Zuverlässigkeitsbewertung komplexer Systeme mit Stochastischen Petri-Netzen und TimeNET

Prof. Dr.-Ing. **A. Zimmermann**, Technische Universität Ilmenau;
Dr. **A. Hildebrandt**, Pepperl+Fuchs GmbH, Mannheim

Kurzfassung

Die Zuverlässigkeit komplexer technischer Systeme hängt oft von ihrem dynamischen Verhalten ab. Petri-Netze sind sehr viel besser als klassische Modelle zuverlässiger Systeme dazu geeignet, derartige Abläufe zu beschreiben. Vorschläge dazu macht die Norm IEC 62551 (Analysis techniques for dependability — Petri net techniques). Für hinreichend komplexe bzw. nicht rein Markovsche Systeme lässt sich nur Simulation zur Berechnung der Zuverlässigkeitskenngrößen einsetzen. Ein offenes Problem ist hier aber die sehr lange Laufzeit für die statistisch abgesicherte Bestimmung sehr kleiner (Ausfall-)Wahrscheinlichkeiten. Der Beitrag beschreibt stochastische Petri-Netze in der Zuverlässigkeitsmodellierung und das RESTART-Verfahren zu ihrer beschleunigten Simulation. Das Softwarewerkzeug TimeNET wird vorgestellt, mit dem stochastische Petri-Netze für zuverlässige Systeme benutzerfreundlich modelliert und effizient untersucht werden können. Ein Anwendungsbeispiel zeigt Möglichkeiten und Vorteile der Methode.

1. Einleitung

Für einen systematischen Entwurf von Automatisierungssystemen ist in vielen Anwendungsbereichen die Vorhersage nichtfunktionaler Eigenschaften vor der Realisierung notwendig. Das ist gerade in frühen Entwurfsphasen schwierig, da sich derartige Eigenschaften meist erst aus dem Zusammenspiel aller Komponenten ergeben (sog. *emergent properties*). Andererseits sind sie gerade in sicherheitskritischen Anwendungen ähnlich wichtig wie funktionale Anforderungen, insbesondere wenn eine Zertifizierung vorgeschrieben ist.

Mit Hilfe eines Modells und geeigneter Analyseverfahren können Entscheidungen fundiert gefällt und die Auswirkung von Entwurfsparametern untersucht werden. Dafür stehen einige erfolgreiche Modelle und Verfahren zur Verfügung [1]; in der Praxis werden vor allem statische Modelle wie Fehlerbäume eingesetzt. Die Zuverlässigkeit komplexer technischer Systeme lässt sich aber unter anderem durch Fehlertoleranzmaßnahmen wie Rekonfiguration oder *cold standby* erhöhen, deren Auswirkung nur mit ihrem dynamischen Verhalten analysiert werden können [10]. Darüber hinaus können das dynamische Zusammenspiel von Systemverhalten und Steuerung sowie eine veränderliche Umgebung Einfluss auf die Zuverlässigkeit haben. Ohne die Betrachtung dieser Eigenschaften werden Modelle und ihre

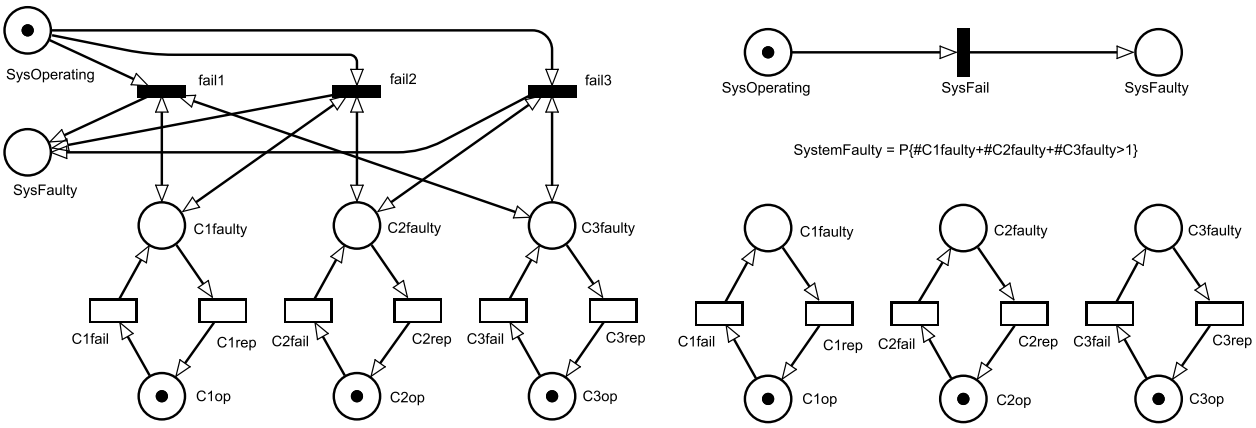
Aussagen unrealistisch oder unnötig konservativ, so dass die gesuchte Systemvariante mit dem besten Kosten-Nutzen-Verhältnis nicht gefunden werden kann. Petri-Netze [14] und ihre Erweiterung um stochastische Zeit sind sehr gut dazu geeignet, derartige komplexe dynamische Abläufe zu beschreiben [11]. Vorschläge dazu macht die IEC 62551 (Analysis techniques for dependability — Petri net techniques). Einen Überblick über verschiedene Modelle sowie Kombinationen von statischen und dynamischen Beschreibungen und ihre Modellierungsmächtigkeit geben unter anderem [4, 12]. Für einfache dynamische Modelle mit gedächtnislosem Zeitverhalten können Markov-Ketten eingesetzt und numerisch analysiert werden [2].

Für Modelle mit großem Zustandsraum oder solchen mit nicht rein Markovschen Zeitverteilungen lässt sich nur Simulation zur Berechnung der Zuverlässigkeitskenngrößen einsetzen [5]. Letzteres ist in vielen technischen Systemen durch Taktzyklen, deterministische Wartungsintervalle oder Weibull-verteilte Lebensdauern häufig der Fall. Ein offenes Problem ist hier aber die sehr lange Laufzeit für die statistisch abgesicherte Bestimmung von Kenngrößen der Zuverlässigkeit, da die interessierenden Ereignisse im Modell nur sehr selten auftreten. Dieses Problem ist als Simulation seltener Ereignisse (*rare-event simulation*) bekannt, und wird in der Literatur vor allem mit Techniken des *importance sampling* [9] oder *splitting* [8] gelöst. Den Verfahren ist gemeinsam, dass die interessierenden Ereignisse in der Simulation forciert werden, um aus der gleichen Anzahl simulierter Ereignisse mehr signifikante Information zu gewinnen. Die Ergebnisse müssen geeignet umgerechnet werden, um die den normalen Simulationsablauf verfälschenden Änderungen zu korrigieren. Splitting-Verfahren und dabei insbesondere der RESTART-Algorithmus [16] haben den Vorteil, weniger vom simulierten Modell abhängig zu sein. Sie sind leichter automatisierbar und daher geeigneter für eine Implementierung in einem Softwarewerkzeug, das auch für Systementwickler ohne detailliertes Hintergrundwissen der mathematischen Zusammenhänge anwendbar ist. Aktuelle Forschungsarbeiten in der Arbeitsgruppe System- und Software-Engineering der TU Ilmenau beschäftigen sich mit der Simulation komplexer zuverlässiger Systeme, die über die Ausnutzung struktureller Eigenschaften von Petri-Netzen um mehrere Größenordnungen beschleunigt werden kann [18, 21].

