Applied RESTART Estimation of General Reward Measures

Armin Zimmermann
Real-Time Systems and Robotics
Technische Universität Berlin, Germany*
Sekr. EN10, Einsteinufer 17, 10587 Berlin
Email: azi@cs.tu-berlin.de

Abstract

The RESTART method is a robust and efficient technique for the simulation of systems which are subject to significant rare events. Just like other rare-event simulation techniques, it estimates the probability of reaching a certain event or of being in a specific state set. Quantitative evaluation of discrete-event systems however requires a much broader type of performance measures, usually specified in terms of *rate* and *impulse rewards* on the stochastic process. This paper presents an extension of RESTART allowing this type of general reward measures. A hypothetical revenue-optimal setup of the future ETCS train control system is computed with the method as an example, based on a stochastic Petri net model and a prototype implementation.

Key words

RESTART/Splitting, General reward measures, Stochastic Petri nets, European Train Control System (ETCS)

1 Introduction

Model-based evaluation of safety-critical embedded systems is an important aid in their design. This is especially true for non-functional properties; examples are real-time capabilities, fault-tolerance, and performance while taking failures as well as heavy load situations into consideration.

Their computer-based evaluation requires a model with performance measures and an evaluation technique implemented in some software tool. In the following we adopt stochastic Petri nets, which have proved to be applicable to a wide area of technical systems. They are considered to describe discrete event systems in a concise and appropriate way. An additional advantage is the availability of many different analysis and simulation techniques as well as software tools. The method demonstrated here is however not restricted to Petri nets as the underlying model.

Rare-event simulation is the only tractable method to evaluate such models if they are subject to multiple non-Markovian activity delays and low probabilities of the states under inspection. Several approaches have been investigated in the literature to overcome this problem; overviews are e.g. given in [2, 3, 4]. They have the common goal to make the

*Until September 2006: Hasso-Plattner-Institute of IT Systems Engineering at the University of Potsdam

rare event happen more frequently in order to gain more significant samples out of the same number of generated events. We consider the RESTART method [11, 10] here because of its robustness and wide range of applicability.

Rare-event simulation approaches concentrate on an estimation of the probability of a rare state set A in transient or steady-state. Others derive the probability of reaching a state or event before another state is hit again. This is however a significant restriction in the context of embedded systems evaluation and their respective models. A much wider applicability can be achieved if general quantitative measures are derivable, which heavily depend (perhaps only in some of their terms) on rare events or states.

The paper shows how RESTART can be used to estimate more general reward measures. Reward variables are functions that return some value of interest from the stochastic process of a stochastic discrete-event model. Following the characterization given in [6], both *rate rewards* and *impulse rewards* (which are associated to *states* and *events* of the process, respectively) are considered. The standard probability of a rare event (or state) set is a special case. The method is based on a variant of RESTART described in [8].

A part of the future European Train Control System (ETCS) is used as an application example belonging to the class of distributed, safety-critical real-time systems. Its planned operation overcomes track space splitting into fixed blocks, and is based on mobile communication. The underlying GSM-R radio communication is a crucial factor for safe and efficient operation. ETCS (level 3) real-time behavior under inevitable link failures is modeled and evaluated to demonstrate the proposed technique. The paper uses a recently developed [7] stochastic Petri net model of location and movement authority data packet exchange between trains and radio block centers. It is adapted here to capture hypothetical revenues and costs related to train operation, allowing the derivation of an economically optimal train distance. A prototype extension of the software tool TimeNET [14] is used for the experiments.

The paper is structured as follows. The RESTART method is briefly revisited in the subsequent Section 2. Its standard usage for the estimation of a low probability is extended in Section 3 to allow arbitrary reward measure types. Section 4 demonstrates the method with an application example.

2 The RESTART Method

Recent treatments of the RESTART technique can be found in [9, 10]. In the following the topic is only briefly touched to introduce some notation.

Assume that the goal of a simulation is to estimate the probability $\mathrm{P}\{A\}$ of being in a set of state A in steady state, and that significant samples are generated only rarely due to the model. Let the set of all reachable states of a model be denoted by B_0, and the initial state of the system by σ_0.

A standard simulation would require a very long run time until A has been visited sufficiently often to estimate $\mathrm{P}\{A\}$. A is visited more frequently by concentrating on promising paths in the state set.

