

Anwendungsspezifische Optimierung von Methoden der dynamischen probabilistischen Sicherheitsanalyse

Yubo Liu

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. habil. A. W. Koch

Prüfer der Dissertation: 1. Univ.-Prof. Dr. R. Macián-Juan

2. Univ.-Prof. Dr. phil., Dr.-Ing. E. h. A. Birkhofer (i. R.)

Die Dissertation wurde am 06.12.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 06.06.2008 angenommen.

Zusammenfassung

Probabilistische Sicherheitsanalyse (PSA) ist ein systematisches Verfahren zur Bewertung der Systemsicherheit und stützt sich heute meistens auf die Fehlerbaum- und Ereignisablaufanalyse. Während die Systemverfügbarkeit mit der Fehlerbaumanalyse berechnet wird, werden die Unfallabläufe mit der Ereignisablaufanalyse modelliert und die Eintrittswahrscheinlichkeit des Schadenszustandes ermittelt. Da sich die Fehlerbaumanalyse und Ereignisablaufanalyse nicht direkt an die Systemdynamik z.B. Prozessgröße koppeln, können die Fehlerbaumanalyse und Ereignisablaufanalyse die Wechselwirkungen zwischen den zufälligen Komponentenausfällen und der Systemdynamik nicht berücksichtigen. Eine genauere probabilistische Analyse eines dynamischen Systems erfordert daher die Anwendung der Methoden der dynamischen PSA.

Die vorliegende Arbeit beschäftigt sich mit dem Thema Optimierung von Methoden der dynamischen PSA. Nach der Untersuchung der konventionellen und dynamischen PSA-Methoden wird ein Optimierungsverfahren - Monte Carlo Simulation mit iterativer Anpassung der Stichprobenfunktionen – für die dynamische PSA vorgestellt. Mittels dieses Optimierungsverfahrens können die Unfallabläufe realitätsnah modelliert und analysiert werden. Durch die gezielte Anpassung der Stichprobenfunktionen wird der Rechenaufwand der Monte Carlo Simulation stark reduziert und die Genauigkeit der Schätzung verbessert. Das Potential dieses Optimierungsverfahrens wird für eine DWR-Transiente untersucht. Zur Implementierung wird der Systemcode ATHLET und der statistische Code SUSANA verwendet.

Abstract

Probabilistic Safety Analysis (PSA) is a systematic approach to evaluate system safety. It is based on Fault Trees Analysis (FTA) which is used for evaluation of the system availability and Event Trees Analysis (ETA) which is implemented for the modeling of accident scenarios and the calculation of the probability of the critical state. Since the Fault Tree Analysis and Event Tree Analysis are not directly linked with the system dynamics e.g. process variable, the interaction between the random system component failure and system dynamics can not be considered. To resolve this limitation of the traditional PSA methods, dynamic methods are required.

This thesis deals with the optimization of dynamic PSA methods. After investigating the conventional and dynamic PSA methods, an optimization method – Monte Carlo Simulation (MCS) using Adaptive Importance Sampling (AIS) - is presented. Using this optimization method the accident scenarios can be modeled and analyzed more precisely. By adjusting the sampling density functions, the computational time can be reduced and the accuracy of the estimate can be improved. The applicability of this optimization method in the field of Reactor Safety has been demonstrated with a transient in a pressurized water reactor. The optimization method is implemented using the advanced thermal hydraulic code ATHLET and the statistical code SUSAN.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Reaktordynamik und -sicherheit der Technischen Universität München.

Großer Dank gilt Herrn Professor Adolf Birkhofer, der mich bei meiner wissenschaftlichen Arbeit am Lehrstuhl für Reaktordynamik und -sicherheit sehr unterstützt hat. Ich möchte auch Herrn Professor Rafael Macian-Juan für das entgegengebrachte Interesse an dieser Arbeit danken.

Ebenfalls möchte ich mich herzlich bei Herrn Schäfer für die vielfältigen Anregungen, bei Frau Martina Kloos, Herrn Joerg Peschke und Herrn Bernhard Krzykacz-Hausmann für die Diskussionen bei den mathematischen Fragenstellungen. Bei Herrn Papukchiv für die Fragen zur Anlagentechnik möchte ich mich ebenfalls bedanken. Ihre Hinweise und Ratschläge in den Diskussionen waren mir eine wertvolle Hilfe.

Weiterhin bedanke ich mich bei allen Kollegen am Institut – Markus Baier, Sebastian Hanig, Alexander Kerner sowie bei unseren Sekretärinnen für die gute Zusammenarbeit und das kollegiale Arbeitsklima.

Schließlich möchte ich mich bei meiner Frau Xingying Pang und meinen Eltern sehr bedanken, die mich mit Liebe und Verständnis während meiner Dissertation unterstützt haben.

Abkürzungen

ADS	Accident Dynamic Simulation
AIS	Adaptive Importance Sampling
ATHLET	Analyse der Thermohydraulik, Lecks und Transienten
DETAM	Dynamic Event Tree Analysis Methodology
DYLAM	Dynamic Logical Analytical Methods
DSA	Deterministische Sicherheitsanalyse
DWR	Druckwasserreaktor
ETA	Event Trees Analysis
FTA	Fault Trees Analysis
FORM	First Order Reliability Method
MCDET	Monte Carlo Dynamic Event Tree
MCS	Monte Carlo Simulation
MCSIAS	Monte Carlo Simulation mit iterative Anpassung der Stichprobenfunktionen
PSA	Probabilistische Sicherheitsanalyse
SORM	Second Order Reliability Method
SUSA	Software-System für Unsicherheits- und Sensitivitätsanalysen

Inhaltverzeichnis

1	Einleitung	11
1.1	Problemstellung	11
1.2	Zielsetzung	11
1.3	Aufbau der Arbeit	11
2	Probabilistische Sicherheitsanalyse (PSA)	15
2.1	Historischer Hintergrund der PSA	15
2.2	Herkömmliche PSA-Methoden	16
2.2.1	Fehlerbaumanalyse	17
2.2.2	Ereignisablaufanalyse	18
2.2.3	Zusammenfassung	19
2.3	Notwendigkeit der dynamischen PSA	20
3	Dynamische Wechselwirkungen	22
3.1	Arten der dynamischen Wechselwirkungen	22
4	Dynamische probabilistische Sicherheitsanalyse (DPSA)	24
4.1	Historische Entwicklung der dynamischen PSA	24
4.2	Definition der dynamischen PSA	24
4.3	Dynamische PSA-Methoden	25
4.3.1	Theorie der dynamischen PSA	26
4.3.2	Erweiterung der Fehlerbaumanalyse und Ereignisablaufanalyse	27
4.3.2.1	Phased Mission Methoden	28
4.3.3	Zustandsübergangsmethoden	29
4.3.3.1	Explizite Zustandsübergangsmethoden	29
4.3.3.1.1	Markov-Modell	29
4.3.3.2	Implizite Zustandsübergangsmethoden	31
4.3.3.2.1	Diskrete dynamische Ereignisablaufdiagramme	31
4.3.3.2.2	Monte Carlo Simulation (MCS)	32
4.4	Graphische Werkzeuge	33

4.4.1	Dynamische Fehlerbaumanalyse	33
4.4.2	GO-FLOW Diagramme	34
4.4.3	Petrinetze	35
4.4.4	Event Sequence Diagramme	35
4.4.5	Zusammenfassung	36
5	Aufbau der MCS und statistische Auswertung der Stichproben	40
5.1	Wichtige Definitionen	40
5.1.1	Zufallsvariablen	40
5.1.2	Verteilungsfunktion der Zufallsvariablen	40
5.1.3	Zufallsvektoren	42
5.2	Zufallszahlengenerator	43
5.3	Inversionsmethode	44
5.4	Schätzwert des Erwartungswertes	44
5.5	Konfidenzintervall des geschätzten Mittelwertes	45
5.6	Zusammenfassung	45
6	Modellierung des „Hold-Up-Tank“-Systems und Analyse mit der MCS	47
6.1	Beschreibung des „Hold-Up-Tank“-Systems	47
6.1.1	Systemparameter und Anfangsbedingungen	49
6.1.2	Dynamik des „Hold-Up-Tank“-Systems	50
6.1.3	Regelsystem	50
6.2	Analyse des „Hold-Up-Tank“-Systems	51
6.2.1	Zufällige Ausfälle der Systemkomponenten und des Regelsystems	51
6.2.1.1	Zufällige Komponentenausfälle	51
6.2.1.2	Vom Wasserfüllstand abhängige Komponentenausfälle	52
6.2.2	Analyse des „Hold-Up-Tank“-Systems mit herkömmlichem Ereignisablaufdiagramm	52
6.2.3	Analyse des „Hold-Up-Tank“-Systems mit MCS	54
6.2.3.1	Realisierung	54
6.2.3.2	Ergebnisse aus der MCS	55
6.2.3.3	Quantifizierung der stochastischen Größen	58

6.2.4	Quantifizierung der epistemischen Größen	62
6.2.4.1	Quantifizierung der epistemischen Größen	62
6.3	Zusammenfassung	68
7	Rechenzeitbeschleunigende Methoden der MCS	69
7.1	First Order Reliability Method (FORM)	69
7.1.1	Anwendbarkeit	70
7.1.2	Schätzungsgenauigkeit	70
7.1.3	Verbesserungsmöglichkeit der FORM	71
7.1.4	Zusammenfassung	71
7.2	Monte Carlo Dynamic Event Tree (MCDET)	72
7.2.1	Grundlagen des MCDET Verfahrens	72
7.2.2	Vorteile und Nachteile des MCDET	74
7.2.3	Zusammenfassung	74
7.3	Weg zur weiteren Optimierung der MCS	75
7.4	MCS	76
7.4.1	MCS mit gewichteter Stichprobenfunktion (MCSGS)	78
7.4.2	MCS mit iterativer Anpassung der Stichprobenfunktion (MCSIAS)	79
8	Modellierung der DWR-Transiente und Analyse mit der MCS und MCSIAS	81
8.1	Beschreibung der Transiente	81
8.2	Stochastische Größen	81
8.3	Systemcode ATHLET	82
8.4	Statistischer Code SUSAS	83
8.5	Anwendung der MCS auf die DWR-Transiente	83
8.6	Ergebnisse aus der MCS	84
8.7	Anwendung der MCSIAS auf die DWR-Transiente	86
8.7.1	Definition der Schätzfunktionen	87
8.8	Realisierung des MCSIAS Systems	88
8.9	Ergebnisse aus der MCSIAS	89
8.10	Anbindung der MCSIAS ins herkömmlichen Ereignisablaufdiagramm	93
8.11	Schätzwertverbesserung mit der MCSIAS	95

9	Zusammenfassung	97
10	Literaturverzeichnis	101
	Abbildungsverzeichnis	104
	Tabellenverzeichnis	107
	Anhang A - Analyse des „Hold-Up-Tank“-Systems mit der Monte Carlo Simulation	108
	Anhang B - Modellierung der zufälligen Komponentenausfälle im „Hold-Up-Tank“-Systems	121
	Anhang C - Modellierung des zufälligen Ausfalls des Regelungssystems im „Hold-Up-Tank“-System	122

1 Einleitung

1.1 Problemstellung

Die Unfallabläufe in den Kernkraftwerken entwickeln sich innerhalb des zeitlichen Verlaufs, in dem sich die zufälligen Komponentenausfälle und die Systemdynamik gegenseitig beeinflussen. Zur Analyse der Unfallabläufe in Kernkraftwerken wird die probabilistische Sicherheitsanalyse (PSA) verwendet. Die Probabilistische Sicherheitsanalyse ist ein systematisches Verfahren zur Bewertung der Systemsicherheit und stützt sich heute meistens auf die Fehlerbaum- und Ereignisablaufanalyse. Während die Systemverfügbarkeit mit der Fehlerbaumanalyse berechnet wird, werden die Unfallabläufe mit der Ereignisablaufanalyse modelliert und wird die Eintrittswahrscheinlichkeit des Schadenszustandes ermittelt. Da die Fehlerbaum- und Ereignisablaufanalyse nicht direkt an die Systemdynamik z.B. Prozessgröße gekoppelt sind, können die Fehlerbaum- und Ereignisablaufanalyse die Wechselwirkungen zwischen den zufälligen Komponentenausfällen und Systemdynamik nicht realitätsnah modellieren und genau analysieren.

1.2 Zielsetzung

Das übergeordnete Ziel der vorliegenden Arbeit ist die Entwicklung eines Optimierungsverfahrens der dynamischen PSA-Methoden. Dieses Optimierungsverfahren soll zum einen die Kopplung zwischen den zufälligen Komponentenausfällen und der Systemdynamik verbessern und zum anderen die Einschränkung der dynamischen PSA-Methoden beseitigen. Der Schwerpunkt wird auf die Untersuchung der dynamischen PSA-Methoden, die Erstellung des Optimierungskonzeptes, Realisierung des Optimierungskonzeptes und die Anwendung auf eine DWR-Transiente gelegt.

1.3 Aufbau der Arbeit

Um die oben genannten Ziele umzusetzen, wurden folgende Untersuchungen in dieser Arbeit durchgeführt:

- **Untersuchung der PSA-Methoden**

Nach der Erläuterung des historischen Hintergrundes der PSA in Kapitel 2 werden die herkömmlichen PSA-Methoden - Fehlerbaumanalyse und Ereignisablaufanalyse - genau untersucht. Die Ziele dieser Untersuchung sind, die Einschränkungen der Fehlerbaum- und Ereignisablaufanalyse zu identifizieren und die Fragen zu beantworten, warum die dynamische PSA nötig ist.

- **Untersuchung der möglichen dynamischen Wechselwirkung**

Da die Haupteinschränkung der Fehlerbaum- und Ereignisablaufanalyse die Berücksichtigung der dynamischen Wechselwirkungen ist, werden die möglichen dynamischen Wechselwirkungen, die für die PSA von Bedeutung sind, in Kapitel 3 festgestellt.

- **Untersuchung der dynamischen PSA-Methoden**

In Kapitel 4 wird einen Überblick über die publizierten dynamischen PSA-Methoden gegeben. Ihre Nachteile, Vorteile und die Anwendbarkeit werden untersucht und verglichen. Die Untersuchungen zeigen, dass die Monte Carlo Simulation (MCS) für die probabilistische Bewertung der dynamischen Systeme sehr gut geeignet ist.

- **Methodischer Aufbau der MCS und statistische Bewertung der Stichproben**

In Kapitel 5 werden die mathematischen Grundlagen zum Aufbau der MCS und die statistische Auswertung der Stichproben erläutert. Die Ergebnisse der Untersuchung bilden die mathematische Basis für die Analyse des dynamischen Systems mit der MCS und die weiteren Optimierungsmaßnahmen der MCS.

- **Modellierung eines dynamischen Systems und Analyse mit der MCS**

Ein dynamisches „Hold-Up-Tank“-System wurde in Kapitel 6 modelliert. Die Ermittlung der Eintrittswahrscheinlichkeit der Schadenszustände „DRYOUT“ und „OVERFLOW“ erfolgte mit der MCS. Die Ziele dieser Untersuchung sind zu demonstrieren, wie die zufälligen Ausfälle und die Prozessgröße sich gegenseitig beeinflussen und wie weit die MCS die dynamischen Wechselwirkungen zwischen den

zufälligen Komponentenausfällen und der Systemdynamik berücksichtigen.

- **Untersuchung der rechenzeitbeschleunigenden Methoden der MCS**

Da die größte Einschränkung der MCS ist, dass die notwendige Anzahl der Simulationen bei gegebener Genauigkeitsanforderung mit der Zuverlässigkeit des untersuchten Systems stark ansteigt, werden in Kapitel 7 die rechenzeitbeschleunigenden Methoden der MCS untersucht. Die Ergebnisse der Untersuchung zeigen, dass die Anwendbarkeit der entwickelten rechenzeitbeschleunigenden Methoden eingeschränkt sind. Ein Konzept zur weiteren Optimierung der MCS wurde erstellt. Anhand dieses Konzeptes wurde eine rechenzeitbeschleunigende Methode - die Monte Carlo Simulation mit iterativer Anpassung der Stichprobenfunktionen (MCSIAS) - für die dynamische PSA vorgestellt.

- **Modellierung der DWR-Transiente und Analyse mit der MCSIAS**

Das Potential der MCSIAS wurde für eine Transiente nach der Turbinenschnellabschaltung mit blockierter Umleitstation in Kapitel 8 untersucht. Um die MCSIAS automatisch durchzuführen, wurde ein MCSIAS-System implementiert, in dem die Transiente mit dem Systemcode ATHLET modelliert und die Zufallszahlen mit dem statistischen Code SUSA generiert wurden. Die Ergebnisse zeigen, dass die Rechenzeit bei der Anwendung der MCSIAS im Vergleich zur herkömmlichen MCS bei gleicher Genauigkeit, um weit mehr als eine Größenordnung sinkt. Außerdem kann die MCSIAS leicht ins klassische Ereignisablaufdiagramm angebunden werden, wo eine genaue Analyse durchgeführt werden muss. Die MCSIAS kann auch eingesetzt werden, um den Schätzwert aus der MCS zu verbessern.

- **Vergleich mit anderen rechenzeitoptimierten Methoden der MCS**

Die MCSIAS wurde mit anderen rechenzeitoptimierenden Methoden der MCS verglichen. Der Hauptvorteil der MCSIAS ist, dass die Kenntnisse aus den Simulationsergebnissen und der Expertenschätzung genutzt werden kann, um die Stichprobenfunktion gezielt anzupassen. Dadurch können die Unfallabläufe mit einer kleinen Eintrittswahrscheinlichkeit schnell entdeckt und der Rechenaufwand stark reduziert werden. Außerdem kann auch die Größe der Schätzung mit dem gewöhnlichen statistischen Verfahren 95% - Vertrauensintervall quantifiziert werden. In

Kapitel 9 werden auch die wesentlichen erzielten Ergebnisse zusammengefasst.

2 Probabilistische Sicherheitsanalyse (PSA)

2.1 Historischer Hintergrund der PSA

Die Probabilistische Sicherheitsanalyse (PSA) ist ein systematisches Verfahren zur Bewertung der Systemsicherheit. Seitdem die erste PSA-Studie im Jahre 1975 für das amerikanische Kernkraftwerk durchgeführt wurde, kam die PSA schon in verschiedenen Gebieten z.B. Kernkraftwerk, Luftfahrt und Raumfahrt erfolgreich zum Einsatz. Die Ergebnisse der PSA kann für die Unterstützung der Entscheidungsfindung in behördlicher Aufsicht verwendet werden.

In den Kernkraftwerken wurde ursprünglich die Anlagensicherheit mit der deterministischen Sicherheitsanalyse (DSA) bewertet. Mit der DSA wird untersucht, ob die festgelegte Sicherheitsanforderung unter definierten Voraussetzungen erfüllt wird. Die PSA ergänzt die DSA. Mit der PSA werden die wichtigen Informationen z.B. Komponentenzuverlässigkeit, Systemzuverlässigkeit und die menschliche Handlung zu einer Gesamtbetrachtung der Anlagensicherheit zusammengeführt und die Anlagensicherheit wird quantitativ ermittelt. Da die PSA, die bei der DSA nicht gewonnenen Kenntnisse über die Anlagensicherheit liefern kann, wird die PSA als wichtiges Instrument der Sicherheitsbewertung benutzt.

Im Kernkraftwerk wird die PSA üblicherweise in drei Stufen geteilt.

PSA Stufe 1

In der PSA Stufe 1 werden die Unfallabläufe vom auslösenden Ereignis bis zum Kernschmelzen untersucht und die Eintrittswahrscheinlichkeit des Kernschmelzens ermittelt. Die Unfallabläufe können durch die Sicherheitssysteme in den Kernkraftwerken entweder beherrscht oder nicht beherrscht werden. Die Verfügbarkeit der Sicherheitssysteme wird mit der Fehlerbaumanalyse berechnet. Mit der Ereignisablaufanalyse können die Unfallabläufe modelliert werden und die Eintrittswahrscheinlichkeit des Schadenszustandes z.B. Kernschmelzen ermittelt werden. Ein typischer Verlauf der PSA Stufe 1 in der Abbildung 2.1 dargestellt.

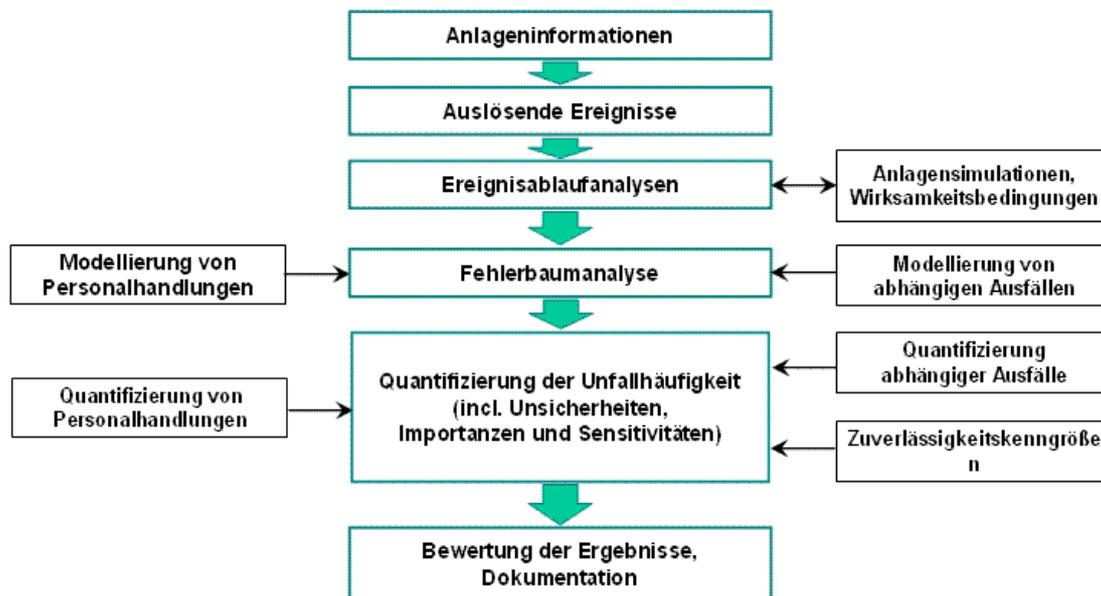


Abbildung 2.1: Ablauf der PSA Stufe 1

PSA Stufe 2

In der PSA Stufe 2 werden die Unfallabläufe nach dem Kernschmelzen untersucht und die Versagenswahrscheinlichkeit des Sicherheitseinschlusses und das Ausmaß der freigesetzten radioaktiven Stoffe in die Umgebung ermittelt. Nach dem Kernschmelzen ist die Anlage weit vom normalen Zustand entfernt und die Unfallabläufe hängen von der Entwicklung der physikalischen-chemischen Prozesse (z.B. Dampfexplosion) ab. Die Analyse der Unfallabläufe kann sich nur auf die rechnerische Simulation der Unfallabläufe abstützen.

PSA Stufe 3

In der Stufe 3 der PSA werden das Ausmaß und die Eintrittswahrscheinlichkeit der Schadensfolgen in der Umgebung ermittelt.

Da der Stand der Technik der PSA die Fehlerbaum- und Ereignisablaufanalyse ist, werden sie in nächsten Abschnitt genau beschrieben.

2.2 Herkömmliche PSA-Methoden

Die Methodik der Stufe 1 der PSA unterscheidet sich in deduktive Verfahren und induktive Verfahren. Ein deduktives Verfahren startet mit der Annahme eines

Systemzustands und versucht diesen auf die Ursachen zurückzuführen, die zu diesem Zustand führen können. Ein induktives Verfahren startet mit der Annahme einer Ursache und untersucht alle möglichen Auswirkungen auf das Systemverhalten.

2.2.1 Fehlerbaumanalyse

Die Fehlerbaumanalyse ist ein deduktives Verfahren. Mit der Fehlerbaumanalyse wird die Verfügbarkeit der Sicherheitssysteme ermittelt.

Der Fehlerbaum stellt in graphischer Form den logischen Zusammenhang zwischen den Ausfällen der Komponenten und dem unerwünschten Ereignis z.B. Ausfall der Sicherheitssysteme dar. Die Spitze des Fehlerbaums bildet das unerwünschte Ereignis. Mit Hilfe einer logischen Verknüpfung wird dieses Ereignis auf die Ausfälle der Teilsysteme und schrittweise auf die Ausfälle der Elemente, die keine gemeinsamen Abhängigkeiten besitzen, zurückgeführt.

Als Beispiel zeigt die Abbildung 2.2 einen Fehlerbaum für ein unerwünschtes Ereignis, dem „Überlauf“. In diesem Fehlerbaum werden alle zum Ereignis „Überlauf“ führenden Ursachen durch logische Verknüpfungen dargestellt. Eine mögliche Ursache für das Ereignis „Überlauf“ ist z.B. ein nicht geöffnetes Ventil VX und ein defekter Überlaufsensor.

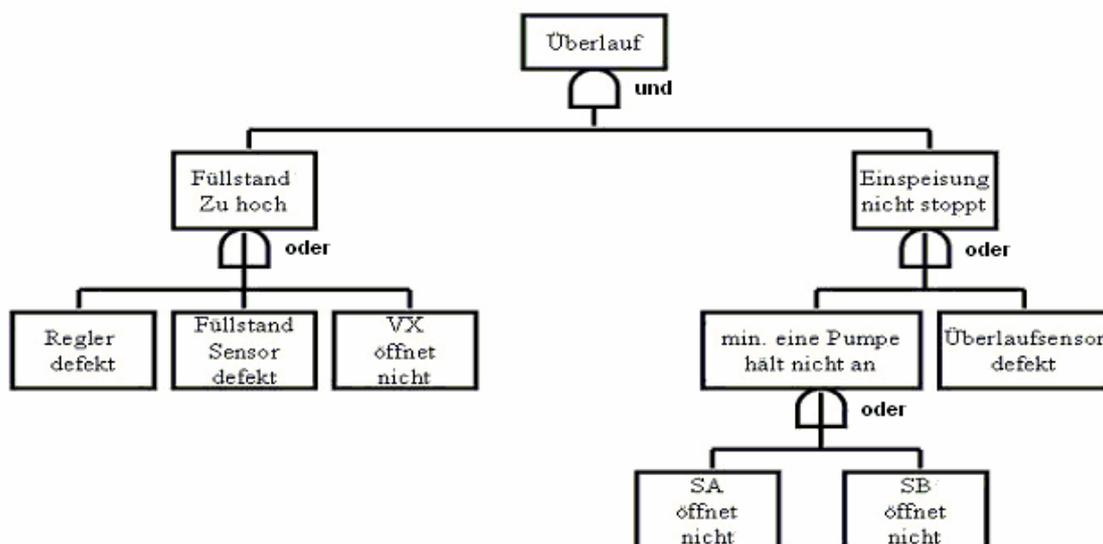


Abbildung 2.2: Fehlerbaum-Beispiel

Um die Eintrittswahrscheinlichkeit des unerwünschten Ereignisses zu ermitteln, müssen zuerst die Minimalschnitte definiert werden. Die Minimalschnitte bestehen aus der Kombination der Komponenten, die gerade zum Ausfall eines Systems führten. Die Eintrittswahrscheinlichkeit des unerwünschten Ereignisses lässt sich aus der Aufsummierung der Wahrscheinlichkeit aller Minimalschnitte approximieren.

Die Fehlerbaumanalyse allein ist bei der Modellierung komplexer Systeme unübersichtlich. Durch die Kombination von der Fehlerbaum- und Ereignisablaufanalyse lassen sich extrem komplexe Systeme auf systematische Weise übersichtlich darstellen.

2.2.2 Ereignisablaufanalyse

Die Ereignisablaufanalyse ist ein induktives Verfahren. Mit der Ereignisablaufanalyse werden die Unfallabläufe modelliert und die Eintrittswahrscheinlichkeit des Schadenszustandes z.B. Kernschmelzen ermittelt.

Bei der Erstellung der Ereignisabläufe müssen zuerst die auslösenden Ereignisse festgelegt werden. Auslösende Ereignisse sind solche Ereignisse, die zu einer Anforderung der Sicherheitssysteme führen. Ausgehend vom auslösenden Ereignis wird ein Ereignisablaufdiagramm erstellt, wobei die zeitliche Reihenfolge der Sicherheitssysteme durch die Anregekriterien der Anlage festgelegt werden. Grundsätzlich wird nur zwischen der Verfügbarkeit und Nichtverfügbarkeit der Sicherheitssysteme unterschieden. Alle Unfallabläufe, die durch die Sicherheitssysteme entweder beherrscht oder nicht beherrscht werden, sind im Ereignisablaufdiagramm zusammengefasst. Im Kernkraftwerk sind vor allem solche Unfallabläufe von Bedeutung, z.B. die zum Kernschmelzen, zum Versagen des Sicherheitseinschlusses und zur Freisetzung von radioaktiven Stoffen in die Umgebung führen.

Als Beispiel zeigt die Abbildung 2.3 ein Ereignisablaufdiagramm für das auslösende Ereignis „Füllstand zu hoch“. Ausgehend vom auslösenden Ereignis „Füllstand zu hoch“ werden die möglichen Unfallabläufe dargestellt. Der Schadenszustand „Überlauf“ wird z.B. bei nicht funktionierendem Überlaufsensor und nicht anhaltenden Einspeisungspumpen erreicht.