Der vorliegende Beitrag beschreibt die Vorteile von Petri-Netzen für die Zuverlässigkeitsmodellierung und -bewertung komplexer technischer Systeme. Der folgende Abschnitt stellt stochastische Petri-Netze anhand eines Zuverlässigkeitsmodells vor. Es wird kurz gezeigt, wie die in der IEC 62551 vorgeschlagenen Modelle typischer Zuverlässigkeitsmuster mit üblichen Modellerweiterungen vereinfacht werden können. Abschnitt 3 gibt einen Überblick über das RESTART-Verfahren zur beschleunigten Simulation vom Modellen mit seltenen Ereignissen. Der folgende Abschnitt stellt das Softwarewerkzeug TimeNET vor, das Analyseverfahren und seltene Ereignissimulation für normale und farbige Petri-Netze unterstützt. Der Nutzen der Methode wird dann in Abschnitt 5 mit einem Anwendungsbeispiel (Zuverlässigkeit eines Hybridantriebs, nach [13]) gezeigt.

2. Modellierung zuverlässiger Systeme mit stochastischen Petri-Netzen

Dieser Abschnitt gibt einen kurzen Einstieg in stochastische Petri-Netze. Für Details, die hier aus Platzgründen nicht beschrieben werden, wird auf die verfügbare Literatur dazu verwiesen [11, 3, 17]. Begleitend zu den Erklärungen wird das kleine Beispiel links in Bild 1a verwendet, dem Vorschlag der IEC 62551 zur Modellierung eines Systems mit 2-aus-3 redundanten Komponenten.



(a) Modell entsprechend IEC 62551

(b) Vereinfachtes Modell

Abbildung 1: Petri-Netz-Modelle für ein 2-aus-3-System

Ein *stochastisches Petri-Netz* kann formal als ein 7-Tupel SPN $= (P, T, \vec{Pre}, \vec{Post}, \Lambda, W, \vec{m}_0)$ definiert werden [17]. Die *Stellen* $p_i \in P$ werden als Kreise dargestellt und entsprechen diskreten Zustandsvariablen, deren aktueller Wert in einem Zustand durch die Anzahl der in ihnen enthaltenen Marken (schwarze Kreise bzw. Zahl) angegeben wird. Im Beispiel modellieren die Stellen `SysOperating` und `SysFaulty` den Zustand des Gesamtsystems, während die weiter unten liegenden Stellen `C1op` und `C1faulty` (bzw. ihre Entsprechungen für die anderen Komponenten C2 und C3) den lokalen Zustand der redundanten Komponenten abbilden. Die *Markierung* $\vec{m}$ eines Petri-Netzes ordnet jeder Stelle eine natürliche Zahl zu $\vec{m} : P \to \mathbb{N}$, und beschreibt so den *Zustand des Modells* (und damit des dadurch definierten stochastischen Prozesses).

Die Menge aller theoretisch möglichen *Markierungen* (nicht alle sind tatsächlich erreichbar) wird mit $M = \{\vec{m} \mid \forall p \in P : \vec{m}(p) \in \mathbb{N}\}$ bezeichnet. Die grafische Darstellung eines Petri-Netzes zeigt wie in Bild 1a den Anfangszustand (die *initiale Markierung*) $\vec{m}_0 \in M$. Komponente C2 ist am Anfang intakt, daher ist $\vec{m}(\texttt{C2op}) = 1$.

Die Menge T repräsentiert die *Transitionen*, d.h. aktiven Elemente des Modells, die als Rechtecke verschiedener Art — abhängig von ihrem Typ — gezeichnet werden. `C1fail` und `C1rep` modellieren im Beispiel die Ausfall- und Reparaturvorgänge der ersten Komponente. Ein Modellelement kann nicht Stelle *und* Transition sein ($T \cap P = \emptyset$), und ein gültiges Petri-Netz sollte nicht leer sein $T \cup P \neq \emptyset$.

Kanten können nur Stellen mit Transitionen (und umgekehrt) verbinden, und beschreiben die Voraus-

setzungen für die Aktivierung der Transition sowie die Veränderung des Zustands bei Ablauf der durch eine Transition modellierten Aktivität. *Eingangskanten* $\vec{Pre} : P \times T \to \mathbb{N}$ verbinden Stellen mit Transitionen, und geben deren Multiplizität (oder Kantengewicht) an. Diese wird grafisch nur angegeben, wenn sie ungleich 1 ist. Wenn es keine Kante von Stelle p_i zu Transition t_j gibt, gilt $\vec{Pre}(p_i,t_j) = 0$. Die Kantengewichte der *Ausgangskanten* sind dementsprechend durch $\vec{Post} : P \times T \to \mathbb{N}$ gegeben. Kanten mit Pfeilspitze auf beiden Seiten im Bild sind nur die Darstellung zweier regulärer Kanten in beiden Richtungen übereinander; hier wird so die Funktion einer Testkante nachgebildet, die den Inhalt der verbundenen Stelle nicht verändert.

Die *Schaltzeit* Λ einer Transition gibt die Zeit an, die nach der Aktivierung der Transition vergehen muss, bevor sie tatsächlich schaltet. Diese Zeit kann zufällig entsprechend einer gegebenen Verteilungsfunktion verteilt sein. Ein Sonderfall sind *zeitlose Transitionen*, die sofort nach ihrer Aktivierung schalten ($\Lambda(.) = 0$) und grafisch durch gefüllte, dünne Rechtecke dargestellt werden (Beispiel: `fail1`, die schaltet, wenn Komponenten 1 und 3 ausgefallen sind). W spezifiziert für alle zeitlosen Transitionen eine reelle Zahl, die das *Schaltgewicht* angibt. Dieses entspricht der relativen Wahrscheinlichkeit, dass die Transition schaltet, falls mehrere in Konflikt befindliche Transitionen gleichzeitig schalten können.