Formally, define M subsets $B_1 \ldots B_M$ of the overall state space B_0 such that

$$A = B_M \quad \text{and} \quad B_M \subset B_{M-1} \subset \ldots \subset B_1 \subset B_0$$

The conditional probabilities $\mathrm{P}\{B_{i+1} \mid B_i\}$ of being in an enclosed set B_{i+1} under the precondition of being in B_i are much easier to estimate than $\mathrm{P}\{A\}$, because every one of them is not rare if the B_i are chosen properly. The measure of interest can then be obtained from the product of the conditionals (obviously $\mathrm{P}\{B_0\} = 1$).

$$\mathrm{P}\{A\} = \prod_{i=0}^{M-1} \mathrm{P}\{B_{i+1} \mid B_i\}$$

States visited during a simulation must be mapped to the respective sets B_i. An *importance function* f_I returns a real value for each state $\sigma \in B_0$.

$$f_I : B_0 \to \mathbb{R}$$

A set of *thresholds* (denoted by $Thr_i \in \mathbb{R}, i = 1 \dots M$) divides the range of importance values such that the state set B_i can be obtained for a state[1].

$$\begin{aligned} \forall i \in \{0 \dots M\} \quad &: \quad Thr_{i+1} > Thr_i \\ \sigma \in B_i \quad &\iff \quad f_I(\sigma) \geq Thr_i \end{aligned}$$

We say that the simulation is in a *level* i if the current state σ belongs to $B_i \setminus B_{i+1}$.

An importance splitting simulation measures the conditional probability of reaching a state out of set B_{i+1} after starting in B_i by a Bernoulli trial. If B_{i+1} is hit, the entering state is stored and the simulation trial is split into R_{i+1} trials. The simulation follows each of the trials to see whether B_{i+2} is hit and so on. A trial starting at B_i is canceled after leaving B_i if it did not hit B_{i+1}. Simulation of paths inside B_0 and $B_M = A$ is not changed.

An estimator of $\mathrm{P}\{A\}$ using R_0 independent replications is then [8]

$$\widehat{\mathrm{P}\{A\}} = \frac{1}{R_0 R_1 \dots R_{M-1}} \sum_{i_0=1}^{R_0} \dots \sum_{i_{M-1}=1}^{R_{M-1}} \mathbf{1}_{i_0} \mathbf{1}_{i_0 i_1} \dots \mathbf{1}_{i_0 i_1 \dots i_{M-1}}$$

if we denote by $\mathbf{1}_{i_0 i_1 \dots i_j}$ the result of the Bernoulli trial at stage j, which is either 1 or 0 depending on its success.

The reduction in computation time results from estimating the conditional probabilities $\mathrm{P}\{B_{i+1} \mid B_i\}$, which are not rare if the sets B_i are selected properly. Experiences show that the technique works robustly for a wide range of applications [10, 9], even if the parameters are not chosen optimally following the rules given in the mentioned papers.

Several variants of RESTART have been considered in the literature [1]. We follow the approach taken in [5, 8], which can be characterized as *fixed splitting* and *global step* according to [1]. The first aspect corresponds to the number of trials into which a path is split when it reaches a higher level. The second issue governs the sequence in which the different trials are executed. Global step has the advantage to store fewer intermediate simulation states.

Following the presentation in [8], the steady-state value of our example measure $\mathrm{P}\{A\}$ is for a standard simulation given by

$$\mathrm{P}\{A\} = \lim_{T \to \infty} \frac{1}{T} \int_0^T \mathbf{1}_A(t) \, \mathrm{d}t$$

if we denote by $\mathbf{1}_A(t)$ the indicator variable that is either one or zero, depending on whether the current state of the simulation at time t is in A.

An estimator for this steady-state measure for a RESTART implementation needs correction factors that take into account the splitting. We adopt the method of [8], where *weights* ω are maintained during the simulation run, which capture the relative importance of the current path elegantly. This makes the division by $R_0 R_1 \dots R_{M-1}$ obsolete.

The weights are computed as follows: A simulation run starting from the initial state $\sigma_0 \in (B_0 \setminus B_1)$ has an initial weight of 1, because it is similar to a "normal" simulation run without splitting. Whenever the simulation path currently in level i crosses the border to an upper level u, the path is split into R_u paths, which are simulated subsequently. The weight is obviously divided by R_u upon splitting. Paths leading to a level $< i$ are discarded except for the last one, which is followed further using the stated rules. The weight of the last path

[1] We assume $Thr_0 = -\infty$ and $Thr_{M+1} = \infty$ here to simplify notation.

is multiplied by R_i when it leaves level i downwards. The weights of the previously discarded paths are thus taken back into consideration, to maintain an overall path probability of one. This procedure is repeated until the required result quality is achieved. This technique has the additional advantage of allowing "jumps of levels" over more than one threshold compared to the original method.

