponding signature vector is given by \( s = (0, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, 0) \). Forming the difference table (see Table 4) and reading its first row, we obtain the vector \( d = (0, 2, -5, 2) \) and hence the function \( h(x) = 2x^2 + 2x^3 - 5x^4 + 2x^5 \).

|     | 0 | 1 | 2 |     |
|-----|---|---|---|-----|
| 0   |   |   | 2 |     |
| 1   |   | 4 | 2 | -5  |
| 3   | -8| 2 |     |     |
| 1   | -4| 5 |     |     |
| 0   | -2| 2 |     |     |

Table 4. Computation of \( d \) from \( s \) (Example 3)

The converse transformation (18) can then be computed efficiently by the following algorithm, in which we compute the quantities

\[
s_{j,k} = \frac{1}{n} \sum_{i=1}^{k} \binom{k-1}{i-1} \binom{i+j-1}{i-1} d_{i+j-1}.
\]

Algorithm 4. The following algorithm inputs vector \( d \) and outputs vector \( s \). It uses the variables \( s_{j,k} \) for \( k = 1, \ldots, n \) and \( j = 1, \ldots, n - k + 1 \).

- **Step 1.** For \( j = 1, \ldots, n \), set \( s_{j,1} := \frac{1}{n} d_j \).
- **Step 2.** For \( k = 2, \ldots, n \)
  - For \( j = 1, \ldots, n - k + 1 \)
    - \( s_{j,k} := \frac{j+1}{n} s_{j+1,k-1} + s_{j,k-1} \)
- **Step 3.** For \( k = 1, \ldots, n \), set \( s_{n-k+1} := s_{1,k} \).

2.4. Conversions between \( \mathcal{S} \) or \( s \) and \( h(x) \). The conversion formulas between vectors \( s \) and \( d \) show that the diagonal section \( h(x) \) of the reliability function encodes exactly the signature (or equivalently, the tail signature), no more, no less. Even though the latter can be computed from vector \( d \) using Eqs. (14) and (18), we will now see how we can compute it by direct and simple algebraic manipulations of function \( h(x) \).

Let \( f \) be a univariate polynomial of degree \( \leq n \),

\[
f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.
\]

The \textit{n-reflected} of \( f \) is the polynomial \( R^nf \) defined by

\[
(R^nf)(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n,
\]
or equivalently, \( (R^nf)(x) = x^n f(1/x) \).

Combining Eq. (6) with Eq. (13), we obtain (see also [3])

\[
(21) \quad h(x) = \sum_{k=0}^{n} \binom{n}{k} x^k (1-x)^{n-k}.
\]
From this equation it follows, as it was already observed in [7], that
\[
(R^n h)(x + 1) = \sum_{k=0}^{n} \binom{n}{k} S_k x^k.
\]
(22)

Thus, \((n)S_k\) can be obtained simply by reading the coefficient of \(x^k\) in the polynomial function \((R^n h)(x + 1)\). Denoting by \([x^k]f(x)\) the coefficient of \(x^k\) in a polynomial function \(f(x)\), Eq. (22) can be rewritten as
\[
\binom{n}{k} S_k = [x^k](R^n h)(x + 1), \quad k = 0, \ldots, n.
\]
(23)

From Eq. (23) we immediately derive the following algorithm (see also [7]).

**Algorithm 5.** The following algorithm inputs the number \(n\) of components and the diagonal section of reliability function \(h\) and outputs the tail signature \(S\) of the system.

**Step 1.** For \(k = 0, \ldots, n\), let \(a_k\) be the coefficient of \(x^k\) in the \(n\)-degree polynomial
\[
(R^n h)(x + 1) = (x + 1)^n h\left(\frac{1}{x+1}\right).
\]

**Step 2.** We have
\[
S_k = \frac{a_k}{(k)!}, \quad k = 0, \ldots, n.
\]

The following proposition yields the analog of Eqs. (22) and (23) for the signature. Here and throughout we denote by \(h'(x)\) the derivative of \(h(x)\).

