The Effect of Rare Events on the Evaluation and Decoding of Hidden non-Markovian Models

Claudia Krull and Graham Horton

Department of Simulation and Graphics,
Otto-von-Guericke-University of Magdeburg, 39106 Magdeburg, Germany
(claudia,graham)@sim-md.de

Abstract. Hidden non-Markovian Models (HnMM) are a modeling paradigm based on Hidden Markov Models. They extend the existing Hidden Markov Models by changing the hidden model from a memoryless discrete-time Markov chain to a more flexible discrete stochastic model involving time dependent transition rates. HnMM enable the analysis of not completely observable real systems only based on their interaction with the environment. Possible application areas of HnMM are the analysis of machine behavior based on protocol data, or diagnosis support of a disease based on symptoms of a patient; both of which are hard or impossible to answer using existing modeling paradigms. Some of the questions that can be answered using HnMM touch failure or unusual system behavior, which might both involve rare events. This paper is aimed at investigating the effect of the existence of rare events on HnMM and on the performance of the analysis methods. By doing this, we want to test the applicability of HnMM to the analysis of problems involving rare events.

1 Introduction

Hidden non-Markovian Models (HnMM) are a recent extension of the well known Hidden Markov Models (HMM). HMM are widely used in speech and pattern recognition [Jel76,JBM75,Koser]. They enable the modeling and analysis of processes, which are not directly observable, but only through their interaction with the environment. HMM contain discrete-time Markov chains as hidden models. Therefore, they are mathematically tractable and verified algorithms exist for the solution of the three HMM tasks (see Section 2.1). However, the restriction to DTMCs as hidden models imposes that they can only directly represent Markovian processes, which involve constant transition rates. Therefore, the modeling power of HMM is severely limited. Attempts to overcome this limitation are aimed at more flexible state durations and have been developed with the focus on better speech recognition capabilities [RC87,RM85]. To date we have not found an application of HMM or their extensions to the analysis of dynamic processes such as they exist in the field of engineering and logistics.

The extension of HMM to HnMM enables the analysis of more realistic models of dynamic systems [Sim05,WIS+06,IWH06]. By allowing time dependent transition rates, the modeling power is greatly increased. At the same time, the analysis of the models becomes more complicated and expensive regarding computational resources.

Some possible application areas of HnMM are the analysis of machine failure protocols or symptoms of a patient. One could use the failure protocols produced by a machine to predict the possible source of the problem and estimate the necessary repair effort and cost. A record of symptoms and physical observations of a patient could be used for diagnosis support or prediction of the reactions to a treatment. This is not easily possible using existing modeling and analysis methods, when continuous models with time dependent transition rates are involved.

In previous research we have focused on the so-called Proxel-based simulation. It uses supplementary variables to build a DTMC from a discrete stochastic model involving time dependent transition rates [LM05]. The method is particularly well suitable for the analysis of problems involving rare events, since it deterministically discovers all possible system developments within a certain accuracy [WH06]. Thereby rare events and system states are discovered easily and at no extra expense. Using this knowledge we have developed a solution algorithm based on Proxels that can solve two of the three tasks associated with HMM. We assume that the existence of rare events in the general discrete stochastic models has an influences on some of the relevant parameters of HnMM and on the developed analysis algorithms. In this paper we want to investigate these effects.

2 State of the Art

2.1 Hidden Markov Models

An HMM is a doubly stochastic process that is sometimes also called a signal model [Rab89]. The internal/hidden process is a DTMC that emits signals in every step according to given probabilities that characterize the second stochastic process. An HMM can be formally described by a 5-tuple (S, V, A, B, Π):

$S = \{s_1 \dots s_N\}$ is the set of DTMC states.
$V = \{v_1 \dots v_M\}$ is the set of output symbols.
$A = \{a_{ij}\}_{N \times N}$ is the transition probability matrix of the hidden DTMC.
$\Pi = \{\pi_i\}_N$ is the initial probability vector of the hidden DTMC.
$B = \{b_i(k)\}_N$ contains the symbol output probabilities for each. DTMC state

Two other HMM constructs are a series of DTMC states $Q = \{q_1 \dots q_T\}$ (also called a state sequence) and a corresponding sequence of output symbols $O = \{o_1 \dots o_T\}$ of the same length (also called a trace).

