Modeling Operational Risk Using Linear Algebra and Monte Carlo Simulation: Enabling Innovative Service Concepts with Simulated Throughput Computations*

Larissa Laumann[1], Daniel Jaroszewski[1], Benedikt Sturm[1], Kathrin Rose[1], Wolfgang Mergenthaler[1] and Gunnar Markert[2]

[1]*FCE Frankfurt Consulting Engineers GmbH, Bessie Coleman-Strasse 7, 60549 Frankfurt am Main, Germany*
[2]*Thyssenkrupp Industrial Solutions, Graf-Galen-Strasse 17, 59269 Beckum, Germany*
E-mail: wolfgang.mergenthaler@frankfurtconsultingengineers.de; gunnar.markert@thyssenkrupp.com

Received 04 December 2018; Accepted 05 December 2018
Publication 18 December 2018

Abstract

Industrial maintenance as a service provided by the plant manufacturer is receiving increasing attention throughout the community of plant operators, notably in the cement and mining industry. The main reason is the fact that manufacturers have the deepest knowledge of their own plants. Furthermore, plant operators do not want to worry about maintenance issues and are rather willing to outsource this task to a service provider. The question then is, at what price the operator and the manufacturer are willing to close a service contract. The service provider must make sure that the price covers his expected cost including an eventual insurance fee against extreme damage and that the risk of outliers can be managed. The operator, in turn, must make sure that the price is covered by his income leaving an appropriate profit. Furthermore, the

*FCE Frankfurt Consulting Engineers GmbH, ThyssenKrupp Industrial Solutions.

Journal of Industrial Engineering and Management Science, Vol. 1, 15–38.
doi: 10.13052/jiems2446-1822.2018.002

maintenance service provider can ask an insurance company for protection against very large damages. And finally, the investor, granting a loan to the operator, is interested in his own risk.

Keywords: Reliability Engineering, Engineering Applications of Graph Theory, Monte Carlo Simulation, Matrix Inversion.

1 The Model

In a favourable market situation, where every unit of production of the commodity under consideration can be sold, operators of large industrial plants, such as in the cement or the mining industry, crucially depend on high availability characteristics of their systems. Every unplanned system malfunction reduces revenue. The current paper describes an approach to estimate the output distribution of a multicommodity plant based on parallel, alternating renewal processes, representing component defects and subsequent repairs. Estimation makes use of the Monte-Carlo Simulation technique. For each time interval in each of a set of simulations, the plant must be modeled by an appropriate technique.

Reliability engineering, in the past, has largely concentrated on systems with stochastically independent binary components, see [4], for instance. Systems with a continuum of states call for a different approach. Kuei-Lin [1] deals with the reliability of manufacturing networks. So does this paper by modeling the manufacturing network as a graph with nodes, which linearly transform real-valued input vectors into real-valued output vectors. The transformation or yield matrix implements this transformation and is subject to failure and repair. This way, computing the output vector calls for an efficient matrix inversion algorithm, as this operation must be performed for each time interval and each simulation. Such algorithms are vital to the purpose of this paper, as also discussed in [2], for instance or [6]. Juan-Li [3] studies stochastic network problems, when malfunctions arise from physical failures and emphasizes insurance and reinsurance aspects. Stochastic networks can also be viewed from an entirely different angle. Thus, continuous and constant in-flow into the system may be replaced by discrete arrivals of customers to be routed through the individual stations in the graph and to join queues in front of the stations. This type of problem is investigated, for example by [5].

The maintenance task in plant operation is increasingly being outsourced to professional maintenance service providers, such as the plant manufacturer himself. The reason is the deep knowledge, the plant manufacturer has on his own systems. Providing maintenance, however, against a fixed fee, usually calls for protection against rare, but very large damages. The idea presented here is to find insurance coverage beyond a certain deductible. Finally, the investor who grants a loan to the operator in order to build the plant, has a vital interest in the probability that the operator will fail to pay back his loan. Computing the insurance premium and every kind of risk incurring to the operator, the manufacturer and the investor, requires knowledge of the output distribution.

This paper is organized as follows: In the current chapter we outline the model and deal with the master equations relating output to input in a network of nodes. We illustrate this with an example and describe how the stochastic processes consisting of failure and repair act on the yield matrices. Next we look at output and loss and define both in terms of the stochastic properties of the yield matrix. In Chapter 2 the necessity to invert certain matrices is the main topic. Various ways to accelerate or avoid matrix inversion are investigated. For instance, if the graph of the system under consideration contains no cycles, the precomputation of certain graph related coefficients will be possible. Also, if the yield matrix has a certain modular structure, matrix inversion collapses into a series of smaller inversions. This, together with a special disturbance model avoids repeated matrix inversion altogether by making use of the eigenvalues and the eigenvectors of the undisturbed matrix. It is shown how Monte-Carlo simulation, in connection with matrix inversion, can be used to estimate the output distribution and the damage distribution. Chapter 3 uses these distributions to calculate certain risks on the various players in the maintenance cycle. Chapter 4, finally illustrates the computational process by means of a small example.

1.1 Definitions and Assumptions

Assume the plant under consideration can be modeled as a graph with n nodes and an appropriate number of edges. Assume node $i \in \{1, \ldots, n\}$ has $M(i)$ inputs and $K(i)$ outputs. Let

$$\begin{aligned} x^i &= (x^i_1, \ldots, x^i_{M(i)})^T \\ y^i &= (y^i_1, \ldots, y^i_{K(i)})^T \end{aligned}$$

be the vector of input and output flows of node i, respectively and let A^i be a matrix mapping $\mathbb{R}^M$ onto $\mathbb{R}^K$ such that

$$y^i = A^i * x^i$$

Below we will refer to the individual inputs and outputs of a node also as the input and output ports. Concatenating vectors x^i und y^i into one single vector x and y each and defining a matrix

$$A = \Psi(A^1, \dots, A^n)$$

whereby Ψ is a matrix valued function of the matrix valued arguments $A^1, \dots, A^n$ one can write

$$y = A * x \tag{1}$$

We will call A the yield matrix in this paper. The inputs to each of the nodes are composed of certain outputs of predecessor nodes and eventual external inputs, whereby cycles may occur. x, therefore can be written as

$$x = B * y + \alpha \tag{2}$$

for some appropriate connection matrix B and a vector α, representing the net inflow to the system. Equation (2) makes sure that inputs to the source nodes can be represented by certain outputs of other nodes and components of the vector α.

