Advanced discrete event simulation methods with application to importance measure estimation

Arne B. Huseby
University of Oslo, Norway

Kristina A. Eide
FFI, Kjeller, Norway

Stefan L. Isaksen
DNV Energy, Høvik, Norway

Bent Natvig
University of Oslo, Norway

Jørund Gåsemyr
University of Oslo, Norway

In the present paper we use discrete event simulation in order to analyze a *binary monotone system* of repairable components. Asymptotic statistical properties of such a system, e.g., the asymptotic system availability and component criticality, can easily be estimated by running a *single* discrete event simulation on the system over a sufficiently long time horizon, or by working directly on the stationary availabilities. Sometimes, however, one needs to estimate how the statistical properties of the system evolve over time. In such cases it is necessary to run many simulations to obtain a stable curve estimate. At the same time one needs to store much more information from each simulation. A *crude* approach to this problem is to sample the system state at fixed points of time, and then use the mean values of the states at these points as estimates of the curve. Using a sufficiently high sampling rate a satisfactory estimate of the curve can be obtained. Still, all information about the process between the sampling points is thrown away. To handle this issue, we propose an alternative sampling procedure where we utilize process data between the sampling points as well. This simulation method is particularly useful when estimating various kinds of component importance measures for repairable systems. As explained in (Natvig and Gåsemyr 2008) such measures can often be expressed as weighted integrals of the time-dependent Birnbaum measure of importance. By using the proposed simulation methods, stable estimates of the Birnbaum measure as a function of time are obtained and combined with the appropriate weight function, and thus producing the importance measure of interest.

1 INTRODUCTION

Discrete event models are typically used in simulation studies to model and analyze pure jump processes. A discrete event model can be viewed as a system consisting of a collection of elementary pure jump processes evolving asynchronously and interacting at random points of time. The interactions are modelled as a sequence of *events* typically affecting the state of some or all the elementary processes. Between these events, however, the states are considered to be constant. Hence, in order to describe how the system evolves, only the points of time where events happen need to be considered. For an extensive formal introduction to discrete event models see (Glasserman and Yao 1994).

Stationary statistical properties of a system, can easily be estimated by running a *single* discrete event simulation on the system over a sufficiently long time horizon, or by working directly on the stationary availabilities. Sometimes, however, one needs to estimate how the statistical properties of the system evolve over time. In such cases it is necessary to run many simulations to obtain stable results. Moreover, one must store much more information from each simulation. A *crude* approach to this problem is to sample the system state at fixed intervals of time, and then use the mean values of the states at these points as estimates of the corresponding statistical properties. Using a sufficiently high sampling rate, i.e., short intervals between sampling points, a satisfactory estimate of the full curve can be obtained. Still, all information about the process between the sampling points is thrown away. Thus, we propose an alternative sampling procedure where we utilize process data between the sampling points as well.

In order to illustrate the main ideas we use discrete events in order to analyze a *multicomponent binary monotone system* of repairable components. In a companion paper (Natvig, Eide, Gåsemyr, Huseby, and Isaksen 2008) the simulation technology developed in the present paper, is used to estimate the Natvig measures of component importance in an offshore oil and gas production system.

2 BASIC RELIABILITY THEORY

We start out by briefly reviewing basic concepts of reliability theory. See (Barlow and Proschan 1981). A *binary monotone system* is an ordered pair (C, ϕ) where $C = \{1, \ldots, n\}$ is a nonempty finite set, and ϕ is a binary function. The elements of C are interpreted as components of some technological system. Each component, as well as the system itself can be either functioning or failed. We denote the state of component i at time $t \geq 0$ by $X_i(t)$, where $X_i(t) = 1$ if i is functioning at time t, and zero otherwise, $i = 1, \ldots, n$. We also introduce the component state vector $\boldsymbol{X}(t) = (X_1(t), \ldots, X_n(t))$. The function ϕ is called the *structure function* of the system, and expresses the state of the system as a function of the component state vector, and is assumed to be nondecreasing in each argument. Thus, $\phi = \phi(\boldsymbol{X}(t)) = 1$ if the system is functioning at time t and zero otherwise.

