Computing Consecutive-Type Reliabilities Non-Recursively

Galit Shmueli

The author is with the Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA (e-mail: galit.shmueli@cmu.edu)

Abstract

The reliability of consecutive-type systems has been approached from different angles. We present a new method for deriving the generating functions and reliabilities of various consecutive-type systems. Our method, which is based on Feller's run theory, is easy to implement, and leads to both recursive and non-recursive formulas for the reliability. The non-recursive expression is especially advantageous for systems with numerous components. We show how the method can be extended for computing generating functions and reliabilities of systems with multi-state components as well as systems with statistically dependent components. To make our theoretical derivations practical to practitioners, we include short computer programs that do the non-recursive computations yielding the reliabilities of such systems.

Keywords

Consecutive systems, system reliability, recurrence relations, partial fraction expansion, non-recursive reliability.

I. Notation

S	Operative component
F	Failed component
p	Probability of a failed component
q	$1-p$
R_n	Reliability of a system with n components
$u_{t,i}$	Probability that pattern i is completed on the tth component
$U_i(s)$	generating function of $u_{t,i}$
$\mathcal{G}(s)$	Reliability generating function
WT	Waiting time
$\mathcal{G}_{WT}(s)$	Waiting time generating function
s_i	ith root of the polynomial in the generating function's denominator
S	Number of possible states of a component
λ	Probability of a failed component given that the previous component failed.

II. Introduction

Consecutive-k-out-of-n and similar systems usually have a higher reliability than series systems, and are less expensive than parallel systems (Chao et al., 1995).

Within the consecutive-type family, we consider the reliability of systems with linearly arranged components, which are labeled $1, 2, ..., n$. Usually, the breakdown of the system is reached by progressive transitions through several levels of deterioration (Koutras, 1996). In other words, components break down successively, until a critical number of failed components k causes the system to fail. We focus on three well-known consecutive-type systems:

1. *Consecutive-k-out-of-n*:F systems, which fail when k consecutive components fail.
2. *m-consecutive-k-out-of-n*:F systems, which fail when m non-overlapping groups of k consecutive components fail.
3. *r-within-consecutive-k-out-of-n*:F systems, which fail when r ($r < k$) components fail within a window of k consecutive components.

These definitions describe the relation between the component failures and the system failure. Methods for evaluating the reliability, its generating function, and long-run behavior in such systems lead to either approximations or exact values. Two major approaches have been the combinatorial approach (Derman et al., 1982; Aki & Hirano, 1984) and the Markov chain embedding approach (Chao & Lin, 1984; Fu, 1987; Koutras, 1996). A comprehensive review is given by Chao et al. (1995). In all cases, the expression for the reliability is recursive in n, the number of components.

In this paper we introduce a new method for computing the exact reliability (and its generating function) for systems with on/off or multiple-outcome components. Our method leads to both a recursive and a non-recursive expression for the reliability. We illustrate how to obtain a recursive expression for the reliability, but focus on the method that leads to a non-recursive expression, which is especially desirable for systems with a large number of components.

In section III we describe the method and use it to derive expressions for the reliability of the three consecutive-type systems mentioned above.

Next, we apply our method to systems that have components that do not exhibit a simple on/off behavior, but rather have multiple outcomes. This means that each component can be in one of S states ($S \geq 2$). We consider the two cases where either the states are ordered (e.g., gradual degradation), or they are mutually exclusive (e.g., different types of failure). When $S = 2$ our results reduce to the ordinary on/off component structure.

Some properties of consecutive-type systems with multi-state components exist in the literature: Haim & Porat (1991) gave mean values and bounds for consecutive-k-out-of-n systems with ordered multi-state components. Fu (1996) used the Markov-chain embedding method for computing probabilities that are associated with multi-state trials, which are related to *consecutive-k-out-of-n*:F and *m-consecutive-k-out-of-n*:F systems. In section IV we show how our method is applied to consecutive systems with these two types of multi-state components. Section V addresses systems with statistically dependent components. We include short computer programs that yield the reliability throughout the different sections. Concluding remarks are given in section VI.