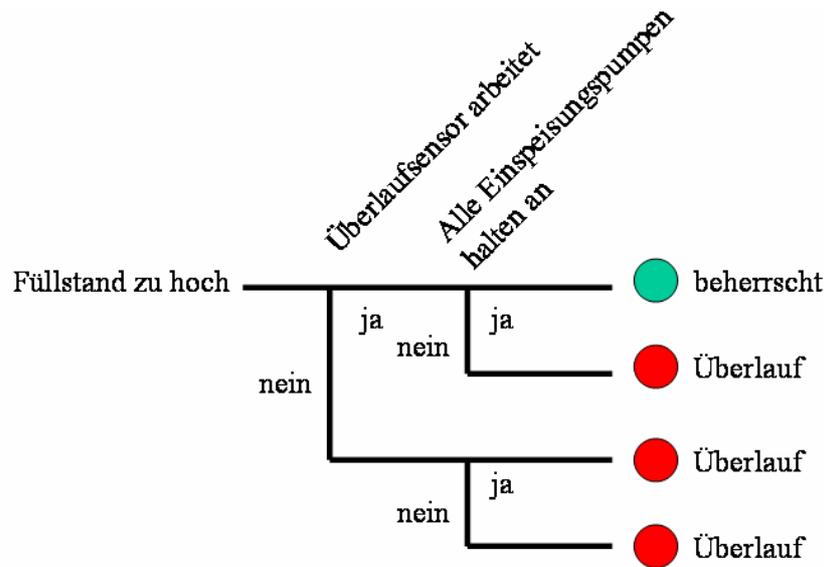


Abbildung 2.3: Ereignisablaufdiagramm-Beispiel

Die Übergangswahrscheinlichkeit eines Unfallablaufes ist gleich dem Produkt aus der Verzweigungswahrscheinlichkeit entlang des Unfallablaufes. Die Übergangswahrscheinlichkeit vom auslösenden Ereignis zu einem bestimmten Schadenszustand ist gleich der Aufsummierung der Übergangswahrscheinlichkeiten der Unfallabläufe, die den gleichen Schadenszustand haben. Die Eintrittswahrscheinlichkeit des Schadenszustandes ist gleich dem Produkt der Eintrittswahrscheinlichkeit des auslösenden Ereignisses, mit der Übergangswahrscheinlichkeit vom auslösenden Ereignis zu einem bestimmten Schadenszustand.

2.2.3 Zusammenfassung

Die herkömmliche PSA stützt sich heute meistens auf die Fehlerbaum- und Ereignisablaufanalyse. Durch die Kombination der Fehlerbaum- und Ereignisablaufanalyse können die komplexen Systeme übersichtlich modelliert und analysiert werden.

Die Fehlerbaum- und Ereignisablaufanalyse gehören zu den statischen Verfahren, in dem die Zeit als Einflussfaktor nur sehr grob berücksichtigt wird. Im Fehlerbaum und Ereignisablaufdiagramm sind die Komponentenverfügbarkeit und die Systemverfügbarkeit zeitlich konstant. Das System wird im Fehlerbaum zu einem bestimmten Zeitpunkt modelliert und analysiert. Die Unfallabläufe werden Im

Ereignisablaufdiagramm nach der vorgegebenen Ereignisfolge, ohne Angabe von Zeitpunkten und zeitlichen Abständen modelliert und analysiert. Die Fehlerbaum- und Ereignisablaufanalysen können in solchen Situationen eingesetzt werden,

- wenn die Änderung der Ereignisabfolge keinen Einfluss auf den Eintritt des Schadenszustandes hat,
- wenn die Änderung der Eintrittszeit der Ereignisse keinen Einfluss auf den Eintritt des Schadenszustandes hat,

Durch die vereinfachte Modellierung der Unfallabläufe mit Fehlerbaum und Ereignisablaufdiagramm z.B. grobe Berücksichtigung der Zeit kann der Rechenaufwand der Fehlerbaum- und Ereignisablaufanalyse zwar stark reduziert werden, es besteht auch die Gefahr bei der Anwendung der Fehlerbaum- und Ereignisablaufanalyse [GRS-A-3269,2005], dass

- viele Zusammenhänge der Unfallabläufe unbekannt und damit unberücksichtigt bleiben,
- wichtige Unfallabläufe, die sich gerade aus den Details in der Zeit und in der Ereignisfolge ergeben, nicht erfasst werden,
- unrealistische Unfallabläufe aufgrund der vorgegebenen Ereignisfolge modelliert werden.

2.3 Notwendigkeit der dynamischen PSA

In den Kernkraftwerken entwickeln sich die Unfallabläufe im zeitlichen Verlauf, wobei die zufälligen Komponentenausfälle die Systemdynamik z.B. die Entwicklung der Prozessvariablen und die menschlichen Handlungen beeinflussen. Umgekehrt können auch die Systemdynamik die zufälligen Komponentenausfälle beeinflussen. Als Beispiel betrachten wir ein System, in dem eine Komponente zuerst ausfällt. Der Ausfall der Komponente kann einen Temperaturanstieg verursachen. Wenn die Temperatur den kritischen Wert übersteigt, kann die Temperatur das Ausfallverhalten der Komponenten so beeinflussen, so dass die Ausfallwahrscheinlichkeit der anderen Komponenten im System ansteigen.

Eine andere Vereinfachung der Fehlerbaum- und Ereignisablaufanalyse neben der Berücksichtigung der Zeit ist, dass die Fehlerbaum- und Ereignisablaufanalyse nicht direkt an die Systemdynamik gekoppelt ist. Die Folgen bei der Anwendung der Fehlerbaum- und Ereignisablaufanalyse sind, dass der Einfluss der zufälligen Komponentenausfälle auf die Systemdynamik nicht erfasst und der Einfluss der Systemdynamik auf die zufälligen Komponentenausfälle nicht berücksichtigt werden kann.

Zur Beseitigung der Einschränkungen der Fehlerbaum- und Ereignisablaufanalyse wird die dynamischen PSA eingeführt. Die dynamische PSA berücksichtigt die Eintrittszeitpunkte der Ereignisse, die Zeitdauer der Ereignisse, die Prozessvariable und die dynamischen Wechselwirkungen zwischen den zufälligen Komponentenausfällen und der Systemdynamik. So ermöglicht die dynamische PSA eine realistische und genaue Modellierung der Unfallabläufe.

Im folgenden Kapitel werden die dynamischen Wechselwirkungen, die für die PSA von Bedeutung sind identifiziert.

3 Dynamische Wechselwirkungen

Die Untersuchung im Kapitel 2 zeigt, dass die größte Einschränkung der Fehlerbaum- und Ereignisablaufanalyse die Berücksichtigung der Wechselwirkungen zwischen den zufälligen Komponentenausfällen und der Systemdynamik ist. Es werden die dynamischen Wechselwirkungen, die für die PSA von Bedeutung sind, festgestellt.

3.1 Arten der dynamischen Wechselwirkungen

Folgende dynamische Wechselwirkungen, die in der Fehlerbaum- und Ereignisablaufanalyse nicht oder nicht genau berücksichtigt werden, können identifiziert werden.

Komponentenverfügbarkeit und Systemverfügbarkeit

Im Fehlerbaum und Ereignisablaufdiagramm sind die Komponentenverfügbarkeit und die Systemverfügbarkeit zeitlich konstant. Zur Berechnung der Komponentenverfügbarkeit wird die konstante Ausfallrate λ verwendet. Die Annahme der konstanten Ausfallrate entspricht nicht der Realität. Die Ausfallrate λ der technischen Komponenten wird abhängig von der Lebensdauer in einer Badewannenkurve in der Abbildung 3.1 dargestellt. Die Ausfallrate λ ist zu Beginn und am Ende der Lebensdauer besonders hoch, während die Ausfallrate λ im Zentrum der Lebensdauer niedriger und recht konstant ist.

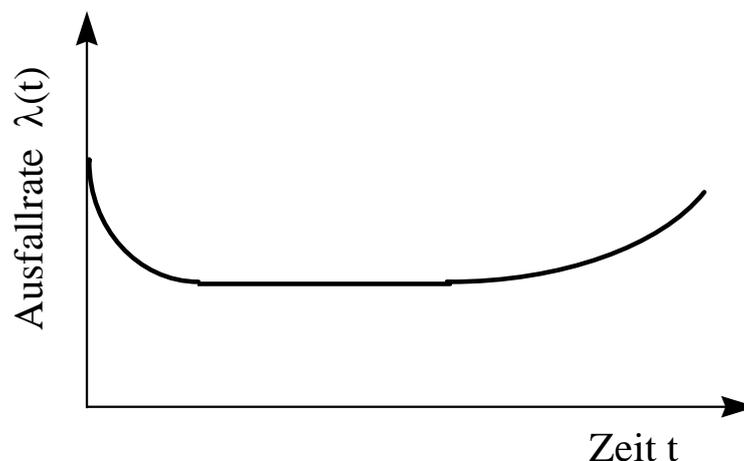


Abbildung 3.1: Badewannenkurve

Will man die Unfallabläufe genau modellieren und analysieren, ist es wichtig die Änderung der Ausfallrate der Komponenten während ihrer Lebensdauer zu berücksichtigen.

Zeitliche Änderung der logischen Struktur während des Unfallablaufes

Die logische Struktur des Systems kann sich in dem zeitlichen Verlauf des Unfalls ändern. Als Beispiel betrachten wir den Kühlmittelverlustunfall. Um die Nachwärme abzuführen, reichen am Anfang zwei Pumpen und später eine Pumpe aus.

Versagen der Menschen während des Unfalls

Der zeitliche Verlauf der Prozessgröße kann das Versagen der Menschen beeinflussen. Z.B. eine menschliche Handlung wird angefordert, wenn die Prozessgröße z.B. der Druck einen kritischen Wert überschreitet. Die menschliche Handlung kann nach der Anforderung entweder erfolgreich oder nicht erfolgreich durchgeführt werden.

Versagen von Schutzsystem während des Unfalls

Der zeitliche Verlauf der Prozessgröße kann das Versagen des Schutzsystems beeinflussen. Z.B. das Schutzsystem wird angefordert, wenn die Prozessgröße einen kritischen Wert überschreitet. Das Schutzsystem kann nach der Anforderung entweder erfolgreich oder nicht erfolgreich gestartet werden.

Einfluss der Vorgeschichte auf menschliche Handlung

Die menschliche Handlung kann auch von der Vorgeschichte abhängen, z.B. die Bedienmannschaft hält an einer falschen Annahme fest.

4 Dynamische probabilistische Sicherheitsanalyse (DPSA)

4.1 Historische Entwicklung der dynamischen PSA

Da die Fehlerbaum- und Ereignisablaufanalyse die dynamischen Abläufe nicht realitätsnah modellieren und genau analysieren, wurden verschiedenen Konzepte entwickelt. Das erste Konzept zur Analyse der dynamischen Abläufe wurde von Herrn Amendola und Herr Reina im Jahre 1981 erstellt. Basierend auf diesem Konzept wurden die DYLAM (Dynamic Event Logic Analytical Methodology) und ADS (Analyzer of Dynamic Systems) entwickelt. Im Jahre 1992 haben Herr Devooght und Herr Smidts die allgemeine Theorie der dynamischen PSA eingeführt. Auch im Jahre 1992 wurden die dynamischen PSA-Methoden, ihre Vorteile, ihre Nachteile und ihre Anwendungsgebiete in dem ersten Workshop für die dynamische Zuverlässigkeitsanalyse diskutiert ([Smidts, 1998], [Devooght, 1998], [IWDR-Discussion-1 1998], [IWDR-Discussion-2 1998], [IWDR-Discussion-1 1998]). Nach diesem Workshop wurde die dynamische PSA ein wichtiger Teil der PSA.

4.2 Definition der dynamischen PSA

Die dynamische PSA wird auch als dynamische Zuverlässigkeitsanalyse bezeichnet. Dies ist ein Verfahren zur probabilistische Bewertung der Mensch-Maschine-Software-Systeme, welche sich dynamisch verhalten. Unter dem Begriff "dynamisch" versteht man, dass das System sich dynamisch über die Zeit entwickelt, zufällig auftretende Ausfälle und Reparaturen sowie menschliche Handlungen beeinflussen die dynamische Entwicklung des Systems. Gleichzeitig verursacht die dynamische Entwicklung des Systems die Ausfälle, Reparaturen und menschliche Handlungen. Bei der Anwendung in Kernkraftwerken hat Herr Hofer „dynamisch“ so definiert, dass die Unfallabläufe ausgehend vom auslösenden Ereignis mit der Zeit durch die Wechselwirkungen zwischen der Systemdynamik und den zufälligen Komponentenausfällen von Mensch, Maschine, Prozess und Umgebung entwickeln. Die dynamische PSA kann die Unfallabläufe unter der Berücksichtigung der zeitlichen Wechselwirkungen realitätsnah analysieren ([Hofer & Kloos -1, 2001], [Hofer & Kloos -1, 2001]).

4.3 Dynamische PSA-Methoden

Das Ziel der dynamischen PSA ist die Unfallabläufe unter Berücksichtigung der zeitlichen Wechselwirkungen realitätsnah zu modellieren und genau zu analysieren. Um dieses Ziel zu erreichen, sollen die folgenden Informationen ständig bereitgestellt werden:

- **Zustände der Komponenten**

Ist das Ventil offen oder geschlossen? Ist die Pumpe eingeschaltet oder ausgeschaltet?

- **Zustände der Prozessgrößen**

Wie Hoch ist der Druck? Wie Hoch ist die Temperatur?

- **Zustand der Bedienmannschaft**

Wie lange braucht die Bedienmannschaft, um den Schadenszustand zu erkennen und die Sicherheitsmaßnahmen durchzuführen?

- **Zeit**

Wann und wie lange wird das Sicherheitssystem angefordert?

- **Vorgeschichte des Unfallablaufes**

Die menschliche Handlung kann auch von der Vorgeschichte des Szenarios abhängen.

Die verschiedenen dynamischen PSA-Methoden unterscheiden sich darin,

- welche oben genannten Informationen berücksichtigt werden,
- wie die oben genannten Informationen berücksichtigt werden,
- ob die dynamischen Abläufe intuitiv dargestellt werden können.

Nach oben genannten Kriterien wurden die dynamischen PSA-Methoden wie folgt klassifiziert.

- Theorie der dynamischen PSA
- Erweiterung der Fehlerbaumanalyse und Ereignisablaufanalyse
 - Phased Mission Methoden
- Graphische Werkzeuge
 - Dynamische Fehlerbaumanalyse
 - GO-FLOW Diagramme
 - Petrinetze
 - Event Sequence Diagramme
- Zustandsübergangsmethoden
 - Markov-Modelle
 - Diskrete dynamische Ereignisablaufdiagramme
 - Monte Carlo Simulation

4.3.1 Theorie der dynamischen PSA

Herr Devooght und Herr Smidts haben im Jahre 1992 die Theorie der dynamischen PSA eingeführt [Devooght & Smidts 1992]. Diese Theorie bietet eine sehr allgemeine mathematische Formulierung an. In der Formulierung werden diskrete Basiszustände z.B. die Zustände der Hardware, die Zustände der Software oder die Zustände der Betriebsmannschaft durch einen Index $i = 1, 2, 3, \dots, n$ betrachtet. Der Vektor der Prozessgrößen wird als x bezeichnet. Die zeitliche Entwicklung der Prozessgrößen x wird deterministisch durch ein System von Differentialgleichungen erster Ordnung modelliert.

$$\frac{d\bar{x}}{dt} = f_i(\bar{x}, t) \quad (4.1)$$

Dieses Differentialgleichungssystem ist ein Modell, so dass die physikalischen Zusammenhänge des Systems im Anlagenzustand i gelten. Der Zustand des Systems i kann aufgrund des Komponentenausfalls, Eingriffen des Regelsystems oder menschlichen Handlungen in einen anderen Zustand des Systems j wechseln. Die Übergangsrates $\lambda(i \rightarrow j | x)$ vom Zustand i zum Zustand j kann von den Prozessgrößen abhängen.

Das Ziel der Theorie der dynamischen PSA ist die Wahrscheinlichkeitsdichte $\pi(x, i, t)$ zu ermitteln, die von den Prozessgrößen, den Basiszuständen und der Zeit abhängen. Die Wahrscheinlichkeitsdichte $\pi(x, i, t)$ genügt der Chapman-Kologorov-Gleichung [Devooght & Smidts 1992] :

$$\frac{\partial \pi(\bar{x}, i, t)}{\partial t} + \text{div}[\bar{f}_i(\bar{x}) \cdot \pi(\bar{x}, i, t)] + \lambda_i(\bar{x}) \cdot \pi(\bar{x}, i, t) = \sum_{j \neq i} \lambda(j \rightarrow i | x) \cdot \pi(x, j, t) \quad (4.2)$$

mit $\lambda_i(x) = \sum_{j \neq i} \lambda(i \rightarrow j | x)$

Um die Theorie der dynamischen PSA für praktische Probleme anwenden zu können, muss sie vereinfacht werden. Für das vereinfachte System mit der Markov-Eigenschaft kann die Wahrscheinlichkeitsdichte $\pi(x, i, t)$ aus der Theorie der dynamischen PSA ermittelt werden. Für das komplexe System kann die Wahrscheinlichkeitsdichte $\pi(x, i, t)$ aus der Theorie der dynamischen PSA schwer ermittelt werden.

4.3.2 Erweiterung der Fehlerbaumanalyse und Ereignisablaufanalyse

Da die Fehlerbaum- und Ereignisablaufanalyse auf verschiedenen Gebieten schon erfolgreich eingesetzt und akzeptiert wurden, konnten die Fehlerbaum- und Ereignisablaufanalyse für die Analyse der dynamischen Abläufe erweitert werden.

4.3.2.1 Phased Mission Methoden

Da sich die logische Struktur der Systeme z.B. die Versagenswahrscheinlichkeit der Systemkomponenten und die Anzahl der erforderlichen Systeme zur Beherrschung des Unfalls im Verlauf des Unfalls ändern kann, versuchen die Phased Mission Methoden [Kohda & Nakagawa, 2006], den zeitlichen Unfallablauf in eine Zahl von Phasen mit einer bestimmten Dauer einzuteilen. In jeder Phase wird ein eigener Fehlerbaum erstellt. Der Vorteil der Phased Mission Methoden besteht darin, dass in jeder Phase die Ausfallwahrscheinlichkeit der Komponenten und die Anzahl der erforderlichen Systeme unterschiedlich sein können. Der Nachteil der Phased Mission Methoden liegt darin, dass die gesamte Struktur der Phased Mission Methoden, aufgrund der Erstellung des Fehlerbaums in jeder Phase sehr komplex und umfangreich werden kann. Deshalb bleibt die Anwendung der Phased Mission Methoden nur auf wenige Phasen beschränkt. In der Abbildung 4.1 wird ein Beispiel für die Phased Mission Methoden gezeigt, wobei der zeitliche Unfallablauf in 3 Phasen eingeteilt wird und ein eigener Fehlerbaum in jeder Phase erstellt wird.

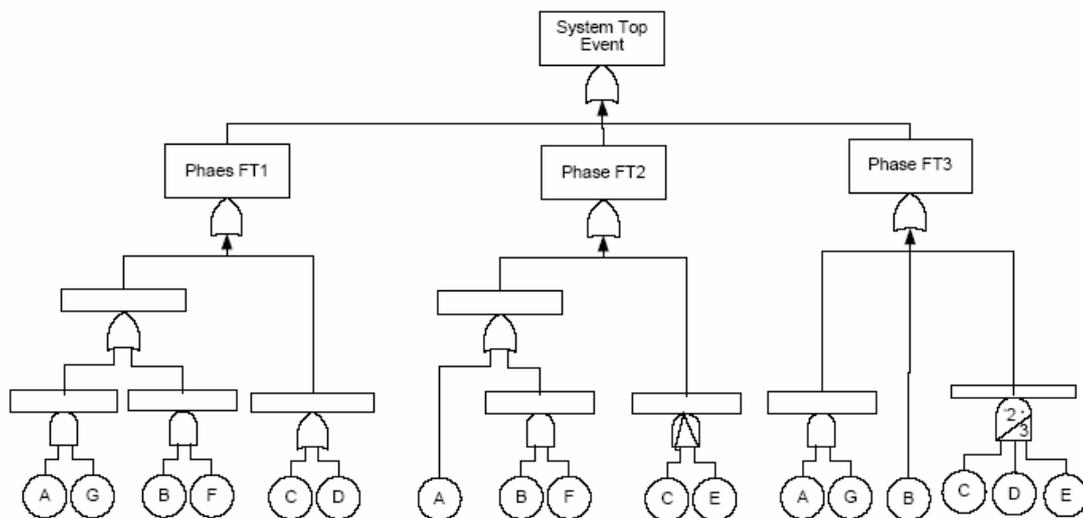


Abbildung 4.1: Beispiel - Phased Mission Methoden

4.3.3 Zustandsübergangsmethoden

Zur Analyse der dynamischen Unfallabläufe wurden auch die Verfahren entwickelt, die als Zustandsübergangsmethoden bezeichnet werden. Die Zustandsübergangsmethoden berücksichtigen die zeitlichen Wechselwirkungen zwischen den Komponentenausfällen und der Systemdynamik gut. Das Prinzip der Zustandsübergangsmethoden ist, dass die Prozessgröße x sich vom Anlagenzustand (x, i, t_0) deterministisch weiterentwickelt, bis zum Zeitpunkt $t_0 + t_n$ ein Zustandsübergang zum neuen Basiszustand j erfolgt. Das Ziel der Zustandsübergangsmethoden ist die Eintrittswahrscheinlichkeit eines bestimmten Zustandes z.B. des Schadenszustandes zu ermitteln. Die Zustandsübergangsmethoden unterscheidet sich darin,

- ob alle möglichen Zustände und die Zustandsübergänge vom Analytiker von Anfang an definiert werden müssen,
- welche Arten von den Zustandsübergängen berücksichtigt werden,
- mit welchem Lösungsalgorithmus die Eintrittswahrscheinlichkeit eines bestimmten Zustandes ermittelt wird.

4.3.3.1 Explizite Zustandsübergangsmethoden

Bei expliziter Zustandsübergangsmethode muss der Analytiker alle möglichen Zustände und Zustandsübergänge vorab explizit definieren. Das Markov-Modell ist explizite Zustandsübergangsmethode.

4.3.3.1.1 Markov-Modell

Das Markov-Modell ist ein klassisches Verfahren, um die zeitlichen Wechselwirkungen in dynamischen Systemen zu analysieren. In der Abbildung 4.2 wird ein Beispiel für das Markov-Modell gegeben.

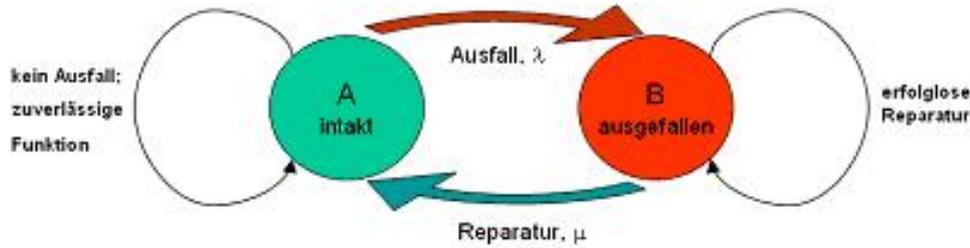


Abbildung 4.2: Beispiel - Markov

Um die zeitliche Entwicklung der Prozessgröße zu berücksichtigen, werden die Prozessgrößen so diskretisiert, dass jede Prozessgröße nur die diskreten Werte annehmen kann. Die Übergangswahrscheinlichkeit hängt bei dem Markov-Modell nur von dem jetzigen Zustand ab, es hängt aber nicht davon ab, wie und wann der momentane Zustand erreicht wird. Diese Eigenschaft des Markov-Modells wird als gedächtnislos bezeichnet. Bei der Markov-Kette ist der Übergang von einem Zustand zu einem anderen nur zu diskreten Zeitpunkten möglich. Die zeitabhängige Zustandswahrscheinlichkeit $P_{(t)}$ ergibt sich aus der folgenden Gleichung

$$P_{(t)} = M_p P(t - \Delta t) \quad (4.3)$$

wobei M_p die Zustandsübergangsmatrix ist. Die Zustandsübergangsmatrix, welche die Übergangswahrscheinlichkeiten beinhaltet, muss vom Analytiker vor der Ermittlung der Eintrittswahrscheinlichkeit des Schadenszustandes aufgestellt werden. Die Dimension der Übergangsmatrix bei n Prozessgrößen und m Komponenten wird aus folgender Gleichung

$$\prod_{i=1}^m k_i \times \prod_j^n l_j \quad (4.4)$$

berechnet, wobei k_i die Anzahl der möglichen Werte der Prozessgrößen i und l_j die Anzahl der möglichen Zustände der Komponenten j sind.

Nachdem die Zustandsübergangsmatrix vom Analytiker aufgestellt wurde, können die Eintrittswahrscheinlichkeiten der seltenen Szenarien genauso leicht, wie die Eintrittswahrscheinlichkeit der häufigen Unfallabläufe berechnet werden. Der Nachteil des Markov-Modells liegt darin, dass die Anzahl der Zustände exponentiell mit der

Anzahl der Komponenten wächst. Die Übergangsmatrix kann bei Systemen mit einer moderaten Größe schnell zu einer nicht mehr handhabbaren Größe anwachsen, so dass die Anwendung des Markov-Modells auf einfache Systeme mit wenigen Systemkomponenten und wenigen Prozessgrößen beschränkt bleibt.

4.3.3.2 Implizite Zustandsübergangsmethoden

Bei den impliziten Zustandsübergangsmethoden muss der Analytiker alle möglichen Zustände und Zustandsübergänge vorab nicht explizit definieren. Die Zustände und die Zustandsübergänge können durch, die vom Analytiker formulierten Regeln während der Simulation generiert wird. Zu den expliziten Zustandsübergangsmethoden gehören die diskreten dynamischen Ereignisablaufdiagramme und die Monte Carlo Simulation.

4.3.3.2.1 Diskrete dynamische Ereignisablaufdiagramme

Die diskreten dynamischen Ereignisablaufdiagramme gehen auf die Entwicklung der DYLAM (Dynamic Logical Analytical Methods) zurück [Cojazzi 1996]. Bei der DYLAM werden die Basiszustände durch ein „Branching set“, dass aus den diskreten Zuständen z.B. den Zuständen der Systemkomponenten und den Zuständen der menschlichen Handlungen definiert wird. Die DYLAM geht, wie das herkömmliche Ereignisablaufdiagramm von einem auslösenden Ereignis aus. Ausgehend vom auslösenden Ereignis wird die zeitliche Entwicklung der Prozessgröße mit Hilfe eines Systemcodes innerhalb eines Intervalls Δt berechnet. Am Ende des ersten Intervalls $(0, \Delta t)$ werden alle möglichen Verzweigungen identifiziert und die Verzweigungswahrscheinlichkeiten zu allen möglichen Nachfolgeständen analytisch berechnet. Die Simulation wird für jeden Zeitschritt neu berechnet bis der Endzustand erreicht wird. Wenn alle i_{\max} verschiedenen Basiszustände am Ende eines jeden Intervalls weiterverfolgt werden, ergeben sich nach k Zeitschritten i_{\max}^k verschiedene Unfallabläufe. Um ein Aufblähen des Ereignisablaufdiagramms zu vermeiden, schneidet man unwahrscheinliche Unfallabläufe gemäß der „Truncation Rules“ ab. Dies kann allerdings zur Vernachlässigung von den Unfallabläufen mit geringer Wahrscheinlichkeit aber zu schweren Folgen führen.

Der Vorteil der DYLAM besteht darin, dass alle möglichen Unfallabläufe systematisch

zu implementiert ist. Die zeitliche Entwicklung der Prozessgrößen und die menschlichen Handlungen können durch ein Systemmodell gut berücksichtigt werden. Da nur die zufällig ausgewählten Unfallabläufe simuliert werden, hängt der Rechenaufwand nur von der Anzahl der Simulationsläufe ab. Außerdem ist der Rechenaufwand von der Komplexität und der Dimension des Systems nicht abhängig, d.h. Probleme mit mehreren stochastischen Variablen benötigen genauso viel Zeit, wie Probleme mit wenigen stochastischen Variablen. Daher gibt es kaum eine realisierbare Alternative zur Monte Carlo Simulation für größere Systeme. [Marsequerra&Zio, 2001].

Die Einschränkung der Monte Carlo Simulation liegt in der Bewertung hoch zuverlässiger Systeme. Eine sehr hohe Anzahl von Simulationsläufen wird bei hoch zuverlässigen Systemen gefordert, um den Schadenszustand mit geringer Eintrittswahrscheinlichkeit mit einer gewissen Genauigkeit abzuschätzen.

4.4 Graphische Werkzeuge

Um die dynamischen Abläufe leicht zu analysieren und die relevanten dynamischen Abläufe zu identifizieren, wurden graphische Werkzeuge entwickelt. Die graphischen Werkzeuge können die dynamischen Abläufe intuitiv darstellen. Zu den graphischen Werkzeugen gehören die dynamische Fehlerbaumanalyse, Petrinetze, GO-FLOW Diagramme und das Dynamic Event Sequence Diagram.

4.4.1 Dynamische Fehlerbaumanalyse

Die dynamische Fehlerbaumanalyse beseitigt die Einschränkungen der Fehlerbaum- und Ereignisablaufanalyse, bei der Modellierung der zeitlichen Ereignisfolge. Um die zeitliche Ereignisfolge zu modellieren, wurden zusätzliche Verknüpfungen, neben der Und-Verknüpfung und der Oder-Verknüpfung definiert, z.B. Verzögerungs-Verknüpfung, Ersatz-Verknüpfung, Prioritäts-Verknüpfung und die Verknüpfung für die funktionale Abhängigkeit. In der Tabelle 4.1 werden die im dynamischen Fehlerbaum definierten Verknüpfungen zusammengefasst.

Dynamische Fehlerbäume können leicht in das Markov-Modell konvertiert werden. Bei großen Systemen können enorme Anzahlen von Zuständen erzeugt werden. Um die Explosion der Zustände zu vermeiden, versucht man den Fehlerbaum in dynamische

Teile und statische Teile zu zerlegen. Während die statischen Teile mit den Binary Decision Diagrammen berechnet werden, berechnet man die dynamischen Teile mit dem Markov-Modell.