In the present paper we consider systems with repairable components. Thus, for $i = 1, \ldots, n$ and $j = 1, 2, \ldots$ let:

$$U_{ij} = \text{The } j\text{th lifetime of the } i\text{th component.}$$

$$D_{ij} = \text{The } j\text{th repair time of the } i\text{th component.}$$

We assume that U_{ij} has an absolutely continuous distribution with mean value $\mu_i < \infty$, while D_{ij} has an absolutely continuous distribution with mean value $\nu_i < \infty$, $i = 1, \ldots, n$, $j = 1, 2, \ldots$. All lifetimes and repair times are assumed to be independent. Thus, in particular the component processes $\{X_1(t)\}, \ldots, \{X_n(t)\}$ are independent of each other.

Let $A_i(t)$ be the availability of the ith component at time t, i.e., the probability that the component is functioning at time t. That is, for $i = 1, \ldots, n$ we have:

$$A_i(t) = \Pr(X_i(t) = 1) = E[X_i(t)].$$

The corresponding stationary availabilities are given by:

$$A_i = \lim_{t \to \infty} A_i(t) = \frac{\mu_i}{\mu_i + \nu_i}, \qquad i = 1, \ldots, n. \quad (1)$$

Introduce $\boldsymbol{A}(t) = (A_1(t), \ldots, A_n(t))$ and $\boldsymbol{A} = (A_1, \ldots, A_n)$. The system availability at time t is given by:

$$A_\phi(t) = \Pr(\phi(\boldsymbol{X}(t)) = 1) = E[\phi(\boldsymbol{X}(t))] = h(\boldsymbol{A}(t)),$$

where h is the system's reliability function. The corresponding stationary availability is given by:

$$A_\phi = \lim_{t \to \infty} A_\phi(t) = h(\boldsymbol{A}) \quad (2)$$

The component i is said to be *critical* at time t if $\psi_i(\boldsymbol{X}(t)) = \phi(1_i, \boldsymbol{X}(t)) - \phi(0_i, \boldsymbol{X}(t)) = 1$. We will refer to $\psi_i(\boldsymbol{X}(t))$ as the *criticality state* of component i at time t. The Birnbaum measure of importance of component i at time t, is defined as the probability that component i is critical at time t, and denoted $I_B^{(i)}(t)$. See (Birnbaum 1969). Thus,

$$\begin{aligned} I_B^{(i)}(t) &= \Pr(\psi_i(\boldsymbol{X}(t)) = 1) = E[\psi_i(\boldsymbol{X}(t))] \quad (3) \\ &= h(1_i, \boldsymbol{A}(t)) - h(0_i, \boldsymbol{A}(t)). \end{aligned}$$

The corresponding stationary measure is given by:

$$I_B^{(i)} = \lim_{t \to \infty} I_B^{(i)}(t) = h(1_i, \boldsymbol{A}) - h(0_i, \boldsymbol{A}). \quad (4)$$

3 DISCRETE EVENT SIMULATION

Since the component state processes of a binary monotone system are pure jump processes with events corresponding to the failures and repairs of the components, they are well suited to be described as a discrete event model. Thus, let (C, ϕ) be a binary monotone system with component state processes $\{X_1(t)\}, \ldots, \{X_n(t)\}$. For $i = 1, \ldots, n$ we denote the events affecting the process $\{X_i(t)\}$ by $E_{i1}, E_{i2}, \ldots$. Moreover, let $T_{i1}, T_{i2}, \ldots$ be the corresponding points of time for these events. Note that since we assumed that all lifetimes and repair times have absolutely continuous distributions, all the events happen at distinct points of time almost surely. That is, all the T_{ij}s are distinct numbers. We assume that the events are sorted with respect to their respective points of time, so that $T_{i1} < T_{i2} < \cdots$.

At the system level the event set is the *union* of all the component event sets. Let $E^{(1)}, E^{(2)}, \ldots$ denote the system events sorted with respect to their respective points of time, and let $T^{(1)} < T^{(2)} < \cdots$ be the corresponding points of time. Thus, each system event corresponds to a unique component event.