Based on the weight factors, an estimator for the steady-state probability of A is

$$\widehat{\mathrm{P}\{A\}} = \frac{1}{T} \int_0^T \omega(t)\, \mathbf{1}_A(t)\ \mathrm{d}t \tag{1}$$

for a large T. T counts in this context only the time spent in final paths, i.e. in the last path of each split.

3 RESTART for Extended Reward Measures

Instead of estimating $\mathrm{P}\{A\}$, the goal is to obtain an estimation of a reward variable *rvar*. This extension is useful for all performance measures that significantly depend on rewards gained in areas of the state space which are only visited rarely. For simplicity of notation, we restrict ourself to one (possibly complex) measure which is assumed to be analyzed in steady state.

3.1 Reward Measures

Reward variables $rvar(SProc)$ are functions that return some value of interest from the stochastic process *SProc* of a stochastic discrete-event model. This process describes the state σ and possibly happening events E at time t, $SProc = \{(\sigma(t), E(t)), t \in \mathbb{R}^{0+}\}$. Reward measures may be defined by numerical expressions containing reward measures, but in the following we use the two terms equally to simplify presentation.

Reward variables describe combinations of a (positive) bonus or (negative) penalty associated to elements of the stochastic process. Two types of elements of such a reward variable have been identified in the literature [6]. This was based on the basic observation that the stochastic process of a discrete event system remains in a state for some time interval and then changes to another state due to an activity execution, which takes place instantaneously. The natural way of defining a reward variable thus includes rate rewards $rrate(\sigma)$ which are accumulated over time in a state σ, and impulse rewards $rimp(e)$ which are gained instantaneously at the moment of an event $e \in E$.

We first introduce an intermediate function $R_{inst}(t)$. This value can be interpreted as the instantaneous reward gained at a point in time t. It is a generalized function containing a Dirac impulse Δ if there is at least one impulse reward collected in t.

$$R_{inst}(t) = \underbrace{rrate\big(\sigma(t)\big)}_{\text{rate rewards}} + \underbrace{\Delta \cdot \sum_{e \in E(t)} rimp(e)}_{\text{impulse rewards}} \tag{2}$$

We define the reward variable value in steady-state

$$rvar(SProc) = \lim_{x \to \infty} \frac{1}{x} \int_0^x R_{inst}(t)\, \mathrm{d}t \tag{3}$$

This leads to a simulation estimator $\widehat{rvar}$ in the sense of Equation (1).

$$\widehat{rvar} = \frac{1}{T} \int_0^T \omega(t)\, R_{inst}(t)\ \mathrm{d}t \tag{4}$$

where T is the (sufficiently large) maximum simulation time spent in final paths, and $R_{inst}(t)$ denotes the instantaneous reward gained at time t which is derived by the simulation. $\omega(t)$ denotes the weight as described in the previous section.

It should be noted that the RESTART algorithm stores states with simulation times after a new level has been reached to restart there, which is not visible in Equation 4. Specifically, the algorithm may visit a simulation time t several times with possibly different current states and weights. The equation should thus be read as taking the integral over all paths visited until the global time T (counting only final paths) is reached.

4 An Application Example — ETCS Train Control

The future European Train Control System (ETCS) will be based on mobile communication and overcome the standard operation with fixed blocks. It is introduced in order to increase track utilization and interoperability throughout Europe while reducing trackside equipment cost. With level 3 of ETCS implementation, trains and control centers are connected by mobile communication links. Classic trackside equipment will be obsolete. The safety of passengers depends on the communication system reliability. It is subject to hard safety requirements, but has to deal with inherent soft real-time aspects (communication delay jitter and packet losses).

An envisioned advantage of ETCS level 3 is an increased track utilization: dropping the standard block synchronization of trains and migrating to a virtual block system has the potential of allowing closer distances between trains. Transmission errors in the communication system influence the minimum possible distance between trains and thus the maximum track utilization. This dependency is addressed and evaluated in [12, 13] for the first time. Communication system, failure behavior and safety braking of trains are modeled and analyzed using different performance evaluation techniques in the following. The results show that with the current state of specifications, shorter distances than with today's technology will not be possible.