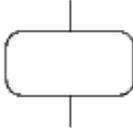
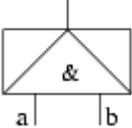
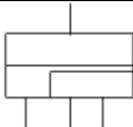
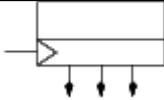
	
Verzögerung-Verknüpfung	Priorität-Und-Verknüpfung
	
Ersatz-Verknüpfung	Funktionale Abhängigkeit-Verknüpfung

Tabelle 4.1: Zusätzliche Verknüpfungen des dynamischen Fehlerbaumes

4.4.2 GO-FLOW Diagramme

Der Name „GO-FLOW“ kommt von der „GO“ Methode, wobei „FLOW“ den Fluss physikalischer Variablen bedeutet. Bei dieser Methode werden die Signale, die den physikalischen Größen entsprechen, durch das System verfolgt. Eine Reihe von Operatoren z.B. logische Operatoren, Signaloperatoren und verschiedene Möglichkeiten der Systemausfälle werden definiert, um die GO-FLOW Diagramme herzustellen. Die Wahrscheinlichkeit, dass ein Signal z.B. elektrischer Strom zu einem bestimmten Zeitpunkt kommt, kann mit den Standardgesetzen der Algebra berechnet werden.

Der Vorteil der GO-FLOW Diagramme besteht darin, dass die Phased Mission Szenarien mit den GO-FLOW Diagrammen kompakt dargestellt werden können. Die Nachteile der GO-FLOW Diagramme liegen darin, dass die wichtigen Informationen über die Struktur des Systems z.B. die Minimalschnitte nicht leicht erhalten werden können. Die GO-FLOW Diagramme werden in den Situationen eingesetzt, wenn die Ausfallraten und die Reparaturraten konstant sind.

4.4.3 Petrinetze

Petrinetze sind graphische Werkzeuge zur Modellierung und Analyse der dynamischen Abläufe. Typische dynamische Abläufe wie z.B. die Übereinstimmung, der Konflikt, die Synchronisation können mit den Petrinetzen leicht modelliert werden. Ein Petrinetz ist ein gerichteter Graph und besteht aus Stellen und Übergängen. Stellen und Übergänge sind durch gerichtete Kanten verbunden. Ein Zustand ist gegeben, wenn der Platz markiert ist, d.h. er enthält wenigstens einen Token. Die dynamischen Abläufe z.B. Synchronisation können durch die Bewegung der Token (schwarzer Punkt) in dem Graph dargestellt werden.

Neben den zeitlosen Übergängen der klassischen Petrinetze wurden die Übergänge eingeführt, welche beim Schalten Zeit verbrauchen. Die verschiedenen Varianten der Petrinetze unterscheiden sich, welche Art der Zeit beim Schalten die Übergänge verbrauchen. Im stochastischen Petrinetz ist die Zeit exponential verteilt. Im verallgemeinerten stochastischen Petrinetz sind sowohl exponentialverteilte zeitbehaftete Übergänge als auch zeitlose Übergänge enthalten. Im deterministischen stochastischen Petrinetz können auch die Übergänge mit einer deterministischen Schaltzeit verwendet werden.

Petrinetze wurde für den Softwareentwurf eingesetzt. Professor Schneeweiss [Schneeweiss 1999] verwendete die Petrinetze zur Modellierung und Analyse der reparierbaren Systeme und nicht-reparierbaren Systeme. Für die Modellierung und Analyse der dynamischen Systeme wurden die Petrinetze von Herrn Dutuit und Herrn Chatelet eingesetzt.

Zur Bewertung der Petrinetze werden die Petrinetze in das Markov-Modell konvertiert. Um die Explosion der Anzahl der Zustände zu vermeiden, wird die Anwendung der Petrinetze auf ein vereinfachtes System beschränkt.

4.4.4 Event Sequence Diagramme

Die Event Sequence Diagramme können als verallgemeinerte Ereignisablaufdiagramme betrachtet werden, welche die Unfallabläufe graphisch darstellen. Durch die graphische Darstellung können die Analytiker die Unfallabläufe besser verstehen. In den Event Sequence Diagrammen werden die Unfallabläufe dargestellt, die vom auslösenden Ereignis durch mehrere Zustandübergänge zum

Endzustand laufen. Der Unterschied, im Vergleich mit dem herkömmlichen Ereignisablaufdiagramm ist, dass nicht nur die Zustände der Systemkomponenten, sondern auch die Zustände der Prozessgrößen z.B. Wasserfüllstand berücksichtigt werden.

Der Vorteil der Event Sequence Diagrammen besteht darin, dass die komplexen Szenarien übersichtlich dargestellt werden können. Die Event Sequence Diagramme wurden z.B. im Gebiet der Raumfahrt eingesetzt, um die relevanten Unfallabläufe zu identifizieren. Der Nachteil der Event Sequence Diagramme ist, dass alle Zustände und die Zustandsübergänge vom Analytiker definiert werden müssen. Bei großen Systemen kann es zu extrem großen Event Sequence Diagrammen führen. Die Bewertung der Event Sequence Diagramme erfolgt entweder mit dem Markov-Modell oder mit den diskreten dynamischen Ereignisablaufdiagrammen. In der Abbildung 4.4 wird ein Event Sequence Diagramm für das dynamische „Hold-Up-Tank“-System aus der Veröffentlichung des Herrn Siu [Siu, 1994] dargestellt.

4.4.5 Zusammenfassung

Die dynamische PSA-Methode berücksichtigt explizit die zeitlichen Wechselwirkungen zwischen den Komponentenausfällen und der Systemdynamik. Daher können die dynamische PSA-Methode die Unfallabläufe im Vergleich mit den herkömmlichen PSA-Methoden realitätsnah modellieren und genauer analysieren.

Die Theorie der dynamischen PSA bietet eine sehr allgemeine mathematische Formulierung zur Modellierung der dynamischen Unfallabläufe an. Um die Theorie der dynamischen PSA anzuwenden, muss sie vereinfacht werden.

Die Zustandsübergangsmethoden z.B. das Markov-Modell, die diskreten dynamischen Ereignisablaufdiagramme und die Monte Carlo Simulation können die zeitlichen Wechselwirkungen zwischen den zufälligen Komponentenausfällen und der Systemdynamik gut berücksichtigen. Bei dem Markov-Modell müssen alle möglichen Zustände und Zustandsübergänge vorab explizit definiert werden. Da die Anzahl der Zustände exponentiell mit der Anzahl der Komponenten wächst, bleibt die Anwendung auf die Systeme mit sehr wenigen Komponenten und Prozessgrößen beschränkt. Bei den diskreten dynamischen Ereignisablaufdiagrammen und der Monte Carlo Simulation müssen nicht alle Zustände und Zustandsübergänge vorab explizit definiert werden. Beide Verfahren verwenden ein Systemmodell zur Modellierung der zeitlichen

Entwicklung der Prozessgrößen und der menschlichen Handlung, so dass die Wechselwirkungen zwischen den zufälligen Komponentenausfällen und den Prozessgrößen bzw. menschlichen Handlungen berücksichtigt werden können. Der große Unterschied zwischen der Monte Carlo Simulation und den diskreten dynamischen Ereignisablaufdiagrammen liegt in der Darstellung der Unfallabläufe. Während alle möglichen Unfallabläufe systematisch in den dynamischen Ereignisablaufdiagrammen simuliert werden, simuliert die Monte Carlo Simulation nur die zufällig ausgewählten Unfallabläufe. Der Rechenaufwand der diskreten dynamischen Ereignisablaufdiagramme wird sehr hoch bei klein gewählten Zeitschritten. Der Rechenaufwand der Monte Carlo Simulation hängt nur von der Anzahl der Simulationsläufe und nicht von der Komplexität des Systems ab. Daher gibt es für große Systeme kaum eine praktikable Alternative als die Monte Carlo Simulation. Allerdings liegt die größte Einschränkung der Monte Carlo Simulation darin, dass die notwendige Anzahl der Simulationen bei gegebener Genauigkeitsanforderung mit der Zuverlässigkeit des Systems stark ansteigt. Der Rechenaufwand der Monte Carlo Simulation bei der Anwendung auf hoch zuverlässige Systeme z.B. Kernkraftwerke kann sehr hoch werden, so dass rechenzeitbeschleunigende Methoden der Monte Carlo Simulation zum Einsatz kommen müssen.

Die graphischen Werkzeuge, wie z.B. die dynamische Fehlerbaumanalyse, GO-FLOW Diagramme, Petrinetze und Event Sequence Diagramme können die dynamischen Unfallabläufe intuitiv darstellen und die relevanten Unfallabläufe leicht identifizieren. Die graphischen Werkzeuge können den Analytikern helfen, die relevanten Unfallszenarien zu identifizieren. Die erzeugte Graphik wird für die Eingabedaten eines mathematischen Modells verwendet. Die Bewertung der Graphik erfolgt meistens mit dem Markov-Modell. Infolgedessen bleibt die Anwendung der graphischen Werkzeuge, wie das Markov-Modell, auf einfache Systeme beschränkt.

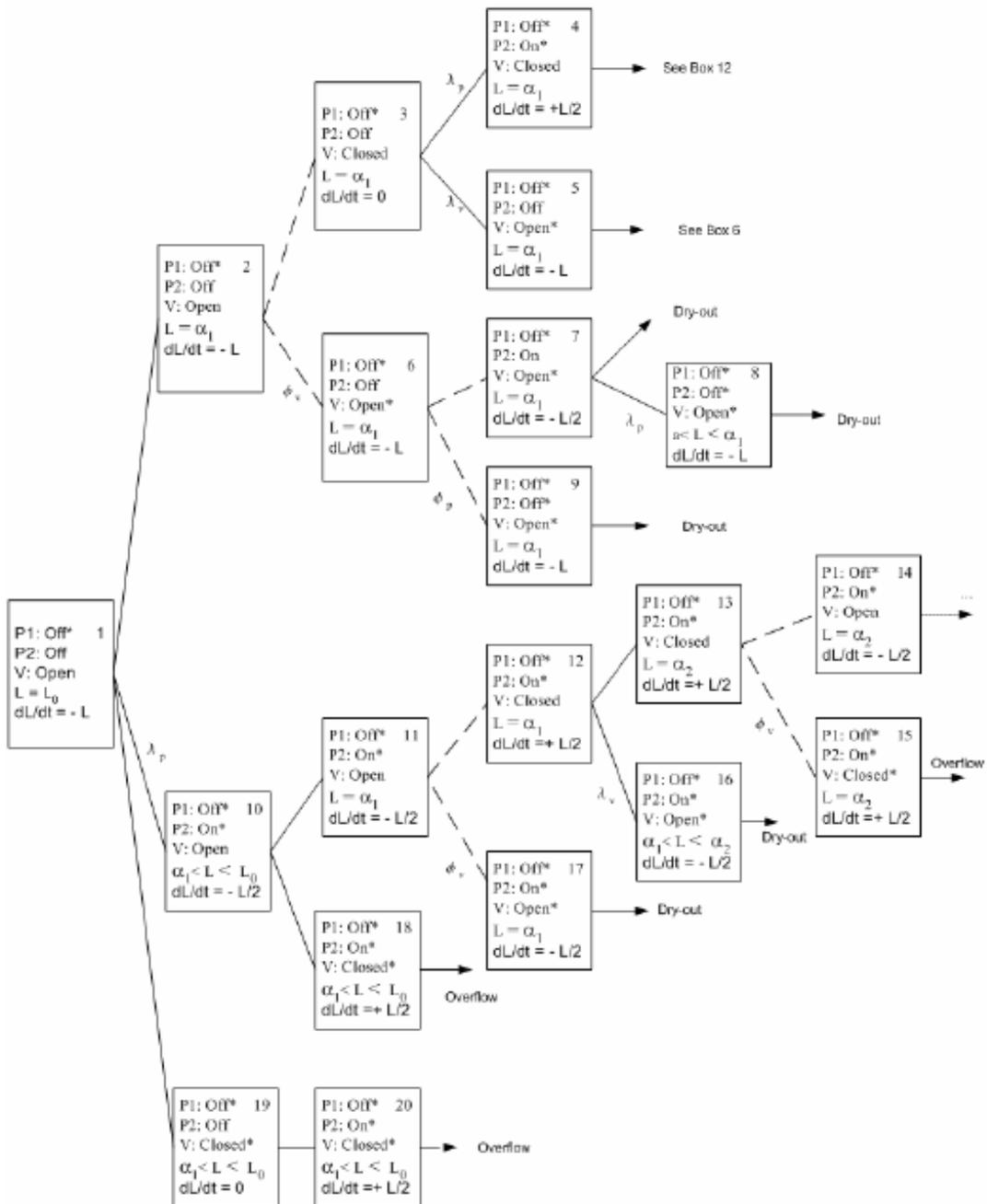


Abbildung 4.4: Beispiel - Event Sequence Diagramm

In Tabelle 4.2 werden alle in diesem Kapitel untersuchten dynamischen PSA-Methoden nach der Anwendbarkeit auf die komplexen dynamischen Systeme zusammengefasst.

	Methoden	Komplexe dynamische Systeme	
		nein	ja
Statische Methoden	Fehlerbaumanalyse	■	
	Ereignisablaufanalyse	■	
Analytische Methode	Theorie der dynamischen PSA	■	
Erweiterung von statischen Methoden	Phased Mission Methoden	■	
Implizite Methode	Markov-Modelle	■	
Explizite Methoden	Dynamische Ereignisablaufdiagramme		■
	Monte Carlo-Simulation		■
Graphische Werkzeuge	Dynamische Fehlerbaumanalyse	■	
	GO-FLOW Diagramme	■	
	Petrinetze	■	
	Event Sequence Diagramme	■	

Tabelle 4.2: Überblick über die dynamischen PSA-Methoden

5 Aufbau der MCS und statistische Auswertung der Stichproben

Die Untersuchung in Kapitel 4 hat gezeigt, dass die MCS für die Sicherheitsbewertung der dynamischen Systeme gut geeignet ist. Daher werden nachfolgend der mathematische Aufbau der MCS und die statistische Auswertung der Stichproben erläutert. Dies sind die mathematischen Grundlagen zur Analyse der dynamischen Systeme und weiterer Optimierung der MCS.

5.1 Wichtige Definitionen

Vor dem Aufbau der MCS und der statistischen Auswertung der Stichproben werden die in dieser Arbeit verwendeten mathematischen Definitionen eingeführt.

5.1.1 Zufallsvariablen

Unter Zufallsvariable versteht man eine Funktion, die zu jedem Ereignis $\omega \in \Omega$ eine reelle Zahl $X(\omega)$ zuordnet:

$$X : \omega \rightarrow X(\omega) \in R \quad (5.1)$$

Die Zufallsvariable kann man als eine Funktion von X definieren, in der alle Realisationen x der Zufallsvariable umfasst werden.

5.1.2 Verteilungsfunktion der Zufallsvariablen

Die Zufallsvariablen können durch ihre Verteilungsfunktion eindeutig charakterisiert werden. Die Eigenschaft der Verteilungsfunktion wird in diesem Abschnitt erläutert [Birolini 1997].

Die Verteilungsfunktion F_x gibt die Wahrscheinlichkeit an, dass die Zufallsvariable X einen bestimmten Wert x nicht überschreitet:

$$F_X = F(x) = P(X < x) = \int_{-\infty}^x f(x)dx \quad (5.2)$$

Die Verteilungsdichte f_x ergibt sich aus der ersten Ableitung der Verteilungsfunktion F_x :

$$f_x = f(x) = \frac{dF(x)}{x} \quad (5.3)$$

Die Verteilungsdichte f_x und die Verteilungsfunktion F_x müssen folgende Bedingungen erfüllen:

$$f_x \geq 0 \quad (5.4)$$

$$\int_{-\infty}^{+\infty} f(x)dx = 1 \quad (5.5)$$

$$\lim_{x \rightarrow -\infty} F(x) = 0 \quad (5.6)$$

$$\lim_{x \rightarrow +\infty} F(x) = 1 \quad (5.7)$$

$$F(x) \text{ ist monoton steigend} \quad (5.8)$$

In der Praxis werden häufig der Mittelwert μ_x und die Varianz σ_x^2 von den Zufallsvariablen verwendet. Der Erwartungswert μ_x und die Varianz σ_x^2 können mit folgender Gleichung berechnet werden:

$$\mu_x = E[X] = \int_{-\infty}^{+\infty} xf(x)dx \quad (5.9)$$

$$\begin{aligned} \sigma_x^2 &= E[(X - \mu_x)^2] \\ &= \int_{-\infty}^{+\infty} (x - \mu_x)^2 f(x)dx \\ &= \int_{-\infty}^{+\infty} x^2 f(x)dx - \mu_x^2 \end{aligned} \quad (5.10)$$

Die Standardabweichung ist ein Maß für die Streuung der Werte einer Zufallsvariable um ihren Mittelwert. Die Standardabweichung σ_x ergibt sich aus folgender Gleichung.

$$\sigma_x = \sqrt{\sigma_x^2} \quad (5.11)$$

Die Normalverteilung ist die wichtigste Verteilungsfunktion ist. Die Bedeutung der Normalverteilung beruht auf dem zentralen Grenzwertsatz. Die Verteilung der Summe von n beliebigen Zufallsvariablen nähert sich unabhängig von den Verteilungstypen der Variablen mit wachsendem n der Normalverteilung. Die Dichtefunktion und Verteilungsfunktion der normalverteilten Zufallsvariablen sind gegeben durch:

$$f(x) = \frac{1}{\sigma_x \sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\frac{x - \mu_x}{\sigma_x}\right)^2\right] \quad (5.12)$$

$$F(x) = \frac{1}{\sigma_x \sqrt{2\pi}} \int_{-\infty}^x \left[-\frac{1}{2} \left(\frac{x - \mu_x}{\sigma_x}\right)^2\right] dx \quad (5.13)$$

Die Normalverteilung kann durch den Mittelwert und die Varianz vollständig beschrieben werden. In der Praxis wird die Normalverteilung durch eine

Transformation $Z = \frac{X - \mu_x}{\sigma_x}$ auf eine Standardnormalverteilung mit $\mu_x = 0$ und σ_x

= 1 zurückgeführt werden. Die Dichtefunktion und Verteilungsfunktion standardnormalverteilter Zufallsvariablen sind gegeben durch:

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \quad (5.14)$$

$$\varphi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z \exp\left(-\frac{z^2}{2}\right) dz \quad (5.15)$$

5.1.3 Zufallsvektoren

Wenn eine Anzahl von Zufallsvariablen betrachtet wird, werden n Zufallsvariablen X_i in einem n-dimensionalen Zufallsvektor X zusammengefasst. Die gemeinsame

Verteilungsfunktion und Dichtefunktion eines Zufallsvektors X ergibt sich wie folgt:

$$F_{X_1 X_2 X_3, \dots, X_n} = F(x_1, x_2, x_3, \dots, x_n) = P[(X_1 < x_1) \cap (X_2 < x_2) \cap \dots \cap (X_n < x_n)] \quad (5.16)$$

$$f_{X_1 X_2 X_3, \dots, X_n} = f(x_1, x_2, x_3, \dots, x_n) = \frac{\partial^n F}{\partial X_1 \partial X_2 \dots \partial X_n}(x_1, x_2, \dots, x_n) \quad (5.17)$$

Die Verteilungsdichte jeder Komponente X_i erhält man aus der gemeinsamen Dichtefunktion durch die Integration aller anderen Variablen:

$$f_{X_i} = f(x_i) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_{i-1} dx_{i+1} \dots dx_n \quad (5.18)$$

Wenn die Zufallsvariablen X_i statistisch unabhängig sind, ergibt sich die gemeinsame Dichtefunktion wie folgt:

$$f_{X_1 X_2 X_3, \dots, X_n} = f_{X_1} f_{X_2} \dots f_{X_n} = \prod_{i=1}^n f_{X_i} \quad (5.19)$$

5.2 Zufallszahlengenerator

Eine der wichtigen Voraussetzungen für die MCS ist der Zufallsgenerator. Als Zufallszahlengenerator bezeichnet man ein Verfahren, das eine Folge von Zufallszahlen erzeugt. Ein weit verbreitetes Verfahren zur Erzeugung der Zufallszahlen, die zwischen 0 und 1 gleich verteilt sind, wird wie folgt berechnet:

$$z_k = (az_{k-1} + c) \bmod (m), \forall k = 1, \dots, n \quad (5.20)$$

z_k lässt sich direkt durch den Startwert z_0 sowie durch die Parameter m, a bzw. c berechnen. Die Zufallszahlen ergeben sich durch die Normierung:

$$u_k = \frac{z_k}{m} \quad (5.21)$$

Um einen guten Zufallszahlengenerator zu konstruieren, sollen die Parameter m, a

bzw. z_0 so ausgewählt werden, dass die Länge der Periode der Zufallszahlen möglichst nahe an der Maximallänge m liegt.

5.3 Inversionsmethode

Die nicht-gleich verteilten Zufallszahlen lassen sich, ausgehend von den gleich verteilten Zufallszahlen zwischen 0 und 1 mit Hilfe der Inversionsmethode erzeugen. Die Zufallszahlen einer beliebigen Verteilungsfunktion sind

$$Z = F^{-1}(u), \quad (5.22)$$

wobei $F(x)$ eine beliebige Verteilungsfunktion ist und u die gleich verteilten Zufallszahlen zwischen 0 und 1 sind.

5.4 Schätzwert des Erwartungswertes

x_1, x_2, \dots, x_n sind die Zufallsstichproben der Zufallsvariable X . Der Mittelwert der Stichproben ist unbekannt und wird geschätzt durch

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \dots + x_n) = \frac{1}{n} \sum_{i=1}^n x_i. \quad (5.23)$$

Die Standardabweichung des Mittelwertes \bar{X} ergibt sich für $n \rightarrow \infty$ aus folgender Gleichung.

$$\sigma_{\bar{x}} = \sqrt{\frac{\text{Var}(X)}{n}} \quad (5.24)$$

Eine Schätzung der Standardabweichung des Mittelwertes \bar{X} erfolgt mit folgender Gleichung.

$$\hat{\sigma}_{\bar{x}} = \frac{\sqrt{\frac{1}{n-1} \sum_i (x_i - \bar{x})^2}}{\sqrt{n}} \quad (5.25)$$

5.5 Konfidenzintervall des geschätzten Mittelwertes

Das Konfidenzintervall schließt den Bereich des geschätzten Parameters ein, der mit einer zuvor festgelegten Wahrscheinlichkeit den wahren Parameter trifft. Das Konfidenzintervall, in dem der wahre Mittelwert mit der Wahrscheinlichkeit $1 - \alpha$ liegt, wird nach folgender Gleichung bestimmt.

$$\left[\bar{x} - \frac{\sqrt{\frac{1}{n-1} \sum_i (x_i - \bar{x})^2}}{\sqrt{n}} z\left(1 - \frac{\alpha}{2}\right) \dots \bar{x} + \frac{\sqrt{\frac{1}{n-1} \sum_i (x_i - \bar{x})^2}}{\sqrt{n}} z\left(1 - \frac{\alpha}{2}\right) \right] \quad (5.26)$$

wobei $z\left(1 - \frac{\alpha}{2}\right)$ das $(1 - \alpha/2)$ Quantil einer Standardnormalverteilung bezeichnet.

In der Praxis wird häufig das 95%-Konfidenzintervall verwendet, in dem der wahre Wert mit der Wahrscheinlichkeit von 95% trifft. Das 95%-Konfidenzintervall ergibt sich aus der folgenden Gleichung.

$$\left[\bar{x} - \frac{\sqrt{\frac{1}{n-1} \sum_i (x_i - \bar{x})^2}}{\sqrt{n}} 1.96 \dots \bar{x} + \frac{\sqrt{\frac{1}{n-1} \sum_i (x_i - \bar{x})^2}}{\sqrt{n}} 1.96 \right] \quad (5.27)$$

5.6 Zusammenfassung

Dieses Kapitel beschäftigt sich mit dem Aufbau der MCS und der statistischen Auswertung der Stichproben. Die Voraussetzung der MCS ist der Zufallsgenerator. Mit dem Zufallsgenerator können die Zufallszahlen zwischen 0 und 1 erzeugt werden. Die Zufallszahlen aus einer beliebigen Verteilung können mit Hilfe der Inversionsmethoden erzeugt werden. Der wesentliche Ansatz der MCS besteht darin, dass theoretische Erwartungswerte durch die Mittelwerte der ausgewählten Stichproben näherungsweise geschätzt werden können. Durch die Definition des Konfidenzintervalls kann auch die Schätzungsgenauigkeit quantifiziert werden.

In folgendem Kapitel wird ein dynamisches „Hold-Up-Tank“-System beschrieben. Das Potential der MCS auf das dynamische „Hold-Up-Tank“-System wird demonstriert. Die Eintrittswahrscheinlichkeiten der kritischen Zustände „DRYOUT“ und „OVERFLOW“ werden mit der MCS ermittelt.

6 Modellierung des „Hold-Up-Tank“-Systems und Analyse mit der MCS

Das dynamische „Hold-Up-Tank“-System wurde als erstes Standardproblem für die dynamischen PSA-Methoden von Herrn Siu und Herrn Labenau vorgeschlagen ([Siu, 1994], [Labeau, 2000]). In diesem Kapitel wurde eine Variante des „Hold-Up-Tank“-Systems ([GRS-A-2786, 2001], [GRS-A-2997,2001]) mit kleinen Änderungen implementiert und die dynamischen Abläufe wurden mit der MCS analysiert.

Das Ziel dieser Untersuchung ist zu demonstrieren,

- wie sich die zufälligen Komponentenausfälle und die Systemdynamik z.B. die zeitliche Entwicklung der Prozessgröße und das Regelsystem gegenseitig beeinflussen,
- wo die Einschränkungen der herkömmlichen PSA-Methoden liegen,
- in wie weit die MCS die dynamischen Unfallabläufe, unter Berücksichtigung der zeitlichen Wechselwirkungen, modellieren und analysieren kann,
- Wo die MCS weiter optimiert werden muss.

6.1 Beschreibung des „Hold-Up-Tank“-Systems

Das „Hold-Up-Tank“-System besteht aus einem mit Wasser gefüllten Speicher-Behälter. Der Wasserfüllstand im Behälter wird durch 3 Systemkomponenten – Pumpe 1, Pumpe 2 und ein Ventil gesteuert. Die Pumpen 1 und 2 haben die Aufgabe, das Wasser in den Behälter einzuspeisen, während das Ventil das Wasser ablässt. Die Abflussmenge des Ventils entspricht der Fördermenge der Pumpe 1. Die Pumpe 2 ist im Normalzustand des Systems ausgeschaltet und befindet sich im Bereitschaftszustand. Die Fördermenge der Pumpe 2 ist halb so groß, wie die Fördermenge der Pumpe 1. Das „Hold-Up-Tank“-System wird in der Abbildung 6.1 dargestellt.

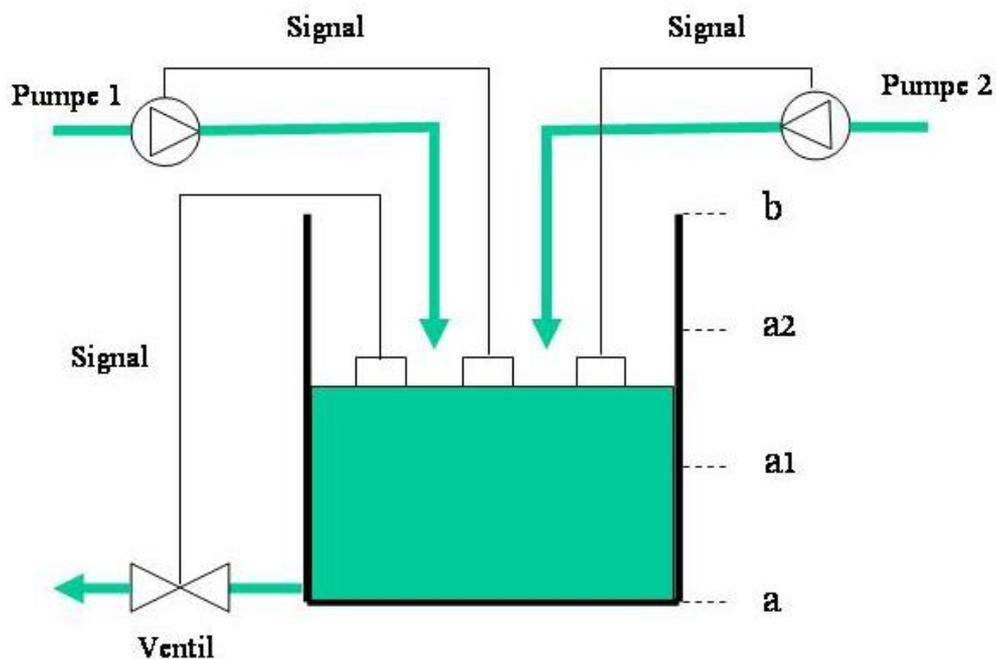


Abbildung 6.1: "Hold-Up-Tank"-System

Wenn der Wasserfüllstand auf ein bestimmtes Niveau a_1 abgesunken ist, sorgen die Pumpen dafür, Wasser in den Behälter nachzufüllen. Wenn der Wasserfüllstand ein bestimmtes Niveau a_2 überschreitet, sorgt das Ventil dafür, dass der Wasserfüllstand bis unter das Niveau a_2 abgelassen wird. Insgesamt hat das „Hold-Up-Tank“-System die Aufgabe, den Wasserfüllstand im Behälter nicht zu tief absinken bzw. zu hoch ansteigen zu lassen.