In order to keep track of the events, they are usually organized in a dynamic queue sorted with respect to the points of time of the events. The component processes place their upcoming events into the queue where they stay until they are processed. More specifically, at time zero each component typically starts out by being functioning, and places its first failure event into the queue. As soon as all these failure events have been placed into the queue, the first event in the queue is processed. That is, the *system time* is set to the time of the first event, and the event is taken out of the queue and passed on to the component owning this event. The component then updates its state, generates a new event, in this case a repair event, which is placed into queue, and notifies the system about its new state so that the system state can be updated as well. Then the next event in the queue is processed in the same fashion, and so forth until the system time reaches a certain predefined point of time. Note that since the component events are generated as part of the event processing, the number of events in the queue stays constant.

3.1 *Sampling events*

Although the system state and component states stay constant between events, it may still be of interest to log the state values at predefined points of time. In order to facilitate this, we introduce yet another type of event, called a *sampling event*. Such sampling events will typically be spread out evenly on the timeline. Thus, if $e_1, e_2, \ldots$ denote the sampling events, and $t_1 < t_2 < \cdots$ are the corresponding points of time, we would typically have $t_j = j \cdot \Delta$ for some suitable number $\Delta > 0$.

The sampling events will be placed into the queue in the same way as for the ordinary events. As a sampling event is processed, the next sampling event will be placed into the queue. Thus, at any time only one sampling event needs to be in the queue.

3.2 *Updating system and criticality states*

In principle one must update the system state every time there is a change in the component states. For large complex systems, these updates may slow down the simulations considerably. Thus, whenever possible one should avoid computing the system state. Fortunately, since the structure function of a binary monotone system is non-decreasing in each argument, it is possible to reduce the updating to a minimum. To explain this in detail, we consider the event E_{ij} affecting component i. Let T_{ij} be the corresponding point of time, and let $\boldsymbol{X}(T_{ij}^-)$ denote the value of the component state vector immediately before E_{ij} occurs.

If E_{ij} is a failure event of component i, i.e., $X_i(T_{ij}^-) = 1$ and $X_i(T_{ij}) = 0$, then the event cannot change the system state if the system is already failed, i.e., $\phi(\boldsymbol{X}(T_{ij}^-)) = 0$. Similarly, if E_{ij} is a repair event of component i, i.e., $X_i(T_{ij}^-) = 0$ and $X_i(T_{ij}) = 1$, this event cannot change the system state if the system is already functioning, i.e., $\phi(\boldsymbol{X}(T_{ij}^-)) = 1$. Thus, we see that we only need to recalculate the system state whenever:

$$\phi(\boldsymbol{X}(T_{ij}^-)) \neq X_i(T_{ij}). \tag{5}$$

Hence, the number of times we need to recalculate the system state is drastically reduced.

In cases where we keep track of the criticality state of each of the components, we can simplify the calculations even further by noting that the system state is changed as a result of the event E_{ij} if and only if component i is critical at the time of the event. Moreover, if i is critical, and E_{ij} is a failure event, it follows that the system fails as a result of this event, i.e., $\phi(\boldsymbol{X}(T_{ij})) = 0$. If on the other hand i is critical, and E_{ij} is a repair event, it follows that the system become functioning as a result of this event, i.e., $\phi(\boldsymbol{X}(T_{ij})) = 1$. Thus, we see that in this setup all the calculations we need to carry out, are related to the updating of the criticality states.

A similar technique can be used when updating the criticality states of the components. Thus, we consider the event E_{ij} affecting the state of component i. We first note that the criticality state function of component i, $\psi_i(\boldsymbol{X}(t)) = \phi(1_i, \boldsymbol{X}(t)) - \phi(0_i, \boldsymbol{X}(t))$ does not depend on the state of component i. Thus, the event E_{ij} does not have any impact on the criticality state of i. However, E_{ij} may still change the criticality state of other components in the system even when the system state remains unchanged. Thus, let $k \neq i$ be another component, and consider its criticality state function $\psi_k(\boldsymbol{X}(T_{ij}))$.

If $X_k(T_{ij}) = 1$ and $\phi(\boldsymbol{X}(T_{ij})) = 0$, it follows that $\phi(1_k, \boldsymbol{X}(T_{ij})) = \phi(0_k, \boldsymbol{X}(T_{ij})) = 0$. Thus, in this case we must have $\psi_k(\boldsymbol{X}(T_{ij})) = 0$. On the other hand, if $X_k(T_{ij}) = 0$ and $\phi(\boldsymbol{X}(T_{ij})) = 1$, it follows that $\phi(1_k, \boldsymbol{X}(T_{ij})) = \phi(0_k, \boldsymbol{X}(T_{ij})) = 1$. Thus, we must have $\psi_k(\boldsymbol{X}(T_{ij})) = 0$ in this case as well. Hence, we see that a necessary condition for component k to be critical at time T_{ij} is that:

$$\phi(\boldsymbol{X}(T_{ij})) = X_k(T_{ij}). \tag{6}$$

Utilizing these observations reduces the need to recalculate the criticality states.