The 3-tuple $\lambda = (A, B, \Pi)$ is also called the model, since the choice of these HMM components determines the behavior of the system, regardless of the naming of the states and output symbols.

There are three main tasks that can be performed using HMM:

- *Evaluation*: Determine the probability of a certain symbol output sequence O for a given model $\lambda = \{A, B, \Pi\}$.
- *Decoding*: Determine the most probable system behavior (in terms of the state sequence Q) of a given model producing a certain sequence of output symbols O.

– *Training*: Given a certain symbol output sequence O, train a model $\lambda = \{A, B, \Pi\}$ such that it produces the output sequence most likely.

Efficient algorithms exist to solve each of these tasks for the original HMM paradigm containing DTMCs as hidden models.

The so-called Forward algorithm is used to solve the Evaluation task. It uses a greedy approach to compute the overall probability that the given output sequence O has been produced by the model λ. This is useful when for a given output (e.g. recorded speech / audio signal) one is trying to find the best matching model of a given set (e.g. possible word meanings of the audio signal).[Rab89]

The Viterbi Algorithm is the solution for the second task. It also uses a greedy approach to find the one most likely state sequence Q of a given model λ that has produced a given symbol sequence O. This information is often used in understanding and tuning models for speech and pattern recognition. [For73]

The last task is by far the most complicated and expensive to solve. The Forward-Backward Algorithm (or Baum-Welch Algorithm) is a local optimization method. It improves an initial guess of the model parameters using a series of successive refinement steps. [BPSW70]

HMM are limited in that they have DTMCs as a discrete hidden model. Most real systems however are continuous and not easily converted into a DTMC. The continuous equivalent of a discrete-time Markov chain is a continuous-time Markov chain (CTMC). These are however restricted to memoryless state changes, which limits their modeling power severely. We want to relieve this shortcoming by using more general hidden models. We call this extension Hidden non-Markovian Models (see Section 3).

2.2 Discrete Stochastic Models

Discrete stochastic models (DSM) are a very powerful modeling paradigm. DSMs are discrete in the models state space and continuous in time. By allowing time dependent transition rates they can exactly represent a number of real life processes. In order to build a DSM, one has to be able to observe the complete system to be modeled. To find the exact parameters of the model, one has to be able to observe and measure the appropriate processes of the real system. Examples for DSMs are stochastic Petri nets (SPN) and their extensions or queuing models. Queuing models are widely used in the analysis of data traffic in networks. The performance measures of a queuing system are usually determined using analytical methods. This limits the application to known classes of queuing models and where the solution formulas are mathematically tractable. SPNs can be found in models for the reliability of components of more complex mechanisms, like a car. Relevant information on the performance measures of an SPN are usually obtained using Monte Carlo simulation methods such as discrete event simulation. DES performs independent replications to determine statistical estimates for the relevant result measures. Using DES, it is not easily possible to determine how exactly an observed output (e.g. error protocol) could have been produced. The variety of application areas and the modeling power make DSM a good candidate for more general hidden models in HnMM.

2.3 Proxel-Based Simulation

The Proxel-based simulation method [Hor02,LM05] is a state space-based simulation method for discrete stochastic models. It deterministically follows all possible system development paths in discrete time steps and calculates their probability. Proxels use supplementary variables [Ger95] to encode the age of an activated transition in the expanded system state and thereby implicitly build a DTMC representing the extended system state space. A Proxel (probability element) P is one such object in the expanded system state space. It contains information about the discrete model state dS, the age of the relevant transitions $\boldsymbol{\tau}$ and the probability of that combination p. In addition one could also store the point in simulation time t and the route R of system states by which this Proxel was generated. However, route R and simulation time t are rarely explicitly included in practical implementations, t is usually global and by omitting R, the reachable model state space is reduced considerably.

$$P = (dS, \boldsymbol{\tau}, t, R, p) \tag{1}$$

The Proxel simulation algorithm works by iteratively computing the next possible system states and generating the appropriate Proxel elements. The result of one simulation run is the probability of each system state at each discrete point in time. This is an advantage over common discrete event simulation, since there one needs to perform replication to provide statistical estimates for relevant system performance measures.