**Proposition 4.** We have
\[
k^n \binom{n}{k} s_k = [x^{k-1}](R^{n-1} h')(x + 1), \quad k = 1, \ldots, n,
\]
(24)

\[
\sum_{k=1}^{n} \binom{n}{k} s_k k x^{k-1} = (R^{n-1} h')(x + 1),
\]
(25)

\[
\sum_{k=1}^{n} \binom{n}{k} s_k x^k = \int_0^{x} (R^{n-1} h')(t + 1) dt.
\]
(26)

**Proof.** By Eq. (23) we have \(h'(x) = \sum_{j=0}^{n-1} (j + 1) d_{j+1} x^j\) and therefore
\[
(R^{n-1} h')(x + 1) = \sum_{j=0}^{n-1} (j + 1) d_{j+1} (x + 1)^{n-1-j}
\]
\[
= \sum_{j=0}^{n-1} (j + 1) d_{j+1} \sum_{k=1}^{n-j} \binom{n-1-j}{k} x^{k-1}
\]
\[
= \sum_{k=1}^{n} x^{k-1} \sum_{j=0}^{n-k} \binom{n-1-j}{k-1} (j + 1) d_{j+1}.
\]

Thus, the inner sum in the latter expression is the coefficient of \(x^{k-1}\) in the polynomial function \((R^{n-1} h')(x + 1)\). Dividing this sum by \(k^n\) and then using Eq. (18), we obtain \(s_k\). This proves Eqs. (24) and (25). Equation (26) is then obtained by integrating both sides of Eq. (25) on the interval \([0, x]\). □

From Eq. (24) we immediately derive the following algorithm.
Algorithm 6. The following algorithm inputs the number \( n \) of components and the diagonal section of reliability function \( h \) and outputs the signature \( s \) of the system.

**Step 1.** For \( k = 1, \ldots, n \), let \( a_{k-1} \) be the coefficient of \( x^{k-1} \) in the \((n-1)\)-degree polynomial

\[
(R^{n-1}h')(x + 1) = (x + 1)^{n-1} h'\left(\frac{1}{x+1}\right).
\]

**Step 2.** We have

\[
s_k = \frac{a_{k-1}}{k\binom{n}{k}}, \quad k = 1, \ldots, n.
\]

Even though such an algorithm can be easily executed by hand for small \( n \), a computer algebra system can be of great assistance for large \( n \).

**Example 4.** Consider again the bridge system described in Example 1. We have

\[
h'(x) = 4x + 6x^2 - 20x^3 + 10x^4 \quad \text{and} \quad (R^4 h')(x) = 10 - 20x + 6x^2 + 4x^3.
\]

It follows that \((R^4 h')(x + 1) = 4x + 18x^2 + 4x^3\) and hence \( s = \left(0, \frac{1}{5}, \frac{3}{5}, \frac{1}{5}, 0\right) \). Indeed, we have for instance \( s_3 = \frac{a_2}{(3\binom{5}{3})} = \frac{3}{5} \).

The following proposition, established in [7], provides a necessary and sufficient condition on the system signature for the reliability function to be of full degree (i.e., \( d_n \neq 0 \)). Here we provide a shorter proof based on Eq. (25).

**Proposition 5 (7).** Let \((C, \phi)\) be an \( n \)-component semicoherent system with continuous and i.i.d. component lifetimes. Then the reliability function \( h(x) \) (or equivalently, its diagonal section \( h(x) \)) is a polynomial of degree \( n \) if and only if

\[
\sum_{k \text{ odd}} \binom{n-1}{k-1} s_k \neq \sum_{k \text{ even}} \binom{n-1}{k-1} s_k.
\]

**Proof.** The function \( h(x) \) is of degree \( n \) if and only if \( g(x) \) is of degree \( n - 1 \) and this condition holds if and only if \( d_n = \frac{1}{n}(R^{n-1}h')(0) \neq 0 \). By Eq. (25) this means that

\[
\sum_{k=1}^{n} \binom{n}{k} s_k (-1)^{k-1} = n \sum_{k=1}^{n} \binom{n-1}{k-1} s_k (-1)^{k-1}
\]

is not zero. \( \square \)

The vectors \( s \) and \( \overline{s} \) can also be computed via their generating functions. The following proposition yields integral formulas for these generating functions.