3.3 *Estimating availability and importance*

Stationary availability and importance measures are typically easy to derive. If the system under consideration is not too complex, these quantities can be calculated analytically using (1), (2) and (4). For larger complex systems one may estimate the availability and importance using Monte Carlo simulations. A fast simulation algorithm for this is provided in (Huseby and Naustdal 2003). Alternatively, estimates can be obtained by running a *single* discrete event simulation on the system over a sufficiently long time horizon

Here, however, we focus on the problem of estimating the system availability $A(t)$ and the component importance measures $I_B^{(1)}(t), \dots, I_B^{(n)}(t)$ as functions of t. Ideally we would like to estimate these quantities for any $t \geq 0$. For practical purposes, however, we have to limit the estimation to a finite set of points. More specifically, we will estimate $A(t)$ for $t \in \{t_1, \dots, t_N\}$, i.e., the set of the N first sampling points. For the points of time between the sampling points, we just use linear interpolation to obtain the curve estimate.

A simple approach to this problem is to run M simulations on the system, where each simulation covers the time interval $[0, t_N]$. In each simulation we sample the values of ϕ and $\psi_1, \dots, \psi_n$ at each sampling point $t_1, \dots, t_N$. We denote the sth simulated result of the component state vector process by $\{\boldsymbol{X}_s(t)\}$, $s = 1, \dots, M$, and obtain the following estimates for $j = 1, \dots, N$:

$$\hat{A}(t_j) = \frac{1}{M} \sum_{s=1}^{M} \phi(\boldsymbol{X}_s(t_j)), \tag{7}$$

$$\hat{I}_B^{(i)}(t_j) = \frac{1}{M} \sum_{s=1}^{M} \psi_i(\boldsymbol{X}_s(t_j)). \tag{8}$$

We will refer to these estimates as *pointwise estimates*. It is easy to see that for $j = 1, \dots, N$, $\hat{A}(t_j)$ and $\hat{I}_B^{(i)}(t_j)$ are unbiased and strongly consistent estimates of $A(t_j)$ and $I_B^{(i)}(t_j)$ respectively. In order to estimate $A(t)$ and $I_B^{(i)}(t)$ between the sampling points, one may use interpolation. Using a sufficiently high sampling rate, i.e., a small value of Δ, a satisfactory estimate of the full curve can be obtained. Still, all information about the process between the sampling points is thrown away.

We now present an alternative approach where we utilize process data between the sampling points as well. As above we assume that the system is simulated M times with $t_1, \dots, t_N$ as sampling points, and let $\{\boldsymbol{X}_s(t)\}$ denote the sth simulated result of the component state vector process, $s = 1, \dots, M$. Then let $E_s^{(1)}, \dots, E_s^{(L_s)}$ denote the events in the interval $[0, t_N]$ in the sth simulation, *including* the sampling events, and let $T_s^{(1)} < \cdots < T_s^{(L_s)}$ be the corresponding points of time, $s = 1, \dots, M$. In particular we assume that the sampling events in the sth simulation are $E_s^{(k_{1s})}, \dots, E_s^{(k_{Ns})}$. Thus, $T_s^{k_{js}} = t_j = j \cdot \Delta$, for $j = 1, \dots, N$ and $s = 1, \dots, M$. It is also convenient to let $T_s^{(0)} = t_0 = k_{0s} = 0$, $s = 1, \dots, M$.