One problem of Proxel-based simulation is the so called state space explosion. The extension of the system state by supplementary variables for the activated transitions increases the size of the state space dramatically. This limits the applicability of Proxels to models with few discrete system states. Proxels can also be viewed as a method to turn a continuous model with arbitrary transition rates into a DTMC. Our knowledge in this area will help in devising solution algorithms for HnMM.

3 Hidden non-Markovian Models

We propose the paradigm of Hidden non-Markovian Models (HnMM) as an extension of HMM to more general discrete stochastic models as hidden component. This extension needs to be done in two steps. In HMM the symbol emissions are associated with the system states. In DSM however, the state changes are of interest, which makes it necessary to associate the symbol emissions with the state transitions. This first step results in expanded HMM, described in [KH07]. The second step, the change from a DTMC as hidden model to a DSM has not yet been formalized. However, solution algorithms for the three HMM tasks based on Proxels and discrete phase-type distributions have been described in [WIS+06,IWH06].

The new modeling paradigm HnMM has two major advantages over existing approaches. Compared to a non-Markovian discrete stochastic model, the real system does not have to be completely observable. The systems interactions with the environment can be used to determine unknown system behavior. Furthermore, observations such as protocol data can be directly linked to internal system behavior. Compared to HMM, a non-Markovian hidden model can represent many real systems more realistically, since it is continuous in time and may contain generally distributed processes. HnMM therefore combine advantages of both HMM and DSM.

The symbol sequences of HMM and HnMM are different. An HMM output sequence contains a list of output symbols, one for each discrete step of the DTMC (e.g $ABDDB$). The decoding task is to match this output sequence to the sequence of model states that produced it most likely. An HnMM output sequence contains symbols which are associated with the time stamp at which they were emitted due to a state change (e.g. $(A,2)(B,5)(D,11)(D,15)(B,21)$). This sequence can now be associated with a series of events (state changes) of the hidden DSM which produced it most likely.

Using HnMM, new questions can be answered, which were not solvable using existing paradigms. Some examples of these questions are the following, each representing a possible application area:

- Service industry: Determine the most probable system behavior that produced a certain failure protocol sequence, in order to estimate cost and duration for the repair of a machine. Of interest is here the unobserved machine behavior producing a certain output. (Task2: Decoding)
- Industrial production: Build a model of not observable machine behavior using protocol data, in order to check manufacturer specifications. This requires to train an HnMM using existing data records (Task3: Training).
- Medical care: Determine a probable course of a disease based in given symptoms of a patient, in order to support the diagnosis of a physician. This might involve matching one of several possible diseases (in the form of HnMMs) to the recorded symptoms of a patient. (Task1: Evaluation)

Especially the analysis of unusual system behavior or failures often involves models containing rare events. The existence of rare events in the hidden model will affect the probabilities of the output sequences and generating paths. We assume that depending on the position of the rare event within the model, this effect can be either positive or negative.

3.1 Evaluation and Decoding Using Proxels

In this paper we are using Proxel-based simulation for the evaluation and decoding of Hidden non-Markovian Models. [WIS$^+$06] describes a very similar method, there are however some differences in the implementation.

The Proxel HnMM algorithm takes as an input a fully specified HnMM and an output symbol sequence with time stamps. In the current implementation the HnMM is described as an SPN with symbol emissions associated to all or some of the transitions.

Given an initial system state, the Proxel algorithm starts to determine the next possible states and their probabilities. At the same time the validity concerning the given output sequence is checked, and only valid successor Proxels are stored. System development paths, which could not have produced the given output sequence are discarded to save memory. This reduces the number of child Proxels produced in each step.