Das Verhalten des Modells bzw. Semantik des Petri-Netzes wird durch die Schaltregeln definiert. Eine Transition t wird als *schaltfähig* in einer Markierung $\vec{m}$ bezeichnet, wenn genug Marken in ihren Eingangsstellen enthalten sind: $\forall p \in P : \vec{m}(p) \geq \vec{Pre}(p,t)$.

Zu Beginn der Schaltfähigkeit einer Transition wird ihr eine *Restschaltzeit* zugeordnet, die zufällig entsprechend der Schaltzeitverteilung der Transition gezogen wird. Die Restschaltzeiten aller Transitionen verringern sich dann mit gleicher Geschwindigkeit. In einem Zustand verbringt der stochastische Prozess daher so lange Zeit, bis die erste Restschaltzeit Null wird.

Die schnellste Transition t (für den Fall mehrerer Transitionen wird zufällig ausgewählt) *feuert*, und verändert den aktuellen Zustand des Petri-Netzes $\vec{m}$ zu einem Neuen $\vec{m}'$ durch Verringern der Markenanzahl in den Eingangsstellen und Erhöhen der Anzahl in den Ausgangsstellen: $\vec{m} \xrightarrow{t} \vec{m}'$ mit $\forall p \in P : \vec{m}'(p) = \vec{m}(p) + \vec{Post}(p,t) - \vec{Pre}(p,t)$.

In der Literatur gibt es viele Erweiterungen für Petri-Netze, unter anderem *hemmende Kanten* zum Testen von Stellen auf Markenanzahl. Diese verbinden Stellen mit Transitionen und enden dort in einem kleinen Kreis (siehe z.B. Bild 5). Wenn die verbundene Stelle mindestens so viele Marken enthält, wie die Multiplizität der Kante angibt, ist die Transition nicht schaltfähig. Die Multiplizität ist gegeben durch $Inh : P \times T \to \mathbb{N}$, und für die Schaltfähigkeit von Transition t in Markierung $\vec{m}$ muss dann zusätzlich gelten: $\forall p \in P : \big(Inh(p,t) > 0\big) \longrightarrow \big(Inh(p,t) > \vec{m}(p)\big)$.

Eine andere Erweiterung sind *guards* zeitloser Transitionen G: boolesche Funktionen über der Markierung, die Wahr sein müssen, damit die Transition schaltfähig ist $G : T \times M \mapsto \{\text{Wahr}, \text{Falsch}\}$.

Für die Schaltfähigkeit einer zeitlosen Transition t in Markierung $\vec{m}$ muss dann zusätzlich gelten $G(t,\vec{m}) = \text{Wahr}$.

Im Standard zur Zuverlässigkeitsmodellierung mit Petri-Netzen [3] werden Subnetze zeitloser Transitionen zur „Implementierung“ der Zuverlässigkeitsabhängigkeiten vom Systemverhalten vorgeschlagen, wie sie sonst klassisch vielleicht in Fehlerbäumen oder RBDs ausgedrückt würden. Das führt zum Modellieren von Abläufen nicht des Systems, sondern der Berechnung des Systemzustands. Besser geeignet und einfacher ist die Verwendung von markierungsabhängigen Ausdrücken, wie sie beispielsweise in Softwarewerkzeugen wie TimeNET (siehe Abschnitt 4) unterstützt werden. Das Beispiel in Bild 1a kann dann beispielsweise so verändert werden, dass der ganze obere Teil der zeitlosen Transitionen weggelassen wird und das interessierende Zuverlässigkeitsmaß direkt in einem Ausdruck wie `SystemFaulty` spezifiziert wird. Für allgemeine m-aus-n-Systeme wären sonst sehr große Teilmodelle für alle kombinatorischen Möglichkeiten nötig. Falls der Systemzustand (intakt / ausgefallen) explizit in extra Stellen realisiert werden muss, kann das wie im rechten Bild 1b oben gezeigt einfach mit einer Transition `SysFail` erreicht werden, deren *guard* dann ebenfalls `#C1faulty + #C2faulty + #C3faulty > 1` ist[1].

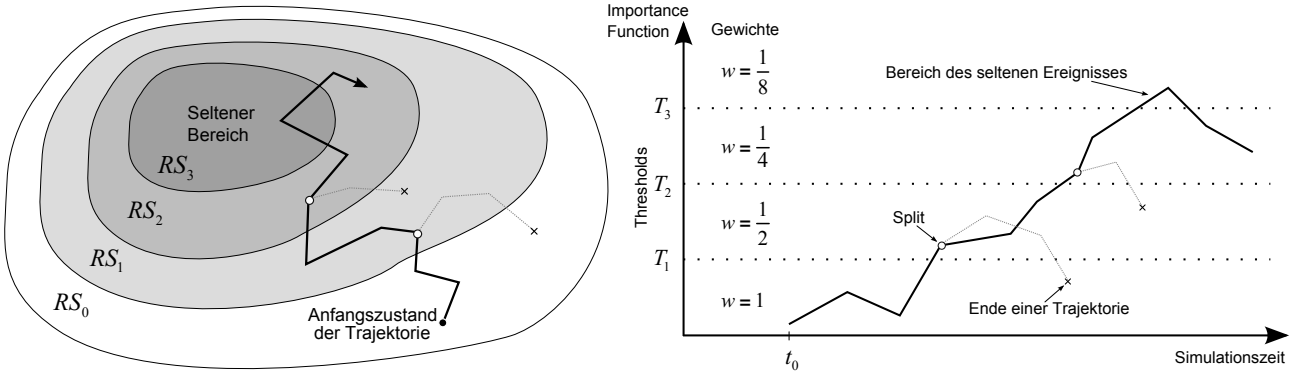
3. Effiziente Simulationsverfahren für Zuverlässigkeitsmodelle

Die Berechnung von Zuverlässigkeits- und Leistungskenngrößen der mit stochastischen Petri-Netzen (oder anderen dynamischen Modellen) spezifizierten Systeme ist aus den eingangs genannten Gründen in vielen Fällen nur mit Simulationsverfahren möglich [3]. Für eine statistisch abgesicherte Aussage sind in vielen sicherheitskritischen Systemen aber unakzeptabel lange CPU-Zeiten nötig, um ausreichend viele interessierende Ereignisse zu erzeugen.

Verfahren zur beschleunigten Simulation seltener Ereignisse können die Effizienz um Größenordnungen steigern. Splitting-Methoden und insbesondere RESTART (*repetitive simulation trials after reaching thresholds*, [16]) haben unter den zur Verfügung stehenden Techniken den Vorteil, relativ unabhängig vom Modell zu funktionieren und robust gegen Parametereinflüsse zu sein. Sie setzen voraus, dass eine heuristische Schätzfunktion (*importance function* $f_I(\vec{m})$) existiert, die für „näher“ am interessierenden Bereich liegende Zustände $\vec{m}$ größer wird. Das ist häufig der Fall, da Systemausfällen oft ein Teilausfall, Alterung oder Abnutzung vorausgeht.