The idea now is to use average simulated availability and criticalities from each interval $(t_{j-1}, t_j]$, $j = 1, \dots, N$ as respective estimates for the availability and criticalities at the midpoints of these intervals. To simplify the formulas slightly, we introduce waiting times between the events. Thus, for $k = 1, \dots, L_s$, $s = 1, \dots, M$ let:

$$W_s^{(k)} = (T_s^{(k)} - T_s^{(k-1)}).$$

We then obtain the following estimates for $j = 1, \dots, N$:

$$\tilde{A}(\bar{t}_j) = \frac{1}{M\Delta} \sum_{s=1}^{M} \sum_{k=k_{(j-1)s}}^{k_{js}-1} \phi(\boldsymbol{X}_s(T_s^{(k)})) W_s^{(k+1)}, \tag{9}$$

$$\tilde{I}_B^{(i)}(\bar{t}_j) = \frac{1}{M\Delta} \sum_{s=1}^{M} \sum_{k=k_{(j-1)s}}^{k_{js}-1} \psi_i(\boldsymbol{X}_s(T_s^{(k)})) W_s^{(k+1)}, \tag{10}$$

where we have introduced the interval midpoints $\bar{t}_j = (t_{j-1} + t_j)/2$, $j = 1, \dots, N$. We will refer to these estimates as *interval estimates*. It is easy to see that for $j = 1, \dots, N$, $\tilde{A}(\bar{t}_j)$ and $\tilde{I}_B^{(i)}(\bar{t}_j)$ are unbiased and strongly consistent estimates of the corresponding average availability and criticality in the intervals $(t_{j-1}, t_j]$ respectively. By choosing Δ so that the availabilities and criticalities are relatively stable within each interval, the interval estimates are approximately unbiased estimates for $A(\bar{t}_j)$ and $I_B^{(i)}(\bar{t}_j)$ as well. In fact the resulting interval estimates tend to stabilize much faster than the pointwise estimates. In order to estimate $A(t)$ and $I_B^{(i)}(t)$ between the interval midpoints, one may again use interpolation. Note that since all process information is used in the estimates, satisfactory curve estimates can be obtained for a much higher value of Δ than the one needed for the pointwise estimates. In the next section we will demonstrate this on some examples.

4 NUMERICAL RESULTS

In order to illustrate the methods presented in Section 3 we consider a simple bridge structure shown in Figure 1. The components of this system are the five edges in the graph, labeled $1, \dots, 5$. The system is functioning if the source node s can communicate with the terminal node t through the graph. All the components in the system have exponential lifetime and repair time distributions with mean values 1 time unit. The objective of the simulation is to estimate $A(t)$ and $I_B^{(1)}(t), \ldots, I_B^{(5)}(t)$ for $t \in [0, 1000]$.

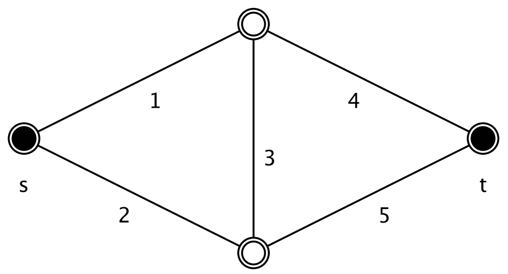

Figure 1: *A bridge system.*

All the simulations were carried out using a program called *Eventcue*[1] This program has an intuitive graphical user interface, and can be used to estimate availability and criticality of any undirected network system.

Since all the lifetimes and repair times are exponentially distributed with the *same mean*, it is easy to derive explicit analytical expressions for the component availabilities. To see this, we consider the ith component at a given point of time t and introduce $N_i(t)$ as the number of failure and repair events affecting component i in $[0, t]$. With times between events being independent and exponentially distributed with mean 1 it follows that $N_i(t)$ has a Poisson distribution with mean t. Moreover, component i is functioning at time t if and only if $N_i(t)$ is even. Thus, the ith component availability at time t is given by:

$$A_i(t) = \sum_{k=0}^{\infty} \Pr(N_i(t) = 2k) = \sum_{k=0}^{\infty} \frac{t^{2k}}{(2k)!} e^{-t}. \quad (11)$$

Using (11) one can verify numerically that all the component availabilities converge very fast towards their common stationary value, 0.5. As a result of this the system availability, $A(t)$, converges very fast towards its stationary value, 0.5, as well. In fact, for $t > 20$, numerical calculations show that $|A(t) - 0.5| < 10^{-15}$. Similarly, the Birnbaum measures of importance converges so that for $t > 20$, $|I_B^{(i)}(t) - 0.375| < 10^{-15}$, $i = 1, 2, 4, 5$, while $|I_B^{(3)}(t) - 0.125| < 10^{-15}$. Thus, for $t > 20$ the true values of all the curves are approximately constant. This makes it easy to evaluate and compare the quality of the different Monte Carlo estimates in this particular case.