The aim of the application example analysis here is to find a driving distance for trains that is (in a hypothetical sense) revenue-optimal. We assume for simplification a continuous track without stops, on which trains follow each other with a maximum speed v (current high-speed trains have a maximum speed of 300 km/h) and a distance *distance*. To ensure safety of the system, worst-case assumptions are made for all timings, distances etc. Practical values will be worse because trains have different speeds, need to follow their timetable, and accelerate or brake due to trackside conditions.

Because there is no fixed block assigned to a train, and no physical block borders exist, the train movement is controlled by exchanging messages with the radio block center (RBC). Each train periodically checks its integrity and sends this information together with the current position of the train head to the RBC. We assume the time between two message creations to be 5 sec, because this is the minimum value in the specifications, but has a major impact on the possible distance.

The integrity/position report is sent via GSM-R to the RBC and processed there, which takes 0.5 sec typically. The resulting information is sent to the following train, telling it either that everything is fine to go on driving (by sending a new *movement authority* packet that extends the free track before it) or that an emergency braking is necessary immediately. Messages may be lost on the up- or downlink with an estimated probability of 1.88% (Packets have a typical length of 190 bit, and GSM-R bit error rate is specified as 10^{-4}). The mean time for a complete message transfer between train and RBC (over GSM-R and ISDN link) is 0.45 sec.

If the second train does not receive a new movement authority message for some time, it needs to decide on its own at what point of time emergency braking is inevitable out of

safety reasons. There is obviously a deadline t after the last movement authority has been received, when the train needs to be stopped. The worst-case assumption is that after the last integrity check of train one has been completed, a part of the train's carriages are lost from the main train and stop where they are or there is an accident. The movement authority therefore shall never exceed the "min safe rear end" of the preceding train in moving block operation. The corresponding dependency for the deadline $t = \frac{\mathit{distance} - 3000\,\mathrm{m}}{83\,\mathrm{m/sec}} - 5\,\mathrm{sec}$ is derived in detail in [12, 13], to which the interested reader is referred.

4.1 An ETCS Communication Model

A model of the position report message exchange and emergency braking due to communication problems is shown below. It represents a variant of previous models introduced in [13, 7]. The goal of the mentioned papers was to analyze the dependency between maximum throughput of trains and reliability measures of the communication system. Here we take a look at revenue-optimal train distances. However, our main goal is to demonstrate the use of extended performance measures. Behavior and quantitative parameters of the model may not detailed enough to obtain realistic results. The general issue of an optimal train distance w.r.t. throughput and emergency braking should nevertheless become evident.

Model parameters are chosen based on a set of resources listed in [13], up to a big part representing requirements of the planned ETCS setup. In the following we adopt worst-case assumptions based on the requirements, because there would otherwise be no guarantee of a working integrated system.

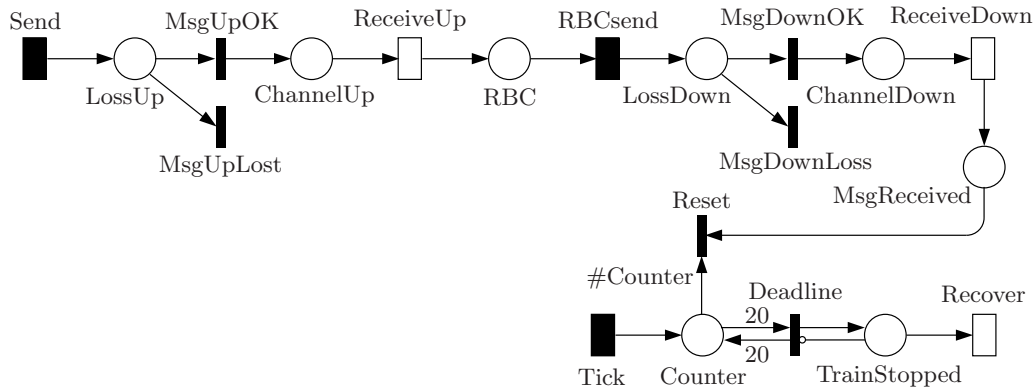

Figure 1: Communication model during moving block operation

Figure 1 shows a deterministic and stochastic Petri net (DSPN) model for the ETCS movement authority data exchange. The upper part models the generation of the position/integrity report and its transmission to the following train via the RCB. Transition `Send` models the generation of a new message that assures train integrity and contains the current position. The message transfer to the RBC is modeled by `ChannelUp` and `ReceiveUp` with exponentially distributed firing time, while a packet is lost when immediate transition `MsgUpLost` fires. Processing at the RBC corresponds to the deterministic transition `RCBsend` with delay 0.5 sec, and the transmission to the following train in the same way as the uplink.