Bild 2a symbolisiert die Funktionsweise: Angenommen, Zustände im Bereich $A = RS_3$ sind gefährlich, und die (sehr kleine) Wahrscheinlichkeit für ihr Auftreten soll berechnet werden. Die Menge aller erreichbaren Zustände sei RS_0, die in Teilmengen $RS_1 \ldots RS_L$ unterteilt wird, so dass $A = RS_L$ und

[1] In allen Fällen kann das Modell sehr vereinfacht werden, falls die Komponenten gleiches Ausfall- und Reparaturverhalten haben, da dann die drei Teilmodelle zu einem Modell mit drei Marken integriert werden können.



(a) Simulationstrajektorien im Zustandsgraphen

(b) Schwellen und Gewichte

Abbildung 2: RESTART-Simulation

$RS_L \subset RS_{L-1} \subset \ldots \subset RS_1 \subset RS_0$.

Die Zugehörigkeit eines Zustands zu einer der Teilmengen RS_i wird durch die Schätzfunktion $f_I : RS_0 \to \mathbb{R}$ festgelegt, gemeinsam mit einer Menge Schwellwerte $Thr_i \in \mathbb{R}, i = 1 \ldots L$. Die Teilmengen sind dann spezifiziert durch $RS_i = \{\vec{m} \in RS_0 \mid f_I(\vec{m}) \geq Thr_i\}$. Ein Zustand $\vec{m}$ gehört zu einem Schwellwert i genau dann wenn $\vec{m} \in RS_i \setminus RS_{i+1}$.

Die Idee hinter Splitting-Verfahren ist nun, statt der direkten Schätzung der Wahrscheinlichkeit $\mathrm{P}\{A\} = \mathrm{P}\{RS_L\}$ eines Aufenthalts in RS_L die einzelnen bedingten Wahrscheinlichkeiten für das Erreichen des jeweils nächsthöheren Schwellwerts $\mathrm{P}\{RS_{i+1} \mid RS_i\}$ zu berechnen. Das gewünschte Ergebnis lässt sich dann aus deren Produkt berechnen.

Die Simulation folgt normalerweise mit hoher Wahrscheinlichkeit Trajektorien im nicht seltenen Bereich, da sie ja vor allem das normale Systemverhalten durchläuft. Mit RESTART wird die simulierte Trajektorie immer dann, wenn die Schätzfunktion eine Schwelle überschreitet, in mehrere Trajektorien aufgeteilt. Trajektorien, die eine Schwelle unterschreiten, werden (bis auf die letzte) nicht weiter verfolgt. Im Algorithmus kann dazu einfach der neue Zustand gespeichert werden (kleine Kreise im Bild 2a), um bei beendeten Trajektorien wieder dort beginnen zu können. Im Bild wird jeweils zweifach geteilt, im Allgemeinen kann dieser Faktor R_i für die Zustandsmenge RS_i auch größer gewählt werden. In den Bereichen RS_0 und $RS_L = A$ läuft die Simulation normal ab.

Eine derartige Simulation benötigt Korrekturfaktoren, die die Aufteilung der Trajektorien berücksichtigt. Jeder Trajektorie wird ein aktuelles Gewicht ω zugeordnet, das anfangs den Wert 1 hat und bei jedem Aufteilen in R_i neue Trajektorien jeder davon nach Division durch R_i zugewiesen wird. Wenn der letzte „überlebende“ Pfad einer Aufteilung einen Schwellwert i unterschreitet, wird sein Gewicht wieder mit dem entsprechenden Faktor R_i multipliziert. Bild 2b skizziert dieses Verhalten.

Eine Simulation sammelt während ihres Ablaufs Teilergebnisse, deren gewichteter Mittelwert den jeweils aktuellen Schätzwert des Zuverlässigkeitsmaßes bildet. Maße können von Zuständen und Ereignissen abhängen, und definieren letztlich eine Funktion F des (durch das Modell beschriebenen)

stochastischen Prozesses bzw. seines Zustands zu jedem Zeitpunkt t. Im Unterschied zu einer normalen Simulation müssen nach dem hier beschriebenen Verfahren lediglich die Teilergebnisse mit dem jeweils aktuellen Gewicht $\omega(t)$ der Trajektorie multipliziert werden. Eine stationäre Simulation berechnet so letztlich eine Näherung des durch F definierten Leistungsmaßes mit

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T \omega(t)\, F(t)\, \mathrm{d}t,$$

wobei T die Simulationszeit angibt.

Für den genauen Ablauf des Algorithmus gibt es verschiedene Varianten [6]. Optimale Beschleunigung erzielt das Verfahren mit theoretisch bestimmbaren Schwellwerten und geeigneter Schätzfunktion [16], aber auch sonst ist eine Beschleunigung um mehrere Größenordnungen erreichbar. Aktuelle Forschungsarbeiten zeigen, dass die Parameter einer RESTART-Simulation für Petri-Netze mit Hilfe struktureller Eigenschaften zum Teil automatisch bestimmt werden können [21]. Das ist ein prinzipieller Vorteil dieser Modellart gegenüber Markov-Ketten oder Simulationssprachen.

4. Modellierung und Bewertung mit dem Softwarewerkzeug TimeNET

Für die Modellierung und Zuverlässigkeitsbewertung mit Hilfe stochastischer Petri-Netze kann das Softwarewerkzeug TimeNET [19] eingesetzt werden. Es unterstützt unter anderem farblose Petri-Netze der Modellklasse *extended deterministic and stochastic Petri nets*, (eDSPN [7]) und verschiedene numerische Analyse- und Simulationsverfahren dafür. Eine Besonderheit ist die numerische Analyse von nicht-Markovschen Modellen. Das Werkzeug wurde für die Berechnungen in der IEC 62551 eingesetzt [3].