[1]Eventcue is a java program developed at the Department of Mathematics, University of Oslo. The program is freely available at http://www.riscue.org/eventcue/.

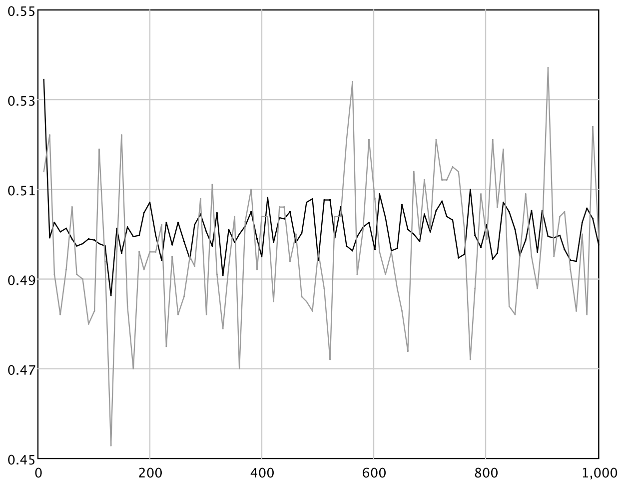

Figure 2: *Availability curve estimates*

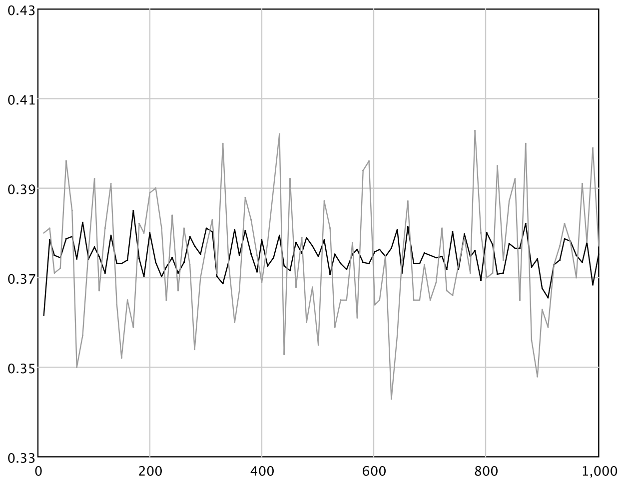

Figure 3: *Importance curve estimates*

Figure 2 and Figure 3 show respectively the availability curve and the criticality curve of component 1. The black curves are obtained using the interval estimates, while the gray curves show the corresponding pointwise estimate curves. In all cases we have used $M = 1000$ simulations and $N = 100$ sample points.

The plots clearly show the difference between the two methods. The black interval estimate curves are much more stable, and thus much closer to the true curve values, compared to the gray pointwise estimates.

One may think that increasing the number of sampling points would make the pointwise curve estimate better as more information is sampled. However, it turns out that the main effect of this is that the curve jumps more and more up and down. In fact with shorter intervals between sampling points the interval estimate becomes more unstable as well, and in the limit where the interval lengths go to zero, the two methods become equivalent. The only effective way of stabilizing the results for the pointwise curve estimate is to increase the number of simulations, i.e., M.

In Table 1 we have listed estimated standard devi-

Table 1: Standard deviations for the pointwise curve estimates

M	2000	4000	6000	8000
St.dev.	0.0121	0.0076	0.0062	0.0054

ations for pointwise curve estimates for different values of M. We see that the standard deviation shows a steady decline as M increases. The corresponding numbers for $M = 1000$ are 0.0055 for the interval curve estimate and 0.0148 for the pointwise estimate. Thus, in this particular case we see that to obtain a pointwise curve estimate with a comparable stability to the interval curve estimate, one needs about eight times as many simulations.

For the interval curve estimate it is possible to obtain an even smoother curve simply by increasing Δ. Still, in general Δ should not be made too large, as this could produce a curve where important effects are obscured. Thus, in order to obtain optimal results, one should try out different values for Δ, and balance smoothness against the need of capturing significant oscillation properties of the curve.