For the application of the RESTART simulation method, place `Counter` is introduced in which the number of tokens corresponds to possible thresholds. Every time a new movement authority message arrives at the second train (place `MsgReceived`), the current elapsed time is set to zero: transition `Reset` fires and removes all tokens from place `Counter`. The train stops after an exceeded deadline (i.e. 20 tokens in place `Counter`, followed by a firing of

Deadline), and we assume an exponentially distributed Recover time of 30 minutes before the train can move on.

We define a performance measure to calculate the hypothetical revenue of train operation per second and track kilometer, depending on the train's head-to-head distance.

- In normal operation, 7 € are gained per train and second from passenger fares
- Whenever an emergency braking occurs, singular costs of 200.000 € are assumed
- In addition to that, a stopped train leads to a cost of 600 € per second, which also includes stopping of following trains
- Assuming a volume-based cost structure of the communication channel, one cent (0.01 €) per transferred message in a channel is paid per second
- To take track utilization into account, the above influences have to be multiplied by 1000 m/*distance* (the number of trains per km)

Applied to the model, the resulting performance measure can be informally expressed as

$$\big(7 * (1 - \#\texttt{TrainStopped}) - 600 * (\#\texttt{TrainStopped}) - 200000 * (\#\texttt{Deadline}) \\ -0.01 * (\#\texttt{ChannelUp} + \#\texttt{ChannelDown})\big) * 1000/\mathit{distance}$$

where #P means a rate reward equaling the number of tokens in place P, and #T an impulse reward of one for the firing of transition T. As we are interested in the average value per model time unit, the performance measure is computed by accumulating the reward over the simulation run and dividing it by the simulation time.

4.2 Numerical Results

Acceptable stop probabilities are naturally very small (e.g. 10^{-12} for *distance* = 8000 m), and thus a rare-event simulation technique is necessary to successfully derive the measure.

The model of the moving block operation has been evaluated using a prototype implementation that is planned to be included in TimeNET [5, 14]. Thresholds are defined based on the number of tokens in place Counter. The number of thresholds is manually selected in the prototype[2], based on the formulas in [11] and an estimation of the probability of the rare event.

Figure 2 shows the resulting revenue per track kilometer and second, depending on the train head-to-head distance. The optimal distance is 5800m, but the results in the range from 5600 to 5900 do not differ significantly. The two main influences are the rare event of train stops and the reciprocal linear effect of train distance on track utilization. All simulation runs were executed until the rare event was hit at least 1000 times, which required 68 billion events to be simulated in the hardest case of *distance* = 8400 m. This took only a few minutes run time on a Pentium Mobile with 1.86 MHz under Windows XP. The same prototype with the RESTART technique switched off is not able to generate any rare event for interesting settings of *distance* in an acceptable time.

The analysis shows a significant impact of communication delays and packet losses on an economical train operation under ETCS level 3.

[2]TimeNET calculates them based on a prior standard simulation run with limited computation time