III. A New Method for Obtaining the Reliability of Consecutive-Type Systems

Our method is based on Feller's (1968) theory for runs. The method consists of five steps, which lead to the reliability probability and generating functions:

Step 1: Specify all the component-patterns that cause the system to fail (For example, in a simple series system, the pattern that causes the entire system to fail is a failure of a single component).

Step 2: Write a recursive relation for the probability that each pattern is completed on the tth component (denoted by u_t).

Step 3: Multiply each equation by s^t and sum over t to infinity. Denote by $U_i(s) = \sum_{t=0}^{\infty} s^t u_t$ the generating function of u_t, for each pattern i $(i = 1, \ldots, a)$.

Step 4: Solve the set of a linear equations to obtain $U_i(s)$, and combine the solutions to get the reliability generating function, $\mathcal{G}(s)$, by using the formula

$$\mathcal{G}(s) = \left[(1-s)\left(\sum_{i=1}^{a} U_i(s) - a + 1\right)\right]^{-1}, \tag{1}$$

where a is the number of distinct failure-causing patterns.

In some cases the reliability is the probability that a specific pattern does not occur more than m times within n components. The generating function in such cases is closely related to that of the waiting time for the mth occurrence of the pattern in an infinite series of Bernoulli trials. In particular, the reliability is equal to the probability that the mth occurrence of the pattern is after n trials. Denote by WT the waiting time for the mth occurrence of the pattern, and its probability generating function by $\mathcal{G}_{WT}$. The relation between the generating functions of the reliability and the waiting time is given by

$$\mathcal{G}(s) = \frac{1 - \mathcal{G}_{WT}}{1 - s} \tag{2}$$

(Feller, p. 265). This relation is used for systems such as the *m-consecutive-k-out-of-n* system.

The procedure that was described above (steps 1-4) leads to generating functions of a special form, namely, rational functions. In many cases this ratio of polynomials is very complicated, and the ordinary way of obtaining the reliability by differentiation is laborious. One alternative is to use a combinatorial method (Stanley, 1994) that leads to a recursive formula for the reliability. However, a recursive expression means that computing the reliability of a system with n components involves computing all the reliabilities of similar systems with $1, 2, \ldots, n-1$ components. We suggest a different technique, that is suitable for rational functions and leads to a non-recursive expression for the system reliability:

Step 5: Expand the rational generating function into partial fractions (Feller, 1968).

This method is especially advantageous for systems with many components, as it involves fewer computations.

The partial fraction expansion method proceeds as follows: The generating function is expressed as a ratio of polynomials $\mathcal{G}(s) = \frac{N(s)}{D(s)}$, that do not have common roots. Next, the ratio is expanded into partial fractions:

$$\mathcal{G}(s) = \sum_{i} \sum_{j=1}^{m_i} \frac{\rho_{ij}}{(s_i - s)^j} \tag{3}$$

where s_i are the distinct roots of $D(s)$, each of multiplicity m_i, and ρ_{ij} are functions of the roots. Using an adequate infinite geometric series, $\mathcal{G}(s)$ can then be expressed as

$$\mathcal{G}(s) = \sum_{n=0}^{\infty} R_n s^n \tag{4}$$

where R_n is the required reliability, and is a function of s_i and ρ_{ij} (for more details, see appendix A and Shmueli & Cohen, 2000). The partial fraction expansion method, which was proposed by Feller for rational generating function, did not lead to exact probabilities due to computational difficulty. However, today many standard software packages (e.g.

Matlab, Maple, and Mathematica) have a partial fraction expansion procedure, which yields very accurate numerical reliabilities for the above roots and the constants. We take advantage of these advances to make Feller's theoretical method practical.

In the next subsections we apply our method to several consecutive-type systems and obtain an expression for their reliabilities and generating functions. We also give simple and efficient computer programs that yield numerical results for a given system.