**Proposition 6.** We have the generating functions

\[
\sum_{k=0}^{n} S_k x^k = \int_0^1 (n+1) R^n_x \left((R^n_h)((t-1)x+1)\right) dt,
\]

\[
\sum_{k=1}^{n} s_k x^k = \int_0^1 x R_x^{n-1} \left((R^{n-1}_t h')(t-1)x+1\right) dt,
\]

where \( R^n_x \) is the \( n \)-reflection with respect to variable \( t \).
Proof. By Eq. (22), we have
\[(R^n h)((t - 1)x + 1) = \sum_{k=0}^{n} \binom{n}{k} \overline{S}_k (t - 1)^k x^k\]
and hence
\[R^n ((R^n h)((t - 1)x + 1)) = \sum_{k=0}^{n} \binom{n}{k} \overline{S}_k t^{n-k} (1-t)^k x^k.\]
Integrating this expression from \(t = 0\) to \(t = 1\) and using the identity
\[\int_0^1 t^{n-k} (1-t)^k dt = \frac{1}{(n+1)(\binom{n}{k})},\]
we finally obtain Eq. (27). Formula (28) can be proved similarly by using Eq. (25). \(\square\)

From Eq. (28) we immediately derive the following algorithm for the computation of the generating function of the signature. The algorithm corresponding to Eq. (27) can be derived similarly.

**Algorithm 7.** The following algorithm inputs the number \(n\) of components and the reliability function \(h\) and outputs the generating function of the signature \(s\) of the system.

**Step 1.** Let \(f(t, x) = x (R^{n-1} h')((t - 1)x + 1).\)

**Step 2.** We have
\[\sum_{k=1}^{n} s_{k} x^k = \int_0^1 f(t, x) dt.\]

The computation of \(h(x)\) from \(s\) or \(\overline{S}\) can be useful if we want to compute the system reliability \(h(p)\) directly from the signature and the component reliability \(p\).

We already know that Eq. (21) gives the polynomial \(h(x)\) in terms of vector \(\overline{S}\). The following proposition yields simple expressions of \(h(x)\) and \(h'(x)\) in terms of vector \(s\). This result was already presented in [5, Sect. 4] and [7, Rem. 2] in alternative forms.

**Proposition 7.** We have
\[(30) \quad h'(x) = \sum_{k=1}^{n} s_{k} k \overline{inom{n}{k}} x^{n-k} (1-x)^{k-1},\]
\[(31) \quad h(x) = \sum_{k=1}^{n} s_{k} I_x(n - k + 1, k) = \sum_{k=1}^{n} s_{k} \sum_{i=n-k+1}^{n} \overline{inom{n}{i}} x^i (1-x)^{n-i},\]
where \(I_x(a, b)\) is the regularized beta function defined, for any \(a, b, x > 0,\) by
\[I_x(a, b) = \frac{\int_0^x t^{a-1} (1-t)^{b-1} dt}{\int_0^1 t^{a-1} (1-t)^{b-1} dt}.\]

**Proof.** Formula (30) immediately follows from Eq. (25). Then, from Eqs. (29) and (30) we immediately derive the first equality in Eq. (31) since \(h(x) = \int_0^x h'(t) dt.\) The second equality follows from Eqs. (24) and (25). \(\square\)

The following proposition provides alternative expressions of \(h(x)\) and \(h'(x)\) in terms of \(\overline{S}\) and \(s\), respectively.
Proposition 8. We have
\begin{align}
\label{eq:32}
  h(x) &= \left( (x\Delta_i + I)^n \overline{S}_{n-i} \right)_{i=0}, \\
\label{eq:33}
  h'(x) &= n \left( (x\Delta_i + I)^{n-1} s_{n-i} \right)_{i=0},
\end{align}
where $I$ denote the identity operator.

Proof. By Eq. \eqref{eq:17} we have
\begin{align}
  h(x) &= \sum_{k=0}^{n} d_k x^k = \sum_{k=0}^{n} \binom{n}{k} x^k \left( \Delta^k_i \overline{S}_{n-i} \right)_{i=0},
\end{align}
which proves Eq. \eqref{eq:32} as we can immediately see by formally expanding the binomial operator expression \((x\Delta_i + I)^n\). Equation \eqref{eq:33} then immediately follows from Eq. \eqref{eq:32}.

Proposition 8 shows that the functions $h(x)$ and $h'(x)$ can be computed from difference tables. Setting $D_{j,k}(x) = \left( (x\Delta_i + I)^k \overline{S}_{n-i} \right)_{i=j}$ and $d_{j,k}(x) = n \left( (x\Delta_i + I)^{k-1} s_{n-i} \right)_{i=j-1}$, we can derive the following algorithms for the computation of $h(x)$ and $h'(x)$.