In order to perform the decoding task, it is necessary to store the route information, which contains all state changes that led to a certain Proxel. This increases the storage space needed per Proxel and brings with it another problem; in the original algorithm, Proxels with the same discrete state and age are combined by adding their probability, reducing the number of Proxels per time step considerably without loosing information. When adding the route information, this is no longer possible. Therefore we developed a method of merging routes for two Proxels with the same discrete system state and age vector. This keeps the number of Proxels per time step low, but the storage space needed per Proxel is still larger than in the original Proxel algorithm.

The result of the Proxel HnMM algorithm is a list of possible end states of system development paths that could have produced the given the output sequence. For Evaluation the probability of these paths needs be added up, resulting in the overall probability to produce the given sequence with the given model. For Decoding, the most likely development path and its probability need to be found. A backtracking step unwinds the merged paths and results in a list of all possible generating paths discovered and their respective probability. This enables the investigation of more than just the one most likely development path. This is an advantage, since the difference in path probability is often only marginal and the next probable paths could also be of interest.

All of these steps in the algorithm could be influenced by the existence of rare events in the model. This will be investigated in the next section.

4 Experiments

The experiments are aimed at investigating the influence of rare events on the HnMM Proxel algorithm. The HnMM result parameters of interest are the *probability of the output sequence*, *the number of possible generating paths* and the *maximum path probability*. The *relative path probability* is the maximum path probability divided by the total trace probability. It shows how probable the most likely generating path is in relation to the other possible generating paths. The relevant algorithm performance measures are the *number of Proxels* generated per time step and the *computation time* and *memory requirement* of the calculation.

4.1 The Example Model

The experiments were performed using the simple machine model shown in Figure 1. The model is a stochastic Petri net of a machine that can be either *OK*, *Failed* or under Maintenance (*Maint*). The processes in the model are distributed according to Normal distributions, except for the time to failure, which is exponentially distributed. The costs of the maintenance and the repair represent the observable output of the model and are generated when finishing either process, meaning the firing of the appropriate transitions.

In the experiments we varied the rate of the failure rate from being comparable to the racing maintenance cycle to being several orders of magnitude slower than the other processes in the model. We also tested the symbol output as shown in Figure 1 and the output being directly associated to the rare event. In the second set of experiments we investigated symbols that are produced with decreasing probabilities, and which are associated to events that are not necessarily rare. The output of such a rare symbol can also be viewed as a rare event in the scope of HnMM.

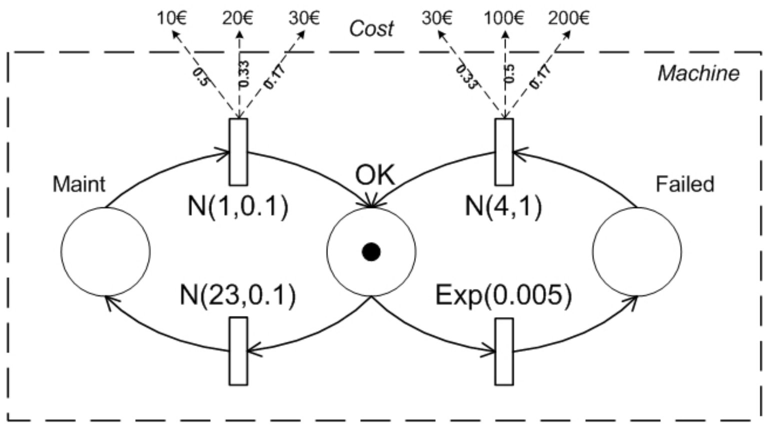

Fig. 1. Example HnMM of a Machine Maintenance Model

4.2 Example Probability Development Path

Figure 2 shows the development of the number of generating paths and the number of concurrent Proxels during the simulation. The output sequence that produced the development shown is: $(10\,€, 24)(30\,€, 48)(100\,€, 72)$. The first symbol is produced by the maintenance cycle, the third by a failure and the second one can be produced by either. The figure shows that the number of possible generating paths and Proxels rises steadily and reaches a maximum, as long as no symbol is observed. At the points in time, when an output is observed, these measures drop sharply. This is due to the fact, that the symbol output renders many of the discovered generating paths invalid, thereby also reducing the number of Proxels to only one. This result of the pruning of no longer possible development paths speeds up the methods performance and reduces memory requirement. In the long run however, the number of possible development paths increases as the simulation time advances.