A. Consecutive-k-out-of-n:F Systems

In *consecutive-k-out-of-n*:F systems, the component-pattern that causes the system to fail is a sequence of k failures, which we denote by $\underbrace{FF\ldots F}_{k\text{ times}}$. We denote the probability that this pattern will be completed on the tth component by u_t ($t = k, \ldots, n$). A recurrence relation for u_t is given by (Feller, 1968):

$$u_t = p^k - pu_{t-1} - p^2 u_{t-2} - \cdots - p^{k-1} u_{t-k+1} \tag{5}$$

where $p = Pr(F)$ and $u_0 = 1$. To obtain $U(s)$, the generating function of u_t, we multiply by s^t and sum from zero to infinity. The reliability generating function is then

$$\mathcal{G}(s) = [(1-s)U(s)]^{-1} = \frac{1-(ps)^k}{1-s+qp^k s^{k+1}} \tag{6}$$

A recursive formula for the reliability (as a function of n) can then be obtained directly, since the generating function is a rational function (Stanley, 1994). In this case the expression is given by

$$R_n = \begin{cases} 1 & ,0 \le n < k \\ 1-p^k & ,n = k \\ R_{n-1} - qp^k R_{n-k-1} & ,n > k \end{cases} \tag{7}$$

To obtain a non-recursive formula, which is especially advantageous for systems with many components, we use partial fraction expansion. Expanding the generating function (6), leads to the expression for the reliability

$$R_n = \sum_{i=1}^{k+1} \frac{\rho_i}{s_i^{n+1}} \tag{8}$$

where $s_1, \ldots, s_k$ are the k distinct roots of the polynomial $1-s+qp^k s^{k+1}$ with the additional root $s_{k+1} = 1/p$, and ρ_i are given by

$$\rho_i = \frac{(1-(ps_i)^k)/(1-ps_i)}{\Pi_{i\neq i'}(s_i - s_{i'})} \tag{9}$$

In practice, this can be computed directly by using a partial fraction procedure that exists in many standard software packages. To illustrate the simplicity of such a procedure, we present a short program in Matlab that evokes the `residue` function (which computes the roots s_i and coefficients ρ_i) and uses it to compute the system reliability for a *consecutive-k-out-of-n*:F system:

```
[R, P, K] = residue( [-p^k,zeros(1,k-1),1] ,
            [(1-p)*p^k,zeros(1,k-1),-1,1] )


sum = 0;
for ( i = 1:k+1 )
    sum = sum + R(i) / P(i)^(n+1) ;
end
Reliability = abs(sum) ;
```

(replace k, p, and n above with the required numerical values). This simple program can be easily modified for use with any other software package that has a partial fraction expansion procedure.

B. m-consecutive-k-out-of-n:F Systems

The reliability in this case is derived through the relation to the waiting time for the mth pattern of k consecutive failed components in an infinite series of Bernoulli trials (for more details, see appendix B). The generating function is given by:

$$\mathcal{G}(s) = \frac{(1-s+qp^k s^{k+1})^m - (ps)^{mk}(1-ps)^m}{(1-s)(1-s+qp^k s^{k+1})^m} \tag{10}$$

Since it is still a rational function, we can obtain the reliability using one of the two methods discussed above. In this case it is easier to obtain a non-recursive formula, as the roots of the denominator are 1 and the same roots as the *consecutive-k-out-of-n*:F case (excepts now, they are of multiplicity m). In this case, the generating function is expanded into an expression of the form given in (3), and the reliability is then expressed as the sum:

$$R_n = \sum_{j=1}^{m} (-1)^j \binom{n+j-1}{n} \sum_{i=1}^{k+2} \frac{\rho_{ij}}{s_i^{n+j}} \tag{11}$$

Where $s_1, \ldots, s_{k+1}$ are the roots computed in the previous subsection, with the addition of the root $s_{k+2} = 1$, and ρ_{ij} are functions of the roots. Like in the *consecutive-k-out-of-n* case, a practical method for computing this expression is by using a built-in partial fraction expansion procedure in a standard software package. Below is an example of a program in Matlab that yields the reliability of a *m-consecutive-k-out-of-n*:F system:

```
s = sym('s');
NumCoeffs =  sym2poly( (1-s+(1-p)*p^k *s^(k+1))^m- (p*s)^(k*m)*(1-p*s)^m);
DenomCoeffs = sym2poly ( (1-s)*(1-s+(1-p)*p^k *s^(k+1))^m );
[R, P, K] = residue( NumCoeffs, DenomCoeffs )


sum2 = 0
for ( j = 1:m )
   mult =  (-1)^j * factorial(n+j-1) / (factorial(n)*factorial(j-1)) ;
   sum1 = 0;
   for ( i = j:m:m*(k+1)+1 )
      sum1 = sum1 + R(i) / P(i)^(n+j) ;
   end
   sum2 = sum2 + mult * sum1 ;
end
Reliability = sum2 ;
```

(replace k, p, and n above with the required numerical values).