It is assumed, without loss of generality, that a unit of production costs a unit price of 1\$.

1.2 The Main Equation

Plugging Equation (2) into Equation (1) yields

$$\begin{aligned} y &= AB * y + A\alpha \text{ or, equivalenly} \\ y &= (I - AB)^{-1} A\alpha \end{aligned} \tag{3}$$

provided matrix $I - AB$ is nonsingular, which will henceforth be assumed and can be tested as needed.

1.3 Example

Figure 1 shows the flow of material in a plant having all the graph related properties as described above. Defining the vectors

$$x = \begin{pmatrix} 0 \\ x_1^1 \\ x_1^2 \end{pmatrix} y = \begin{pmatrix} y_1^1 \\ y_1^2 \\ y_2^2 \end{pmatrix} \alpha = \begin{pmatrix} \alpha_1 \\ 0 \\ 0 \end{pmatrix}$$

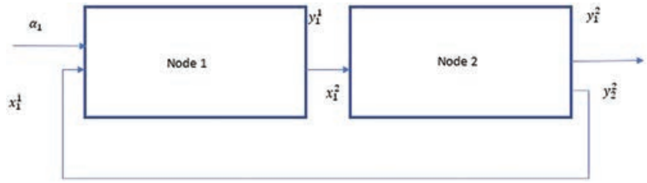

Figure 1 Example layout.

then, with the matrices

$$A^1 = \left(a^1_{1,1}, a^2_{1,2}\right), \quad A^2 = \begin{pmatrix} a^2_{1,1} \\ a^2_{2,1} \end{pmatrix}$$

$$A = \begin{pmatrix} a^1_{1,1}, a^2_{1,2}, 0 \\ 0, 0, a^2_{1,1} \\ 0, 0, a^2_{2,1} \end{pmatrix}, \quad B = \begin{pmatrix} 0, 0, 0 \\ 0, 0, 1 \\ 1, 0, 0 \end{pmatrix}$$

we see that Equation (3) is satisfied.

1.4 The Random Process

Let us now follow the system over a time period $[0, T] \subseteq \mathbb{R}$. Throughout this time period we will see a sequence of events. The k-th event $t_k \in [0, T]$ occurs in node $i \in \{1, \ldots, n\}$, say. If the event is a failure, then the affected node will experience a modified yield matrix $A^i_{t_k}$, which differs by a small matrix $\Delta^i_{t_k}$ from A^i such that $A^i_{t_k} = A^i + \Delta^i_{t_k}$. The up times of each node are supposed to be – without loss of generality – exponentially distributed with an event rate κ_i. The repair times are supposed – again without loss of generality – to be deterministic with a constant repair time τ_i. Starting with an entirely intact system, the first event must necessarily be a failure of a node. Each subsequent event produces a new matrix A_{t_k} whereby submatrix $A^i_{t_k}$ is equal to submatrix A^i, if the event at time t_k was a repair and differs from A^i, if the event was a failure. Each event is represented by

- the time
- the node affected
- and the mode (Up/Down)

and creates its own successor until time T has been exhausted.

1.5 Output and Loss Distributions

The vectorial output y in the system under consideration becomes time dependent through the randomly changing yield matrices A_t of the nodes, i.e.

$$y_t = (I - A_t B)^{-1} A_t \alpha \tag{4}$$

The entire output over the whole period can be expressed as a sum of output rates over the list of events multiplied by the time expired since the last event, i.e.

$$Y := \sum_{k \in \{1,2,3,\ldots\}, t_k \in [0,T]} y_{t_{k-1}} * (t_k - t_{k-1}) \tag{5}$$

Below we single out one particular component of y – henceforth called y^s denoting a sink – and we are interested in its distribution and its density

$$\begin{aligned} F(\eta) &= P\{y^s \leq \eta\} \\ f(\eta) &= \frac{dF(\eta)}{d\eta} \end{aligned} \tag{6}$$

Let y^s_{max} be the maximum achievable output of the plant, when the system is intact, i.e. when

$$\begin{aligned} A &= A_0 \\ y^s_{max} :&= ((I - A_0 B)^{-1} A_0 \alpha)^s \end{aligned} \tag{7}$$

with the symbol s on the right hand side again denoting the sink and define the defect as the missing yield

$$Z := Y^s_{max} - y^s$$

As a consequence the distribution function of Z is given by

$$\begin{aligned} G(\zeta) :=& \; P\{Z \leq \zeta\} = P\{Y^s \geq y^s_{max} - \zeta\} \\ =& \; 1 - P\{Y^s \leq y^s_{max} - \zeta\} = 1 - F(y^s_{max} - \zeta) \end{aligned} \tag{8}$$

assuming F is continuous. The distribution density of the defect is given by

$$g(\zeta) := \frac{dP\{Z \leq \zeta\}}{d\zeta} = f(y^s_{max} - \zeta) \tag{9}$$

The main objective in this paper is the numerical estimation of $G(z)$ and $g(z)$ via a Monte-Carlo simulation of $F(\eta)$ and $f(\eta)$.

2 The Numerical Challenge

The purpose of this paper is to use Equations (4) and (5) in a loop over many simulations and time steps within each simulation to generate possible lifetime trajectories. Below we will simulate the global output y over the time period $[0, T]$.

In the pursuit of finding the damage distribution we are seeking ways to accelerate the algorithm which consists in many consecutive matrix inversions. We process the sequence of events in the following manner: In the beginning y^s is given by Equation (7). As time proceeds y_t is given by Equation (4).