Die Schadenszustände des „Hold-Up-Tank“-Systems werden erreicht,

- wenn der Wasserfüllstand innerhalb einer Anforderungszeit auf das Niveau a sinkt („DRYOUT“-Zustand),
- oder wenn der Wasserfüllstand innerhalb einer Anforderungszeit über das Niveau b ansteigt („OVERFLOW“-Zustand).

In Tabelle 6.1 werden die möglichen Schadenszustände des „Hold-Up-Tank-Systems“ dargestellt.

Schadenzustand	
Zustand 1	„DRYOUT“-Zustand
Zustand 2	„OVERFLOW“-Zustand

Tabelle 6.1: Schadzustände des "Hold-Up-Tank"-Systems

Die Pumpe1 und 2 und das Ventil arbeiten unabhängig voneinander. In Tabelle 6.2 werden die möglichen Komponentenzustände definiert.

Zustand	Ventil	Pumpe 1	Pumpe 2
1	offen	aus	aus
2	geschlossen	an	an
3	schließt nicht	startet nicht	startet nicht
4	öffnet nicht	stoppt nicht	stoppt nicht

Tabelle 6.2: Mögliche Komponentenzustände des "Hold-Up-Tank"-Systems

Der Zustand 1 und der Zustand 2 werden als funktionierende Zustände bezeichnet, d.h. die Systemkomponenten reagieren nach der Anforderung, wenn die Systemkomponenten sich im Zustand 1 und im Zustand 2 befinden. Der Zustand 3 und der Zustand 4 werden als nicht funktionierende Zustände bezeichnet, d.h. die Systemkomponenten reagieren nicht nach der Anforderung, wenn die Systemkomponenten sich im Zustand 3 und im Zustand 4 befinden.

6.1.1 Systemparameter und Anfangsbedingungen

Zum Zeitpunkt $t=0$ ist der Behälter zur Hälfte mit Wasser gefüllt, d.h. der Wasserfüllstand beträgt $WF=5$ m. In Tabelle 6.3 werden die Anfangsbedingungen der Systemparameter des „Hold-Up-Tank“-Systems dargestellt.

Systemparameter	a	a1	a2	b	FH
Anfangsbedingungen	0m	3m	7m	10m	5m

Tabelle 6.3: Systemparameter des "Hold-Up-Tank"-Systems

Zum Zeitpunkt $t = 0$ befinden sich die Systemkomponenten im Zustand Ventil = offen, Pumpe 1 = startet nicht, Pumpe 2 = aus. Dieser Anfangszustand der Systemkomponenten wurde ausgewählt, weil die Unfallabläufe ausgehend von diesem Anfangszustand schneller als die anderen Anfangszustände den Schadenszustand „DRYOUT“ erreichen können. In Tabelle 6.4 werden die Anfangszustände der Systemkomponenten zusammengefasst.

Systemkomponenten	Ventil	Pumpe 1	Pumpe 2
Anfangszustand	offen	startet nicht	aus

Tabelle 6.4: Anfangszustände der Systemkomponenten

6.1.2 Dynamik des „Hold-Up-Tank“-Systems

Die Entwicklung des Wasserfüllstandes in Abhängigkeit von den Zuständen der Systemkomponenten pro Zeitschritt Δt werden nach den folgenden Gleichungen in Tabelle 6.5 berechnet.

	Systemkomponenten	Wasserfüllstand
1	Ventil =offen, Pumpe 1=aus, Pumpe 2=aus	$FH(t)=FH(t-1)-0.1*5*\Delta t$
2	Ventil =offen, Pumpe 1=an, Pumpe 2=aus	$FH(t)=FH(t-1)$
3	Ventil =offen, Pumpe 1=aus, Pumpe 2= an	$FH(t)=FH(t-1)-0.05*5*\Delta t$
4	Ventil =offen, Pumpe 1=an, Pumpe 2= an	$FH(t)=FH(t-1)+0.05*5*\Delta t$
5	Ventil =geschlossen, Pumpe 1=aus, Pumpe 2= aus	$FH(t)=FH(t-1)$
6	Ventil =geschlossen, Pumpe 1=an, Pumpe 2= aus	$FH(t)=FH(t-1)+0.1*5*\Delta t$
7	Ventil =geschlossen, Pumpe 1=aus, Pumpe 2= an	$FH(t)=FH(t-1)+0.05*5*\Delta t$
8	Ventil =geschlossen, Pumpe 1=an, Pumpe 2= an	$FH(t)=FH(t-1)+0.15*5*\Delta t$

Tabelle 6.5: Dynamik des "Hold-Up-Tank"-Systems

6.1.3 Regelsystem

Der Wasserfüllstand kann durch das Regelsystem zwischen dem Niveau a_1 und dem Niveau a_2 geregelt werden. Die Komponentenzustände können durch das

Regelsystem in Abhängigkeit von der Entwicklung des Wasserfüllstandes, nach den in Tabelle 6.6 dargestellten Regeln geändert werden.

	Wasserfüllstand und Komponentenzustände	Regel
1	$FH > a_2$ und Ventil= geschlossen	Das Ventil wird geöffnet.
2	$FH > a_2$ und Pumpe 1= an	Die Pumpe 1 wird ausgeschaltet.
3	$FH > a_2$ und Pumpe 2= an	Die Pumpe 2 wird ausgeschaltet.
4	$FH < a_1$ und Ventil= offen	Das Ventil wird geschlossen.
5	$FH < a_1$ und Pumpe 1= aus	Die Pumpe 1 wird angeschaltet.
6	$FH < a_1$ und Pumpe 2= aus	Die Pumpe 2 wird angeschaltet.
7	$a_1 \leq FH \leq a_2$, Ventil= offen und Pumpe 1=aus	Die Pumpe 1 wird angeschaltet.
8	$a_1 \leq FH \leq a_2$, Ventil= geschlossen und Pumpe 1=an	Das Ventil wird geöffnet.
9	$a_1 \leq FH \leq a_2$, Pumpe 1= an und Pumpe 2=an	Die Pumpe 2 wird ausgeschaltet.

Tabelle 6.6: Regelung des "Hold-Up-Tank"-Systems

6.2 Analyse des „Hold-Up-Tank“-Systems

Das Ziel der PSA ist die Wahrscheinlichkeit, mit der der „DRYOUT“-Zustand und „OVERFLOW“-Zustandes innerhalb einer bestimmten Betriebszeit erreicht wird, zu ermitteln. Dafür müssen zuerst die stochistischen Elemente und die Wahrscheinlichkeitsdichtefunktion der stochastischen Elemente festgelegt werden.

6.2.1 Zufällige Ausfälle der Systemkomponenten und des Regelsystems

Das Ventil, die Pumpe 1 und die Pumpe 2 im „Hold-Up-Tank“-System werden als die stochastischen Elemente bezeichnet. Die stochastischen Elemente können entweder zufällig oder durch das Regelsystem abhängig vom Wasserfüllstand ausfallen.

6.2.1.1 Zufällige Komponentenausfälle

Das Ventil, die Pumpe 1 und die Pumpe 2 können zu zufälligen Zeitpunkten ausfallen. Die Wahrscheinlichkeitsdichtefunktion der Ausfallzeitpunkte wird für das Ventil, die Pumpe 1 und die Pumpe 2 wie folgt angenommen.

- Die Ausfallzeitpunkte des Ventils sind exponentiell mit dem Parameter $5 \cdot 10^{-3}$.
- Die Ausfallzeitpunkte der Pumpe 1 und der Pumpe 2 sind exponentiell mit dem Parameter $1 \cdot 10^{-3}$.

6.2.1.2 Vom Wasserfüllstand abhängige Komponentenausfälle

Das Regelsystem kann auch in Abhängigkeit vom Wasserfüllstand versagen. Nach der Anforderung des Regelsystems können die Komponentenzustände durch das Regelsystem entweder mit einer Wahrscheinlichkeit p erfolgreich oder mit einer Wahrscheinlichkeit $1-p$ nicht erfolgreich umgeschaltet werden. Die Wahrscheinlichkeit für die erfolgreiche Umschaltung der Systemkomponenten und die nicht erfolgreiche Umschaltung der Systemkomponenten nach den Regeln des Regelsystems wird wie folgt angenommen.

- Die Wahrscheinlichkeit, dass das Ventil nicht mehr geöffnet werden kann, beträgt $P = 0.08$
- Die Wahrscheinlichkeit, dass das Ventil nicht mehr geschlossen werden kann, beträgt $P = 0.05$
- Die Wahrscheinlichkeit, dass die Pumpe 1 oder Pumpe 2 nicht mehr eingeschaltet werden kann, beträgt $P = 0.05$
- Die Wahrscheinlichkeit, dass die Pumpe 1 oder Pumpe 2 nicht mehr ausgeschaltet werden kann, beträgt $P = 0.03$

6.2.2 Analyse des „Hold-Up-Tank“-Systems mit herkömmlichem Ereignisablaufdiagramm

Um die Eintrittswahrscheinlichkeit des Schadenszustandes „DRYOUT“-Zustandes und „OVERFLOW“-Zustands zu ermitteln, wird zuerst die Ereignisablaufanalyse eingesetzt. Die Unfallabläufe werden, ausgehend vom Anfangszustand in dem folgenden Ereignisablaufdiagramm in der Abbildung 6.2 dargestellt.

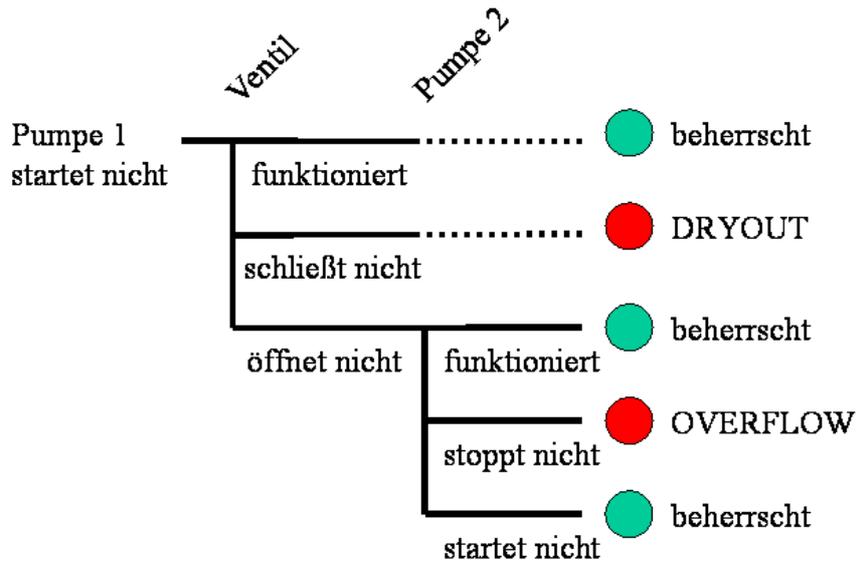


Abbildung 6.2: Analyse des "Hold-Up-Tank"-Systems mit herkömmlichem Ereignisablaufanalyse

Im Ereignisablaufdiagramm werden die Unfallabläufe so vereinfacht modelliert, dass die Systemkomponenten, das Ventil und die Pumpe 2, nur einmal angefordert werden. In der Realität kann das Ventil und Pumpe 2 mehrmals angefordert werden, so dass eine große Anzahl der Unfallabläufe erzeugt werden kann. Ohne direkte Kopplung an die Systemdynamik kann das Ereignisablaufdiagramm die zeitlichen Wechselwirkungen zwischen den zufälligen Komponentenausfällen und der Systemdynamik schwer erfassen. Die Anzahl der modellierten Unfallabläufe ist klein und die dynamischen Abläufe sind einfach, aber unrealistische Unfallabläufe können generiert werden.

Die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes ergibt sich aus der folgenden Gleichung. Ohne Berücksichtigung des Wasserfüllstandes kann die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes nicht genau ermittelt werden, da der Ausfall des Regelsystems vom Wasserfüllstand abhängig ist.

$$P(\text{DRYOUT}) = P(\text{Ventil schließt nicht}) \approx P(\text{Ausfall des Regelsystem}) + P(\text{zufälliger Ausfall des Ventils})$$

Die Analyse der Unfallabläufe mit der Ereignisablaufanalyse hat gezeigt, dass die Ereignisablaufanalyse die Unfallabläufe nicht realitätsnah modellieren und die Eintrittswahrscheinlichkeit des Schadenszustandes nicht genau ermitteln kann.

6.2.3 Analyse des „Hold-Up-Tank“-Systems mit MCS

Da die MCS für die Sicherheitsbewertung der dynamischen Systeme gut geeignet ist, wird das Potential der MCS für das „Hold-Up-Tank“-System untersucht. Mit der MCS sollen die Unfallabläufe, unter Berücksichtigung der zeitlichen Wechselwirkungen zwischen den zufälligen Komponentenausfällen und der Systemdynamik, modelliert und die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes und des „OVERFLOW“-Zustandes ermittelt werden.

6.2.3.1 Realisierung

Zur Modellierung der Entwicklung des Wasserfüllstandes, des Regelsystems und der zufälligen Komponentenausfälle wurden das Programm Perl und Maple verwendet. Während die zeitliche Entwicklung des Wasserfüllstandes und das Regelsystem mit dem Programm Perl modelliert werden, modellierte man die stochastischen Ausfälle der Systemkomponenten mit dem Programm Maple. Um die MCS automatisch durchzuführen wurde ein Simulationskonzept in der Abbildung 6.3 erstellt. Da das Programm Perl und das Programm Maple zu jedem Zeitschritt miteinander gekoppelt wurde, ist das Konzept in der Lage, die Unfallabläufe unter Berücksichtigung der zeitlichen Wechselwirkungen zwischen den Komponentenausfällen und der Systemdynamik zu modellieren. Das Programm Perl und Maple für das „Hold-Up-Tank“-System wird dem Anhang beigefügt.

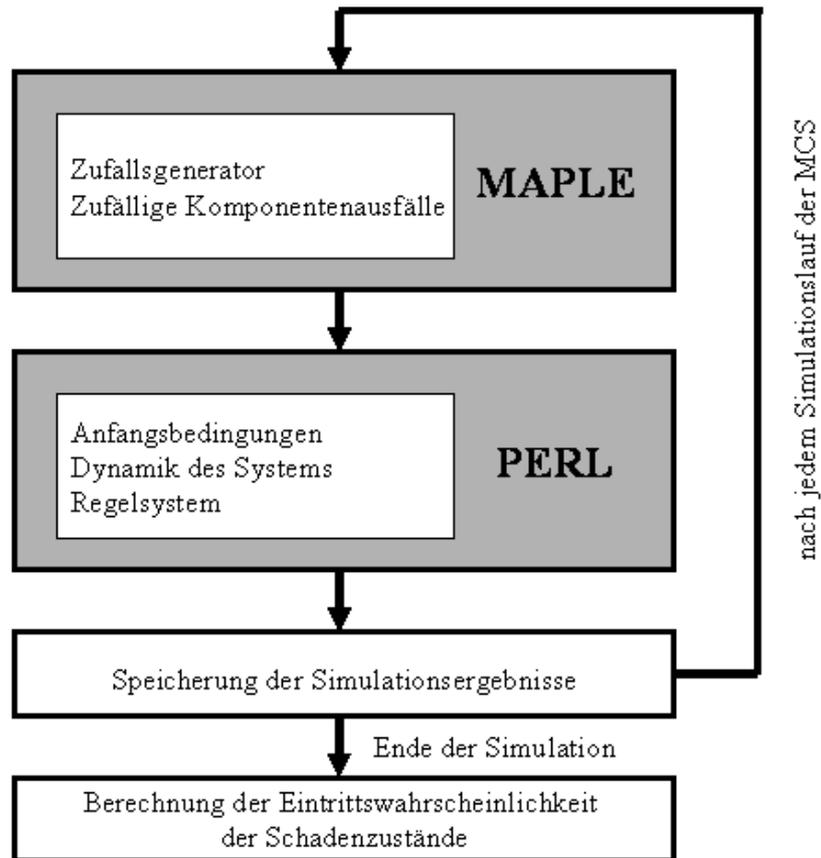


Abbildung 6.3: Analyse des "Hold-Up-Tank"-Systems mit der MCS

6.2.3.2 *Ergebnisse aus der MCS*

Zur Ermittlung der Eintrittswahrscheinlichkeit der Schadenzustände wurden die zufällig ausgewählten Unfallabläufe aus der MCS simuliert. Um die Unfallabläufe in Ereignisablaufdiagramm zu vergleichen, werden vier Unfallabläufe aus der MCS in den Abbildung 6.4 bis Abbildung 6.7 graphisch dargestellt. Die Unfallabläufe aus der MCS sind kompliziert im Vergleich mit dem Ereignisablaufdiagramm. Die Unfallabläufe, in denen das Ventil und die Pumpe 2 mehrfach angefordert werden, wurden auch simuliert. Daher kann die MCS die Unfallabläufe unter der Berücksichtigung der zeitlichen Wechselwirkungen realitätsnah modellieren und genau analysieren.

Lauf 1

Nach dem Start der Simulation sinkt der Wasserfüllstand. Wenn der Wasserfüllstand niedriger als 3 m ist, startet das Regelsystem. Das Ventil wird durch das Regelsystem nicht erfolgreich geschlossen und die Pumpe 2 wird durch das Regelsystem nicht erfolgreich eingeschaltet. Der Wasserfüllstand sinkt weiter, bis der „DRYOUT“-Zustand eintritt.

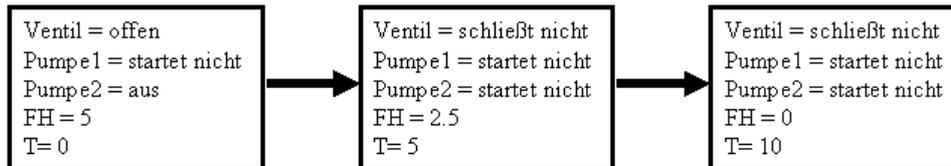


Abbildung 6.4: Unfallablauf 1 des "Hold-Up-Tank"-Systems

Lauf 2

Nach dem Start der Simulation sinkt der Wasserfüllstand. Wenn der Wasserfüllstand niedriger als 3 m ist, startet das Regelsystem. Das Ventil wird durch das Regelsystem erfolgreich geschlossen und die Pumpe 2 wird durch das Regelsystem erfolgreich eingeschaltet. Der Wasserfüllstand steigt. Zum Zeitpunkt $t = 9$ fällt das Ventil zufällig aus. Der Wasserfüllstand sinkt, bis der Zustand „DRYOUT“-Zustand eintritt.

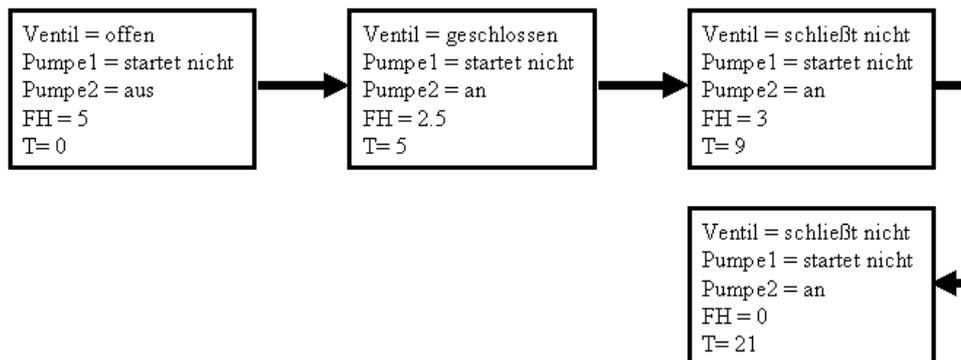


Abbildung 6.5: Unfallablauf 2 des "Hold-Up-Tank"-Systems

Lauf 3

Nach dem Start der Simulation sinkt der Wasserfüllstand. Wenn der Wasserfüllstand niedriger als 3 m ist, startet das Regelsystem. Das Ventil und die Pumpe 2 werden durch das Regelsystem erfolgreich geschlossen bzw. angeschaltet. Der Wasserfüllstand steigt. Wenn der Wasserfüllstand höher als 7 m ist, werden das Ventil und die Pumpe 2 durch das Regelsystem erfolgreich geöffnet bzw. ausgeschaltet. Der Wasserfüllstand sinkt. Wenn der Wasserfüllstand wieder niedriger als 3 m ist, wird das Ventil und die Pumpe 2 durch das Regelsystem erfolgreich geschlossen bzw. eingeschaltet. Der Wasserfüllstand steigt. Zum Zeitpunkt $t = 42$ fällt das Ventil zufällig aus. Der Wasserfüllstand sinkt, bis der Zustand „DRYOUT“-Zustand eintritt.

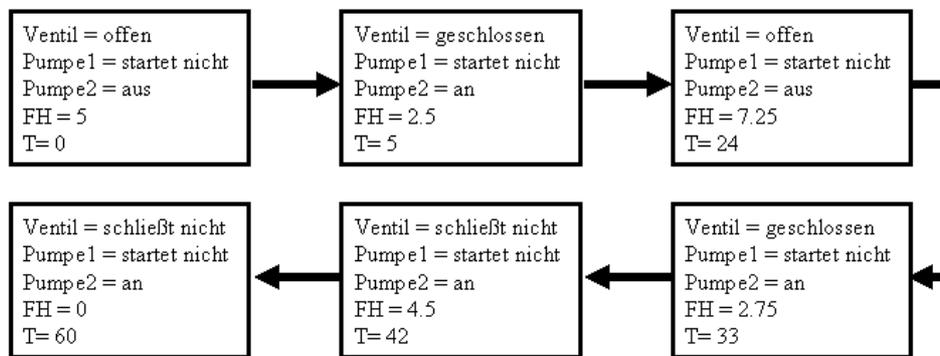


Abbildung 6.6: Unfallablauf 3 des "Hold-Up-Tank"-Systems

In der Abbildung 6.7 wurde noch ein komplizierter Störfallablauf dargestellt.

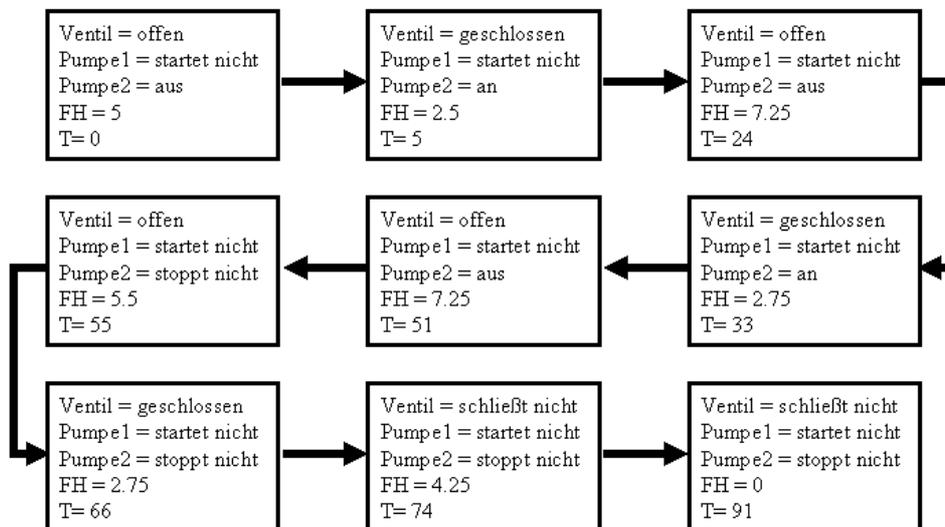


Abbildung 6.7: Unfallablauf 4 des "Hold-Up-Tank"-Systems

6.2.3.3 Quantifizierung der stochastischen Größen

Die Quantifizierung der stochastischen Größen beantwortet die Frage, wie wahrscheinlich das Eintreten des „DRYOUT“-Zustandes oder des „OVERFLOW“-Zustandes innerhalb einer bestimmten Betriebszeit ist. Die Quantifizierung wurde für eine Betriebsdauer von 100 Stunden und von 10000 Stunden betrachtet.

In den Abbildung 6.8 und Abbildung 6.9 wurden die Verläufe des Wasserfüllstandes innerhalb von 100 Stunden und 10000 Stunden aufgezeigt. Der Wasserfüllstand wurde durch das Regelsystem in einem Bereich zwischen 3 m und 7 m geregelt. Der „DRYOUT“-Zustand tritt zwischen 0 Stunden und 800 Stunden ein, während der „OVERFLOW“-Zustand zwischen 0 Stunden und 5000 Stunden eintritt. Da der Wasserfüllstand vom Anfang an sinkt, ist die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustands größer als die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustands.

In den Abbildung 6.10 und Abbildung 6.11 wurden die Verläufe der Wahrscheinlichkeit, dass der „DRYOUT“-Zustand innerhalb von 100 Stunden und 10000 Stunden eintritt, aufgezeigt. Die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes zum Zeitpunkt $T=100$ beträgt 27%. Die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes zum Zeitpunkt $T=10000$ beträgt 62%.

In den Abbildung 6.12 und Abbildung 6.13 wurden die Verläufe der Wahrscheinlichkeit, dass der „OVERFLOW“-Zustand innerhalb von 100 Stunden und 10000 Stunden eintritt, dargestellt. Die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes zum Zeitpunkt $T=100$ Stunden ist ca. 1% hoch und die des „OVERFLOW“-Zustandes zum Zeitpunkt $T=10000$ Stunden liegt bei 38%.

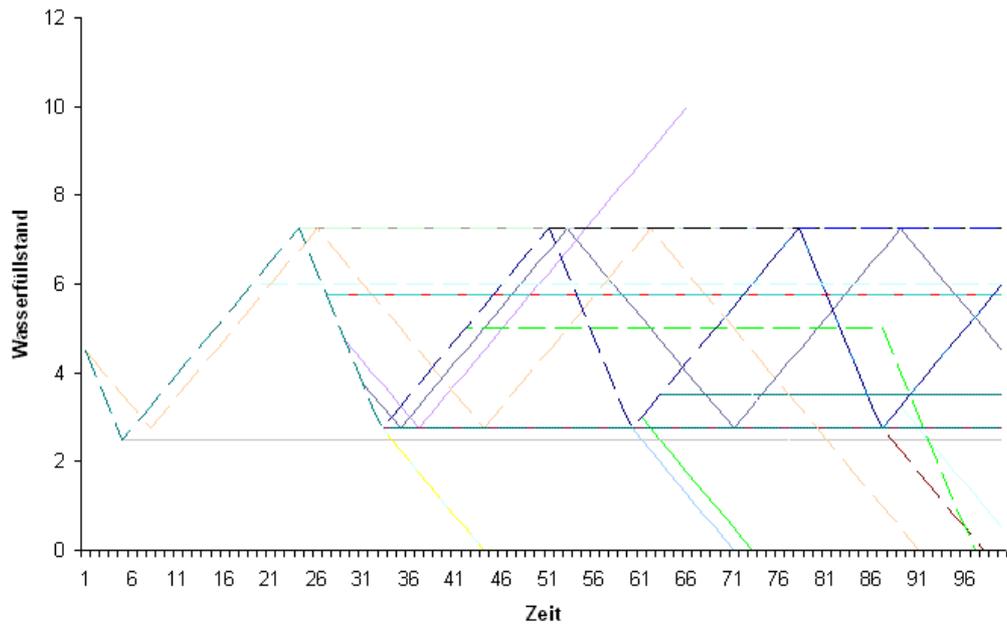


Abbildung 6.8: Verläufe des Wasserfüllstandes innerhalb von 100 Stunden

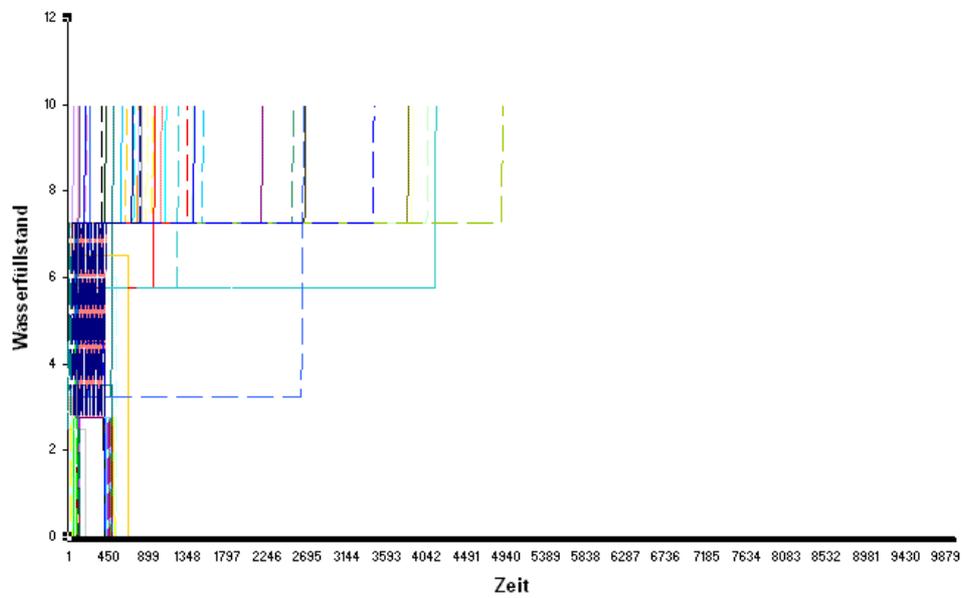


Abbildung 6.9: Verläufe des Wasserfüllstandes innerhalb von 10000 Stunden

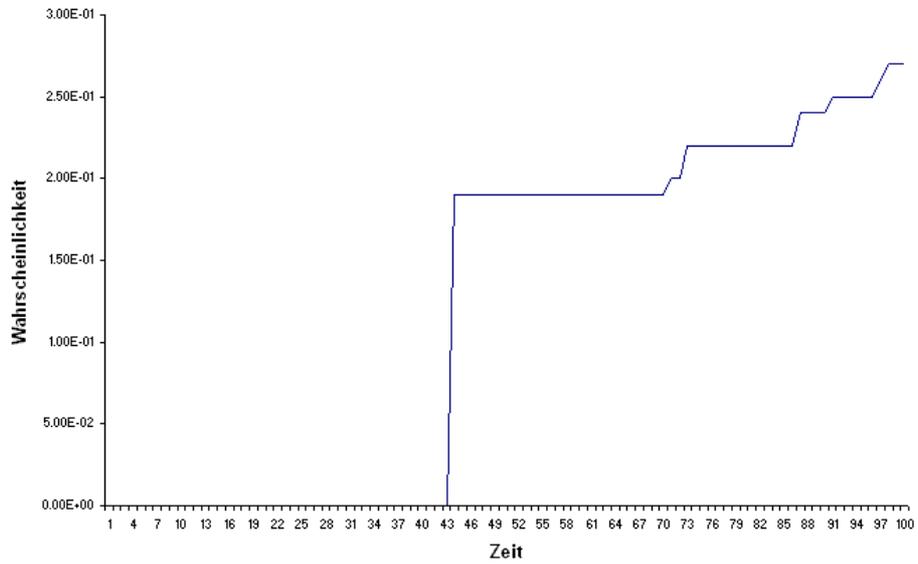


Abbildung 6.10: Eintrittswahrscheinlichkeit des "DRYOUT"-Zustandes innerhalb von 100 Stunden