Darüber hinaus können farbige stochastische Petri-Netze für komplexere Systeme genutzt wer-

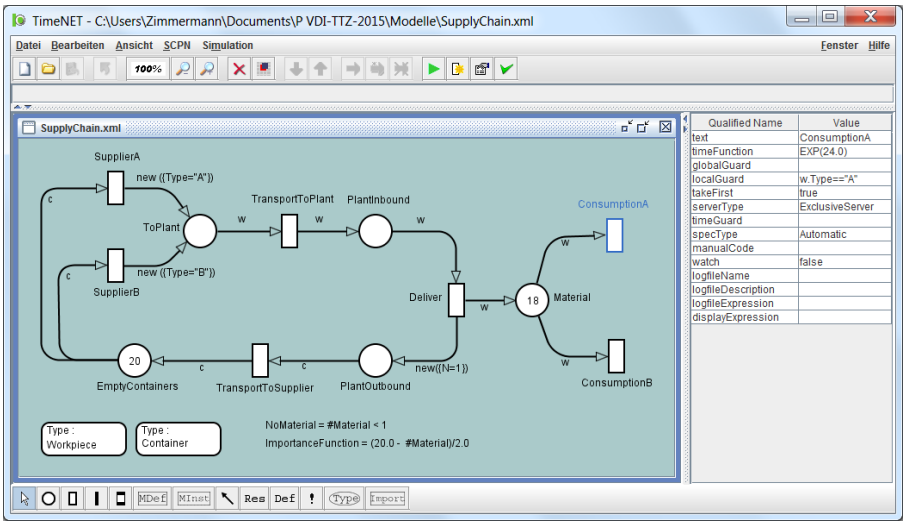


Abbildung 3: Benutzungsoberfläche von TimeNET

den [17]. Für beide Modellklassen sind effiziente Simulationsverfahren für stationäre und transiente Fragestellungen sowie das oben beschriebene RESTART-Verfahren implementiert. TimeNET ist (soweit bekannt) das einzige Werkzeug, das die Simulation seltener Ereignisse für stochastische farbige Petri-Netze unterstützt. Das Werkzeug wurde ursprünglich an der TU Berlin realisiert, seit 2008 läuft die Weiterentwicklung an der TU Ilmenau.

Bild 3 zeigt ein Beispiel der Benutzungsoberfläche mit einem farbigen Petri-Netz eines Logistikmodells. Das Werkzeug ist aktuell für Windows 7 und Linux (jeweils 32 und 64 Bit-Versionen) verfügbar. Es kann für nichtkommerzielle Zwecke kostenlos genutzt werden, weitere Informationen unter `http://www.tu-ilmenau.de/TimeNET`. Für einen Überblick über andere Softwarewerkzeuge sei hier aus Platzgründen auf [20] und die Petri-Netz-Homepage `http://www.informatik.uni-hamburg.de/TGI/PetriNets/` verwiesen.

5. Ein Anwendungsbeispiel

Der Einsatz von Hybridantrieben in Kraftfahrzeugen kann zur Einsparung fossiler Energieträger und zur Verringerung des innerstädtischen Ausstoßes an Luftschadstoffen beitragen. Der effiziente Betrieb derartiger Antriebssysteme muss zahlreiche Randbedingungen wie Energieverbrauch, Batterielebensdauer und Zuverlässigkeit beachten, und ist nur mit einer komplexen Steuerungssoftware und hohem Entwurfs- und Testaufwand möglich. Modellbasierter Entwurf und Bewertung von Leistungsfähigkeit und Zuverlässigkeit haben daher einen großen Stellenwert für frühe Entwurfsphasen.

Im Folgenden wird ein paralleler Vollhybridantrieb betrachtet, bei dem Verbrennungsmotor und Elektromotor einzeln oder gemeinsam zur Beschleunigung beitragen können. Der grundsätzliche Aufbau ist in Bild 4 dargestellt; er ist wie die Modellannahmen und das Fahrmodell [13] entnommen.

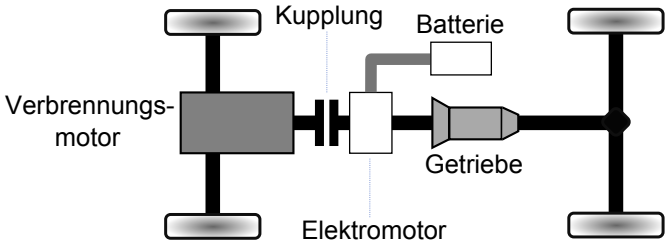


Abbildung 4: Vereinfachter Aufbau eines parallelen Vollhybridantriebs nach [13]

Diese Art Hybridantrieb kann in mehreren Modi verwendet werden: solange Batteriekapazität und durch den Elektromotor lieferbare Kraft ausreichen, kann im reinen Elektrobetrieb (*e-drive*) gefahren werden, und der Verbrennungsmotor ist abgeschaltet. Fällt die Ladung unter einen Schwellwert, muss die Batterie geladen werden. Der Verbrennungsmotor wird dann gestartet und mit einem höheren Drehmoment betrieben, als für die reine Fahrt nötig wäre. Ein Teil der Energie wird über den Elektromotor im Generator-Betrieb in elektrischen Strom gewandelt, mit dem die Batterie geladen

wird (sog. *charge mode*).

Beim kurzfristigen Beschleunigen können Elektro- und Verbrennungsmotor gemeinsam betrieben werden (*boost mode*). Die kinetische Energie kann bei Bremsvorgänge zum Teil über den Elektromotor zurückgewonnen und in der Batterie gespeichert werden (*regenerative braking*).

Die Lebensdauer der Batterie als oft teuerstem Bauteil des Fahrzeugs hängt entscheidend von den Ladezyklen ab. Diese lassen sich zum Teil von der Steuerlogik beeinflussen, hängen aber natürlich auch vom unvorhersehbaren Fahrverhalten und der Lastsituation ab. Für eine lange Lebensdauer sollten aktuelle Batterietypen nicht über 80% geladen und nicht unter 40% ihrer Kapazität entladen werden[2]. Ein gebräuchlicher Typ sind Lithium-Ionen-Batterien mit einer Energiedichte von über 200 Wh/kg, die für eine Lebensdauer von 5 Jahren und 100 000 Meilen ausgelegt sind[3].

Im Systementwurf können die nichttrivialen Abhängigkeiten zwischen Fahrprogramm, Ladesteuerung, Systemzuverlässigkeit und Batterielebensdauer kaum ohne eine modellbasierte Untersuchung beurteilt werden. Details zur Größe der Energieflüsse, dem angenommenem Fahrverhalten und weiteren numerischen Parametern des Beispiels wurden [13] entnommen. Im Unterschied zu der dort vorgenommenen Untersuchung verschiedener Batteriegrößen wird im Folgenden eine Kapazität von 1310 Wh angenommen, wie sie beispielsweise im Toyota Prius verbaut ist.

Ein Petri-Netz-Modell des Hybridantriebs

Das beschriebene Anwendungsbeispiel wird in diesem Abschnitt mit einem stochastischen Petri-Netz modelliert, um im Anschluss Zuverlässigkeitskenngrößen zu berechnen. Im Unterschied zu [13] wird hier kein farbiges Modell benutzt, da die zusätzlichen Möglichkeiten farbiger Petri-Netze für das Modell nicht unbedingt nötig sind[4].