2.1 A Matrix Series Expansion

Setting

$$A_t = A_{t-1} + \Delta_t \tag{10}$$

for a certain matrix Δ_t, one obtains from (4) through a series expansion of the matrix $(I - A_{t-1}B)^{-1}\Delta_t B$ the following expression:

$$\begin{aligned} y_t &= (I - (A_{t-1} + \Delta_t)B)^{-1} A_t \alpha \\ &= (I - A_{t-1}B - \Delta_t B)^{-1} A_t \alpha \\ &= ((I - A_{t-1}B)(I - (I - A_{t-1}B)^{-1}\Delta_t B))^{-1} A_t \alpha \\ &= (I - (I - A_{t-1}B)^{-1}\Delta_t B)^{-1}(I - A_{t-1}B)^{-1} A_t \alpha \\ &= (I + (I - A_{t-1}B)^{-1}\Delta_t B)(I - A_{t-1}B)^{-1} A_t \alpha \\ &\quad + o(\|(I - A_{t-1}B)^{-1}\Delta_t B\|) \end{aligned} \tag{11}$$

assuming the norm of $(I - A_{t-1}B)^{-1}\Delta_t B$ is small. This creates hope that – by tolerating a small error of order $o(\|(I - A_{t-1}B)^{-1}\Delta_t B\|)$ – one could do the matrix inversion only in every other step and thereby save on computational work.

2.2 An Iterative Approximation

Another method to save on the effort of inversion arises by considering

$$\begin{aligned} y_{t+1} &= (I - A_{t+1}B)^{-1} A_{t+1} \alpha \\ (I - A_{t+1}B) y_{t+1} &= A_{t+1} \alpha \\ (I - (A_t + \Delta_{t+1})B) y_{t+1} &= A_{t+1} \alpha \\ (I - A_t B) y_{t+1} &= A_{t+1} \alpha + \Delta_{t+1} B y_{t+1} \\ y_{t+1} &\approx (I - A_t B)^{-1}[A_{t+1} \alpha + \Delta_{t+1} B y_t] \end{aligned} \tag{12}$$

2.3 An Iterative Solutions

Equation (12) suggests to look for an iterative solution for y_{t+1} in the following form:

$$\begin{aligned} y_{t+1}^{k+1} &= (I - A_t B)^{-1}[A_{t+1}\alpha + \Delta_{t+1} B y_{t+1}^k], k = 0, 1, 2, 3... \\ y_{t+1}^0 &= y_t \end{aligned} \tag{13}$$

The question now is, whether iteration (13) converges. Applying (13) for indices k and $k+1$ therefore yields

$$y_{t+1}^{k+1} - y_{t+1}^k = (I - A_t B)^{-1}\Delta_{t+1} B(y_{t+1}^k - y_{t+1}^{k-1}), k = 1, 2, 3 \dots \tag{14}$$

If the operator $U_t := (I - A_t B)^{-1}\Delta_{t+1} B$ represents a contracting map, then iteration (13) converges.

Lemma 1 *Let the matrix* U_t *have* n *different eigenvalues* $\chi_1, \dots, \chi_n$. *If*

$$\begin{aligned} 0 \leq \chi_i < 1, i \in 1, \dots, n, \\ R := \max_{1,\dots,n} \chi_i < 1 \end{aligned} \tag{15}$$

then (13) *is a contracting map.*

Proof. With

$$\Delta y_{t+1}^{k+1} := y_{t+1}^{k+1} - y_{t+1}^k, k = 1, 2, 3, \dots$$

Then, from (14) one obtains $\Delta y_{t+1}^{k+1} := U_t * \Delta y_t^k$. Now express Δy_t^k in terms of the eigenvectors $v_i, i \in \{1, \dots, n\}$ of U_t, i.e. $\Delta y_t^k := \sum_{i \in 1,\dots,n} w_i * v_i$ for some coefficients $w_i, i \in \{1, \dots, n\}$. Then the following holds

$$\begin{aligned} \|\Delta y_t^{k+1}\| &\leq \max_{i \in \{1,\dots,n\}} \chi_i * \| \sum_{i \in \{1,\dots,n\}} w_i * v_i \| \\ &= \max_{i \in \{1,\dots,n\}} \chi_i * \|\Delta y_t^k\| \leq R \|\Delta y_t^k\| \\ \|\Delta y_t^{k+1}\| &\leq R^k * \|\Delta y_t^1\| \end{aligned}$$

and therefore $\lim_{k->\infty} \|\Delta y_t^k\| = 0$ which proves the claim.

Equivalently, starting from Equation (3) one can set

$$\begin{aligned} y_t^{k+1} &= A_t B * y_t^k + A_t \alpha, k \in 0, 1, 2, 3... \\ y_t^0 &= A_t \alpha \end{aligned} \tag{16}$$

Lemma 2 *It is straightforward to show that*

$$y_t^k = \sum_{\nu=0}^{k} (A_t B)^\nu * A_t \alpha \tag{17}$$

Also, if

$$\|A_t B\| < 1 \tag{18}$$

then iteration (16) *converges.*

Proof. The proof of (17) follows by induction. (17) certainly holds for $k = 0$ using (16). Now, assume it holds for any $k > 0$. Then

$$\begin{aligned} y_t^{k+1} &= A_t B * \sum_{\nu=0}^{k} (A_t B)^\nu * A_t \alpha + A_t \alpha \\ &= \sum_{\nu=1}^{k+1} (A_t B)^\nu * A_t \alpha + A_t \alpha \\ &= \sum_{\nu=1}^{k+1} (A_t B)^\nu * A_t \alpha + (A_t B)^0 * A_t * \alpha \\ &= \sum_{\nu=0}^{k+1} (A_t B)^\nu * A_t \alpha \end{aligned} \tag{19}$$

and the lemma is proven.

2.4 A Special Case

Lemma 3 *If*

$$\kappa_i = 0 \vee \tau_i = 0, i \in \{1, \ldots, n\}$$

then

$$P\{Y_t = Y_{max}\} = 1, t \in \{0, 1, 2, .., T\}$$

Proof. Straightforward.