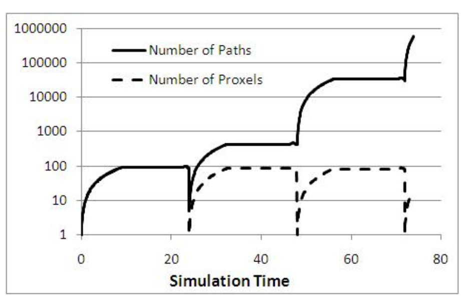

Fig. 2. Number of Possible Development Paths and Proxels During the Simulation

4.3 Experiment Using Original Model

In the first set of experiments using the model from Figure 1, the rate of the failure event is varied from 0.05 to 0.000001. Several different output symbol sequences (traces) were tested.

The trace contains symbols that can only be generated by a path containing the rare event. As the failure event becomes rarer, the total trace probability and maximum path probability first increase and then decreases again (see Figure 3 (left)). The relative path probability decreases and levels out at 0.0116 (see Figure 3 (right)). The computation time is not affected by the rareness of the event but the memory needed for the computation decreases steadily (see Figure 4 (left)).

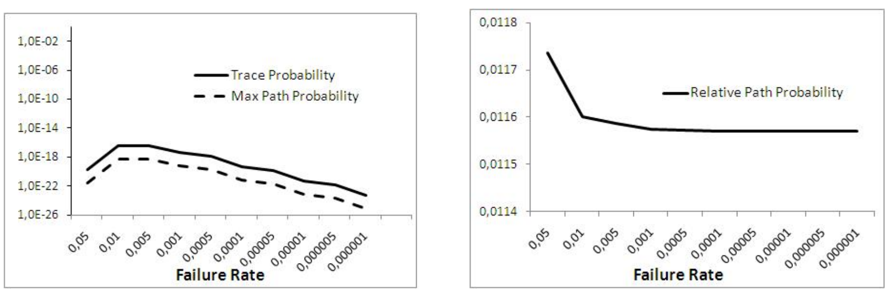

Fig. 3. Development of Trace Probability, Maximum Path Probability (left) and Relative Path Probability (right) for Decreasing Failure Rate

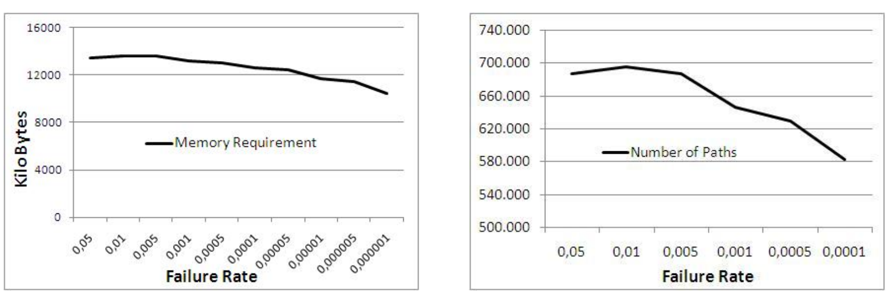

Fig. 4. Development of Algorithm Memory Requirement (left) and Number of Generating Paths (right) for Decreasing Failure Rate

The trace contains symbols that can be produced by paths containing rare or no rare events. As the failure event becomes rarer, the total trace probability and maximum path probability first increase and then decreases again (compare Figure 3). The number of possible generating paths steadily decreases, probably due to the pruning of paths below a given probability level (see Figure 4 (right)). The relative path probability decreases and levels out around 0.01 (compare Figure 3 (right)). The computation time and memory needed for the computation decrease steadily.

The trace contains no symbols generated by a path containing the rare event. As the failure event becomes rarer, the trace and maximum path probabilities rise and seem to reach a maximum (see Figure 5). But the probability range is not nearly as large as when the generating path contains rare events. The relative path probability on the other hand decreases and levels out at about 0.15. The memory needed for the computation decreases steadily, but the computation time is not affected.