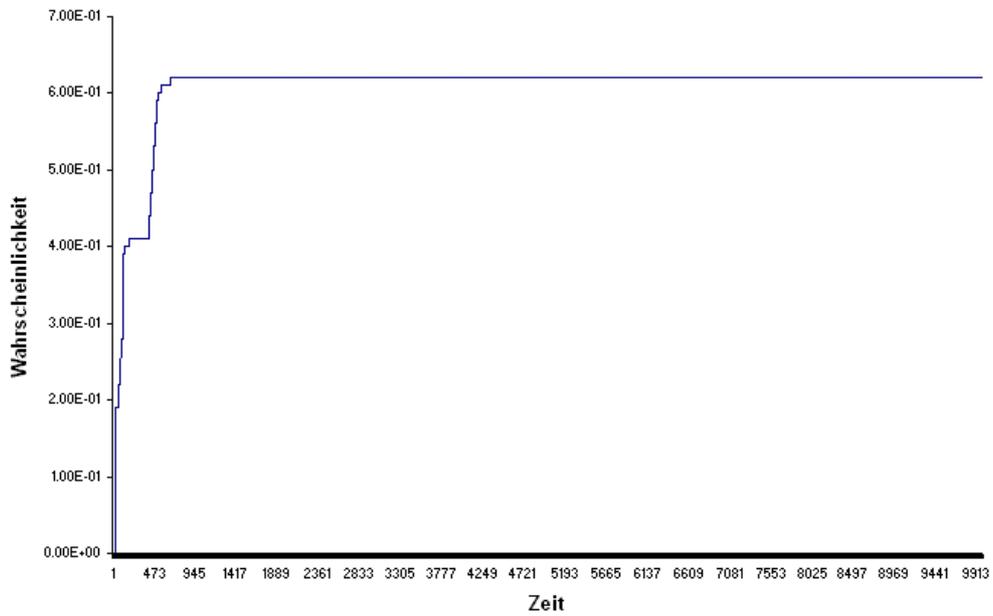


Abbildung 6.11: Eintrittswahrscheinlichkeit des "DRYOUT"-Zustandes innerhalb von 10000 Stunden

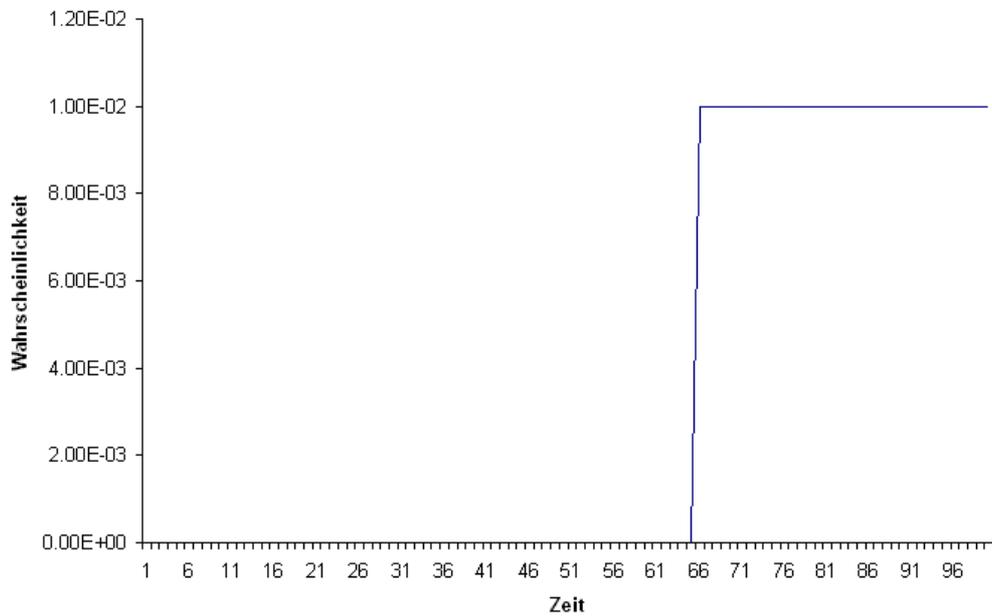


Abbildung 6.12: Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 100 Stunden

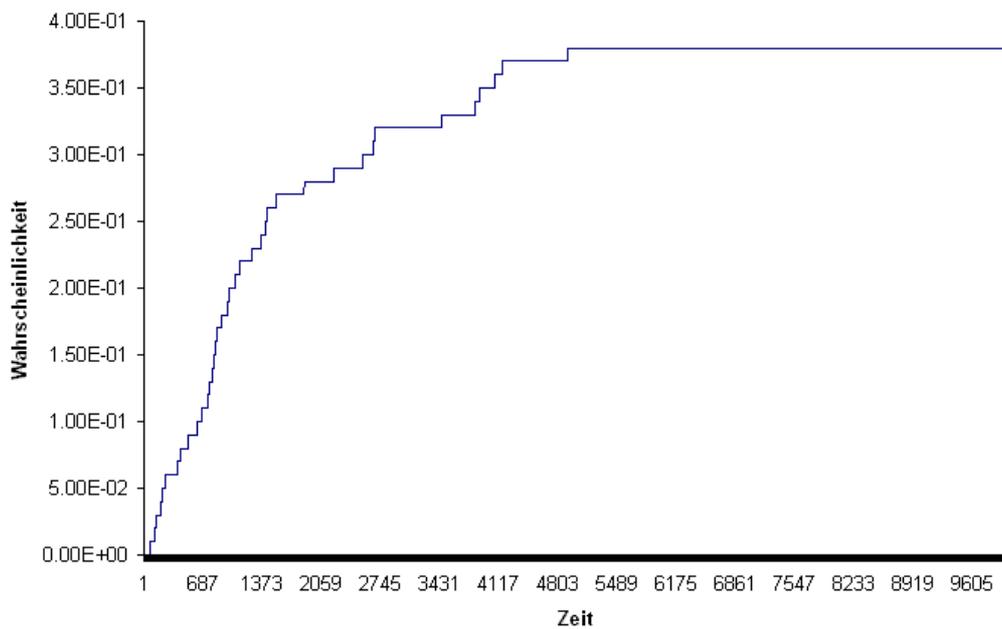


Abbildung 6.13: Eintrittswahrscheinlichkeit des "OVERFLOW"-Zustandes innerhalb 10000 Stunden

6.2.4 Quantifizierung der epistemischen Größen

Die stochastischen Größen sind die Größen, die echt zufällig variieren. Die epistemischen Größen sind die Größen, die feste Werte besitzen aber ungenau bekannt sind. Die epistemischen Größen sind die Modellparameter der stochastischen Komponentenausfälle z.B. die Ausfallrate der Wahrscheinlichkeitsdichtefunktion, die Erfolgswahrscheinlichkeit bei der Umschaltung. Die stochastischen Größen liefern die Ergebnisse der PSA. Die epistemischen Größen bestimmen, wie genau die PSA-Ergebnisse aus den stochastischen Größen berechnet werden können [NRC-RG-1174, 2002].

Um den Einfluss der epistemischen Größen auf die PSA-Ergebnisse zu demonstrieren, werden die epistemischen Größen durch die subjektive Wahrscheinlichkeitsverteilung ausgedrückt. Die Wahrscheinlichkeitsdichtefunktionen der epistemischen Größen werden wie folgt angenommen.

- Die Wahrscheinlichkeit, dass das Ventil nicht mehr geschlossen werden kann, ist die Betaverteilung mit Beta (5,95).
- Die Wahrscheinlichkeit, dass das Ventil nicht mehr geöffnet werden kann, ist die Betaverteilung mit Beta (8,92).
- Die Wahrscheinlichkeit, dass die Pumpe 1 oder Pumpe 2 nicht mehr eingeschaltet werden kann, ist die Betaverteilung mit Beta (5,95).
- Die Wahrscheinlichkeit, dass die Pumpe 1 oder Pumpe 2 nicht mehr ausgeschaltet werden kann, ist die Betaverteilung mit Beta (3,97).
- Die Ausfallrate des Ventils ist die Gammaverteilung mit Gamma (5,1000).
- Die Ausfallrate der Pumpe 1 und der Pumpe 2 sind die Gammaverteilung mit Gamma (1,1000).

6.2.4.1 Quantifizierung der epistemischen Größen

Zur Quantifizierung des Einflusses der epistemischen Größen auf die PSA-Ergebnisse wird normalerweise das Zweischleifenverfahren verwendet. Das

Zweischleifenverfahren sind zwei ineinander verschachtelte MCS. In der äußeren Schleife spielt zuerst ein Zufallsgenerator für jede epistemischen Größe einen Wert aus. In der inneren Schleife lässt sich die MCS mit n Simulationläufen hinsichtlich des generierten Wertes der epistemischen Größen durchlaufen. Nach n Simulationläufen erhält man einen Schätzwert für die Eintrittswahrscheinlichkeit des Schadenszustandes unter der Bedingung der ausgewählten epistemischen Größen. In der äußeren Schleife spielt ein Zufallsgenerator für jede epistemische Größe wieder einen neuen Wert und der neue Schätzwert für die Eintrittswahrscheinlichkeit des Schadenszustandes, unter der Bedingung der neu ausgewählten epistemischen Größen wird ermittelt. Dies wird m -mal wiederholt. Um das Zweischleifenverfahren für das „Hold-Up-Tank“-System automatisch durchzuführen, wurde das Konzept im Abschnitt 6.2.3.1 erweitert. In der Abbildung 6.14 wird das erweiterte Konzept dargestellt.

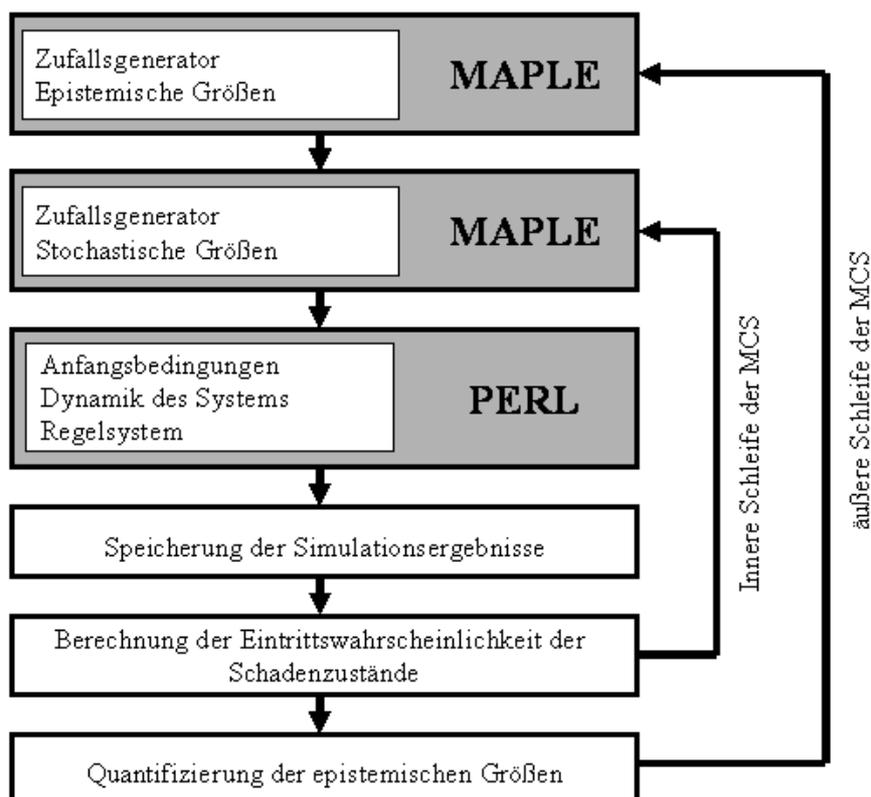


Abbildung 6.14: Analyse des "Hold-Up-Tank"-Systems mit dem Zweischleifenverfahren

Der Einfluss der epistemischen Größen auf die Ergebnisse der PSA wurden mit 100 Läufen in der inneren Schleife und 100 Läufe in der äußeren Schleife für die

Betriebszeit T=100 Stunden und T=10000 Stunden ermittelt.

In den Abbildung 6.15 bis Abbildung 6.18 werden die Einflüsse der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes und „OVERFLOW“-Zustandes innerhalb von 100 Stunden und 10000 Stunden präsentiert.

In den Abbildung 6.19 bis Abbildung 6.20 werden die Einflüsse der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes und „OVERFLOW“-Zustandes innerhalb von 100 Stunden und 10000 Stunden mit dem Maximum und Minimum, dem 25%-Quantil, dem 50%-Quantil und dem 75%-Quantil quantifiziert.

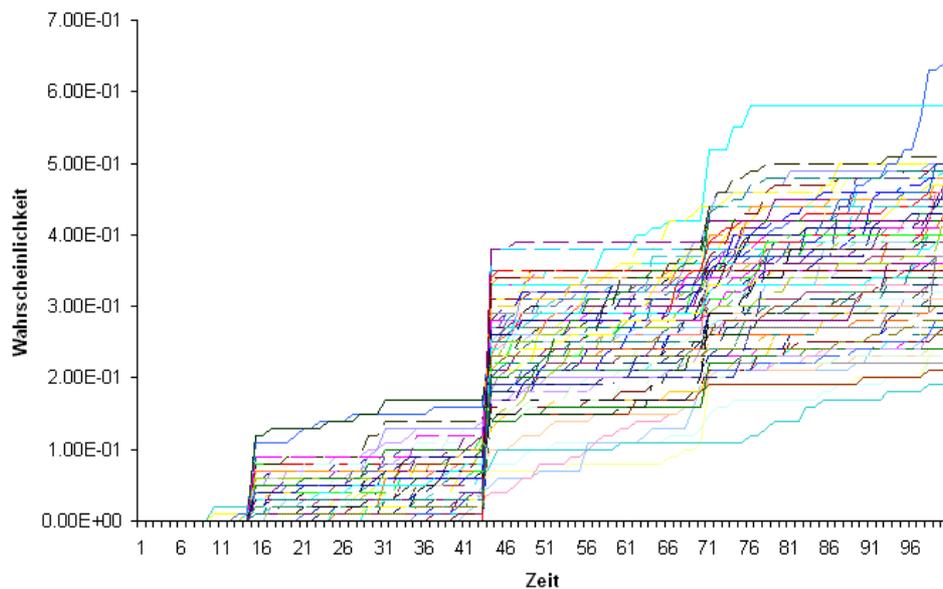


Abbildung 6.15: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des "DRYOUT"-Zustandes innerhalb von 100 Stunden

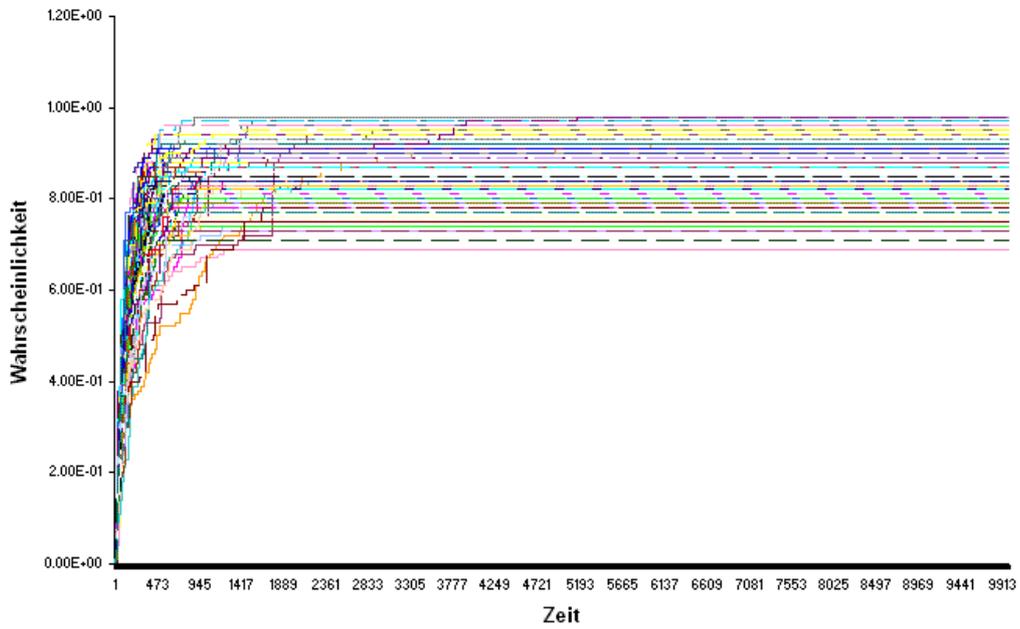


Abbildung 6.16: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des "DRYOUT"-Zustandes innerhalb von 10000 Stunden

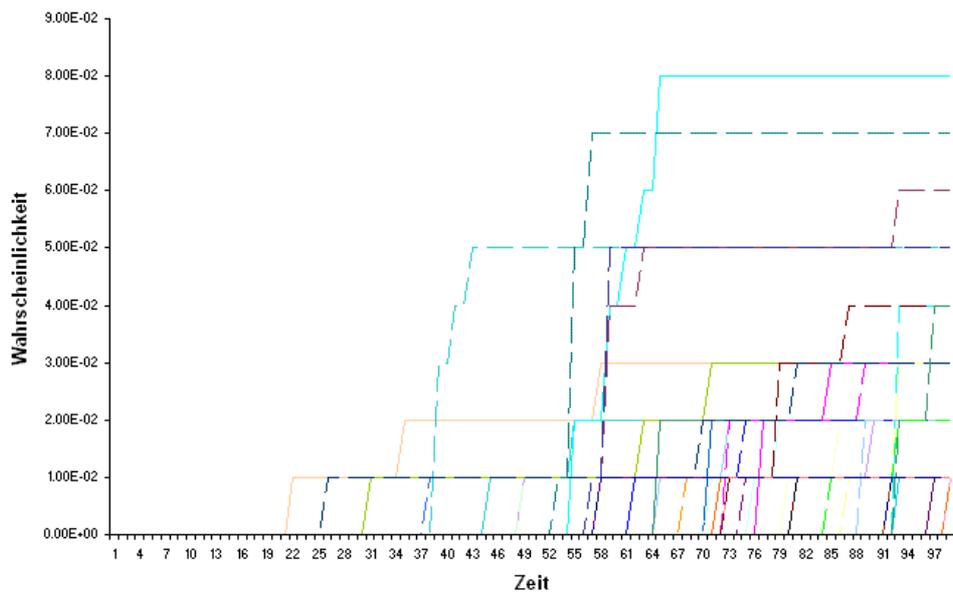


Abbildung 6.17: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 100 Stunden

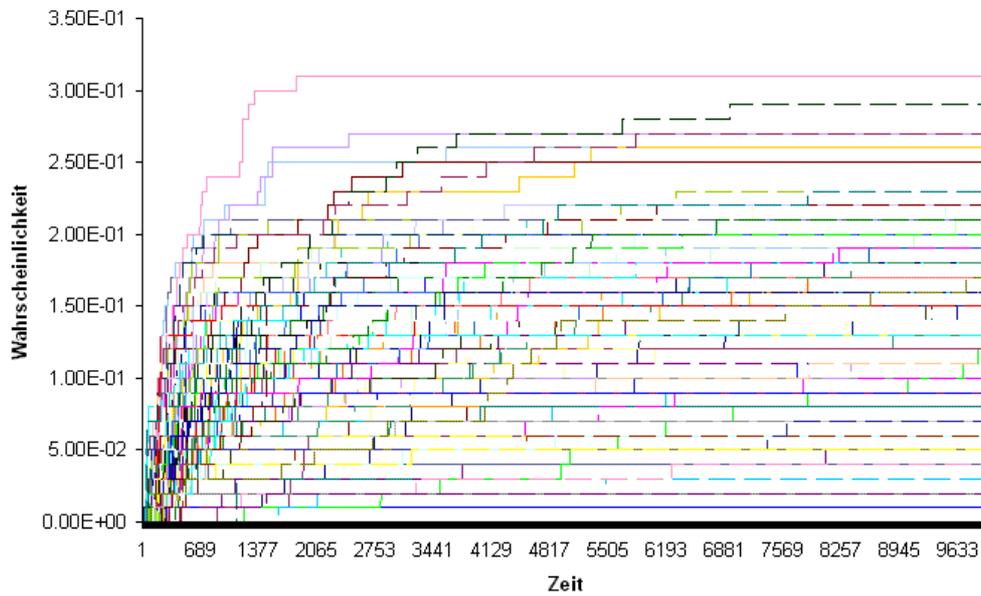


Abbildung 6.18: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 10000 Stunden

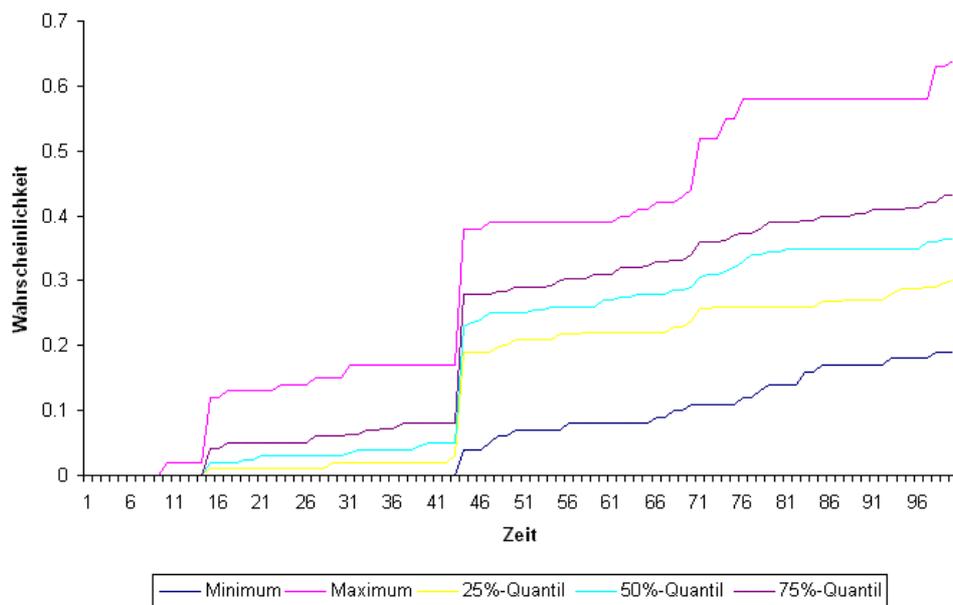


Abbildung 6.19: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes innerhalb von 100 Stunden

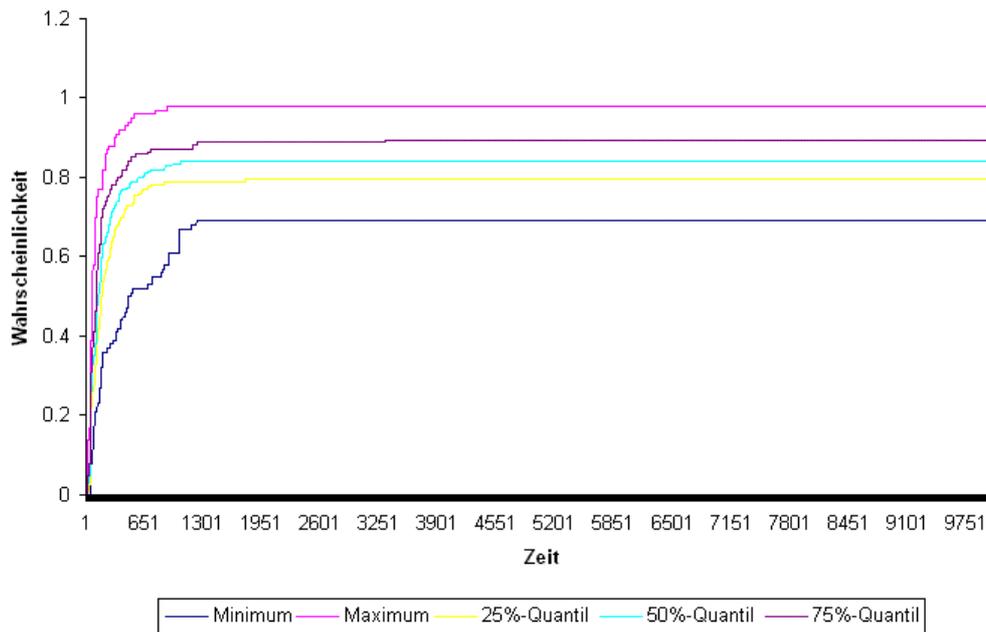


Abbildung 6.20: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes innerhalb von 10000 Stunden

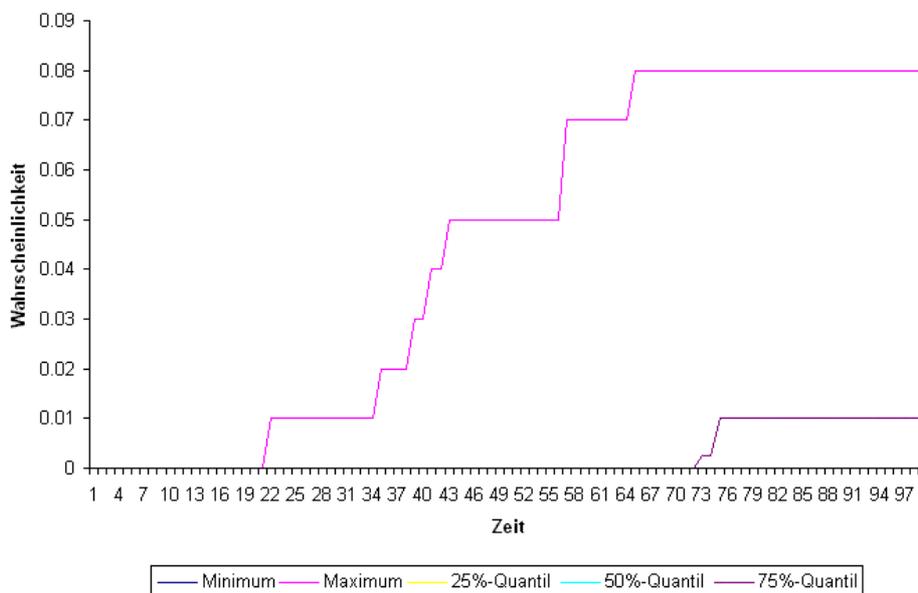


Abbildung 6.21: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 100 Stunden

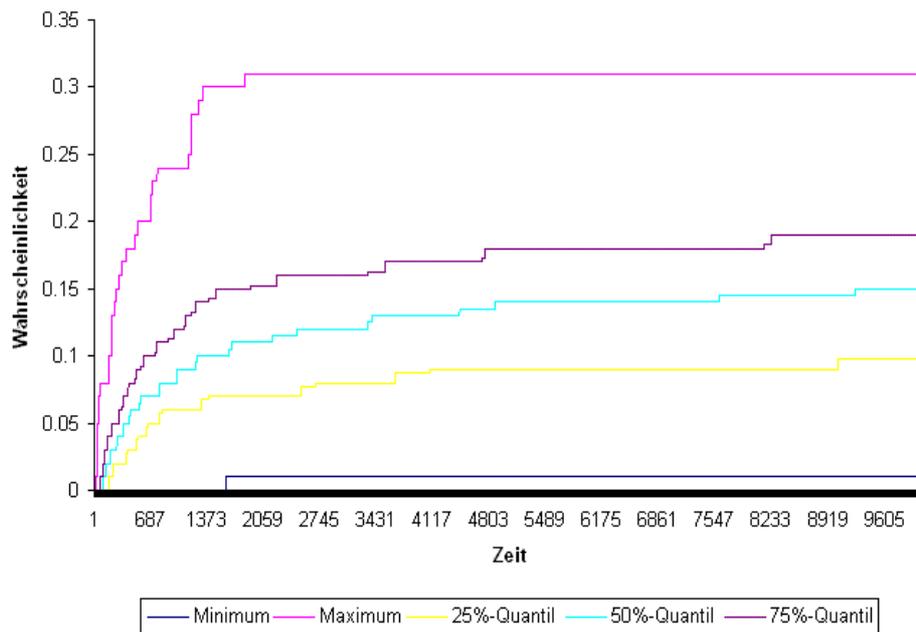


Abbildung 6.22: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 10000 Stunden

6.3 Zusammenfassung

In diesem Kapitel wurde ein dynamisches „Hold-Up-Tank“-Systems modelliert. Anhand des Beispiels wurden die zeitlichen Wechselwirkungen zwischen den zufälligen Komponentenausfällen und der Systemdynamik demonstriert.