Dropping the time index in the matrix elements and the input and output vectors for now one can show, that the following holds:

Lemma 4 *Let, for each node* $i \in \{1, \ldots, n\}$ *and for each input port* $k \in \{1, \ldots, M(i)\}$ $V(i, k)$ *and* $P(i, k)$ *denote the predecessor attached to* *input port k of node i and the corresponding output port on node $V(i,k)$, respectively. Also, let $A^i_{t_k} = \xi^i_{t_k} * A^i$ for some random variable $\xi^i_{t_k}, 0 \leq \xi^i_{t_k} \leq 1$. Finally set $V(i,k_1,k_2) := V(V(i,k_1),k_2)$ and $P(i,k_1,k_2) := P(V(i,k_1),k_2)$ and so forth. If the network contains no cycles, then each output of each node can be written as*

$$y^i_l = \sum_{k_1=1}^{M(i)} \sum_{k_2=1}^{M(V(i,k_1))} \cdots \sum_{k_N=1}^{M(V(i,k_1,\ldots,k_{N-1}))} a^i_{l,k_1} \prod_{s=2}^{N} a^{V(i,k_1,\ldots,k_{s-1})}_{P(i,k_1,\ldots,k_{s-1}),k_s} x^{V(i,k_1,\ldots,k_N)}_{k_N} \quad (20)$$

Proof. By definition the following holds

$$y^i_l = \sum_{k=1}^{M(i)} a^i_{l,k} x^i_k \quad (21)$$

Observing that

$$x^i_k = \begin{cases} y^{V(i,k)}_{P(i,k)}, & \text{if } M(V(i,k)) > 0 \\ \alpha_{L(i,k)}, & \text{else - for some function } L(i,k) \end{cases}$$

and plugging this result into Equation (21), one obtains

$$y^i_l = \sum_{k=1}^{M(i)} a^i_{l,k} y^{V(i,k)}_{P(i,k)} \quad (22)$$

as long as $M(V(i,k)) > 0$. Continuing this process once shows

$$y^i_l = \sum_{k=1}^{M(i)} \sum_{r=1}^{M(V(i,k))} a^i_{l,k} a^{V(i,k)}_{P(i,k),r} y^{V(V(i,k),r)}_{P(V(i,k),r)} \quad (23)$$

The process stops, when $M(V(i,k_1,\ldots,k_N)) = 0$, in which case $x^{V(i,k_1,\ldots,k_N)}_N = \alpha(L(i,k_1,\ldots,k_N))$ for some appropriate function $\alpha(L(i,k_1,\ldots,k_N))$.

Corollary 1 *Formula* (20) *has an important consequence. By letting i represent a sink node, i.e. a node which has no successors and by continuing the* *recursion on the right hand side until* $V(i, k_1, \ldots, k_s), P(i, k_1, \ldots, k_s)$ *point to an external input* $\alpha(L(i, k1, \ldots, k_N))$*,* y_i^l *can be represented as a linear function of the inputs, whereby the coefficients are composed of a product of matrix elements.*

$$y_l^i = \sum_{k \in N^N} \psi(l, i, k)\alpha(L(l, i, k)) \prod_{\rho \in \{1,\ldots,n\}, \rho \text{ defective}} \xi_t^\rho \tag{24}$$

where the summation index satisfies

$$\begin{aligned} 1 \le k_\nu \le M(V(i, k_1, \ldots, k_{\nu-1})), 1 \le \nu \le N, \\ M(V(i, k_1, \ldots, k_\nu)) = 0 \end{aligned} \tag{25}$$

and

$$\psi(l, i, k) = a_{l,k_1}^i \prod_{s=2}^N a_{P(i,k_1,\ldots,k_{s-1}),k_s}^{V(i,k_1,\ldots,k_{s-1})}$$

Proof. Starting from Equation (20) one verifies that, upon using $k^{s-1} := (k_1, \ldots, k_{s-1})$

$$y_{l,t}^i = \sum_{k \in N^N} a_{l,k_1,t}^i \prod_{s=2}^N a_{P(i,k^{s-1}),k_s,t} \alpha(L(l, i, k))$$

with the above mentioned condition on the summation index. Now, upon observing that $a_{r,k,t}^i = \xi_t^i a_{r,k}^i, 1 \le k \le M(i), 1 \le k \le K(i)$, one proves the corollary.

Formula (24) allows for a precomputation of all the factors ψ_k^i and thereby avoids the necessity to invert the yield matrix for each epoch throughout the simulation. Instead, only the scalars ξ_t^l must be simulated.

2.5 Network Modularization

Dropping the time index for the purpose of the follwing result, one can show

Lemma 5 *If the graph representing the system can be partitioned into a set of L subgraphs such that subgraph* $i \in \{1, \ldots, L\}$

- *receives a net external inflow represented by* $\alpha^{(i)}$ *from subgraphs with lower indices*
- *has its own yield matrix* $A^{(i)}$
- *has a set of connection matrices* $B^{(i,j)}$

- *has an output vector given by* $y^{(i)}$

and there are no cycles between subgraphs, while each subgraph may possess internal cyles, then the following holds:

$$\begin{aligned} y^{(1)} &= (I - A^{(1)}B^{(1,1)})^{-1}A^{(1)}\alpha^{(i)} \\ y^{(2)} &= (I - A^{(2)}B^{(2,2)})^{-1}A^{(2)}\left(\sum_{j=1}^{1} B^{(2,j)}y^{(j)} + \alpha^{(2)}\right) \\ &\dots \\ y^{(L)} &= (I - A^{(L)}B^{(L,L)})^{-1}A^{(L)}\left(\sum_{j=1}^{L-1} B^{(L,j)}y^{(j)} + \alpha^{(L)}\right) \end{aligned} \tag{26}$$

Proof. With the above definitions the following equations hold:

$$\begin{aligned} y^{(i)} &= A^{(i)}x^{(i)} \\ x^{(i)} &= \sum_{j=1}^{i-1} B^{(i,j)}y^{(j)} + B^{(i,i)}y^{(i)} + \alpha^{(i)} \end{aligned} \tag{27}$$