Algorithm 8. The following algorithm inputs vector $\overline{S}$ and outputs function $h(x)$. It uses the functions $D_{j,k}(x)$ for $k = 0, \ldots, n$ and $j = 0, \ldots, n - k$.
\begin{enumerate}
  \item [Step 1.] For $j = 0, \ldots, n$, set $D_{j,0}(x) := \overline{S}_{n-j}$.
  \item [Step 2.] For $k = 1, \ldots, n$
    \begin{enumerate}
      \item [For $j = 0, \ldots, n - k$]
        \begin{enumerate}
          \item $D_{j,k}(x) := x D_{j+1,k-1}(x) + (1-x) D_{j,k-1}(x)$
        \end{enumerate}
    \end{enumerate}
  \item [Step 3.] $h(x) := D_{0,n}(x)$.
\end{enumerate}

Algorithm 9. The following algorithm inputs vector $s$ and outputs function $h'(x)$. It uses the functions $d_{j,k}(x)$ for $k = 1, \ldots, n$ and $j = 1, \ldots, n - k + 1$.
\begin{enumerate}
  \item [Step 1.] For $j = 1, \ldots, n$, set $d_{j,1}(x) := n s_{n-j+1}$.
  \item [Step 2.] For $k = 2, \ldots, n$
    \begin{enumerate}
      \item [For $j = 1, \ldots, n - k + 1$]
        \begin{enumerate}
          \item $d_{j,k}(x) := x d_{j+1,k-1}(x) + (1-x) d_{j,k-1}(x)$
        \end{enumerate}
    \end{enumerate}
  \item [Step 3.] $h'(x) := d_{1,n}(x)$.
\end{enumerate}

Table 5 summarizes the main conversion formulas obtained thus far. They are given by the corresponding equation numbers. For instance, formulas to compute $s$ from $d$ or $h(x)$ are given in Eqs. \eqref{eq:18}, \eqref{eq:24}, \eqref{eq:26}, and \eqref{eq:28}.

| d or $h(x)$ | $s$ | $\overline{S}$ |
|-------------|-----|----------------|
| 18          | 19  | 20  | 31  |
| 15          | 17  | 24  | 32  |
| 18          | 24  | 20  | 28  |
| 2           | 12  |     |     |

Table 5. Conversion formulas
2.5. Conversions based on the dual structure. We end this section by giving conversion formulas involving the dual structure of the system. Let $\phi^D: \{0,1\}^n \rightarrow \{0,1\}$ be the dual structure function defined as $\phi^D(x) = 1 - \phi(1-x)$ and let $h^D: [0,1]^n \rightarrow \mathbb{R}$ be its corresponding reliability function, that is, $h^D(x) = 1 - h(1-x)$.

Straightforward computations yield the following conversion formulas, where the upper index $D$ always refers to the dual structure and $\delta$ stands for the Kronecker delta:

\begin{align}
(34) \quad d_k^D &= \delta_{k,0} - (-1)^k \sum_{j=k}^n \binom{j}{k} d_j, \quad k = 0, \ldots, n, \\
(35) \quad d_k &= \delta_{k,0} - (-1)^k \sum_{j=k}^n \binom{j}{k} d_j^D, \quad k = 0, \ldots, n, \\
(36) \quad \mathcal{S}_k &= 1 - \mathcal{S}^D_{n-k} = 1 - \sum_{j=0}^k \binom{k}{j} d_j^D, \quad k = 0, \ldots, n, \\
(37) \quad s_k &= \phi^D_{n-k+1} = \sum_{j=1}^k \binom{k-1}{j-1} d_j^D, \quad k = 1, \ldots, n, \\
(38) \quad d_k^D &= \delta_{k,0} - \binom{n}{k} (\Delta^k \mathcal{S})_{i=0}, \quad k = 0, \ldots, n, \\
(39) \quad d_k^D &= \binom{n}{k} (\Delta^k s)_{i=1}, \quad k = 1, \ldots, n.
\end{align}