Die Unfallabläufe wurden sowohl mit dem herkömmlichen Ereignisablaufdiagramm als auch mit der MCS modelliert und analysiert. Die MCS können die Unfallabläufe realitätsnah modellieren und die Eintrittswahrscheinlichkeit der Schadenszustände genauer ermitteln. Außerdem kann auch der Einfluss der epistemischen Größen auf die Ergebnisse der PSA durch das Zweischleifenverfahren quantifiziert werden.

Wenn die Eintrittswahrscheinlichkeit des Schadenszustandes klein ist oder das Zweischleifenverfahren durchgeführt werden muss, ist der Rechenaufwand der MCS sehr hoch. Für die realistische Anwendung müssen die rechenzeitbeschleunigenden Methoden der MCS zum Einsatz kommen.

7 Rechenzeitbeschleunigende Methoden der MCS

Da die Eintrittswahrscheinlichkeit des Schadenszustandes in den Kernkraftwerken recht gering ist, muss die rechenzeitbeschleunigende Methode der MCS zum Einsatz kommen. Zur Beschleunigung der MCS hat Herr Woltereck die „First Order Reliability Method - FORM“ [Woltereck, 2000] und hat Herr Hofer die „Monte Carlo Dynamic Event Tree“ [Hofer & Kloos -1, 2001] vorgeschlagen. Die Anwendbarkeit und die Einschränkungen beider Methoden werden zuerst untersucht.

7.1 First Order Reliability Method (FORM)

Die Zuverlässigkeitsmethode erster Ordnung (FORM) ist seit langem auf dem Gebiet der Strukturmechanik angewandt worden. FORM ist ein Annäherungsverfahren, um die Versagenswahrscheinlichkeit zu ermitteln. Die Idee der FORM ist, die eine Annäherung für die Versagenswahrscheinlichkeit aus der Gleichung (7-1) zu finden.

$$P_f = \int_{g \leq 0} f(x) dx \quad (7.1)$$

Das Gebiet $g \leq 0$ wird als das Versagensgebiet bezeichnet. Um diese Annäherung zu finden, müssen alle stochastischen Variablen mit beliebiger Verteilungsfunktion zuerst in einen Standardnormalraum transformiert werden, in dem alle stochastischen Variablen standard-normalverteilt sind. Falls die stochastischen Variablen von einander unabhängig sind, kann jede stochastische Variable separat transformiert werden

$$\begin{aligned} u_i &= \Phi^{-1}(F_{x_i}(x_i)) \\ x_i &= F_{x_i}^{-1}(\Phi(u_i)), \end{aligned} \quad (7.2)$$

wobei Φ die Verteilungsfunktion der Standardnormalverteilung ist und F die Verteilungsfunktion der beliebigen Verteilung ist. Falls die stochastischen Variablen von einander abhängig sind, kann die *Rosenblatt*-Transformation verwendet werden.

Wenn das Versagensgebiet vom Ursprung im standardnormalen Raum weiter entfernt ist, nimmt die Dichtefunktion der transformierten Standardnormalverteilung sehr schnell ab. Der Punkt u^* , der auf der Grenzfläche $g(u) = 0$ und der am nächsten am Ursprung liegt, hat die größte Eintrittswahrscheinlichkeit und den größten Beitrag für

die Eintrittswahrscheinlichkeit P_f .

Durch die Annäherung der ursprünglichen Grenzfläche durch eine Tangentialebene durch den Punkt u^* , kann die Versagenswahrscheinlichkeit in der nachfolgenden Gleichung näherungsweise abgeschätzt werden.

$$p_f = \Phi(-|u^*|) \quad (7.3)$$

Zur Lösung der Gleichung (7.3) muss zuerst der Punkt u^* gefunden werden. Dieses Problem wird als eine Optimierungsaufgabe mit Nebenbedingungen formuliert, d.h. der Punkt u^* liegt auf der Grenzfläche und am nächsten vom Ursprung.

$$\begin{array}{ll} \text{Minimiere} & |u| \\ \text{unter der Bedingung} & g(u) = 0 \end{array} \quad (7.4)$$

7.1.1 Anwendbarkeit

Um die FORM anzuwenden und den Punkt u^* leicht zu finden, müssen zuerst folgende Voraussetzungen erfüllt werden:

- Die Dichtefunktion der stochastischen Größen muss stetig sein.
- Die Grenzfunktion $g(u)$ muss stetig sein und monoton in Bezug auf jede stochastische Größe.
- Die Grenzfunktion $g(u)$ soll differenzierbar sein.

7.1.2 Schätzungsgenauigkeit

Die Schätzungsgenauigkeit der FORM hängt davon ab,

- wie genau der Optimierungsalgorithmus u^* zu finden ist und
- wie groß der Fehler durch die Annäherung der wahren Grenzfläche $g(u) = 0$

durch die Tangentialebene am Punkt u^* ist.

7.1.3 Verbesserungsmöglichkeit der FORM

Bei einer stark gekrümmten Grenzfunktion stellt die Linearisierung der Grenzfläche in der FORM eine grobe Annäherung dar. Die Genauigkeit der FORM kann verbessert werden, wenn die Grenzfläche durch eine Taylor-Reihe bis zur zweiten Ableitung angenähert wird. Dies wird unter dem Begriff Zuverlässigkeitstheorie zweiter Ordnung oder „Second Order Reliability Method - SORM“ zusammengefasst. Mit der SORM kann auch die Krümmung der Grenzfläche am Punkt u^* berücksichtigt werden. Die Voraussetzung für die Anwendung der SORM ist, dass die Grenzfunktion im Bereich des Punktes u^* zweimal differenzierbar ist.

7.1.4 Zusammenfassung

Die FORM ist ein analytisches Annäherungsverfahren zur Ermittlung der Versagenswahrscheinlichkeit. Um die FORM anzuwenden, müssen zuerst die Voraussetzungen z.B. die Stetigkeit, Monotonie und die Differenzierbarkeit der Grenzfunktion erfüllt werden. Dagegen ist die MCS allgemein anwendbar und es müssen keine weiteren Voraussetzungen erfüllt sein. Wenn alle Voraussetzungen erfüllt sind, kann die FORM den Punkt u^* mit wenigen Simulationsläufen finden und die Versagenswahrscheinlichkeit schneller abschätzen. Der Vorteil der FORM ist, dass kleine Versagenswahrscheinlichkeit, auch mit wenigen Simulationsläufen, abgeschätzt werden kann. Der Rechenaufwand der FORM ist von der Anzahl der stochastischen Größen abhängig. Wenn die Anzahl der stochastischen Größen ansteigt, steigt auch die Anzahl der Simulationsläufe an und der entsprechende Rechenaufwand wird hoch. Dagegen ist der Rechenaufwand der Monte Carlo Simulation von der Anzahl der stochastischen Größen unabhängig. Die Aussagegenauigkeit der FORM hängt sowohl vom Optimierungsalgorithmus als auch von der Annäherung der wahren Grenzfläche $g(u)$ durch die Tangentialebene am Punkt u^* ab. Daher kann die Schätzungsgenauigkeit schwer quantifiziert werden. Bei der Monte Carlo Simulation kann die Schätzungsgenauigkeit mit dem Standardverfahren z.B. 95%-Vertrauensintervall quantifiziert werden. In der Tabelle 7.1 werden die FORM und die MCS miteinander verglichen.

	FORM	MCS
Anwendungsvoraussetzungen	ja	nein
Art der Verteilungsfunktion der stochastischen Größen	beliebig	beliebig
Quantifizierung der Schätzungsgenauigkeit	nein	ja
Anwendbarkeit	Systeme mit kleiner Anzahl der stochastischen Größen	allgemeine Anwendbarkeit

Tabelle 7.1: Vergleich zwischen FORM und MCS

7.2 Monte Carlo Dynamic Event Tree (MCDET)

Der MCDET (Monte Carlo Dynamic Event Tree) ist eine Kombination des diskreten dynamischen Ereignisbaums und der MCS. Während die diskreten Übergänge z.B. Ausfall des Regelsystems in dem diskreten dynamischen Ereignisablaufdiagramm berücksichtigt werden, werden die stetigen Übergänge z.B. Ausfall des Ventils zu jedem Zeitpunkt mit der MCS betrachtet.

Der MCDET ist eine varianzreduzierende MCS, dieser erlaubt eine Reduktion der Varianz bei gleichem Stichprobenumfang, d.h. eine Erhöhung der Genauigkeit bei der Schätzung von der Eintrittswahrscheinlichkeit eines Ereignisses.

7.2.1 Grundlagen des MCDET Verfahrens

Die Idee des MCDET ist, die Erwartung des Ereignisses Y mit einer geeigneten Auswahl der stochastischen Größen zu betrachten.

$$E[Y | Z] \tag{7.5}$$

Aufgrund der bekannten Beziehung

$$E(E[Y | Z]) = EY \tag{7.6}$$

$$\text{var}(E[Y | Z]) = \text{var} Y - E(\text{var}[Y | Z]) \tag{7.7}$$

hat die Erwartung $E[Y | Z]$ denselben Erwartungswert aber eine kleine Varianz im

Vergleich mit dem Ereignis Y .

Die stochastischen Größen Z in dem MCDET sind die stetigen Übergänge. Mit der MCS werden die zufällig ausgewählten Übergänge für die stetigen Übergänge generiert. Die sämtlichen Kombinationsmöglichkeiten aus den zufällig ausgewählten Übergängen und den diskreten Übergängen werden als „Pfade“ eines Ereignisbaumes behandelt. Unter Berücksichtigung der Wahrscheinlichkeiten für alle Pfade, kann die bedingte Verteilung der Versagenswahrscheinlichkeit Y und der bedingte Erwartungswert $E[Y | Z]$ analytisch berechnet werden. Nach den n Simulationsläufen kann ein Schätzer für den Erwartungswert EY durch $(\sum[Y | Z])/n$ ermittelt werden. In der Abbildung 7.1 wird ein von n möglichen dynamischen Ereignisablaufdiagrammen aus dem MCDET für das „Hold-Up-Tank“-Systems dargestellt.

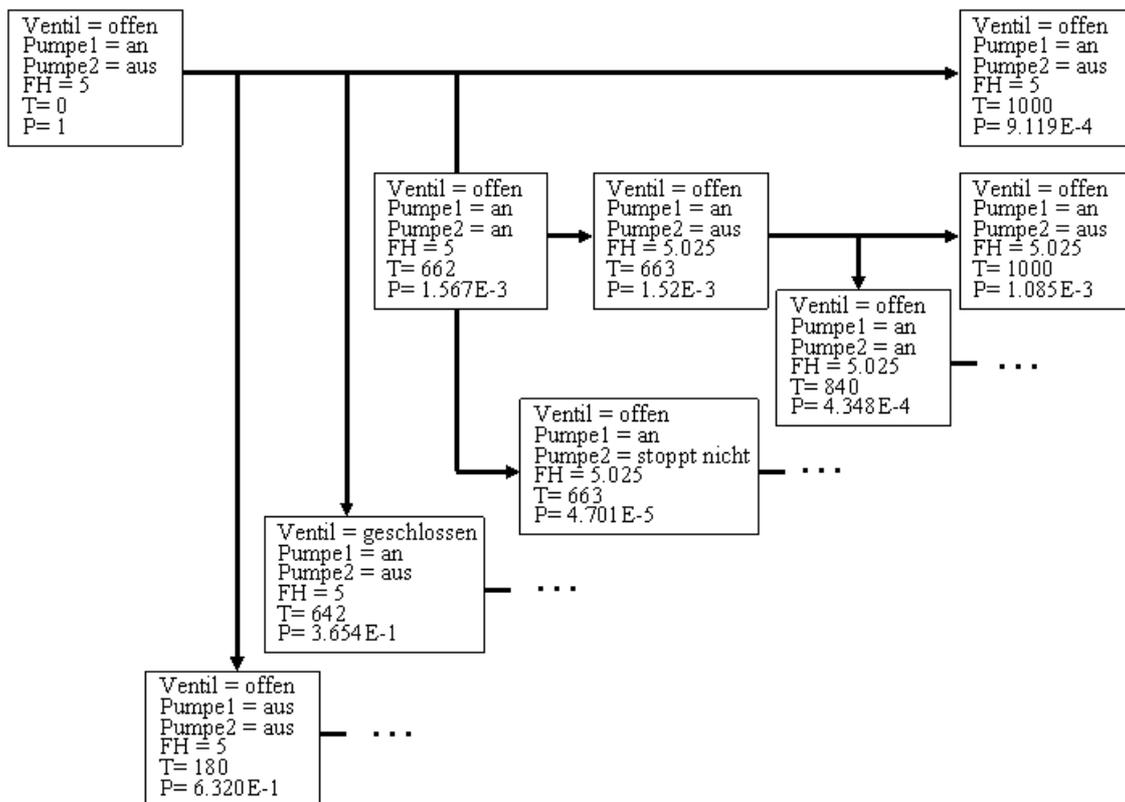


Abbildung 7.1: Beispiel – Ein dynamischer Ereignisablaufdiagramm aus dem MCDET für das "Hold-Up-Tank"-System

7.2.2 Vorteile und Nachteile des MCDET

Durch die Kombination des diskreten dynamischen Ereignisablaufdiagramms und der MCS kann der MCDET die Einschränkungen des dynamischen Ereignisablaufdiagramms bei der Behandlung der stetigen Übergänge beseitigen. Die Kombinationsmöglichkeiten der Übergänge können im Vergleich mit dem diskreten dynamischen Ereignisablaufdiagramm stark reduziert werden.

Die größte Einschränkung des MCDET liegt darin, dass der Stichprobenumfang der stetigen Übergänge klein ist. Daher können die stetigen Übergänge mit einer kleinen Eintrittswahrscheinlichkeit nicht zufällig ausgewählt und simuliert werden. Außerdem wird die ineffiziente MCS verwendet, um die Stichproben der stetigen Übergänge zu generieren.

7.2.3 Zusammenfassung

Sowohl die MCS als auch der MCDET können die stetigen Übergängen und die diskreten Übergängen behandeln. Da in jedem Lauf der MCS mehrere dynamische Rechnungen durchgeführt werden, kann der MCDET die Versagenswahrscheinlichkeit im Vergleich mit der MCS bei gleichem Stichprobenumfang genauer schätzen.

Da in jedem Lauf der MCS mehrere dynamische Rechnungen durchgeführt werden, kann der Stichprobenumfang der stetigen Übergänge aufgrund des hohen Rechenaufwandes bei dem MCDET nicht groß ausgewählt werden. Die stetigen Übergänge mit einer kleinen Eintrittswahrscheinlichkeit können nicht zufällig ausgewählt und simuliert werden. Bei der MCS wird in jedem Simulationslauf nur eine dynamische Rechnung durchgeführt, so kann der Stichprobenumfang der stetigen Übergänge groß ausgewählt werden. Der MCDET wird eingesetzt, wenn die stetigen Übergänge mit einer kleinen Eintrittswahrscheinlichkeit keinen Einfluss auf die Versagenswahrscheinlichkeit haben. Dagegen ist die MCS allgemein anwendbar. In der Tabelle 7.2 werden der MCDET und die MCS miteinander verglichen.

	MCDET	MCS
Varianzreduktion	ja	nein
Behandlung stetiger und diskreter Übergänge	ja	ja
Anwendbarkeit	Wenn die stetigen Übergänge keine Rolle spielen	allgemein anwendbar

Tabelle 7.2: Vergleich zwischen dem MCDET und der MCS

7.3 Weg zur weiteren Optimierung der MCS

Die FORM und der MCDET können die MCS hinsichtlich des Rechenaufwandes und der Genauigkeit optimieren. Die Anwendbarkeit der beiden Methoden ist allerdings eingeschränkt. Daher wird ein weiteres Optimierungsverfahren untersucht.

Ein neues eingesetztes Optimierungsverfahren der MCS soll folgende Punkte berücksichtigen.

1. Da die MCS allgemein einsetzbar und sehr gut für die dynamische PSA geeignet ist, muss die MCS weiter optimiert werden.
2. Das neue Optimierungsverfahren der MCS soll die Unfallabläufe mit einer kleinen Eintrittswahrscheinlichkeit schnell entdecken.
3. Es ist wünschenswert, das neue Optimierungsverfahren der MCS mit geringerem Rechenaufwand im Vergleich mit der klassischen MCS durchführen zu können.
4. Um das neue Optimierungsverfahren der MCS leicht anzuwenden, soll das neue Optimierungsverfahren im Vergleich mit der FORM möglichst wenige Voraussetzungen verlangen.
5. Die Schätzungsgenauigkeit soll mit dem Standardverfahren der Statistik z.B. 95%-Vertrauensintervall quantifiziert werden.
6. Da die Parameter der bisher entwickelten Methoden während der Analyse sich nicht ändern, ist es wünschenswert, ein adaptives Verfahren zu entwickeln, um die Parameter während der Analyse stetig anzupassen.

7. Da die Kenntnisse über das Systemverhalten aus den Simulationsergebnissen und der Expertenschätzung nicht in den bisher entwickelten Methoden verwendet wurden, ist es wünschenswert, die Kenntnisse im Verfahren zu nutzen, um den Rechenaufwand zu reduzieren und die Genauigkeit zu verbessern.

Unter Berücksichtigung der oben erwähnten Punkte wurde ein Optimierungsverfahren der Monte Carlo Simulation - Monte Carlo Simulation mit iterativer Anpassung der Stichprobenfunktion – für die dynamische PSA vorgeschlagen. Die MCS mit iterativer Anpassung der Stichprobenfunktion wurde auf dem Gebiet der Strukturmechanik eingesetzt und stellt eine effiziente Methode zur Berechnung der Versagenswahrscheinlichkeit dar. In dieser Arbeit wurde gezeigt, dass diese Methode auch für die Sicherheitsbewertung der Unfallabläufe in Kernkraftwerken Vorteile bietet.

7.4 MCS

Die Unfallabläufe in Kernkraftwerken können von einem Anfangszustand entweder den Schadenzustand Endzustand $Z_{Versagen}$ oder den Endzustand Z_{ok} erreichen. Das Ziel der dynamischen PSA ist die Eintrittswahrscheinlichkeit des Endzustandes $Z_{Versagen}$ zu ermitteln. Um festzulegen, ob die simulierten Unfallabläufe den Endzustand Z_{ok} oder den Endzustand $Z_{Versagen}$ erreichen, betrachtet man eine Prozessgröße s z.B. die Temperatur oder der Druck. Wenn der Wert der Prozessgröße s eine Schranke der Prozessgröße w überschreitet, tritt der Endzustand $Z_{Versagen}$ ein. Das Versagensgebiet kann durch eine Grenzfunktion g

$$g = w - s \tag{7.8}$$

definiert werden, wenn die Grenzfunktion klein oder gleich Null ist. Die Versagenswahrscheinlichkeit P_f , dass der Endzustand $Z_{Versagen}$ eintritt, wird wie folgt berechnet

$$\begin{aligned} P_f &= \int_{g \leq 0} f(x) \\ &= \int_X I(g(x) \leq 0) f_x(x) dx \end{aligned} \tag{7.9}$$

, wobei $I(g(x))$ die Indikatorfunktion und $f(x)$ die zusammengesetzte Dichtefunktion der stochastischen Größen ist. Je nachdem ob der Ausdruck in der Klammer wahr oder falsch ist, kann die Indikatorfunktion den Wert entweder 1 oder 0 annehmen.

$$I(g(x)) = \begin{cases} 1, & \text{if } x \in [g(x) \leq 0] \\ 0, & \text{if } x \in [g(x) > 0] \end{cases} \quad (7.10)$$

Die Versagenswahrscheinlichkeit aus der Gleichung 7.9 kann leider nicht analytisch gelöst werden, weil die Grenzfunktion g eine nicht analytische Funktion ist.

Mit Hilfe der MCS kann die Versagenswahrscheinlichkeit abgeschätzt werden. In jedem Lauf der MCS werden alle stochastischen Größen zufällig variiert. Die Zufallszahl für jede stochastische Größe wird nach der Wahrscheinlichkeitsdichtefunktion $f(x_i)$ durch einen Zufallsgenerator generiert. Die generierten Zufallszahlen werden an ein Systemmodell weitergegeben. Die Unfallabläufe und die zeitliche Entwicklung der Prozessgröße werden mit Hilfe des Systemmodells modelliert. Nach N Simulationsläufen wird die Anzahl der Simulationsläufen n , in den der Prozessgröße s die Schranke w überschreitet, gezählt. Damit ergibt sich die Versagenswahrscheinlichkeit aus der folgenden Gleichung

$$\bar{P}_f = \frac{1}{N} \sum_{i=1}^N I(g(x)) = \frac{n}{N}. \quad (7.11)$$

Die Varianz dieser Schätzung ergibt sich aus der folgenden Gleichung.

$$\sigma^2[\bar{P}_f] = E[(\bar{P}_f - E(\bar{P}_f))^2] = \frac{P_f - P_f^2}{N} = \frac{P_f}{N} \quad (7.12)$$

Man kann auch das 95%-Konfidenzintervall konstruieren, in dem die wahre Versagenswahrscheinlichkeit P_f mit 95% Wahrscheinlichkeit liegt.

$$\left[\bar{P}_f - 1.96\sigma[\bar{P}_f], \dots, \bar{P}_f + 1.96\sigma[\bar{P}_f] \right] \quad (7.13)$$

Die Varianz dieser Schätzung ist proportional zur N^{-1} . Wenn die Eintrittswahrscheinlichkeit des Schadenszustandes klein ist, ist eine große Anzahl von

Simulationen bei einer bestimmten Genauigkeitsanforderung erforderlich.

7.4.1 MCS mit gewichteter Stichprobenfunktion (MCSGS)

Da die Eintrittswahrscheinlichkeit des Schadenszustandes in Kernkraftwerken klein ist, werden, die meist zufällig generierten Stichproben der stochastischen Größen, außerhalb des Versagensgebietes liegen und können nicht zur Ermittlung der Versagenswahrscheinlichkeit P_f beitragen. Um die MCS zu beeinflussen, dass mehr Stichproben aus dem Versagensgebiet erzeugt werden, wird die gewichtete Stichprobenfunktion eingesetzt.

Die Versagenswahrscheinlichkeit nach der Gleichung (7.9) kann wie folgt erweitert werden.

$$\begin{aligned}
 p_f &= \int_{g(x) \leq 0} f_x(x) dx \\
 &= \int_x I(g(x)) f_x(x) dx \\
 &= \int_x I(g(x)) \frac{f_x(x)}{h_x(x)} h_x(x) dx \\
 &= E\left[I(g(x)) \frac{f_x(x)}{h_x(x)} \right]
 \end{aligned} \tag{7.14}$$

$h_x(x)$ ist die genannte gewichtete Stichprobenfunktion. Die Generierung der Stichproben erfolgt nicht direkt aus der ursprünglichen Stichprobenfunktion $f_x(x)$, sondern aus der gewichteten Stichprobenfunktion $h_x(x)$. Durch gezielte Auswahl der gewichteten Stichprobenfunktion können mehr Stichproben im Versagensgebiet erzeugt werden.

Nach N Simulationsläufen kann die Versagenswahrscheinlichkeit P_f nach der folgender Gleichung abgeschätzt werden.

$$\bar{P}_f = \frac{1}{N} \sum_{i=1}^N I(g(x_i)) \frac{f_x(x_i)}{h_x(x_i)} \approx P_f \tag{7.15}$$

Die Varianz dieser Schätzung ergibt sich aus der folgenden Gleichung.

$$\begin{aligned}\sigma^2[\bar{P}_f] &= E[(\bar{P}_f - E(\bar{P}_f))^2] \\ &= E\left[\left(\frac{1}{N} \sum_{i=1}^N I(g(x_i)) \frac{f_x(x_i)}{h_x(x_i)} - P_f\right)^2\right]\end{aligned}\quad (7.16)$$

Die Varianz der Schätzung nach der Gleichung (7.16) kann durch die Wahl der gewichteten Stichprobenfunktion reduziert werden. Wenn man die optimal gewichtete Stichprobenfunktion

$$\begin{aligned}h_x(x) &= \frac{I(g(x))f_x(x)}{P_f} \\ &= \begin{cases} \frac{1}{P_f} f_x(x) & \text{if } g(x) \leq 0 \\ 0 & \text{if } g(x) > 0. \end{cases}\end{aligned}\quad (7.17)$$

auswählt, beträgt die Varianz der Schätzung Null, d.h. die Versagenswahrscheinlichkeit wird perfekt abgeschätzt.

In der Praxis kann man die optimal gewichtete Stichprobenfunktion nicht direkt auswählen, da die wahre Versagenswahrscheinlichkeit P_f von Anfang an unbekannt ist. Die optimal gewichtete Stichprobenfunktion kann sukzessive angepasst werden.

7.4.2 MCS mit iterativer Anpassung der Stichprobenfunktion (MCSIAS)

Die Idee der MCSIAS ist die Stichprobenfunktion durch zusätzliche Informationen über das spezifische Problem gezielt so anzupassen, dass die Anzahl der Stichproben im Versagensgebiet steigt. Weitere Verbesserungen sind möglich, wenn die Anpassung der Stichprobenfunktionen in mehreren Schritten iterativ erfolgt, indem auch die Informationen aus den vorherigen Iterationen für die Anpassung genutzt werden, so dass die Stichprobenfunktionen von Iteration zu Iteration verbessert werden.

In dieser Arbeit wird das Grundprinzip der MCSIAS nach dem Konzept von Herrn Bucher [Bucher, 1987] erläutert. Die gewichtete Stichprobenfunktion kann von Iteration zu Iteration hinsichtlich des Mittelwertes und der Varianz angepasst werden.

$$\begin{aligned}
E_h[X] &= E_f[X \mid g(x) \leq 0] \\
E_h[(X - E_h[X])^2] &= E_f[(X - E_f[X])^2 \mid g(x) \leq 0]
\end{aligned}
\tag{7.18}$$

Der Mittelwert der gewichteten Stichprobenfunktion ist gleich dem Mittelwert der Stichproben im Versagensgebiet. Die Varianz der gewichteten Stichprobenfunktion ist gleich der Varianz der Stichproben im Versagensgebiet.

In der ersten Iteration der MCSIAS können der Mittelwert und die Varianz der gewichteten Stichprobenfunktion durch die Expertenschätzung gegeben werden. Wenn der Mittelwert der gewichteten Stichprobenfunktion in der Nähe der Grenzfläche liegt, kann die MCSIAS das Versagensgebiet schnell identifizieren. Nach der ersten Iteration der MCSIAS können der Mittelwert und die Varianz der gewichteten Stichprobenfunktion nach der Gleichung (7.18) berechnet werden. Die gewichtete Stichprobenfunktion wird dann angepasst und die neuen Stichproben werden generiert. Die Simulation wiederholt sich.

8 Modellierung der DWR-Transiente und Analyse mit der MCS und MCSIAS

Das Potential des Verfahrens MCSIAS wurde für eine Transiente nach Turbinenschnellabschaltung mit blockierter Umleitstation in einem DWR untersucht. Zur Analyse dieser Transiente wurden sowohl die gewöhnliche MCS als auch die MCSIAS durchgeführt.

8.1 Beschreibung der Transiente

In einem Druckwasserreaktor werden alle Turbinen mit blockierter Umleitstation abgeschaltet. Infolgedessen steigt der primäre Druck. Als der primäre Druck den Wert 16.1 MPa. erreicht, sprüht der Druckhalter. Aufgrund der Turbinenschnellabschaltung wird der Reaktor nach ein paar Sekunden abgeschaltet. Der primäre Druck sinkt ab. Wenn der primäre Druck den Ansprechdruck des Druckhalterventils erreicht, öffnet das Druckhalterventil. Wenn der primäre Druck den Ansprechdruck des Druckhalterventils nicht erreicht, öffnet das Druckhalterventil nicht. Mit Hilfe der dynamischen PSA soll die Öffnungswahrscheinlichkeit des Druckhalterventils ermittelt werden.

8.2 Stochastische Größen

Während der Transiente haben zwei stochastische Größen den Einfluss auf das Öffnungsverhalten des Druckhalterventils. Die erste stochastische Größe ist die Enthalpie des Sprühwassers. Wenn das letzte Sprühwasser lang zurückliegt, kühlt sich das Wasser in der Leitung ab. Das kalte Wasser führt zu einer stärkeren Druckreduktion. Wenn der Einsatz des letzten Sprühwassers kurz zurückliegt, befindet sich das heiße Wasser in der Leitung. Das führt zu einer langsamen Druckreduktion. Die zweite stochastische Größe ist die anfängliche Reaktorleistung, die auch den Einfluss auf den primären Druckanstieg hat.

Parameter	Verteilung	Min.	Max.
Sprühenthalpie	Uniform	0.5	1.0
Anfangsleistung	Uniform	0.9	1.06

Tabelle 8.1 Verteilungsfunktion der stochastischen Größen

8.3 Systemcode ATHLET

Der Systemcode ATHLET (Analyse der Thermohydraulik, Lecks und Transienten) wird von der Gesellschaft für Anlagen- und Reaktorsicherheit im Auftrag des Bundesministeriums für Bildung, Wissenschaft, Forschung und Technologie (BMBF) entwickelt [Lerchl & Austregesilo,2000]. Der Systemcode ATHLET wird angewendet, um die thermohydraulischen Auslegungstörfälle von Transienten und die Kühlmittelverluststörfälle in Druckwasserreaktoren und Siedewasserreaktoren zu simulieren. Die Abbildung 8.1 zeigt einen Simulator aus dem Systemcode ATHLET.