Bild 5 zeigt das Modell, bei dessen Gestaltung versucht wurde, die einzelnen Teile übersichtlich zu gruppieren. Links ist das Modell des Fahrprogramms mit einzelnen Phasen dargestellt. In der Mitte befindet sich das Modell der Betriebsmodi des Hybridantriebs, und rechts das Batteriemodell mit Energieverbräuchen und Steuerung. Teilmodelle weiter rechts sind von den Zuständen der Modelle links abhängig.

Das Modell des Fahrprogramms ist ähnlich einem Automaten aufgebaut. Zustände entsprechen markierten Stellen; zu Beginn parkt das Fahrzeug (Marke in der Stelle `Parking`). Die weiteren entsprechenden Stellen `Maneuvering`, `IntraUrban` und `ExtraUrban` bilden die weiteren Phasen Ein- bzw. Ausparken, innerstädtischer Verkehr sowie außerorts ab. Die sechs dazu gehören-

[2] *state of charge*, SOC

[3] Siehe `http://en.wikipedia.org/wiki/Electric_vehicle_battery`

[4] Das Beispiel in [13] wurde mit dem Softwarewerkzeug REALIST [15] simuliert.

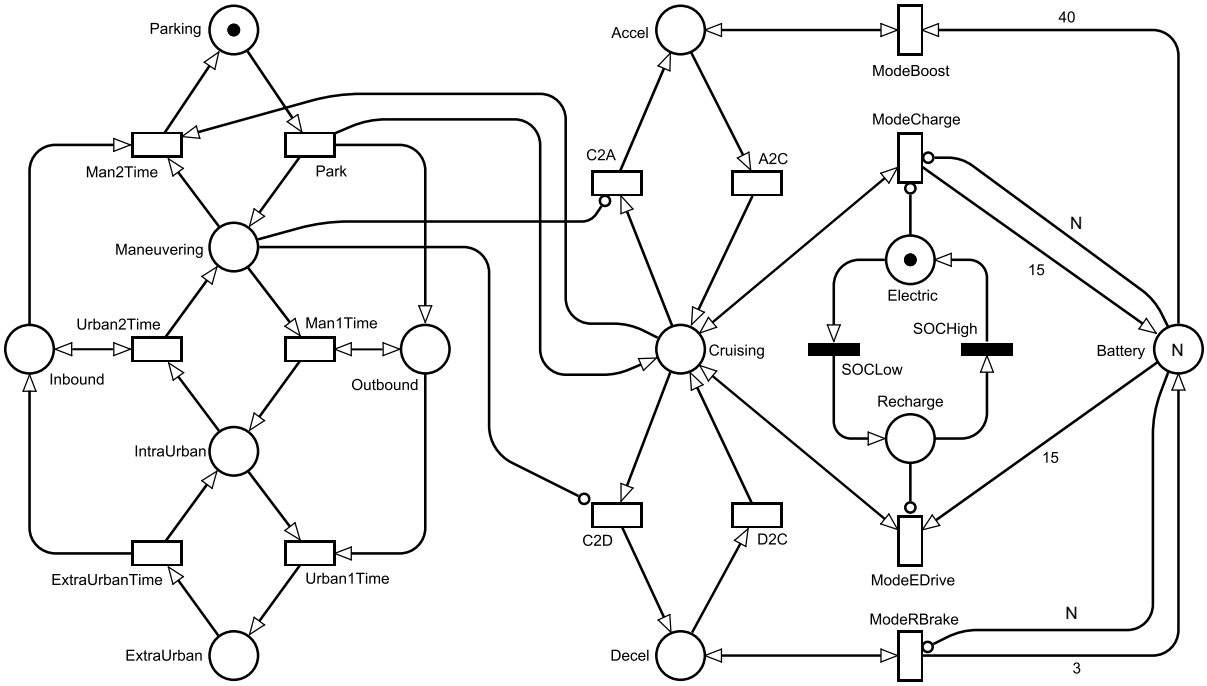


Abbildung 5: Stochastisches Petri-Netz für den Hybridantrieb

den Transitionen modellieren jeweils die Zustandsübergänge und haben eine der Dauer der Phase entsprechende Schaltzeit (Werte übernommen aus [13]). Die Stellen `inbound` und `outbound` erzwingen die gewünschte Abfolge der Phasen; die dem Fahrzeug entsprechende Marke bewegt sich zunächst in die unteren Stellen und danach wieder nach oben zum Parken.

Die aktuelle Phase beeinflusst das in der Mitte dargestellte Teilmodell der Betriebsmodi. Auch hier orientiert sich das Modell an einem Automaten: die drei Zustände Fahren, Bremsen und Beschleunigen entsprechen den Stellen `Cruising`, `Decel` und `Accel`. Das Teilmodell ist während des Parkens leer, erst beim Losfahren wird eine Marke in `Cruising` erzeugt. Die vier Transitionen zwischen den genannten Stellen bilden die Zustandsübergänge ab, ihre zugeordneten Schaltzeiten wurden wiederum entsprechend [13] gewählt.

Der Unterschied zwischen der innerstädtischen Fahrt und außerorts besteht in unterschiedlich häufigen Beschleunigungs- und Bremsvorgängen, was im hier dargestellten Modell einfach mit einer (von der Markierung von Stelle `ExtraUrban`) abhängigen Schaltzeit der Transitionen `C2A` und `C2D` erreicht wird. In [13] wird angenommen, dass während der ersten und letzten Fahrphase (`Maneuvering`) nur elektrisch gefahren wird und keine für das Batteriemodell signifikanten Brems- und Beschleunigungsvorgänge betrachtet werden müssen. Das wird hier mit den hemmenden Kanten an den beiden genannten Transitionen erreicht.

Ganz rechts im Bild 5 befindet sich Stelle `Battery`, die mit ihrer enthaltenen Anzahl Marken den aktuellen Ladezustand der Batterie modelliert. Da es sich um ein zustandsdiskretes Modell handelt, muss diese physikalische Größe mit einer bestimmten Diskretisierung abgebildet werden. Dabei

muss zwischen hoher Genauigkeit einerseits und schneller Zuverlässigkeitsbewertung andererseits abgewogen werden. In den durchgeführten Experimenten (siehe folgender Abschnitt) stellten sich 100 Wattsekunden als sinnvolle Bezugsgröße für eine Marke heraus. Der Parameter N wird daher im Modell für eine Batteriegröße von 1310 Wh bei maximaler Ladung von 80% auf 37 728 festgelegt. Die Modellzeit wird als Sekunden interpretiert.