Therefore, plugging the second equation into the first one yields

$$y^{(i)} = \sum_{j=1}^{i-1} A^{(i)}B^{(i,j)}y^{(j)} + A^{(i)}B^{(i,i)}y^{(i)} + A^{(i)}\alpha^{(i)} \tag{28}$$

Concatenating the output vectors $y^{(i)}, i \in \{1, \dots, L\}$ into one large output vector y one obtains

$$y = \begin{pmatrix} A^{(1)}B^{(1,1)}, \dots\dots\dots \\ A^{(2)}B^{(2,1)}, A^{(2)}B^{(2,2)}, \dots\dots \\ \ddots \\ A^{(L)}B^{(L,1)}, A^{(L)}B^{(L,2)}, \dots, A^{(L)}B^{(L,L)} \end{pmatrix} y + \begin{pmatrix} A^{(1)}\alpha^{(1)} \\ A^{(2)}\alpha^{(2)} \\ \ddots \\ A^{(L)}\alpha^{(L)} \end{pmatrix} \tag{29}$$

This can also be written as

$$y = \begin{pmatrix} A^{(1)}B^{(1,1)}, \dots\dots\dots \\ \dots\dots, A^{(2)}B^{(2,2)}, \dots\dots \\ \ddots \\ \dots\dots\dots, A^{(L)}B^{(L,L)} \end{pmatrix} y$$

$$+\begin{pmatrix} A^{(1)}\alpha^{(1)} \\ A^{(2)}\sum_{j=1}^{2-1} B^{(2,j)}y^{(j)} + A^{(2)}\alpha^{(2)} \\ \ddots \\ A^{(L)}\sum_{j=1}^{L-1} B^{(L,j)}y^{(j)} + A^{(L)}\alpha^{(L)} \end{pmatrix} \tag{30}$$

Equation (30) is equivalent to Equation (26), which proves the lemma.

This result may eventually be used to simplify matters and accelerate the simulations, as smaller matrices must be inverted in comparison to Equation (3). Also, it can be extended to more complex graphs with isolated subgraphs.

In order to save matrix inversion effort we can observe another interesting special case upon reinsertion of the time index t:

Corollary 2 *If the conditions of lemma* (5) *hold for each subgraph, and in addition matrices* $A^{(i)}$ *for all* $i \in \{1, \ldots, M(i)\}$ *have only different eigenvalues and the following definitions are used*

$$\begin{aligned} \tilde{\alpha}_t^{(i)} &:= \sum_{j=1}^{i-1} B^{(i,j)} y_t^{(i)} + \alpha^{(i)} \\ A_t^{(i)} B^{(i,i)} v_t^{(i)} &= \lambda_k^{(i)} v_k^{(i)} \\ A_t^{(i)} \tilde{\alpha}_t^{(i)} &= \sum_{k=1}^{n} r_{k,t}^{i} v_k^{(i)} \end{aligned} \tag{31}$$

where $\lambda_k^{(i)}$ *and* $v_k^{(i)}, k \in \{1, \ldots, M(i)\}$ *are the eigenvalues and the eigenvectors of* $A^{(i)}B^{(i,i)}$ *and* $r_{k,t}^{(i)},\ v_k^{(i)}$ *is the representation of* $A^{(i)}\tilde{\alpha}_t^{(i)}$ *in the basis spanned by the eigenvectors, then* $y_t^{(i)}$ *can be expressed as*

$$y_t^{(i)} = \xi_t^{(i)} \sum_{k=1}^{n} r_{k,t}^{(i)} \frac{v_k^{(i)}}{1 - \xi_t^{(i)} \lambda_k^{(i)}} \tag{32}$$

Proof. The following can be concluded from (26)

$$\begin{aligned} y_t^{(i)} &= (I - A_t^{(i)} B^{(i,i)})^{-1} A_t^{(i)} \tilde{\alpha}_t^{(i)} \\ &= \xi_t^{(i)} (I - \xi_t^{(i)} A^{(i)} B^{(i,i)})^{-1} A^{(i)} \tilde{\alpha}_t^{(i)} \\ &= \sum_{k=1}^{n} r_{k,t}^{(i)} \xi_t^{(i)} (I - \xi_t^{(i)} A^{(i)} B^{(i,i)})^{-1} v_k^{(i)} \end{aligned} \tag{33}$$

using $A_t^{(i)} = \xi_t^{(i)} A^{(i)}$. With the definitions of $\lambda_k^{(i)}, v_k^{(i)}, k \in \{1, \ldots, n\}$ one obtains

$$\begin{aligned} (I - \xi_t^{(i)} A^{(i)} B^{(i,i)}) v_k^{(i)} &= (1 - \xi_t^{(i)} \lambda_k^{(i)}) v_k^{(i)}, k \in \{1, \ldots, n\} \\ (I - \xi_t^{(i)} A^{(i)} B^{(i,i)})^{-1} v_k^{(i)} &= \frac{1}{1 - \xi_t^{(i)} \lambda_k^{(i)}} v_k^{(i)}, k \in \{1, \ldots, n\} \end{aligned} \tag{34}$$

Inserting (34) into (33) one obtains

$$y_t^{(i)} = \xi_t^{(i)} \sum_{k=1}^{n} \frac{r_{k,t}^{(i)} v_k^{(i)}}{1 - \xi_t^{(i)} \lambda_k^{(i)}} \tag{35}$$

which proves the corollary.