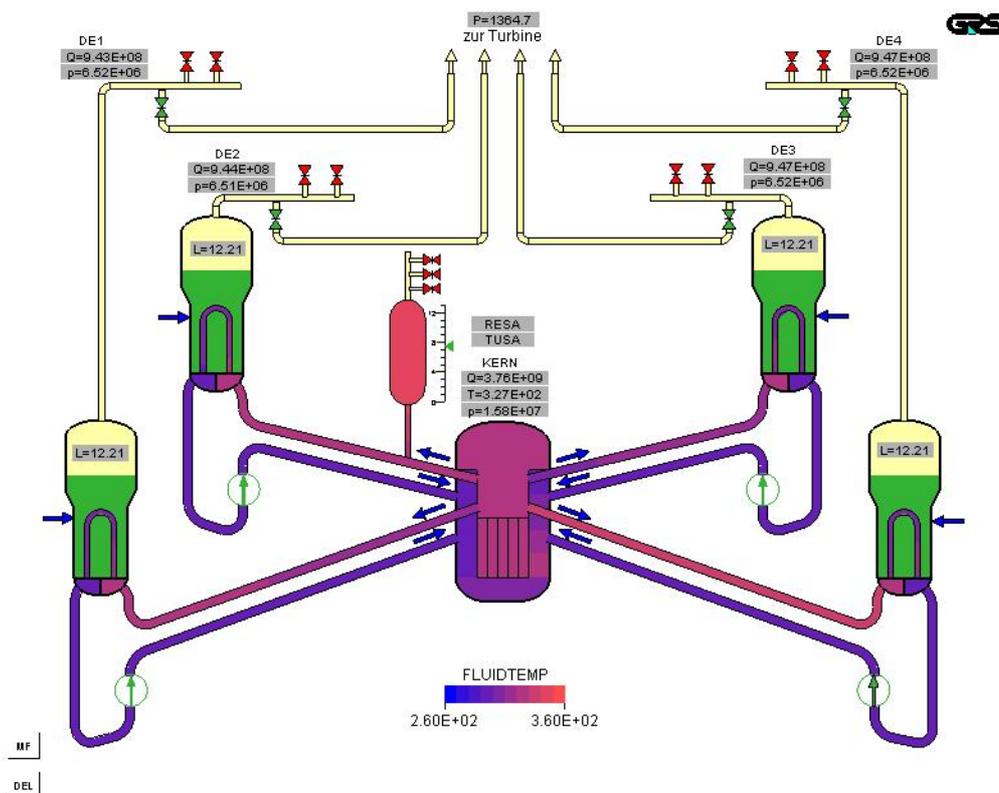


Abbildung 8.1: Beispiel - ATHLET Simulator

Der Systemcode ATHLET besteht aus mehreren Modulen, in denen die verschiedenen Phänomene in einem Druckwasserreaktor oder Siedewasserreaktor behandelt werden. Die Module sind die Thermofluiddynamik, das Wärmeleitmodell, die Neutronenkinetik, die Regelung, die Steuerung und die Spezialmodelle für Ventile und Pumpen.

Die Thermofluiddynamik hat eine zentrale Bedeutung im ATHLET Systemcode. Dieser Modul basiert auf dem Lumped-Parameter-Konzept und kombiniert mit einem eindimensionalen Ausströmmodell. Der Druck, die Enthalpie, der Dampfgehalt und der

Massenstrom können mit dem gewöhnlichen Differentialgleichungssystem, das auf den Erhaltungssätzen für die Dampf- und Wassermasse, die Gemischenergie und den Gemischimpuls basiert, berechnet werden. Zur Integration dieses Differentialgleichungssystems wird ein Integrationsverfahren mit Zeitschritt- und Konvergenzkontrolle eingesetzt. Mit Wärmeleitmodell können die Wärmeleitung in Strukturen und die Wärmeerzeugung in elektrisch und nuklear beheizten Stabbündeln nachgebildet werden. Mit dem Neutronenkinetik-Modell kann die nukleare Wärmeerzeugung simuliert werden. Das Regel- und Steuerungssystemen werden mit dem GCSM-Modell nachgebildet. Die Ventile und die Pumpen werden mit den Spezialmodellen nachgebildet.

8.4 Statistischer Code SUSAN

Der statistische Code SUSAN (Software-System für Größe- und Sensitivitäts-Analysen) wurde von der Gesellschaft für Anlagen- und Reaktorsicherheit entwickelt [Hofer & Kloos 1999]. Mit dem statistischen Code SUSAN kann der Einfluss von stochastischen Größen auf die Simulationsergebnisse untersucht werden und den Experten bei der Quantifizierung der Größen unterstützen.

Dieses Software-System bietet eine große Auswahl von Verteilungsfunktionen für unsichere Parameter. Aus den gegebenen Verteilungsparametern kann die Verteilung ermittelt und graphisch dargestellt werden. Mit dem Zufallsgenerator im SUSAN können die Stichproben der unsicheren Parameter nach der ausgewählten Verteilungsfunktion heraus gezogen werden.

8.5 Anwendung der MCS auf die DWR-Transiente

Da die MCS allgemein anwendbar ist, wird die Öffnungswahrscheinlichkeit des Druckhalterventils mit der MCS als die Referenzlösung ermittelt. Um die MCS automatisch durchzuführen, wird ein MCS-System in der Abbildung 8.2 implementiert.

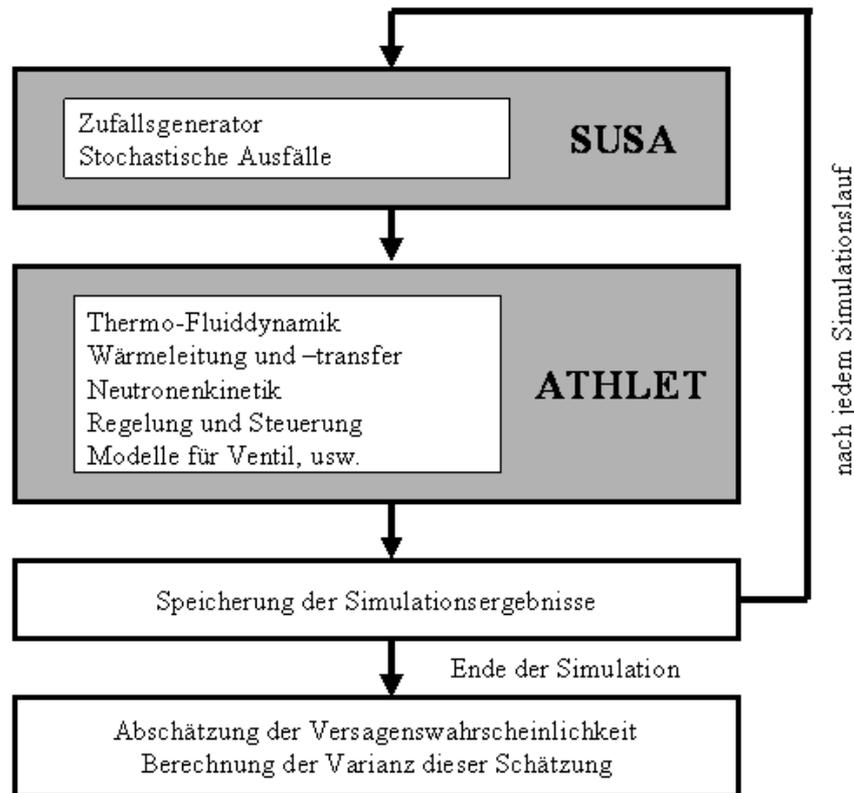


Abbildung 8.2: Aufbau des MCS-Systems

Zu Beginn der MCS müssen die Dichtefunktionen der stochastischen Größen im Code SUSA definiert werden, dann generiert der Code SUSA die Zufallszahlen gemäß der Dichtefunktion der stochastischen Größen. Die generierten Zufallszahlen werden an den entsprechenden Stellen in den ATHLET-Eingabedatensatz übernommen und die MCS startet. Nach jedem Simulationslauf werden die Simulationsergebnisse gespeichert. Nach der Bewertung der Simulationsergebnisse kann die Versagenswahrscheinlichkeit und die Varianz dieser Schätzung ermittelt werden.

8.6 Ergebnisse aus der MCS

In der MCS werden die Sprühenthalpie eine Gleichverteilung zwischen 50% und 100% und die Anfangsreaktorleistung zwischen 90% und 106% angenommen. Die Transiente wird mit ATHLET simuliert und die statistische Analyse erfolgt mit SUSA.

Die MCS mit drei verschiedenen Stichprobenumfängen ($N = 1000, 3000, 5000$) wurde durchgeführt. Der rote Kreis bedeutet, dass das Druckhalterventil nicht öffnet. Das

blaue Kreuz bedeutet, dass das Druckhalterventil öffnet. Um die Ergebnisse mit der MCSIAS leicht zu vergleichen, werden in den Abbildung 8.3 bis Abbildung 8.5 die MCS mit 1000 Läufen, 3000 Läufen und 5000 Läufen sowohl im ursprünglichen Raum als auch im standardnormalen dargestellt.

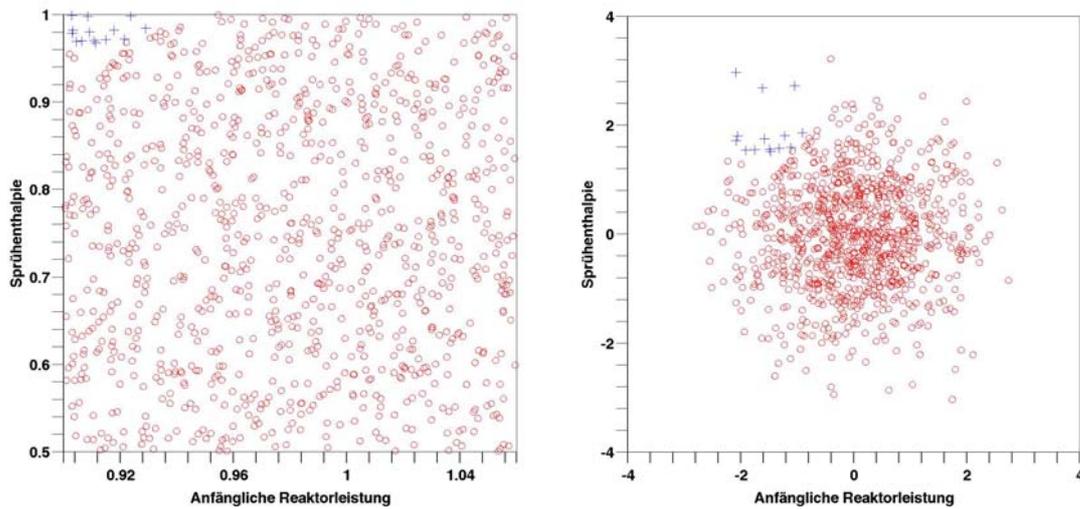


Abbildung 8.3: MCS mit 1000 Läufen im ursprünglichen und standardnormalen Raum

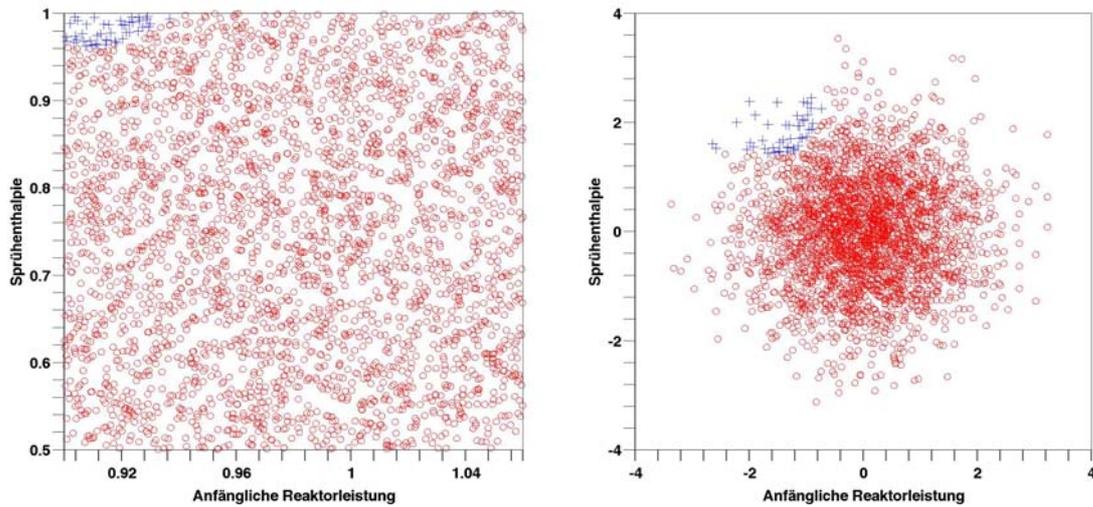


Abbildung 8.4: MCS mit 3000 Läufen im ursprünglichen und standardnormalen Raum

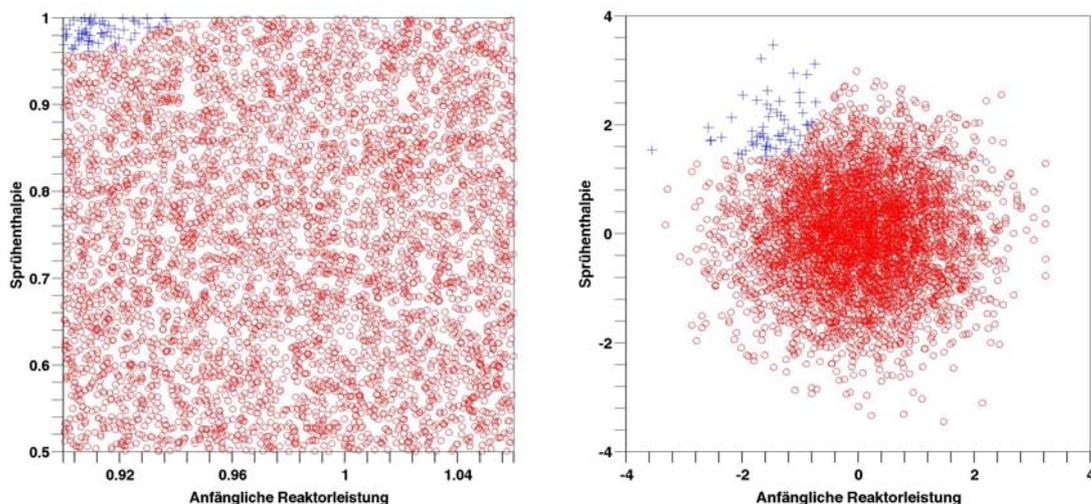


Abbildung 8.5: MCS mit 5000 Läufen im ursprünglichen und standardnormalen Raum

Die Ergebnisse zeigen, dass die MCS ineffizient ist. Viele Stichproben liegen außerhalb des Versagensgebietes. Nach 1000, 3000 und 5000 Läufen öffnet das Druckhalterventil in 14, 45 und 60 Läufen. Damit ergibt sich die Eintrittswahrscheinlichkeit des Druckhalter-Ventils \hat{P}_f gemäß der Gleichung (7.11) 0.014, 0.015, 0.012. Nach 1000, 3000, 5000 Läufen reduziert sich die Varianz der geschätzte Öffnungswahrscheinlichkeit gemäß der Gleichung (7.12) von 1.18E-02 auf 1.54E-03. In der Tabelle 8.2 werden alle Ergebnisse aus der MCS zusammengefasst.

Fall	Methode	P_f	$\sigma[P_f]$	95%-Konfidenzintervall	CPU-Zeit (s)
1	MCS / 1000 Läufe	1.40E-02	3.72E-03	6.71E-03-2.13E-02	17033
2	MCS / 3000 Läufe	1.50E-02	2.22E-03	1.07E-02-1.93E-02	51187
3	MCS / 5000 Läufe	1.20E-02	1.54E-03	9.00E-03-1.50E-02	85196

Tabelle 8.2: Ergebnisse aus MCS

8.7 Anwendung der MCSIAS auf die DWR-Transiente

Um das Potential der MCSIAS zu zeigen, wird auch die Öffnungswahrscheinlichkeit des Druckhalter-Ventils mit der MCSIAS ermittelt. Da es sich um zwei voneinander unabhängige stochastische Größen handelt, werden die Schätzfunktionen für zwei unabhängige stochastische Größen erweitert.

8.7.1 Definition der Schätzfunktionen

Aus den Gleichungen (7.15-7.16) können der Schätzwert der Öffnungswahrscheinlichkeit und die Varianz dieser Schätzung berechnet werden. Um die MCSIAS zu implementieren, brauchen weitere Schätzfunktionen.

- Erwartungswert von der stochastischen Größe x_j im Versagensgebiet

Nach der Gleichung (7.18) ist der Erwartungswert der gewichteten Stichprobenfunktion $h_X(x)$ ist gleich dem Erwartungswert der Stichprobenfunktion $f_X(x)$ im Versagensgebiet. Die j-te Komponente des Erwartungswerts X ergibt sich aus der folgenden Gleichung.

$$\begin{aligned}\mu_{x_j} &= \frac{1}{P_f} \int_{g \leq 0} x_j \cdot f_X(x) \cdot dx \\ &= \frac{1}{P_f} \int_X x_j \cdot I(x) \cdot \frac{f_X(x)}{h_X(x)} \cdot h_X(x) \cdot dx\end{aligned}\quad (8.1)$$

Die Schätzfunktion für den Erwartungswert von der stochastischen Größe x_j im Versagensgebiet bei einem Stichprobenumfang von N wird mit der folgender Gleichung berechnet.

$$\begin{aligned}\bar{\mu}_{x_j} &= \frac{1}{P_f} \cdot \frac{1}{N} \cdot \sum_{i=1}^N x_{j,i} \cdot I[x_i] \cdot \frac{f_X(x_i)}{h_X(x_i)} \\ &= \frac{\sum_{i=1}^N x_{j,i} \cdot I[x_i] \cdot \frac{f_X(x_i)}{h_X(x_i)}}{\sum_{i=1}^N I[x_i] \cdot \frac{f_X(x_i)}{h_X(x_i)}}\end{aligned}\quad (8.2)$$

$x_{j,i}$ ist die i-te Stichprobe der stochastischen Größe x_j

- Varianz von stochastischer Größe x_j im Versagensgebiet

Nach der Gleichung (7.18) ist die Varianz der gewichteten Stichprobenfunktion $h_X(x)$ ist gleich der Varianz der Stichprobenfunktion $f_X(x)$ im Versagensgebiet.

Die j-te Komponente der Varianz ergibt sich aus der folgenden Gleichung.

$$\begin{aligned}
 \sigma^2_{x_j} &= \frac{1}{P_f} \int_{g \leq 0} (x_j - \mu_{x_j})^2 \cdot f_X(x) \cdot dx \\
 &= \frac{1}{P_f} \int_{g \leq 0} x_j^2 \cdot f_X(x) \cdot dx - \mu_{x_j}^2 \\
 &= \frac{1}{P_f} \int_X x_j^2 \cdot I[x] \cdot \frac{f_X(x)}{h_X(x)} h_X(x) \cdot dx - \mu_{x_j}^2
 \end{aligned} \tag{8.3}$$

Die Schätzfunktion für die Varianz von der stochastischen Größe x_j im Versagensgebiet bei einem Stichprobenumfang von N wird mit der folgenden Gleichung berechnet.

$$\begin{aligned}
 \bar{\sigma}^2_{x_j} &= \frac{1}{P_f} \cdot \frac{1}{N} \cdot \sum_{i=1}^N x_{j,i}^2 \cdot I[x_i] \cdot \frac{f_X(x_i)}{h_X(x_i)} - \bar{\mu}_{x_j}^2 \\
 &= \frac{\sum_{i=1}^N x_{j,i}^2 \cdot I[x_i] \cdot \frac{f_X(x_i)}{h_X(x_i)}}{\sum_{i=1}^N I[x_i] \cdot \frac{f_X(x_i)}{h_X(x_i)}} - \bar{\mu}_{x_j}^2
 \end{aligned} \tag{8.4}$$

8.8 Realisierung des MCSIAS Systems

Um die MCSISA automatisch durchzuführen, wird ein MCSIAS-System implementiert. Das MCSIAS-System wird in der Abbildung 2.1 dargestellt.

Da die Simulation von Stichproben aus Normalverteilung besonders einfach ist, werden die Stichprobenfunktionen der stochastischen Größen in den standardnormalen Raum transformiert, in dem die Stichprobenfunktionen weiter angepasst werden. Zu Beginn der ersten Iteration der Simulation werden zuerst der Mittelwert und die Varianz der Normalverteilung jeder stochastische Größe im Code SUSA definiert, dann generiert der statistische Code SUSA die Zufallszahl gemäß der definierten Normalverteilung. Die generierten Zufallszahlen werden zuerst in den ursprünglichen Raum gemäß den Gleichungen (7.2) transformiert und an den entsprechenden Stellen in den ATHLET-Eingabedatensatz eingetragen. Nach jeder Iteration werden alle Simulationsergebnisse gespeichert und der Mittelwert und die Varianz der Stichprobenfunktion für die nächste Iteration berechnet. Der daraus resultierende Mittelwert und die Varianz werden

verwendet und die Stichprobenfunktionen im Code SUSAS werden neu definiert. Die Iteration wiederholt sich. Die Erfahrung auf dem Gebiet der Strukturmechanik hat gezeigt, dass die Ergebnisse nach 3 Iterationen schon sehr genau sind.

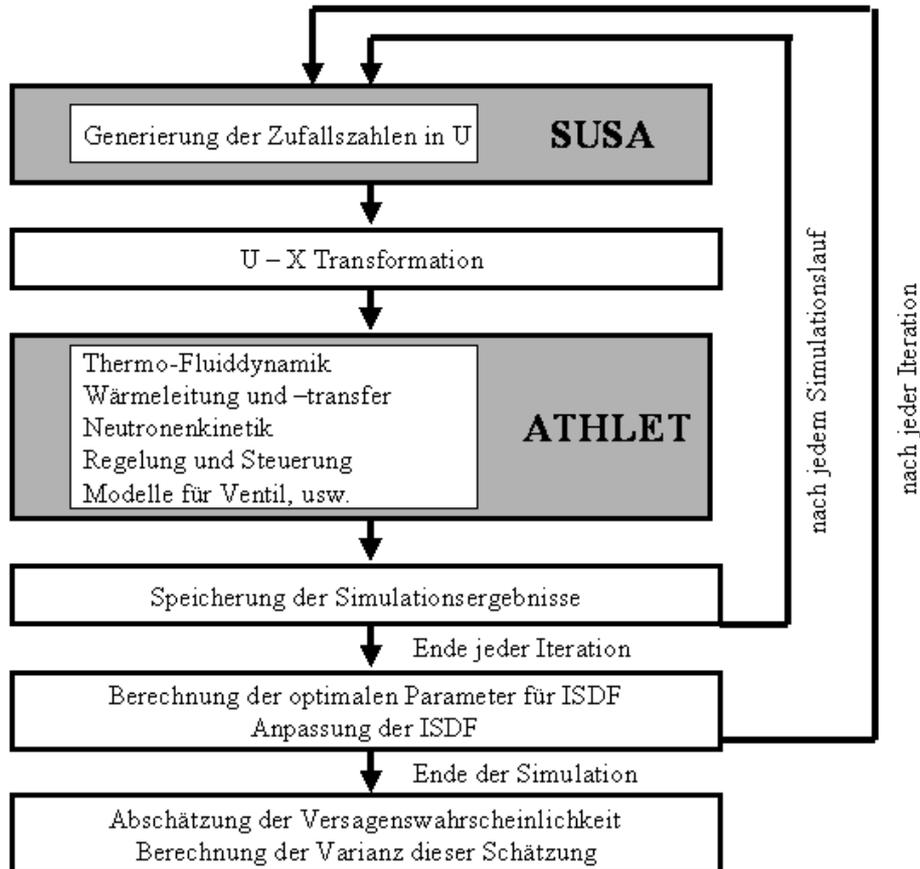


Abbildung 8.6: Aufbau des MCSAIS-Systems

8.9 Ergebnisse aus der MCSIAS

Die MCS mit iterativer Anpassung der Stichprobenfunktion wurde in vier Iterationen bei gleichem Stichprobenumfang (N=100) durchgeführt. Die gewichtete Stichprobenfunktion wurde nach der Bewertung der Simulationsergebnisse der vorherigen Iteration angepasst.

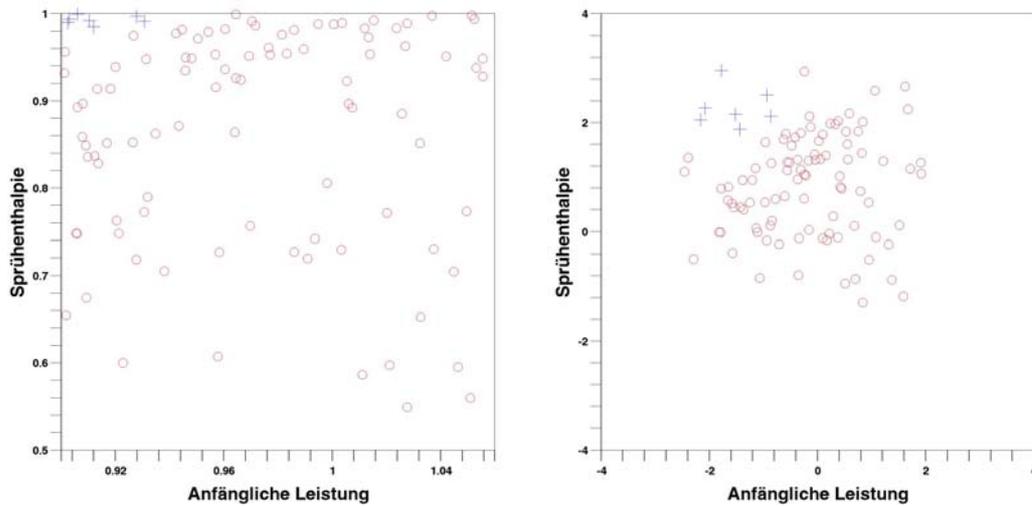


Abbildung 8.7: 1. Iteration der MCSAIS mit 100 Läufen in ursprünglichem und standardnormalem Raum

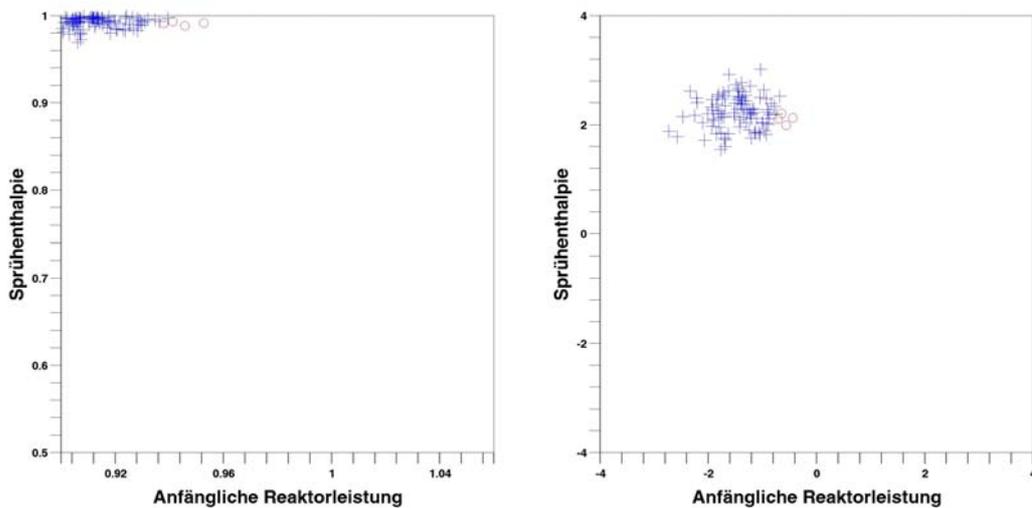


Abbildung 8.8: 2. Iteration der MCSAIS mit 100 Läufen in ursprünglichem und standardnormalem Raum

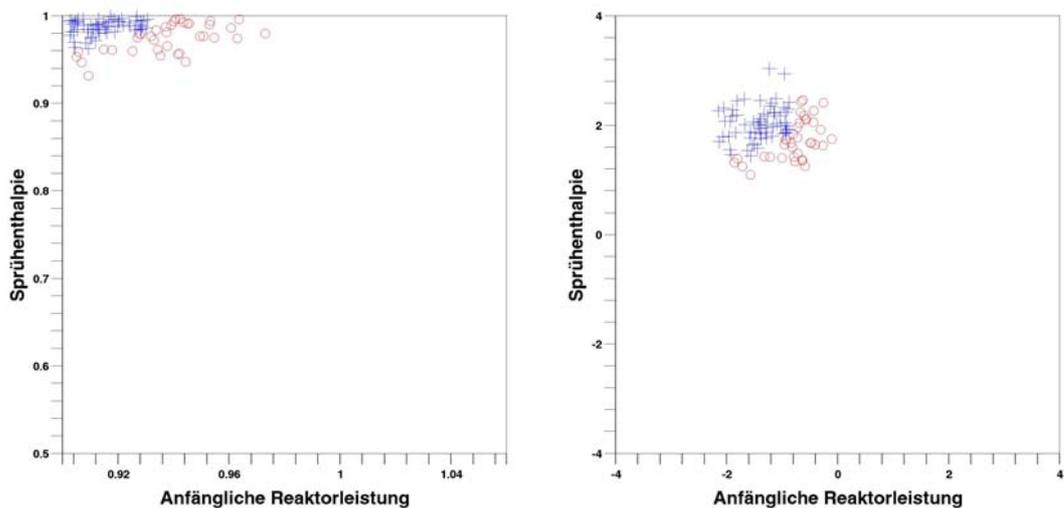


Abbildung 8.9: 3. Iteration der MCSIAS mit 100 Läufen in ursprünglichem und standardnormalem Raum

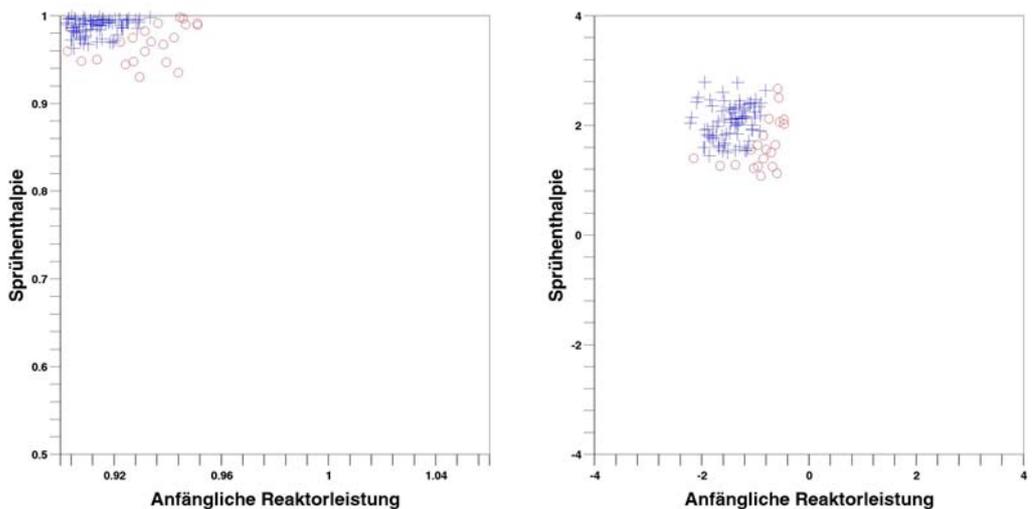


Abbildung 8.10: 4. Iteration der MCSIAS mit 100 Läufen in ursprünglichem und standardnormalem Raum

Im Vergleich mit der MCS ist MCSIAS sehr effizient, da viele Stichproben im Versagensgebiet gezogen werden. In der Tabelle 8.3 werden die Ergebnisse der MCS und der MCSIAS gegenübergestellt.