C. r-within-consecutive-k-out-of-n:F Systems

In this case, the component-types that cause the system to fail include all the possibilities of r failed components within a "window" of k consecutive components. For simplicity, we illustrate the method for $r = 2$ and $k > 2$ (two failed components within a window of k consecutive components). There are $k-1$ patterns that cause the system to fail: $FF, FSF, FSSF, \ldots, FS...SF$. We denote by $u_{t,1}$, $u_{t,k-1}$ the probabilities that each of the above patterns is completed on the tth component, respectively. Recurrence relations for these probabilities are

$$\begin{aligned} u_{t,1} &= p^2 - p\sum_{i=1}^{k-1} u_{t-1,i} \quad \text{for } n \geq 2 \\ u_{t,2} &= p^2 q - pq\sum_{i=1}^{k-1} u_{t-2,i} \quad \text{for } n \geq 3 \\ &\vdots \\ u_{t,k-1} &= p^2 q^{k-2} - pq^{k-1}\sum_{i=1}^{k-1} u_{t-2,i} \quad \text{for } n \geq k \end{aligned} \tag{12}$$

In order to find the generating functions of $u_{t,i}$, which are defined as

$$U_i(s) = \sum_{t=0}^{\infty} u_{t,i} s^t \tag{13}$$

we multiply the equations in (12) by s^t and sum to infinity. Summing the $k-1$ equations and equating the left and right sides leads to the following equation

$$\sum_{i=1}^{k-1} U_i(s) - (k-1) = \frac{p^2 \sum_{j=1}^{k-2} (qs)^j}{1-s} - ps\sum_{j=1}^{k-2}(qs)^j \sum_{i=1}^{k-1} U_i(s). \tag{14}$$

Rearranging this expression and using (1) we obtain the reliability generating function:

$$\mathcal{G}(s) = \frac{1 + ps\sum_{j=0}^{k-2}(qs)^j}{1 - qs - pq^{k-1}s^k} = \frac{1 - s(q-p) - pq^{k-1}s^k}{1 - 2qs + q^2s^2 - pq^{k-1}s^k + pq^k s^{k+1}} \tag{15}$$

A non-recursive expression can be computed using a partial fraction expansion software procedure, such as the following Matlab program:

```
[R, P, K] = residue( [-p*(1-p)^(k-1),zeros(1,k-2),2*p-1,1] ,
             [p*(1-p)^k,-p*(1-p)^(k-1),zeros(1,k-3),(1-p)^2,-2*(1-p),1] )


sum = 0;
for ( i = 1:k+1 )
    sum = sum + R(i) / P(i)^(n+1) ;
end
```

```
Reliability = abs(sum) ;
```

(replace p, k and n above with the required numerical values).

This general method can be applied to any value of r. For the popular case $k = r + 1$, it involves the solution of a linear equation system of rank $(k - 1)$. For small to moderate values of k, this can be done symbolically, using a symbolic software package (e.g. Maple or Mathematica). For $k > r + 1$ and for large values of r and k, the numerical value of p should be plugged into the equations, and the system solved numerically.

D. Deriving the Reliability for General Failure Patterns

The method described in the beginning of this section can be applied to *any* type of failure pattern/s, which cause a system to fail. For example, for a system that fails if the pattern $FSFS$ occurs, we write a recurrence relation for u_m (the probability that $FSFS$ is completed on the mth component). Using the generating function of u_m, the reliability generating function can be obtaind from equation (1), and an expression for the reliability derived.