Recall that $\phi_k$ gives the number of path sets of size $k$. Combining (13) with (22), we obtain the identity

$$\sum_{k=0}^n \phi_{n-k} x^k = (R^n h)(x + 1).$$

Note that the generating function of vector $(\phi_0, \ldots, \phi_n)$ can be obtained by using (13), (39), and the dual version of (22). Indeed, we have

\begin{align*}
\sum_{k=0}^n \phi_k x^k &= \sum_{k=0}^n \binom{n}{k} \mathcal{S}_{n-k} x^k = \sum_{k=0}^n \binom{n}{k} x^k - \sum_{k=0}^n \binom{n}{k} \mathcal{S}^D_k x^k \\
&= (x + 1)^n - (R^n h^D)(x + 1).
\end{align*}

3. The general dependent case

In this final section we drop the i.i.d. assumption and consider the general dependent setting, assuming only that there are no ties among the component lifetimes (i.e., $\Pr(T_i - T_j) = 0$ whenever $i \neq j$).

Two concepts of signature emerge in this general setting. First, we can consider the structural signature, that is, the $n$-vector $s$ whose $k$-th coordinate is given by Boland’s formula (1). Of course, the conversion formulas and algorithms obtained in Section 2 can still be used “as is”, even if the i.i.d. assumption is dropped. Second, we can consider the probability signature, that is, the $n$-vector $p$ whose $k$-th coordinate is given by $p_k = \Pr(T_S = T_{kn})$. In the latter case we show that a modification of the structure function enables us to formally extend the conversion formulas and algorithms to the general dependent setting.
Marichal and Mathonet [6] showed that

\[(40) \quad p_k = \sum_{A \subseteq C, |A| = n-k+1} q(A) \phi(A) - \sum_{A \subseteq C, |A| = n-k} q(A) \phi(A),\]

where the function \(q: 2^{[n]} \to \mathbb{R}\), called the \textit{relative quality function} associated with the system, is defined by

\[q(A) = \Pr\left(\max_{i \in A} X_i < \min_{i \in A} X_i\right),\]

and has the property \(\sum_{|A| = k} q(A) = 1\) for \(k = 0, \ldots, n\). Thus, for any subset \(A \subseteq C\), the number \(q(A)\) is the probability that the best \(|A|\) components of the system are precisely those in \(A\).

In the special case when the component lifetimes are i.i.d., or even exchangeable, the number \(q(A)\) is exactly \(1/\binom{n}{|A|}\) and therefore by comparing Eqs. (1) and (40) we immediately see that \(p\) then reduces to \(s\). As mentioned in [6], this observation motivates the introduction of the \textit{normalized relative quality function} \(\tilde{q}: 2^{[n]} \to \mathbb{R}\), defined by \(\tilde{q}(A) = \binom{n}{|A|} q(A)\). We then have \(\tilde{q}(A) = 1\) whenever the component lifetimes are i.i.d. or exchangeable.

We now assign to the system a pseudo-structure function \(\psi: 2^{[n]} \to \mathbb{R}\) defined so as to have

\[\sum_{A \subseteq C, |A| = k} \frac{1}{\binom{n}{|A|}} \psi(A) = \sum_{A \subseteq C, |A| = k} q(A) \phi(A), \quad k = 0, \ldots, n.\]

**Definition 9.** Let \((C, \phi)\) be an \(n\)-component system with relative quality function \(q\). The \textit{q-structure function} associated with the system is the set function \(\psi: 2^{[n]} \to \mathbb{R}\) defined by

\[\psi(A) = \begin{cases} \binom{n}{|A|} q(A), & \text{if } A \text{ is a path set,} \\ 0, & \text{otherwise.} \end{cases}\]

It is clear that \(\psi\) reduces to \(\phi\) whenever the component lifetimes of the system are i.i.d. or exchangeable. In the general dependent case, the function \(\psi\) is a pseudo-Boolean function, that is, a function from \(\{0, 1\}^n\) to \(\mathbb{R}\). As such, it has the following multilinear form

\[\psi(x) = \sum_{A \subseteq C} \psi(A) \prod_{i \in A} x_i \prod_{i \in C \setminus A} (1 - x_i), \quad x \in \{0, 1\}^n.\]

We can also define its multilinear extension \(g: [0, 1]^n \to \mathbb{R}\) as

\[g(x) = \sum_{A \subseteq C} \psi(A) \prod_{i \in A} x_i \prod_{i \in C \setminus A} (1 - x_i), \quad x \in [0, 1]^n.\]