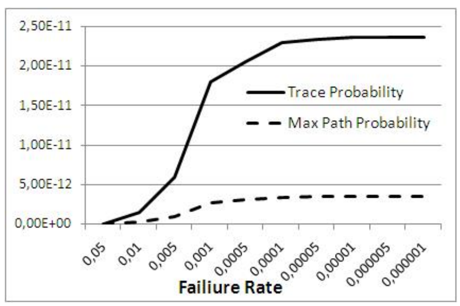

Fig. 5. Development of Trace Probability and Maximum Path Probability for Decreasing Failure Rate

4.4 Symbol Output Associated to Rare Event

In this experiment set, the rate of the failure event is varied from 0.05 to 0.000001. In contrast to the model in Figure 1, the symbol outputs are now associated to the failure and begin maintenance events. The same cases were tested as in the previous experiment set, with the same general behavior. The difference between the two experiment sets is that the relative path probability increases and is in general quite high with about 0.4 percent, as compared to about 0.01 before. The number of possible generating paths is much smaller when the rare event, which has a larger variance, is directly associated with a symbol output and not only part of the path.

4.5 Experiments on Rare Symbols

This set of experiments tests the effect of symbols with very small emission probabilities, which we will call rare symbols. Neither the existence or position of such rare symbols, nor their degree of rareness have an effect on the computation time and memory requirement. Only the relevant HnMM measures are affected.

The trace contains rare symbols. As the symbol gets rarer, the trace and maximum path probability decrease linearly (compare Figure 6). When the rare symbol is attached to a path through the failure state, the relative path probability drops, as the symbol gets rarer. If the rare symbol is generated by a path through the Maintenance state, the relative path probability increases slightly.

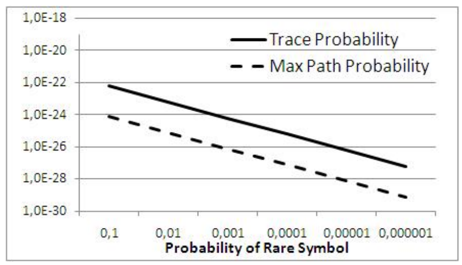

Fig. 6. Development of Trace Probability and Maximum Path Probability for Decreasing Symbol Output Probability

The trace contains a symbol which is rare for some transition but not for the another. As the symbol gets rarer, trace and maximum path probabilities decrease linearly (compare Figure 6). The relative path probability decreases and then drops sharply.

The trace does not contain rare Symbols. The rareness of the symbol has hardly any effect on the result values. Only the higher probability of the other output symbols affects the path and trace probabilities.

4.6 Result Discussion

The experiments showed the following results. If the given output sequence can be generated by a path containing a rare event, which has a larger variance, the path probability decreases with the decrease of the rate of the event. The relative probability of the most likely generating path seems to converge to a fixed value. If the symbol output is directly associated to the rare event in the path, the number of possible generating paths, runtime and memory requirement decrease considerably. In general, the existence of rare events, if these are part of the generating paths, decreases the runtime and memory requirement of the HnMM Proxel algorithm. This is most likely due to the fact, that the algorithm discards possible system developments below a given threshold. This pruning does not affect the HnMM results negatively, since only relatively unlikely paths are discarded, which have no pronounced effect on the overall trace probability. Rare symbols do not affect the algorithm performance measures. They only influence the trace and path probabilities, which decrease, as the event becomes less frequent.

5 Conclusion

The paper gives an introduction to Hidden non-Markovian Models, which are an extension of Hidden Markov Models to more general hidden models. A Proxel based solution algorithm for the evaluation and decoding of HnMM is described. The experiment section investigates the effect of rare events and rare symbols on the algorithm performance and relevant HnMM result measures. In general one can say, that the effect of a rare event in the results of Decoding and Evaluation of HnMM is as expected. The rarer the event, the smaller the probability of the generating paths and traces that involve the rare event. The algorithm performance on the other hand is positively affected by the existence of rare events. Memory requirement and runtime decrease as the event becomes rarer.

The algorithm presented holds improvement potential regarding storage schemes and retrieval strategies. Future work also includes the formalization of HnMM and possibly the adaption of the original HMM solution algorithms introduced in section 2.1. Further challenges will arise when examining possible application areas of the proposed paradigm.