IV. Reliability of Systems with Multi-state components

Our method can be used to obtain the reliability of systems with multi-state components. The only difference from the binary case, where each component is either on or off, is in specifying the component-patterns that lead to the system failure (step 1).

We deal with two types of relations between the possible component states:

1. Exclusive states, where each component can be in one state only.
2. Inclusive, gradual states. Each state is included in the next state.

For simplicity, we illustrate the application of our method to a system with three-state components. We use the following notation:

S	The component is on
F_I	The component failed by type I failure
F_{II}	The component failed by type II failure

A. Exclusive Types of Failures

A simple example is a system with three-state components, where the temperature of the components can vary. Within a certain temperature range the component is operational (state 1), while above or below that, the component fails. In this case a component can either fail because it is below the required temperature (state 2) *or* because it is above the required temperature (state 3). The states are therefore mutually exclusive. Consider a consecutive-k-out-of-n system with mutually exclusive three-state components. Here the system fails when there exist k_1 consecutive failed components of type I, or k_2 failed components of type II. Each component can either be operational (S), failed by type I (F_I) *or* failed by type II (F_{II}). Denote the probabilities of a failure of type I and II by p_1 and p_2, respectively. The probability of an operational component is then $1 - p_1 - p_2$.

For this system, the two distinct component-patterns that cause system failure are $\underbrace{F_I F_I \ldots F_I}_{k_1 \text{ times}}$ and $\underbrace{F_{II} F_{II} \ldots F_{II}}_{k_2 \text{ times}}$. Following steps 2-4, we obtain the reliability generating function:

$$\mathcal{G}(s) = \frac{\left[1-(p_1 s)^{k_1}\right]\left[1-(p_2 s)^{k_2}\right]}{1 - s + q_1 p_1^{k_1} s^{k_1+1} + q_2 p_2^{k_2} s^{k_2+1} - p_1^{k_1} p_2^{k_2} s^{k_1+k_2} - p_1^{k_1} p_2^{k_2} (1-p_1-p_2) s^{k_1+k_2+1}} \quad (16)$$

A non-recursive calculation of the system reliability, that is based on expanding (16) into partial fractions can be obtained via a simple program, such as the following Matlab program:

```
s = sym('s');
 NumCoeffs =  sym2poly( (1-(p1*s)^k1)*(1-(p2*s)^k2) );
 DenomCoeffs = sym2poly ( 1-s+(1-p1)*p1^k1 *s^(k1+1) + (1-p2)*p2^k2 *s^(k2+1) -
             -(p1*s)^k1 * (p2*s)^k2 - p1^k1 * p2^k2 * (1-p1-p2)^(k1+k2+1)  );


[R, P, K] = residue( NumCoeffs, DenomCoeffs )


 sum = 0;
 for ( i = 1:k1+k2+1 )
     sum = sum + R(i) / P(i)^(n+1) ;
 end
```

```
Reliability = abs(sum) ;
```

(replace $k1, k2, p1, p2$, and n above with the required numerical values).

This method is general and can be applied to various consecutive-type systems, with components that can have any fixed number of mutually exclusive states.

B. Gradual Degradation

A simple example for this type of system is one where each component can be perfectly operative (S), have a minor failure (F_I), or a major failure (F_{II}). Each of these three states is considered to be inclusive of the previous state. In other words, the degree of the failure is gradual, from non-existent to totally failed with a medium in-between state.

Let us assume that the system fails following a run of k_2 type II (major) failed components, or a run of k_1 failed components of either type I or II (major/minor). In many cases it makes sense to assume that $k_1 > k_2$. For example, a system with $k_1 = 3$ and $k_2 = 2$ fails iff it consists of two successive major-failed components ($F_{II}F_{II}$), or three consecutive failed components of any type ($F_IF_IF_I$, $F_IF_IF_{II}$,$F_IF_{II}F_I$,or $F_{II}F_IF_I$). Note that combinations such as F_{II}, F_{II}, F_I are excluded from the second set. Since these five patterns are mutually exclusive, step 1 is completed. Following steps 2,3, and 4 will lead to the generating function and reliability of this system.