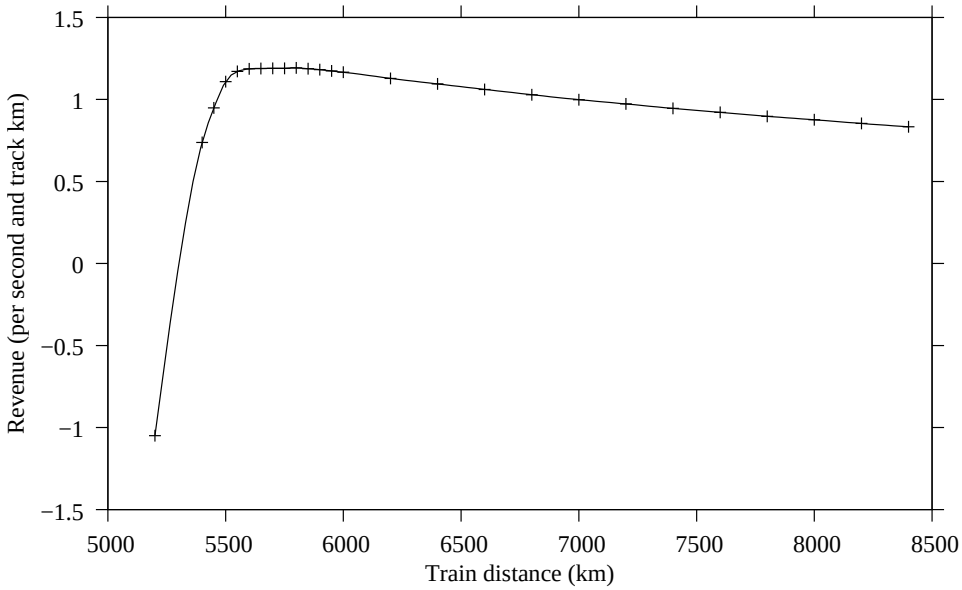

Figure 2: Train revenue versus train distance

5 Conclusion

In this paper we have shown how the RESTART method for rare-event simulation can be applied to stochastic discrete-event systems containing more general performance measures than only the probability of one rare event. We extend the technique to general reward measures containing impulse and rate rewards. The method is demonstrated using the European Train Control System as an application example. A hypothetical revenue-optimal train distance is derived.

References

[1] M. J. Garvels and D. P. Kroese, "A comparison of RESTART implementations," in *Proc. 1998 Winter Simulation Conference*, 1998.

[2] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic, "Multilevel splitting for estimating rare event probabilities," *Operations Research*, vol. 47, pp. 585–600, 1999.

[3] C. Görg, E. Lamers, O. Fuß, and P. Heegaard, "Rare event simulation," Computer Systems and Telematics, Norwegian Institute of Technology, Tech. Rep. COST 257, 2001.

[4] P. Heidelberger, "Fast simulation of rare events in queueing and reliability models," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 5, no. 1, pp. 43–85, 1995.

[5] C. Kelling, "A framework for rare event simulation of stochastic Petri nets using RESTART," in *Proc. of the Winter Simulation Conference*, 1996, pp. 317–324.

[6] W. H. Sanders and J. F. Meyer, "A unified approach for specifying measures of performance, dependability, and performability," in *Dependable Computing for Critical Applications*, Dependable Computing and Fault-Tolerant Systems, A. Avizienis and J. Laprie, Eds. Springer Verlag, 1991, vol. 4, pp. 215–237.

[7] J. Trowitzsch and A. Zimmermann, "Using UML state machines and Petri nets for the quantitative investigation of ETCS," in *Proc. Int. Conf. on Performance Evaluation Methodologies and Tools (VALUETOOLS 2006)*, Pisa, Italy, 2006, accepted for publication.

[8] B. Tuffin and K. S. Trivedi, "Implementation of importance splitting techniques in stochastic Petri net package," in *Computer Performance Evaluation, Modelling Techniques and Tools — 11th Int. Conf., TOOLS 2000*, Lecture Notes in Computer Science, C. U. S. Boudewijn R. Haverkort, Henrik C. Bohnenkamp, Ed., vol. 1786. Schaumburg, IL, USA: Springer Verlag, 2000, pp. 216–pp.

[9] M. Villén-Altamirano and J. Villén-Altamirano, "Analysis of RESTART simulation: Theoretical basis and sensitivity study," *European Transactions on Telecommunications*, vol. 13, no. 4, pp. 373–385, 2002.

[10] ——, "Optimality and robustness of RESTART simulation," in *Proc. 4th Workshop on Rare Event Simulation and Related Combinatorial Optimisation Problems*, Madrid, Spain, Apr. 2002.

[11] M. Villén-Altamirano, J. Villén-Altamirano, J. Gamo, and F. Fernández-Cuesta, "Enhancement of the accelerated simulation method RESTART by considering multiple thresholds." in *Proc. 14th Int. Teletraffic Congress*. Elsevier Science Publishers B. V., 1994, pp. 797–810.

[12] A. Zimmermann and G. Hommel, "A train control system case study in model-based real time system design," in *Proc. 11th Int. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS03)*, Nice, France, 2003.

[13] ——, "Towards modeling and evaluation of ETCS real-time communication and operation," *Journal of Systems and Software*, vol. 77, pp. 47–54, 2005.

[14] A. Zimmermann, M. Knoke, A. Huck, and G. Hommel, "Towards version 4.0 of TimeNET," in *13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB 2006)*, March 2006, pp. 477–480.