2.6 Monte Carlo Simulation

In the sequel we will discuss several details of simulating the output y^s in order to generate the distribution $P\{y^s \leq \eta\}$. Assume we are performing a number $NSIM$ of simulations. According to (6) the probability $P\{y^s \leq \eta\}$ can be approximated by

$$\hat{F}(\eta) = \hat{P}\{y^s \leq \eta\} = \frac{\text{Number of simulations with } y_t^s \leq \eta}{NSIM} \tag{36}$$

From this distribution we derive an approximate distribution density

$$\hat{f}(\eta) = \frac{d\hat{F}(\eta)}{d\eta} \tag{37}$$

such that

$$P\{\eta \leq y^s \leq \eta + d\eta\} \approx \hat{f}(\eta) d\eta + o(d\eta) \tag{38}$$

Also, with the definitions given in (8) and (9) the following relationships can be used.

$$\begin{aligned} \hat{G}(\zeta) &= \hat{F}(y_{max}^s - \zeta) \\ \hat{g}(\zeta) &= \hat{f}(y_{max}^s - \zeta) \end{aligned} \tag{39}$$

3 The Players and Their Risk Levels

Four players are involved in the current concept, i.e. the plant operator, the plant manufacturer – here identical to the maintenance provider – eventually an insurance company and an investor. It is assumed that all of the following agreements between the players prevail:

- The plant operator and the investor negotiate a loan on the investment to build the plant. Repayment plus interest amounts to a sum RI per year.
- The plant operator and the maintenance contractor agree that the contractor will take care of the entire maintenance at a yearly fee *FM*.
- The maintenance contractor and the insurance company close an insurance contract such that, whenever the defect per year exceeds a deductible D, the limit D is covered by the maintenance contractor and the exceeding amount is carried by the insurance. Let the insurance fee be *FI*.
- Let LC be the labor cost
- Let MC be the pure mechanical maintenance cost
- Let E denote other expenditures and, finally
- Let P be the profit before tax

All cost numbers referenced above and below will be yearly cost.

3.1 The Operator's Risk

If the operator and the contractor close a service contract, the operator bears no risk, as he receives a fixed income generated either by his sales or by a reimbursement from the contractor, should the output remain below y^s_{max}. All he has to do is to pay the fee *FM*, which includes the price for having no risk.

3.2 The Contractor's Risk

The contractor's risk – defined as the probability that the defect is greater than the fee paid to him by the operator – is given by

$$\rho_C = \int_{FM}^{y^s_{max}} g(\zeta) d\zeta \tag{40}$$

The risk is mitigated by the insurance contract. The average defect and the variance the contractor sees can be written as

$$\begin{aligned} \mu_C &:= \int_0^D g(\zeta)\zeta d\zeta + D \int_D^{y^s_{max}} g(\zeta) d\zeta \\ \sigma_C^2 &:= \int_0^D g(\zeta)(\zeta - \mu_C)^2 d\zeta + (D - \mu_C)^2 \int_D^{y^s_{max}} g(\zeta) d\zeta \end{aligned} \tag{41}$$

3.3 The Insurer's Risk

Usually the insurance fee the contractor must pay to the insurance company is equal to the expected damage to the insurer plus a certain safety factor SF times the standard deviation. SF typically takes values between 1.0 and 3.0. The expected cost and the variance the insurer sees is therefore

$$\begin{aligned} \mu_I &= \int_D^{y^s_{max}} g(\zeta)(\zeta - D)d\zeta \\ \sigma_I^2 &= \int_D^{y^s_{max}} g(\zeta)(\zeta - \mu_I)^2 d\zeta \end{aligned} \tag{42}$$

The insurance fee is normally set to

$$FI = \mu_I + SF * \sigma_I \tag{43}$$

The insurer's risk is the probability that the damage exceeds the insurance fee paid to him by the maintenance contractor, i.e.

$$\rho_I = \int_{FI}^{y^s_{max}} g(\zeta)d\zeta \tag{44}$$

3.4 The Investor's Risk

3.4.1 No service contract

Figure 2 below shows, how the yearly revenue to the operator is split between the various types of expenditure, if the operator does not close a maintenance contract with the contractor.

No Market Risk: Let P^1_{CS} be the probability that the yearly capital service amounting to RI cannot be fully served, under the assumptions that

- The yearly revenue Y is fixed and no market risk exists, i.e. every unit of production can be sold
- Labor cost is fixed
- Payments must be made in the order $LC \rightarrow MC \rightarrow RI \rightarrow E$

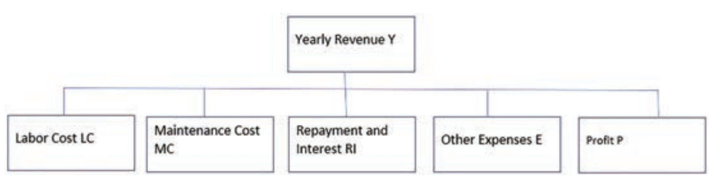

Figure 2 No service contract.

In that case we have

$$\begin{aligned} P^1_{CS} &= \int_{y^s_{max}-(LC+MC+RI)}^{y^s_{max}} g(\zeta)d\zeta \\ &= G(y^s_{max}) - G(y^s_{max} - (LC+MC+RI)) \\ &= 1 - G(y^s_{max} - (LC+MC+RI)) \\ &= 1 - F(LC+MC+RI) \end{aligned} \quad (45)$$

Verbally this means that labor and maintenance cost do not leave any resources to pay the loan back to the investor. Figure 3 illustrates this situation. The integral is the shaded area in the limits between 0 und $y^s_{max} - (LC + MC + RI)$. If this integral grows due to the fact that the density function assumes larger values even for smaller maintenance cost (red area), i.e. if the maintenance cost become “stochastically smaller”, then the probability to serve the capital cost is larger than for smaller values of the integral (blue area).

Hence

- Capital service probability grows with falling maintenance cost and
- Investor’s risk falls with falling maintenance cost as well

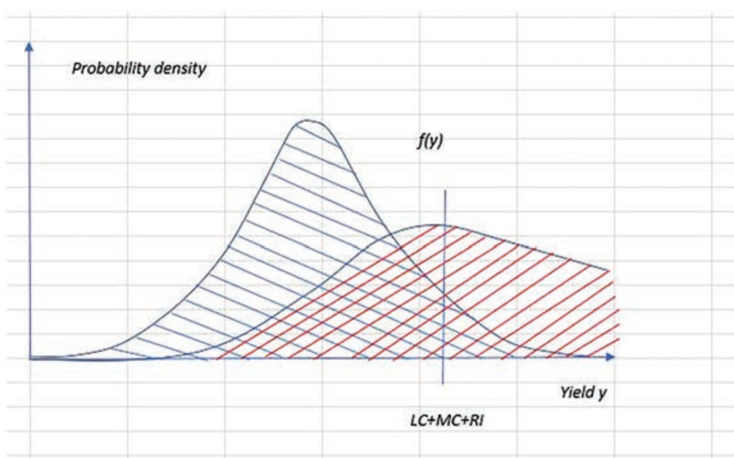

Figure 3 Probability for capital service with service contract.