Energieverbrauch bzw. -erzeugung durch das Hybridfahrzeug während der einzelnen Betriebsmodi modellieren die vier mit `Mode...` bezeichneten Transitionen. Während des Bremsens (Stelle `Decel`) werden durch jedes Schalten der Transition `ModeBrake` drei Marken zur Stelle `Battery` hinzugefügt, falls diese nicht bereits N Marken enthält und die Batterie daher bereits zu 80% gefüllt ist. Die Schaltzeit der Transition ist 0.2, damit ist die mittlere beim Bremsen generierte Energie 1.5kW wie in [13]. Die Schaltzeit der anderen Transitionen ist 0.1, so dass sich die angegebenen Markenanzahlen pro Schaltvorgang für 40 kW bzw. 15 kW ergeben.

Während beim Beschleunigen und Bremsen jeweils der Betriebsmodus eindeutig festgelegt ist, wird dieser im normalen Fahren (Stelle `Cruising`) vom Batteriezustand abhängig gesteuert. Das ist eine Erweiterung gegenüber [13] und orientiert sich an der Modellierung einer (hier sehr einfachen) Steuerung nach [3][5]. Die Steuerung befindet sich entweder im Zustand `Electric` (bei ausreichend hohem Ladestand), oder im Zustand `Charge`, in dem der Verbrennungsmotor dauerhaft läuft und die Batterie lädt. Die Transitionen `SOCLow` und `SOCHigh` schalten beispielsweise bei weniger als 18 864 bzw. mindestens 37 728 Marken für den Bereich $40\% \ldots 80\%$. Dieses Verhalten wird mit zustandsabhängigen Bedingungen (*guards*) realisiert, hätte aber auch durch hemmende und normale Kanten abgebildet werden können. Das sofortige Schalten wird durch zeitlose (*immediate*) Transitionen erreicht.

Im Modell wurden zunächst nur exponentielle Schaltzeiten gewählt, für die Simulation gibt es hier aber keine numerischen Einschränkungen. Das Modell ist im Vergleich zum farbigen Vorbild aus [13] kompakt und enthält nur 28 Stellen bzw. Transitionen gegenüber den dort angegebenen 69.

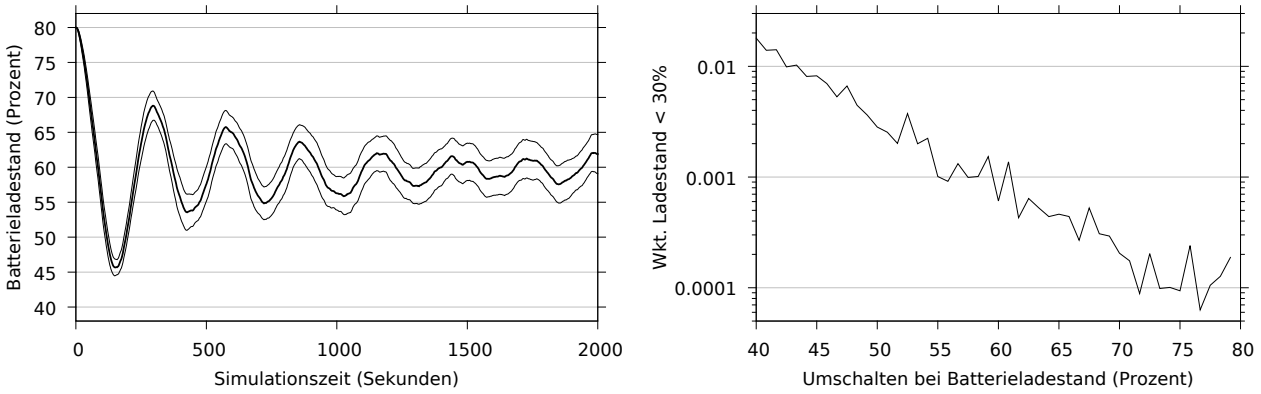
Zuverlässigkeitsbewertung

Anhand des Modells kann nun untersucht werden, wie sich Entwurfsparameter des Systems und Steuerstrategie auf die Leistungs- und Zuverlässigkeitskenngrößen auswirken. Die Ergebnisse können helfen, eine sinnvolle Abwägung zwischen optimaler Batteriesteuerung (Ladestand immer zwischen 40% und 80%) und Sicherheit / Zuverlässigkeit zu treffen. Ein Sicherheitsrisiko kann hier eine unerwartete oder ausbleibende Reaktion auf Fahrereingaben sein, wenn beispielsweise bei einem Beschleunigungsvorgang nicht genug Energie für die elektrische Unterstützung zur Verfügung steht

[5] Die bereits beschriebenen Modellteile entsprechen dem nicht steuerbaren *plant model*.

oder diese während eines Überholvorgangs abgebrochen werden muss. Während in [13] die Batterieladung priorisiert wird, soll hier der umgekehrte Weg untersucht werden: Befehle des Fahrers werden soweit möglich ausgeführt, und die Folgen für die Batterieladung untersucht.

Alle Berechnungen wurden mit TimeNET auf einem Laptop mit einer Intel Core i7-4600U CPU bei 2.1 GHz durchgeführt. Als Abbruchkriterium für die Simulationen wurde jeweils ein Konfidenzintervall von 95% und ein relativer Fehler von 5% gewählt. In der ersten Untersuchung wurde zur Überprüfung der Wirkung der Steuerung eine transiente Simulation im Zeitintervall $[0, 2000]$ durchgeführt, die sich ergebenden mittleren Ladestände zeigt Bild 6a. Das geschieht in TimeNET einfach über die Definition eines Leisungsmaßes `E{#Battery}` für den Erwartungswert der entsprechenden Markenanzahl und Umrechnung in Prozent.



(a) Verlauf der Batterieladung (mit Konfidenzintervall) (b) Einfluss des Umschaltpunktes

Abbildung 6: Simulationsergebnisse

Die in Bild 5 gezeigte, „naive“ Steuerung schaltet erst bei Erreichen der Grenzwerte der Batterieladung um und wird daher recht häufig den Ladestand 40% unterschreiten. Das Leistungsmaß hierfür lautet `P{#Battery<18864}` für die Wahrscheinlichkeit, dass die Anzahl der Marken die 40% entsprechende Menge unterschreitet. Mit Hilfe der Experiment-Funktion von TimeNET wurden automatisch Simulationen mit zwischen 40% und 80% Ladestand variierendem, höherem Umschaltwert für die Steuerung durchgeführt. Die sich ergebende deutlich kleinere Wahrscheinlichkeit für eine zu geringe Batterieladung zeigt Bild 6b; ein ungefähr logarithmischer Zusammenhang ist erkennbar.