V. Dependent Failures of Components

The five-step method described in Section III can be adapted for cases where failures of components are statistically dependent on the state of preceding components. Except for step 2, i.e. creating the recurrence relations, the rest of the steps remain unchanged. To illustrate how the dependence in incorporated into the recurrence relations, we look at a single-step Markovian dependence. This means that the failure of component t depends on the state of component $t-1$. Denoting by X_t the state of the tth component, we specify the conditional probability of a failure by

$$Pr(X_t = F \mid X_{t-1} = F) = \lambda \tag{17}$$

and the unconditional probability of a failure by $Pr(X_t = F) = p$. For a *Consecutive-k-out-of-n*:F System, we already showed that the failure causing pattern is a sequence of k

successive failures. The recurrence relation for u_t in this case is then

$$u_t = p\lambda^{k-1} - \lambda u_{t-1} - \lambda^2 u_{t-2} - \cdots - \lambda^{k-1} u_{t-k+1} \tag{18}$$

Following steps 3 and 4 leads to the generating function

$$\mathcal{G}(s) = \frac{1-(\lambda s)^k}{1-s+(p-\lambda)\lambda^{k-1}s^k + q\lambda^k s^{k+1}} \tag{19}$$

which can then be expanded to obtain the probability function. Note that when $\lambda = p$ this reduces to the statistically independent case.

The same method can be used for dependence of higher order, by defining the relevant conditional probabilities and incorporating them into the recurrence relations.

VI. Concluding Remarks

We have presented a new method that gives exact expressions for the reliability and its generating functions for consecutive-type systems. The method consists of creating recurrence relations and solving a set of linear equations that lead to an expression for the reliability generating function. The reliability is then obtained by partial fraction expansion, which is a standard procedure in many software packages. The short programs included in this paper are written in Matlab, but they are simple enough to be modified into other languages.

Using a non-recursive formula for computing the reliability of a consecutive-type system is advantageous over recursive formulas for large systems. In comparison to the n (=number of components) computations that the recursive formulas require, the non-recursive formula only requires the computation of the roots of a polynomial of order k. In large systems, the magnitude of n is much larger than that of k.

Our method can be used for many generalizations, including systems with multi-state components or statistically dependent components. The five-step reasoning is general and suitable for any type of component-patterns that cause system failure.

Appendix

I. Expanding $\mathcal{G}(s)$ into partial fractions

In order to expand a rational generating function into partial fractions, it is first expressed as an infinite geometric series. If the roots of the denominator are of multiplicity

1, then $\mathcal{G}(s)$ is expressed as the sum:

$$\mathcal{G}(s) = \sum_i \frac{\rho_i}{(s_i - s)} = \sum_i \frac{\rho_i}{s_i} \cdot \frac{1}{1 - s/s_i} = \sum_i \frac{\rho_i}{s_i} \sum_{n=0}^{\infty} \left(\frac{s_i}{s}\right)^n = \sum_{n=0}^{\infty} \sum_i \frac{\rho_i}{s_i^{n+1}} s^n \tag{20}$$

Since the reliability generating function is defined as $\sum_{n=0}^{\infty} R_n s^n$, using (20) the reliability itself is given by

$$R_n = \sum_i \frac{\rho_i}{s_i^{n+1}} \tag{21}$$

If the roots of the denominator are of multiplicity $m > 1$, then the generating function is expanded using a different geometric series:

$$\begin{aligned} \mathcal{G}(s) &= \sum_i \sum_j^m \frac{\rho_{ij}}{s_i - s)^j} = \sum_i \sum_{j=1}^m (-1)^j \cdot \frac{\rho_{ij}}{s_i^j} \cdot \frac{1}{(1 - s/s_i)^j} = && (22) \\ &= \sum_i \sum_{j=1}^m (-1)^j \cdot \frac{\rho_{ij}}{s_i^j} \sum_{n=0}^{\infty} \binom{n+j-1}{n} \left(\frac{s_i}{s}\right)^n = && (23) \\ &= \sum_{n=0}^{\infty} \sum_{j=1}^m (-1)^j \binom{n+j-1}{n} \sum_i \frac{\rho_{ij}}{s_i^{n+j}} \cdot s^n \end{aligned}$$

which then gives the reliability

$$R_n = \sum_{j=1}^m (-1)^j \binom{n+j-1}{n} \sum_i \frac{\rho_{ij}}{s_i^{n+j}} \tag{24}$$