With Market Risk: If, however, $h(x)$ represents the probability density that x units of product can be sold only, then we have

$$\begin{aligned} P_{CS}^2 &= \int_{LC+MC+RI}^{y_{max}^s} h(x)dx \int_{x-(LC+MC+RI)}^{x} g(\zeta)d\zeta \\ &= \int_{LC+MC+RI}^{y_{max}^s} h(x)(G(x) - G(x-(LC+MC+RI)))dx \\ &= \int_{LC+MC+RI}^{y_{max}^s} h(x)(F(y_{max}^s - (x-(LC+MC+RI))) \\ &- F(y_{max}^s - x))dx \end{aligned} \tag{46}$$

3.4.2 With service contract

In contrast, Figure 4 below shows the distribution of the yearly revenue between the various types of expenditure, if the operator does close a maintenance contract with the contractor.

Assume, that in this case payments must be made in the order $LC \to FM \to FI \to RI \to E$

No Market Risk: In this case we have

$$P_{CS}^3 = \begin{cases} 1 & \text{if } Z \geq y_{max}^s - (LC+FM+FI+RI) \\ & \iff Y^s \leq (LC+FM+FI+RI) \\ 0 & \text{else} \end{cases}$$

With Market Risk:

$$P_{CS}^4 = \int_0^{LC+FM+FI+RI} h(x)dx \tag{47}$$

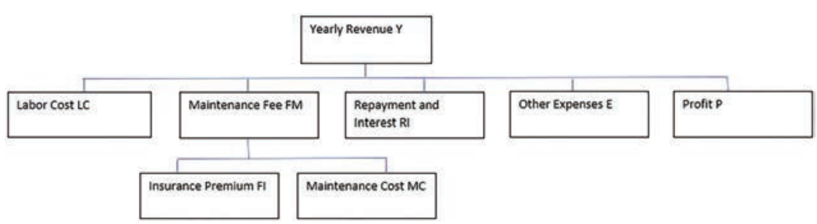

Figure 4 With service contract.

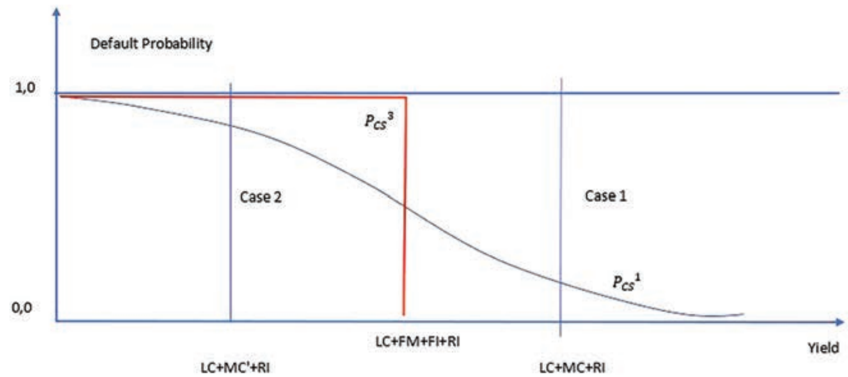

Figure 5 Default probability without market risk.

3.4.3 Conclusion

Corollary 3 *If there is no market risk, then the following holds:*

$$
\begin{aligned}
MC \geq FM + FI &\Rightarrow P^1_{CS} \geq P^3_{CS} \\
MC \leq FM + FI &\Rightarrow P^1_{CS} \leq P^3_{CS}
\end{aligned}
\tag{48}
$$

Therefore, if there is no market risk, and if $MC \geq FM + FI$ (Case 1 in figure 5 below, then a service contract positively influences the investor's risk, otherweise not.

On the other hand, if there is a market risk, and if

$$
\begin{aligned}
&\int_{LC+MC+RI}^{y^s_{max}} h(x)(F(y^s_{max} - (x - (LC + MC + RI))) \\
&-F(y^s_{max} - x))dx \\
&\quad \geq \int_0^{LC+FM+FI+RI} h(x)dx
\end{aligned}
\tag{49}
$$

then $P^4_{CS} \leq P^2_{CS}$ and – again – a service contract has a positive impact.

Proof. Straightforward.

4 Numerical Example

Figure 6 represents the simplified layout of an industrial cement plant. Each of the nodes in this layout is given by the following four items.

- An error rate κ_i
- A downtime τ_i
- A yield matrix $A^{(i)}$ and a
- A reduction factor ξ_i

Table 1 below summarizes the reliability related data on κ_i, τ_i and ξ_i as averages over the different failures. There were two failures that lead to a complete breakdown of the cement plant.

Figure 7 shows the defect distribution as estimated by a Monte-Carlo simulation.

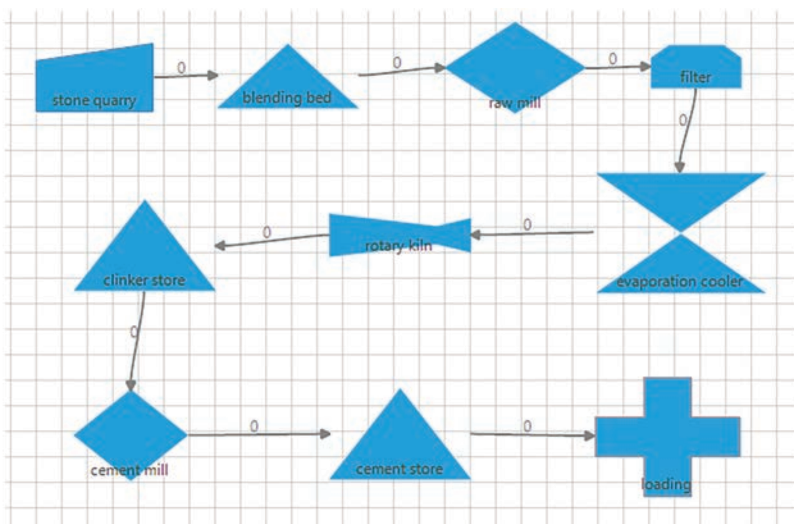

Figure 6 Layout of an industrial cement plant.