Nehmen wir einen milden Hybridbetrieb des Fahrzeugs an, in dem außer beim Beschleunigen immer geladen wird. Dann sollten niedrige Ladezustände noch möglich, aber sehr unwahrscheinlich sein. Angenommen, bestimmte Batteriezustände wie z.B. $SOC < 30\%$ sind sehr kritisch für die Batterielebensdauer, aber bis zu einem Ladezustand von 20% würden aus Sicherheitsgründen trotzdem Beschleunigungsvorgänge noch elektrisch unterstützt werden. Für die Berechnung des für die Lebensdauer kritischen Zeitanteils, in dem $20\% \leq SOC \leq 30\%$ gilt, wurde eine etwas angepasste Modellvariante untersucht und damit das Ergebnis 1.38749E-5 bestimmt. Die normale Simulation von TimeNET benötigt dafür 3183 Sekunden Rechenzeit, während das im Werkzeug implementierte

RESTART-Verfahren lediglich 272 Sekunden brauchte. Erfahrungen zeigen, dass die erreichbare Beschleunigung für noch schwerere Probleme mit deutlich selteneren Ereignissen um mehrere Größenordnungen höher ist [17].

6. Zusammenfassung

Stochastische Petri-Netze sind ein geeignetes Werkzeug zur Modellierung und Analyse der Zuverlässigkeit von komplexen dynamischen Systemen. Die IEC 62551 beschreibt Arbeitsschritte und gibt Hinweise zur Modellierung. Für die Untersuchung nicht Markovscher Modelle zuverlässiger Systeme in akzeptabler Zeit sind Verfahren zur beschleunigten Simulation seltener Ereignisse notwendig. Der vorliegende Bericht beschreibt die Anwendung des RESTART-Verfahrens, das in aktuellen Forschungsarbeiten für Petri-Netze weiterentwickelt wird. Zuverlässige Systeme können so durch die Ausnutzung struktureller Eigenschaften von Petri-Netzen um mehrere Größenordnungen schneller untersucht werden. Außerdem werden das dafür eingesetzte Softwarewerkzeug TimeNET und ein Anwendungsbeispiel (Zuverlässigkeit eines Hybridantriebs) vorgestellt.

Literatur

[1] BERTSCHE, Bernd: *Reliability in Automotive and Mechanical Engineering*. Springer, 2008

[2] Norm DIN EN 00338 Mai 2006. *Application of Markov techniques*

[3] Norm DIN EN 00338 September 2013. *Analysis techniques for dependability — Petri net techniques*

[4] EVERDIJ, M. H. C. ; BLOM, H. A. P.: Petri nets and hybrid state Markov processes in a power hierarchy of dependability models. In: *Proc. IFAC Conf. on Analysis and Design of Hybrid Systems*, 2003, S. 355–360

[5] FAULIN, Javier (Hrsg.) ; JUAN, Angel A. (Hrsg.) ; MARTORELL, Sebastián (Hrsg.) ; RAMÍREZ-MÁRQUEZ, José-Emmanuel (Hrsg.): *Simulation methods for reliability and availability of complex systems*. Springer, 2010. – xvii + 315 S.

[6] GARVELS, Marnix J. ; KROESE, Dirk P.: A Comparison of RESTART Implementations. In: *Proc. 1998 Winter Simulation Conference*, 1998

[7] GERMAN, Reinhard: *Performance Analysis of Communication Systems, Modeling with Non-Markovian Stochastic Petri Nets*. John Wiley and Sons, 2000

[8] GLASSERMAN, Paul ; HEIDELBERGER, Philip ; SHAHABUDDIN, Perwez ; ZAJIC, Tim: Multilevel Splitting for Estimating Rare Event Probabilities. In: *Operations Research* 47 (1999), S. 585–600

[9] GLYNN, Peter W. ; IGLEHART, D. L.: Importance sampling for stochastic simulations. In: *Management Science* 35 (1989), November, Nr. 11, S. 1367–1392

[10] KOLOWROCKI, Krzysztof ; SOSZYNSKA-BUDNY, Joanna: *Reliability and Safety of Complex*

Technical Systems and Processes: Modeling - Identification - Prediction - Optimization. Springer, 2013 (Springer Series in Reliability Engineering)

[11] MALHOTRA, M. ; TRIVEDI, K.S.: Dependability Modeling Using Petri-Net Based Models. In: *IEEE Trans. on Reliability* 44 (1995), Nr. 3, S. 428–440

[12] MUPPALA, Jogesh K. ; FRICKS, Ricardo M. ; TRIVEDI, Kishor S.: Techniques for System Dependability Evaluation. In: GRASSMANN, Winfried K. (Hrsg.): *Computational Probability* Bd. 24. Springer US, 2000. – ISBN 978–1–4419–5100–7, S. 445–479

[13] NEBEL, S. ; DIETER, A. ; MÜLLER, P. ; BERTSCHE, B.: Application of ECSPN to RAMS modeling and analysis of hybrid drive systems. In: *Proc. Reliability and Maintainability Symposium (RAMS 2010)*, 2010, S. 1–6

[14] REISIG, Wolfang: *Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien*. Vieweg+Teubner, 2010 (Leitfäden der Informatik). – 248 pages; ISBN 978-3-8348-1290-2

[15] TROST, M. ; NEBEL, S. ; BERTSCHE, B.: Modellierungs- und Simulationsumgebung REALIST. In: SCHNIEDER, E. (Hrsg.): *Proc. Entwicklung komplexer Automatisierungssysteme (EKA 2006)*. Braunschweig, may 2006

[16] VILLÉN-ALTAMIRANO, Manuel ; VILLÉN-ALTAMIRANO, Jose: Analysis of RESTART Simulation: Theoretical Basis and Sensitivity Study. In: *European Transactions on Telecommunications* 13 (2002), Nr. 4, S. 373–385

[17] ZIMMERMANN, Armin: *Stochastic Discrete Event Systems*. Springer, Berlin Heidelberg New York, 2007

[18] ZIMMERMANN, Armin: RESTART Simulation of Colored Stochastic Petri Nets. In: *Proc. 7th Int. Workshop on Rare Event Simulation (RESIM 2008)*. Rennes, France, September 2008, S. 143–152

[19] ZIMMERMANN, Armin: Modeling and evaluation of stochastic Petri nets with TimeNET 4.1. In: *Proc. 6th Int. Conf on Performance Evaluation Methodologies and Tools (VALUETOOLS)* IEEE, 2012, S. 54–63

[20] ZIMMERMANN, Armin: Reliability Modelling and Evaluation of Dynamic Systems With Stochastic Petri Nets (Tutorial). In: *Proc. 7th Int. Conf. on Performance Evaluation Methodologies and Tools (VALUETOOLS 2013)*. Torino, Italy, Dezember 2013

[21] ZIMMERMANN, Armin ; MACIEL, Paulo: Importance Function Derivation for RESTART Simulations of Petri Nets. In: *9th Int. Workshop on Rare Event Simulation (RESIM 2012)*. Trondheim, Norway, Juni 2012, S. 8–15