Now, if smoothness is important, it is of course possible to apply some standard smoothing technique, such as moving averages or exponential smoothing, to the pointwise curve estimate. While such post-smoothing would clearly make the curve smoother, this technique does not add any new information to the estimate. The main advantage with the interval curve estimates is that such estimates actually use information about all events. Especially in cases where events occur at a very high rate, this turns out to be a great advantage.

5 APPLICATIONS TO IMPORTANCE MEASURE ESTIMATION

In this section we shall explain how the sampling methods developed in Section 3 can be used to estimate more advanced importance measures like e.g., those introduced in (Natvig and Gåsemyr 2008) and applied in (Natvig, Eide, Gåsemyr, Huseby, and Isaksen 2008). In the context of the present paper the general idea can be explained as follows. As before we consider a binary monotone system (C, ϕ). Moreover, let $i \in C$ be a component in the system, and let $E_{i1}, E_{i2}, \ldots$ be the events affecting this component occurring respectively at $T_{i1} < T_{i2}, \cdots$. For each of these events we then introduce new *fictive* events $E'_{i1}, E'_{i2}, \ldots$ occurring respectively at $T'_{i1} < T'_{i2} < \cdots$. We assume that the fictive events always occur after their respective *real* events. That is, $T_{ij} \leq T'_{ij}$, $j = 1, 2, \ldots$. The fictive events could be some sort of action altering the state of the component throughout the interval between the real event and the corresponding fictive event. If E_{ij} is a failure event, then E'_{ij} could e.g., be a fictive failure event occurring as a result of the component being minimally repaired at T_{ij} and then functioning until T'_{ij}. Similarly, if E_{ij} is a repair event, one may consider actions, such as e.g., a fictive minimal failure at T_{ij} that extends the repair interval until T'_{ij}. The effect on the system of such actions typically says something about the importance of the component. In any case, however, unless the component is critical at some point during the interval $[T_{ij}, T'_{ij}]$, the system will not be affected by the action. This motivates the definition of the following random variable ($i = 1, \ldots, n$):

$$Z_i = \int_0^{t_N} [\sum_{j=1}^{\infty} \mathrm{I}(T_{ij} \leq t \leq T'_{ij})] \psi_i(\boldsymbol{X}(t)) dt, \quad (12)$$

where $\mathrm{I}(\cdot)$ denotes the indicator function. The expectation of this variable can then serve as a basis for an importance measure. In particular, it can be shown that the importance measures introduced in (Natvig and Gåsemyr 2008) can be obtained in this way.

Since the variable Z_i involves both real and fictive events, estimating its expectation using standard discrete event simulation can be a very complex task. While the real events represent a single possible sequence of changes in the states of the system and its components, each of the fictive events introduces an alternative sequence of state changes. Note in particular that it may happen that a fictive event, E'_{ij}, occurs after the *next* real event, E_{ij+1}, in which case the intervals $[T_{ij}, T'_{ij}]$ and $[T_{ij+1}, T'_{ij+1}]$ overlap. Hence, keeping track of all the different parallel sequences of events is indeed a challenge. Armed with the methods introduced in the present paper, however, the problem can easily be solved. In order to explain this we first note that since the component processes are assumed to be independent of each other we have:

$$E[Z_i] = \int_0^{t_N} [\sum_{j=1}^{\infty} \Pr(T_{ij} \leq t \leq T'_{ij})] I_B^{(i)}(t) dt \quad (13)$$

$$= \int_0^{t_N} \omega_i(t) I_B^{(i)}(t) dt,$$

where we have introduced the weight function:

$$\omega_i(t) = \sum_{j=1}^{\infty} \Pr(T_{ij} \leq t \leq T'_{ij}). \quad (14)$$