Table 1 Average reliability data

average error rate κ	1/5551,7h
average downtime τ	66,85h
average reduction factor ξ	0,29

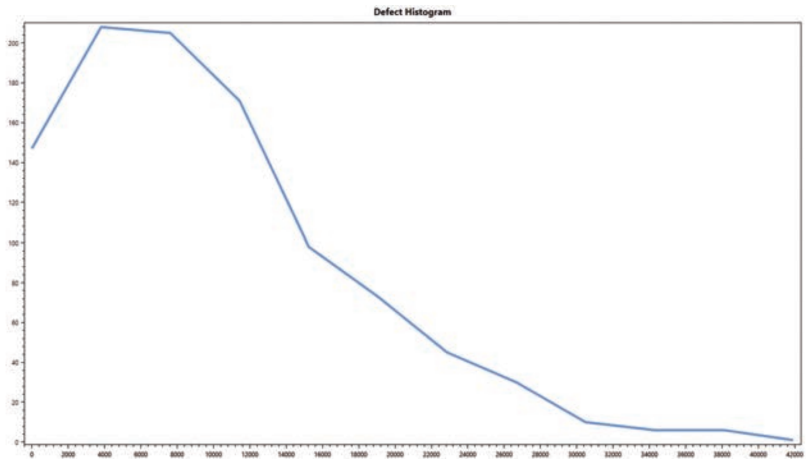

Figure 7 Defect histogram for 1000 simulation steps.

References

[1] Kuei-Lin, Y. and Chang, P. (2012). Evaluate the system reliability for a manufacturing network with reworking actions. *Reliability Engineering and System Safety*, 106, 127–137.

[2] Rashmi, P. and Dinesh Kumar, V. (2014). Design of newton iterative method for matrix inversion. *International Journal of Research in Advent Technology*, 2, 124–126.

[3] Juan, L. (2015). *Stochastic Networks: Modeling, Simulation Design and Risk Control, Dissertation*, Columbia University, New York.

[4] Meyna, A. and Pauli, B. (2010). Zuverlaessigkeitstechnik, Hanser, Muenchen.

[5] Dai, J. and Vande Vate, J. (1996). Global stability of two station queueing networks, in: Glasserman, P., Sigman, K. and Yao, D. (Eds.) *Stochastic Networks*, Lecture Notes in Statistics 117, Springer, New York.

[6] van der Veen, A. and Dewilde, P. (1993). Large Matrix Inversion using State Space Techniques, IEEE Workshop on VLSI Signal Processing, Veldhoven, The Netherlands.

Biographies

Larissa Laumann: M.Sc. (Mathematics) graduated with a Master's degree from Johannes Gutenberg-Universität Mainz. She holds a position as a software developer and a data analyst for FCE Frankfurt Consulting Engineers GmbH. Her focus is on reliability engineering, predictive maintenance, sequencing and scheduling as well as the simulation of plant performance. Another current topic of work is the development, implementation and automation within the .NET environment.

Daniel Jaroszewski (male): Dipl.-Math. (Mathematics) Johann Wolfgang Goethe University Frankfurt. He works as a consultant and software developer for Frankfurt Consulting Engineers GmbH. The consultant expertise is focused on machine learning, probability theory, combinatorial optimization and its applicability in the industry. Moreover he is Software Project Manager for a tool in the area of Predictive Maintenance. The main tasks are the development, implementation and automation of machine learning/prognosis algorithms, as well as the deployment of a monitoring platform in the .NET environment.

Benedikt Sturm received his B.Sc. and M.Sc. degrees in Mathematics from Goethe-University in Frankfurt, Germany. He works for Frankfurt Consulting Engineers GmbH since 2015. His main topics are predictive maintenance and combinatorial optimization such as routing problems. For this, he develops mathematical algorithms within a front-end interface for different costumers.

Kathrin Rose is a student at Johann Wolfgang Goethe University in Frankfurt on her way to her master's degree in mathematics. Kathrin started to work as an intern and a research assistant at FCE Frankfurt Consulting Engineers GmbH. Her specialties are graph theory, numerical and computational mathematics.

Wolfgang Mergenthaler is managing director and owner of FCE Frankfurt Consulting Engineers GmbH in Frankfurt, Germany. He received his diploma in physics from Technische Universität München (TUM) in 1973 and his doctorate in applied mathematics in 1978, also from TUM. In his current position Dr. Mergenthaler leads a team of mathematicians and computer scientists working in the fields of Pattern Recognition, Mathematical Statistics, Probability Calculus and Combinatorial Optimization with Industrial Applications. Plant simulation is one of his strong points of interest, both with regard to operational as well as planning issues. Other important working areas are the domains of sequencing and scheduling in production planning and the generation of response surfaces from process data.

Gunnar Markert graduated from the Technical University of Kaiserslautern with a diploma in industrial engineering (Diplom-Wirtschaftsingenieur). He continued his academic career at the University of Basel in the economics department with a doctoral dissertation in an industrial marketing subject. After assuming functions in business consulting and corporate strategy, since 2016 holds a position as a senior manager for business development and sales with thyssenkrupp Industrial Solutions in the cement and mining sector. Gunnar specializes on the estimation of industrial risks due to plant malfunctions and on innovative ways to mitigate financial damage to the plant owner, in order to improve the overall risk profile and facilitate project financing. He is also interested in the computation of defect distributions and analyzes their impact on the break even point of the plant operator and – thereby – on an investor’s risk to finance a project.