II. The relation between waiting times and the *m-consecutive-k-out-of-n* system reliability

To obtain the reliability generating function for the *m-consecutive-k-out-of-n*:F system, we use a shortcut that is based on the relations between two sets of generating functions. First, we find the generating function for the waiting time for the first sequence of k consecutive failures in a Bernoulli series ($\mathcal{G}_{WT}(s)$), using the relation (2) to the *consecutive-k-out-of-n* reliability generating function ($\mathcal{G}(s)$):

$$\mathcal{G}_{WT}(s) = 1 - (1 - s)\mathcal{G}(s) = \frac{(ps)^k (1 - ps)}{1 - s + qp^k s^{k+1}} \tag{25}$$

Next, we relate this generating function to that for the waiting time for the mth sequence of k consecutive failures in a infinite series of Bernoulli trials, denoted by $\mathcal{G}_{WT}^m(s)$. Since the waiting time for the mth sequence is a convolution of m waiting times for the first such sequence, the relation between the two generating functions is

$$\mathcal{G}_{WT}^m(s) = [\mathcal{G}_{WT}(s)]^m \tag{26}$$

Finally, the the *m-consecutive-k-out-of-n*:F reliability is related to waiting time for the mth sequence of k consecutive failures in Bernoulli trials through (2), thus yielding the expression in (10):

$$\mathcal{G}(s) = \frac{1-\mathcal{G}_{WT}^m(s)}{1-s} = \frac{(1-s+qp^k s^{k+1})^m - (ps)^{mk}(1-ps)^m}{(1-s)(1-s+qp^k s^{k+1})^m} \tag{27}$$

References

[1] Aki, S. & Hirano, K., "Lifetime Distribution and Estimation Problems of Consecutive-k-out-of-n:F Systems", *Ann. Inst. Statist. Math.*, **48**,no. 1, pp. 185-199, 1996.

[2] Chao, M. T., Fu, J. C., and Koutras, M. V., "Survey of Reliability Studies of Consecutive-k-out-of-n:F & Related Systems", *IEEE Transactions on Reliability*, **44**,no. 1, pp. 120-127, 1995.

[3] Chao, M. T. and Lin, G. D., "Econimical Design of Large Consecutive-k-out-of n:F Systems," *IEEE Transactions on Reliability*, vol. R-33, pp. 411-413, 1984.

[4] Derman, C., Liberman, G. J., and Ross, S., "On the Consecutive-k-out-of n:F System," *IEEE Transactions on Reliability*, vol. R-31, pp. 57-63, 1982.

[5] Feller, W., "An Introduction to Probability Theory and Its Applications", vol. I, John Wiley & Sons, Inc., 1968.

[6] Fu, J. C., "On Reliability of a Large Consecutive-k-out-of-n:F System with (k-1)-step Markov Dependence," *IEEE Transactions on Reliability*, vol. R-36, No. 1, pp. 75-77, 1987.

[7] Fu, J. C., "Distribution Theory of Runs and Patterns with a Sequence of Multi-State Trials," *Statistica Sinica* , vol.6, pp. 957-974, 1996.

[8] Koutras, M. V., "On a Markov Chain Approach for the Study of Reliability Structures", *J. Appl. Prob.*, **33**, pp. 357-367, 1996.

[9] Shmueli, G. and Cohen, A., "Run-Related Probability Functions Applied to Sampling Inspection," *Technometrics*, **42**, no. 2, pp. 188-202, 2000.

[10] Stanley, R. P., *Enumerative Combinatorics*, (Vol. I), Wadsworth & Brooks/Cole Advanced Books & Software, 1994.

Galit Shmueli is a Visiting Assistant Professor at the Department of Statistics at Carnegie-Mellon University, Pittsburgh, PA. She received her Ph.D. in Statistics from the Israel Institute of Technology in 2000. Her research interests include industrial applications that are based on runs, and creating Web applications for statistical methods that are used in industry.