Now, by running a separate discrete event simulation for each of the component it turns out that the weight functions, $\omega_1, \ldots, \omega_n$, can easily be estimated using similar techniques as the ones discussed in the previous sections. In particular, one may use both pointwise curve estimates as well as interval estimates, denoted respectively by $\hat{\omega}_i$ and $\tilde{\omega}_i$. Combining these estimates with the respective estimates for $I_B^{(i)}$ we get the following estimates for $E[Z_i]$, $i = 1, \ldots, n$:

$$\hat{E[Z_i]} = \int_0^{t_N} \hat{\omega}_i(t)\hat{I}_B^{(i)}(t)dt, \tag{15}$$

$$\tilde{E[Z_i]} = \int_0^{t_N} \tilde{\omega}_i(t)\tilde{I}_B^{(i)}(t)dt, \tag{16}$$

where the integrals are easily calculated numerically. Note that one should generally not mix pointwise curve estimates and interval curve estimates. The reason for this is that the pointwise curve estimates provide unbiased estimates for the curve values at the sampling points, while interval estimates provide unbiased estimates for the average curve values over the corresponding intervals. Thus, by mixing the two, the result may not be unbiased. In the stationary phase, this issue is negligible. However, in the initial phase, the error resulting from this may be significant.

In cases where several different importance measures are used, each with its own weight function, the above technique allows us to reuse the curve estimate for $I_B^{(i)}$ when calculating each of the measures. This makes it easier and faster to compare the different measures.

For specific examples of the use of this technique see the companion paper (Natvig, Eide, Gåsemyr, Huseby, and Isaksen 2008).

6 CONCLUSIONS

In the present paper we have discussed two different approaches to curve estimation in discrete event simulations. In particular, we have indicated that using interval estimates may produce more stable curve estimates compared to pointwise estimates. The proposed methods are particularly useful in relation to importance measure estimation, especially when several different importance measures are calculated and compared.

An important parameter used in the curve estimates is the distance between the sampling points, i.e., Δ. Finding a suitable value for this parameter, may be challenging as it depends on how fast the underlying processes converge to a stationary state. Note, however, that it is not necessary to use the same distance between the sampling points throughout the sampling period. Instead it is possible to use shorter distances between the sampling points in the early stage, where the processes have not converged, and then use longer distances as soon as all the processes have entered an approximate stationary state. By studying this issue further, we think that the proposed methods can be improved considerably.

ARNE BANG HUSEBY
Department of Mathematics, University of Oslo
P.O. Box 1053 Blindern, N 0316 Oslo, Norway
Phone: +47 22 85 58 60, Fax: +47 22 85 43 49
Email: *arne@math.uio.no*

KRISTINA AALVIK EIDE
FFI, P.O. Box 25, N 2027 Kjeller, Norway
Phone: +47 63 80 75 86, Fax: +47 63 80 74 49
Email: *Kristina-Aalvik.Eide@ffi.no*

STEFAN LANDSVERK ISAKSEN
DNV Energy, Veritasveien 1, N 1322 Høvik, Norway
Phone: +47 67 57 93 71, Fax: +47 67 57 99 11
Email: *Stefan.Isaksen@dnv.com*

BENT NATVIG
Department of Mathematics, University of Oslo
P.O. Box 1053 Blindern, N 0316 Oslo, Norway
Phone: +47 22 85 58 72, Fax: +47 22 85 43 49
Email: *bent@math.uio.no*

JØRUND GÅSEMYR
Department of Mathematics, University of Oslo
P.O. Box 1053 Blindern, N 0316 Oslo, Norway
Phone: +47 22 85 59 60, Fax: +47 22 85 43 49
Email: *gaasemyr@math.uio.no*

REFERENCES

Barlow, R. E. and F. Proschan (1981). *Statistical Theory of Reliability and Life Testing*. To Begin With – Silver Spring MD.

Birnbaum, Z. W. (1969). On the importance of different components in a multicomponent system. In P. R. Krishnaia (Ed.), *Multivariate Analysis - II*, pp. 581–592. Academic Press, New York.

Glasserman, P. and D. D. Yao (1994). *Monotone Structure in Discrete-event Systems*. John Wiley and Sons, Inc.

Huseby, A. B. and M. Naustdal (2003). Improved simulation methods for system reliability evaluation. In *Mathematical and Statistical Methods in Reliability*, pp. 105–121. World Scientific Publishing Co. Pte. Ltd.

Natvig, B., K. A. Eide, J. Gåsemyr, A. B. Huseby, and S. L. Isaksen (2008). The Natvig measures of component importance in repairable systems applied to an offshore oil and gas production system. To be presented at ESREL 2008.

Natvig, B. and J. Gåsemyr (2008). New results on the Barlow-Proschan and Natvig measures of component importance in nonrepairable and repairable systems. Submitted.