This function can always be put in the unique standard multilinear form

\[(41) \quad g(x) = \sum_{A \subseteq C} c(A) \prod_{i \in A} x_i,\]

where, by the Möbius inversion theorem, the coefficient \(c(A)\) is given by

\[(42) \quad c(A) = \sum_{B \subseteq A} (-1)^{|A| - |B|} \psi(B).\]
Thus, in this general setting we see that Eq. (1) extends to Eq. (40), while Eq. (13) extends to the following equation

$$P_k = \sum_{A \subseteq C, |A|=n-k} A \divides A = n-k q(A) \phi(A) = 1 \left( \frac{n}{k} \right) \psi_{n-k}, \quad k = 0, \ldots, n,$$

where $P_k$ is the $k$-th coordinate of the tail probability signature $P = (P_0, \ldots, P_n)$ of the system and is defined as $P_k = \sum_{i=k+1}^n p_i$.

As far as the other equations are concerned, we have the following straightforward theorem.

**Theorem 10.** Equations (2)–(12) and (14)–(33) still hold if we replace $s_k$, $S_k$, $h(x)$, $h(x)$, $\phi(A)$, $d(A)$, $\phi_k$, and $d_k$ with $p_k$, $P_k$, $g(x)$, $g(x)$, $\psi(A)$, $c(A)$, $\psi_k$, and $c_k$, respectively.

This theorem essentially states that we can formally extend our formulas and algorithms mutatis mutandis to the general dependent setting. For instance, from Eq. (26) we immediately derive the identity

$$\sum_{k=1}^n \left( \frac{n}{k} \right) p_k x^k = \int_0^\infty (R^{n-1}y)(t+1) dt.$$

In practice, however, function $g(x)$ is much heavier to handle than function $h(x)$ (consider Example 1). Moreover, this function need not be nondecreasing in each argument and we do not see an easy way to express it as a coproduct over minimal path sets.

In conclusion, even though this formal extension of the conversion formulas is mathematically elegant it seems to be of little interest in practice.

**Acknowledgments**

This research is supported by the internal research project F1R-MTH-PUL-12RDO2 of the University of Luxembourg.

**References**

[1] R. E. Barlow and F. Proschan. *Statistical theory of reliability and life testing. To Begin With*, Silver Spring, MD, 1981.

[2] P. J. Boland. Signatures of indirect majority systems. *J. Appl. Prob.*, 38:597–603, 2001.

[3] P. J. Boland, F. J. Samaniego, and E. M. Vestrup. Linking dominations and signatures in network reliability theory. In: Mathematical and statistical methods in reliability (Trondheim, 2002), pages 89-103. Ser. Qual. Reliab. Eng. Stat., 7, World Sci. Publ., River Edge, NJ, 2003.

[4] I. Gertsbakh, Y. Shpungin, and F. Spizzichino. Signatures of coherent systems built with separate modules. *J. Appl. Probab.*, 48(3):843–855, 2011.

[5] J.-L. Marichal. Cumulative distribution functions and moments of lattice polynomials. *Stat. Prob. Letters*, 76(12):1273–1279, 2006.

[6] J.-L. Marichal and P. Mathonet. Extensions of system signatures to dependent lifetimes: Explicit expressions and interpretations. *J. Multivariate Analysis*, 102(5):931–936, 2011.

[7] J.-L. Marichal and P. Mathonet. Computing system signatures through reliability functions. *Stat. Prob. Letters*, 83(3):710–717, 2013.

[8] K. G. Ramamurthy. *Coherent structures and simple games*, volume 6 of Theory and Decision Library. Series C: Game Theory, Mathematical Programming and Operations Research. Kluwer Academic Publishers Group, Dordrecht, 1990.

[9] F.J. Samaniego. On closure of the IFR class under formation of coherent systems. *IEEE Trans. Reliability Theory*, 34:69–72, 1985.
[10] F.J. Samaniego. *System signatures and their applications in engineering reliability*. Int. Series in Operations Research & Management Science, 110. New York: Springer, 2007.

Mathematics Research Unit, FSTC, University of Luxembourg, 6, rue Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg

E-mail address: jean-luc.marichal[at]uni.lu