Hidden non-Markovian Models enable the solution of important questions, which cannot be answered yet today. Therefore, further research in this area is valuable not only to the academic world. Some application areas of HnMM involve the study of failure or unusual system behavior, leading to models containing rare events. The positive effect of rare events on the behavior of the analysis method, which we observed in our experiments, seems promising. It suggests that HnMM and the described Proxel solution algorithm are very well applicable to the analysis of models containing rare events.

References

[BPSW70] L. E. Baum, T. Petrie, G. Soules and N. Weiss. *A Maximization Technique in the Statistical Analysis of Probabilistic Functions of Markov Chains*. The Annals of Mathematical Statistics, 41(1):164–171, 1970.

[For73] G. D. Forney. *The Viterbi Algorithm*. In: Proceedings of the IEEE, Jahrgang 61, S. 268–278, March 1973.

[Ger95] R. German. *Transient Analysis of Deterministic and Stochastic Petri Nets by the Method of Supplementary Variables*. In: Proceedings of the 3rd International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 95), S. 394–398. IEEE Computer Society, 1995.

[Hor02] G. Horton. *A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times*. In: Proceedings of the European Simulation Symposium 2002, S. 129–136. SCS European Publishing House, 2002.

[IWH06] C. Isensee, F. Wickborn and G. Horton. *Training Hidden Non-Markov Models*. In: Proceedings of 13th International Conference on ANALYTICAL and STOCHASTIC MODELLING TECHNIQUES and APPLICATIONS, Bonn, Germany, S. 105–110, May 2006.

[JBM75] F. Jelinek, L. R. Bahl and R. L. Mercer. *Design of a linguistic statistical decoder for the recognition of continuous speech*. IEEE Transactions on Information Theory, IT-21:250–256, 1975.

[Jel76] F. Jelinek. *Continuous speech recognition by statistical methods*. In: Proceedings of the IEEE, Jahrgang 64, S. 532–536, April 1976.

[KH07] C. Krull and G. Horton. *Expanded Hidden Markov Models: Allowing Symbol Emissions at State Changes*. In: Proceedings of 14th International Conference on ANALYTICAL and STOCHASTIC MODELLING TECHNIQUES and APPLICATIONS, Prague, Czech Republic, 2007.

[Koser] J. Koserski. *Analyse der Ratingmigrationen interner Ratingsysteme mit Markov-Ketten, Hidden-Markov-Modellen und Neuronalen Netzen*. Dissertation, Otto-von-Guericke-Universität Magdeburg, 2006 September.

[LM05] S. Lazarova-Molnar. *The Proxel-Based Method: Formalisation, Analysis and Applications*. Dissertation, Otto-von-Guericke-University Magdeburg, November 2005.

[Rab89] L. R. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. In: Proceedings of the IEEE, Jahrgang 77, S. 257–286, February 1989.

[RC87] M. J. Russel and A. E. Cook. *Experimental evaluation of duration modelling techniques for automatic speech recognition*. In: Proceedings of the ICASSP'87, S. 2376–2379, 1987.

[RM85] M. J. Russel and R. K. Moore. *Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition*. In: Proceedings of the ICASSP'85, Tampa, Florida, S. 5–8, March 1985.

[Sim05] T. H. Simon. *Anwendung des Hidden Markov Modell-Ansatzes auf die Proxel-basierte Simulation*. Thesis, Otto-von-Guericke-Universität Magdeburg, September 2005.

[WH06] F. Wickborn and G. Horton. *Reducing the Effect of Stiffness For a State Space-Based Simulation Method Using Adaptive Time Steps*. In: 6th International Workshop on Rare Event Simulation, Bamberg, Germany, October 2006.

[WIS$^+$06] F. Wickborn, C. Isensee, T. Simon, S. Lazarova-Molnar and G. Horton. *A New Approach for Computing Conditional Probabilities of General Stochastic Processes*. In: Proceedings of 39th Annual Simulation Symposium 2006, Huntsville, USA, S. 152–159, April 2006.