Fall	Methode	P_f	$\sigma[P_f]$	95%-Konfidenzintervall	CPU-Zeit (s)
1	MCS/1000 Läufe	1.40E-02	3.72E-03	6.71E-03-2.13E-02	17033
2	MCS/3000 Läufe	1.50E-02	2.22E-03	1.07E-02-1.93E-02	51187
3	MCS/5000 Läufe	1.20E-02	1.54E-03	9.00E-03-1.50E-02	85196
4	MCSIAS/100 Läufe/ 1. Iteration	9.90E-03	3.77E-03	2.50E-03-1.73E-02	1712
5	MCSIAS/100 Läufe/ 2. Iteration	1.50E-02	2.17E-03	1.07E-02-1.93E-02	1656
6	MCSIAS/100 Läufe/ 3. Iteration	1.14E-02	1.28E-03	8.90E-03-1.39E-02	1685
7	MCSIAS/100 Läufe/ 4. Iteration	1.19E-02	1.08E-03	9.80E-03-1.40E-02	1694

Tabelle 8.3: Vergleich der Ergebnisse aus der MCS und MCSIAS

In der Abbildung 8.11 wird die Reduktion der Standardabweichung für die Schätzung bei der MCS und der MCSIAS dargestellt. Im Vergleich mit der MCS reduziert sich die Standardabweichung der Schätzung bei der MCSIAS schneller. Nach der dritten Iteration mit 300 Läufen ist die Standardabweichung der geschätzten Öffnungswahrscheinlichkeit des Druckhalterventils kleiner als MCS mit 5000 Läufen.

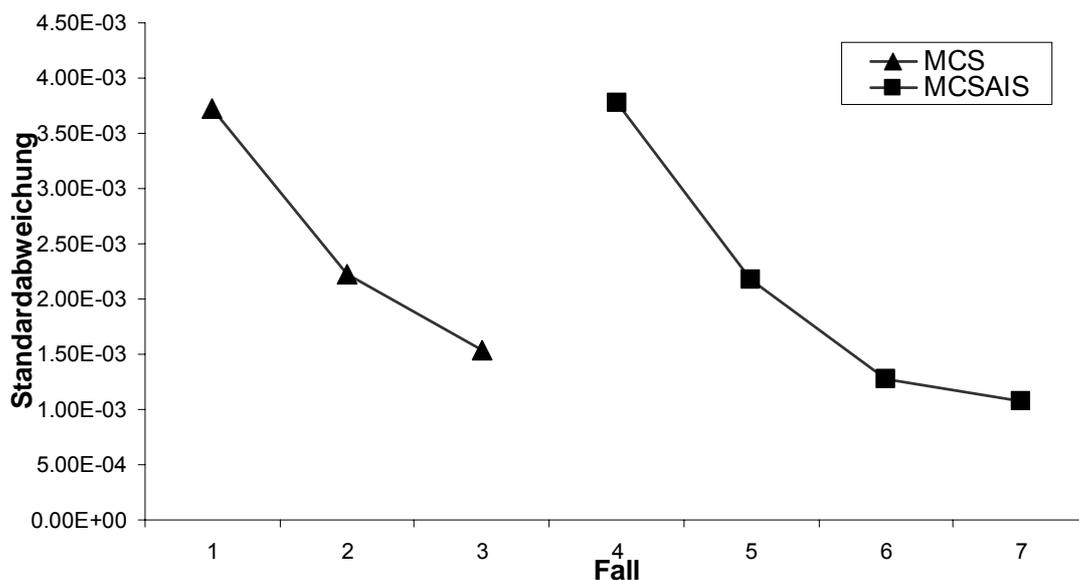


Abbildung 8.11: Standardabweichungen der geschätzten Öffnungswahrscheinlichkeiten

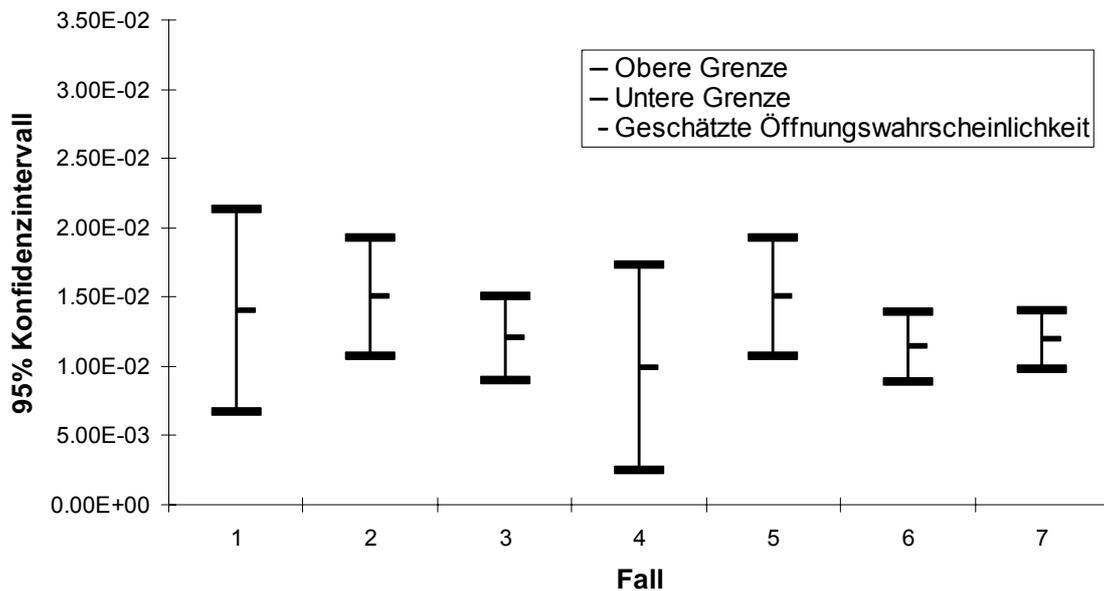


Abbildung 8.12: 95%-Konfidenzintervalle der geschätzten Öffnungswahrscheinlichkeiten

Da die Standardabweichung der Schätzung mit der MCSIAS im Vergleich mit der MCS schneller abnimmt, wird das 95%-Konfidenzintervall der Schätzung in der Abbildung 8.12 kleiner, d.h. die MCSIAS kann die Öffnungswahrscheinlichkeit des Druckhalterventils genauer schätzen.

8.10 Anbindung der MCSIAS ins herkömmlichen Ereignisablaufdiagramm

Die MCSIAS kann ins herkömmliche Ereignisablaufdiagramm an der Stelle angebunden werden, wo die Unfallabläufe sich unter Berücksichtigung der zeitlichen Wechselwirkungen mit der Fehlerbaum- und Ereignisablaufanalyse nicht realitätsnah modellieren und genau analysieren lassen.

Bei den Unfallabläufen in diesem Kapitel wird angenommen, dass das Druckhalterventil sich nach dem Erreichen des Ansprechendruckes öffnet. In der Realität kann sich das Druckhalterventil nach der Anforderung entweder öffnen oder versagen. Die Transiente im Kapitel 8.1 kann weiter untersucht werden. Ausgehend vom auslösenden Ereignis - Turbinenschnellabschaltung mit blockierter Umleitstation

- steigt der primäre Druck – im Reaktor. Als der primäre Druck den Ansprechdruck des Druckhalterventils erreicht, wird das Druckhalterventil angefordert zu öffnen. Das Druckhalterventil kann erfolgreich geöffnet oder nicht erfolgreich geöffnet werden. Die Wahrscheinlichkeit, dass das Druckhalter-Ventil nicht erfolgreich geöffnet wird, beträgt 0.06. Wenn das Druckhalterventil richtig geöffnet wird, senkt sich der primäre Druck. Wenn der primäre Druck wieder den Ansprechdruck des Druckhalterventils erreicht, wird an das Druckhalter-Ventil angefordert zu schließen. Das Druckhalter-Ventil kann erfolgreich oder nicht erfolgreich geschlossen werden. Die Wahrscheinlichkeit, dass das Druckhalter-Ventil nicht erfolgreich geschlossen wird, beträgt 0.06. Nach dem Versagen des Druckhalter-Ventils bei der Anforderung, kann das Druckhalter-Ventil offen bleiben. Dies kann einen Kühlmittelverlust verursachen. Um zu ermitteln, ob das Druckhalter-Ventil nach dem auslösenden Ereignis den Ansprechdruck erreicht oder nicht, wird nun normalerweise eine dynamische Rechnung durchgeführt. Die Ergebnisse hängen stark von der Entwicklung des primären Drucks ab. Um die Unfallabläufe genau zu analysieren, wird an dieser Stelle die dynamische PSA – MCSIAS – eingesetzt. Die weiteren Unfallabläufe können, durch die Anbindung der MCSIAS, ins herkömmliche Ereignisablaufdiagramm analysiert werden. Das Konzept der Anbindung der MCSIAS ins Ereignisablaufdiagramm wird in der Abbildung 8.13 dargestellt. Da die Unfallabläufe nicht zum Endzustand, dass das DH-Ventil offen bleibt, simuliert werden müssen, kann der gesamte Rechenaufwand stark reduziert werden.

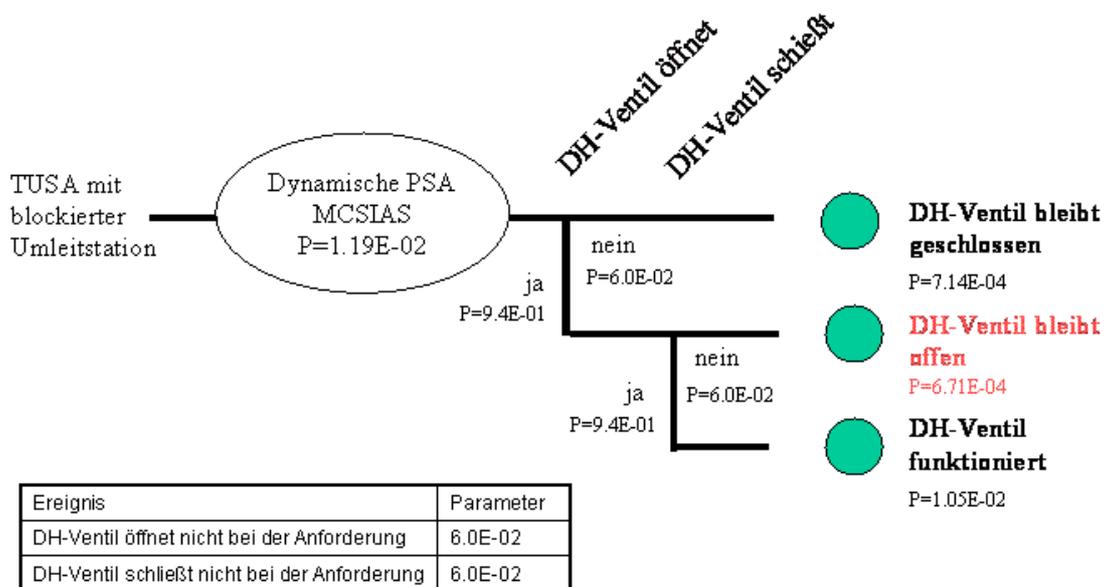


Abbildung 8.13: Einbindung der MCSIAS ins herkömmlichen Ereignisablaufdiagramm

8.11 Schätzwertverbesserung mit der MCSIAS

Im Abschnitt 6.2.4 wird das Verfahren für die Quantifizierung der epistemischen Größen auf die Ergebnisse der dynamischen PSA vorgestellt. Das Zweischleifenverfahren der MCS ist zwar sehr einfach aber sehr zeitaufwendig und praktisch nicht durchführbar. Um den Rechenaufwand des Zweischleifenverfahrens zu reduzieren hat Herr Hofer ein Annäherungsverfahren durch zwei getrennte MCS vorgeschlagen ([Hofer & Kloos -1, 2001], [Hofer & Kloos -2, 2001]).

Das Annäherungsverfahren beruht auf folgender Beziehung:

$$\text{Var}\{Y\} = \text{Var}\{E\{Y | B\}\} + E\{\text{Var}\{Y | B\}\} \quad (8.5)$$

In der ersten MCS werden nur die stochastischen Größen variiert und die epistemischen Größen bleiben auf dem Best-Estimate Wert. Der Schätzwert $\hat{\text{Var}}\{Y | b_{BE}\}$ aus der ersten MCS liefert einen Schätzwert für den Erwartungswert von $\text{Var}\{Y | B\}$ in Gleichung (8.14).

$$E\{\hat{\text{Var}}\{Y | B\}\} \approx \hat{\text{Var}}\{Y | b_{BE}\} \quad (8.6)$$

In der zweiten MCS werden sowohl die stochastischen als auch epistemischen Größen variiert. Nach der MCS werden $\hat{E}\{Y\}$ und $\hat{\text{Var}}\{Y\}$ geschätzt. Ein Schätzwert $\hat{\text{Var}}\{E\{Y | B\}\}$ kann aus der folgenden Beziehung erhalten werden

$$\hat{\text{Var}}\{E\{Y | B\}\} = \hat{\text{Var}}\{Y\} - \hat{E}\{\text{Var}\{Y | B\}\} \quad (8.7)$$

Durch die Annahme, dass die Wahrscheinlichkeitsverteilung von $E\{Y | B\}$ durch eine logarithmische Normalverteilung mit dem Erwartungswert $\hat{E}\{Y\}$ und der Varianz $\hat{\text{Var}}\{E\{Y | B\}\}$ gut beschrieben werden kann, können die epistemischen Größen auf die PSA-Ergebnisse quantifiziert werden.

Die Schätzungsgenauigkeit hängt stark vom den Schätzwerten $\hat{E}\{Y\}$ und $\hat{\text{Var}}\{E\{Y | B\}\}$ ab. Um aussagenkräftige Schätzwerte zu erhalten, muss eine ausreichende Anzahl von Simulationsläufen durchgeführt werden. Hier kann die

MCSIAS eingesetzt, um die Effizienz der MCS zu verbessern, d.h. die Schätzwerte können mit einem geringeren Rechenaufwand genauer geschätzt werden.

9 Zusammenfassung

Probabilistische Sicherheitsanalyse (PSA) ist ein systematisches Verfahren zur Bewertung der Systemsicherheit und wurde erfolgreich in verschiedenen Gebieten eingesetzt. Die Methoden der herkömmlichen PSA stützen sich auf die Fehlerbaum- und Ereignisablaufanalyse. Während die Systemverfügbarkeit mit der Fehlerbaumanalyse berechnet wird, werden die Unfallabläufe mit der Ereignisablaufanalyse modelliert und die Eintrittswahrscheinlichkeit der Schadenszustände mit der Ereignisablaufanalyse ermittelt. Die Fehlerbaum- und Ereignisablaufanalyse sind statische Methoden und diese erfolgen ohne direkte Kopplung an die tatsächliche Systemdynamik, so dass die Wechselwirkungen zwischen den Komponentenausfällen und der Systemdynamik nicht berücksichtigt wird. Daher können die Unfallabläufe nicht realitätsnah modelliert und genau analysiert werden. Eine genauere probabilistische Analyse der dynamischen Systeme sowie der dynamischen Wechselwirkungen erfordert daher die Anwendung der dynamischen PSA-Methode.

Zur Beseitigung der Einschränkungen der Fehlerbaum- und Ereignisablaufanalyse wurden die dynamischen PSA-Methoden entwickelt. Die Phased Mission Methoden können für die verschiedenen Phasen eines Unfalls einen eignen Fehlerbaum erstellen. Dies kann zu sehr komplexen Strukturen führen. Markov Modelle können auch zeitabhängige Prozessgrößen gut berücksichtigen und bieten somit ein analytisches Lösungsverfahren an. Zur Ermittlung der Eintrittswahrscheinlichkeit müssen alle möglichen Zustände und Zustandsübergänge definiert werden. Da die Anzahl der Zustände und Zustandsübergänge sehr schnell zu einer nicht handhabbaren Größe bei den dynamischen Systemen anwachsen kann, bleibt die Anwendung der Markov-Modelle auf Systeme mit wenigen Komponenten und Prozessgrößen beschränkt. Die graphischen Werkzeuge z.B. dynamische Fehlerbaumanalyse, GO-FLOW Diagramme, Petrinetze und Event Sequence Diagramme können die dynamischen Abläufe intuitiv modellieren und die relevanten Unfallabläufe leicht identifizieren. Die quantitative Bewertung der graphischen Werkzeuge erfolgt meistens mit den Markov-Modellen, so dass die Anwendung auch auf Systeme mit wenigen Komponenten und Prozessgrößen beschränkt bleibt. Bei der MCS und den diskreten dynamischen Ereignisablaufdiagrammen müssen nicht alle mögliche Zustände und Zustandsübergänge vorher definiert sein. Die Zustände und Zustandsübergänge werden nach den vom Analytiker formulierten Regeln definiert. Außerdem modelliert ein Systemmodell die zeitliche Entwicklung der Systemdynamik,

so dass die Wechselwirkungen zwischen der Systemdynamik und stochastischen Ereignissen gut berücksichtigt werden können. Dynamische Ereignisablaufdiagramme versuchen alle möglichen Unfallabläufe systematisch darzustellen. Wenn die stochastischen Übergänge stetig sind und die Zeitschritte klein gewählt werden, wird der Rechenaufwand sehr hoch. Die dynamischen Ereignisablaufdiagramme werden eingesetzt, wenn die Anzahl der Unfallabläufe klein ist. Dagegen werden nur zufällig ausgewählte Unfallabläufe bei der MCS simuliert. Der Rechenaufwand ist nur von der Anzahl der Simulationsläufe abhängig aber von der Dimension des Systems unabhängig. Außerdem ist die MCS leicht zu implementieren und allgemein anwendbar, wie z.B. Simulation der stetigen und diskreten Übergänge.

Um die Analysemöglichkeiten der MCS und die gegenseitigen Wechselwirkungen zwischen den Komponentenausfällen und der Systemdynamik zu demonstrieren, wurde ein dynamisches „Hold-Up-Tank“-System implementiert. Die Unfallabläufe werden sowohl mit der herkömmlichen Ereignisablaufanalyse als auch mit der MCS analysiert. Im Vergleich mit dem Ereignisablaufdiagramm kann die MCS die Unfallabläufe realitätsnah modellieren und analysieren, wie z.B. die Modellierung der mehrfachen Anforderung des Ventils während des Unfallablaufes. Mit dem Zweischleifenverfahren kann auch der Einfluss der epistemischen Größen auf die Ergebnisse der dynamischen PSA quantifiziert werden. Der Nachteil der MCS liegt darin, dass eine große Anzahl von Simulationen erforderlich ist, um die kleine Eintrittswahrscheinlichkeit eines Ereignisses bei der gegebenen Genauigkeitsanforderung zu schätzen. Bei hochzuverlässigen Systemen z.B. Kernkraftwerk müssen die rechenzeitbeschleunigenden Methoden der Monte Carlo Simulation zum Einsatz kommen.

Die FORM und die MCDET wurden von Herrn Woltereck und Herrn Hofer vorgeschlagen, um die MCS für die Anwendung in Kernkraftwerken zu optimieren. Die FORM ist ein Annäherungsverfahren, in dem das Problem zur Ermittlung der Versagenswahrscheinlichkeit als ein Optimierungsproblem mit der Zielfunktion und den Nebenbedingungen formuliert wird. Zur Ermittlung der Versagenswahrscheinlichkeit braucht die FORM wenige Simulationsläufe und weniger Rechenaufwand im Vergleich mit der MCS. Um das Optimierungsproblem leicht zu lösen, müssen einige Voraussetzungen erfüllt werden. Bei einer stark gekrümmten Grenzfunktion hat die Schätzung der FORM einen großen Fehler. Die Schätzungsgenauigkeit kann mit der FORM schwer quantifiziert werden. Die MCDET ist eine Kombination der Monte Carlo Simulation und des dynamischen Ereignisbaums. Dieses Verfahren kann die

Versagenswahrscheinlichkeit bei gleichem Stichprobenumfang der MCS genauer schätzen. Die größte Einschränkung bei MCDET ist die Simulation der stetigen Übergänge mit Hilfe von klassischer MCS. Die MCDET wird eingesetzt, wenn die stetigen Übergänge mit einer kleinen Eintrittswahrscheinlichkeit keinen Einfluss auf die Versagenswahrscheinlichkeit haben.

Da die Anwendbarkeit der FORM und MCDET eingeschränkt ist, wird ein Konzept zur weiteren Optimierung der MCS gestellt. Das neue Optimierungsverfahren soll den Schadenszustand mit einer kleinen Eintrittswahrscheinlichkeit schnell entdecken, einen geringen Rechenaufwand haben und die Schätzungsunsicherheit quantifizieren. Außerdem ist es wünschenswert, ein Adaptives Verfahren zu entwickeln und die Simulationsergebnisse und die Expertenschätzung zur Reduzierung des Rechenaufwandes nutzen zu können. Nach diesem Optimierungskonzept wird das Verfahren – Monte Carlo Simulation mit iterativer Anpassung der Stichprobenfunktionen (MCSIAS) – vorgeschlagen.

Die MCSIAS wurde erstmalig für die dynamische PSA eingesetzt. Die MCSIAS kann die Versagenswahrscheinlichkeit mit einer kleinen Varianz bei gleichem Stichprobenumfang schätzen. Die Simulationsergebnisse und Expertenschätzung können genutzt werden, um die Stichprobenfunktion gezielt anzupassen, so dass die Effizienz und die Genauigkeit der MCS verbessert werden. Das Verfahren wurde für eine Transiente in einem DWR nach einer Turbinenschnellabschaltung mit blockierter Umleitstation untersucht. Die Öffnungswahrscheinlichkeit des Druckhalter-Ventils wurde sowohl mit der MCS als auch mit der MCSIAS ermittelt. Im Vergleich mit der MCS sinkt die Rechenzeit bei der Anwendung der MCSIAS um weit mehr als eine Größenordnung. Um die MCSIAS anzuwenden, braucht die MCSIAS wenige Voraussetzungen. Außerdem kann die Schätzungsgenauigkeit mit dem 95%-Konfidenzintervall quantifiziert werden. Im Vergleich mit der FORM braucht die MCSIAS wenige Voraussetzungen. Im Vergleich mit der klassischen MCS können deutlich mehr Stichproben im Versagensgebiet gezogen und simuliert werden. Die Effizienz der MCSIAS hängt von der geeigneten Wahl der gewichteten Stichprobenfunktionen in der ersten Iteration ab. Wenn der Mittelwert der Stichprobenfunktion in der ersten Iteration liegt in der Nähe der Grenzfläche, so kann die MCSIAS den Versagensgebiet schnell identifizieren und die gewichteten Stichprobenfunktionen leicht anpassen.

Die MCSIAS kann auch leicht ins Ereignisablaufdiagramm an der Stelle angebunden werden, wo eine detaillierte und genaue Analyse durchgeführt werden muss. Außerdem kann die MCSIAS auch für solche Verfahren wie z.B. das Annäherungsverfahren im Abschnitt 8.11 eingesetzt werden, um den Schätzwert z.B. die Eintrittswahrscheinlichkeit eines Schadenszustandes genauer schätzen zu können.

10 Literaturverzeichnis

- [Birolini 1997] Alessandro Birolini, Zuverlässigkeit von Geräten und Systemen, Springer-Verlag, Berlin, 1997
- [Bucher, 1987] Bucher, C., Adaptive sampling – an iterative fast monte carlo procedure. Structural Safety 5: 119-126, 1987
- [Cojazzi 1996] Cojazzi G., The DYLAM approach for the dynamic reliability analysis of systems, Reliability Engineering and System Safety 52, page 279-296, 1996
- [Dutuit & Chatelet 1997] Dutuit Y., Chatelet E., Signoret J.P. & Thomas P., Dependability modelling and evaluation by using stochastic Petri nets: application to two test cases, Reliability Engineering & System Safety 55(2): 117-124, 1997
- [Devooght & Smidts 1992] Devooght, J., Smidts, C., Probabilistic Reactor Dynamics 1. The Theory Of Continuous Event Trees, Nuclear Science and Engineering 111(3), 229-240, 1992
- [Devooght, 1998] J.Devooght, Dynamic Reliability: The Challenges Ahead, Fifth International Workshop on Dynamic Reliability: Future Directions, USA, Sept. 1998
- [IWDR-Discussion-1 1998] Discussion, Application of Dynamic Reliability to PRA, Fifth International Workshop on Dynamic Reliability: Future Directions, USA, Sept. 1998
- [IWDR-Discussion-2 1998] Discussion, New Domains of Application, Fifth International Workshop on Dynamic Reliability: Future Directions, USA, Sept. 1998
- [IWDR-Discussion-3 1998] Discussion, Solution Methods and Development of Software Tools, Fifth International Workshop on Dynamic Reliability: Future Directions, USA, Sept.1998

- [Hofer & Kloos 1999] Hofer, E. & Kloos, M., SUSA – User’s Guide and Tutorial, GRS, 1999
- [Hofer & Kloos -1, 2001] Hofer, E., Kloos, M., Methodenentwicklung zur simulativen Behandlung der Stochastik in probabilistischen Sicherheitsanalysen der Stufe 2, Zwischenbericht, GRS-A-2786, März 2001
- [Hofer & Kloos -2, 2001] Hofer, E. & Kloos, M., Methodenentwicklung zur simulativen Behandlung der Stochastik in probabilistischen Sicherheitsanalysen der Stufe 2, Abschlussbericht, GRS-A-2997, 2001
- [Izquierdo & Labeau 2004] Izquierdo, J., & Labeau, P. E., The Stimulus-Driven Theory of Probabilistic Dynamics as a Framework for Probabilistic Safety Assessment, Proceedings of the PSAM7 – ESREL 2004 conference, 2004
- [Kloos & Peschke 2005] Kloos, M., Peschke J., Fassmann W., Methodenentwicklung zur Berücksichtigung menschlicher Eingriffe im Rahmen einer dynamischen PSA der Stufe 1 und Stufe 2, Technischer Bericht, GRS-A-3269, März 2005
- [Kohda & Nakagawa, 2006] Kohda T. & Nakagawa M., Risk evaluation of batch processes in chemical plants using phased-mission analysis, Proceedings of the ESREL 2006 conference, p. 151-156, 2006
- [Labeau, 2000] P.E. Labeau, C. Smidts, Dynamic reliability: towards an integrated platform for probabilistic risk assessment, Reliability Engineering & System Safety 68 219-254, 2000
- [Lerchl & Austregesilo, 2000] Lerchl, G. & Austregesilo H., *ATHLET Mod.1.1 Cycle D, User Manual*, GRS-P-1 /Vol.1 (Updated). Köln: GRS, 1997
- [Marseguerra & Zio 2001] Marseguerra, M. & Zio, E., Principles of monte carlo

simulation for application to reliability and availability analysis. European Safety & Reliability International Conference. Italy, 2001.

[NRC-RG-1174, 2002] An Approach for Using Probabilistic Risk Assessment in Risk-informed Decisions on Plant-Specific Changes to the Licensing Basis, Regulatory Guide 1.174, U.S. Nuclear Regulatory Commission, Washington, 2002

[Woltereck, 1998] Martin Woltereck, Dynamic Reliability Analysis with Quantification of Epistemic Uncertainty: A BWR Application,

[Woltereck, 2000] Woltereck, M., Dynamische Zuverlässigkeitsanalyse mit anlagenspezifischen Störfallsimulation, Dissertation, Lehrstuhl für Reaktordynamik und Reaktorsicherheit, 2000

[Siu, 1994] Siu, N.: Risk assessment for dynamic systems: An overview, Reliability Engineering and System Safety 43, Seite 43-73, 1994

[Smidts, 1998] Smidts, C., White paper for workshop *Dynamic reliability: Further directions*, Proceedings in prep, 1998

[Schneeweiss 1999] Schneeweiss, W., Petri Nets for Reliability Modeling, LiLoLe Publishing, Germany, 1999

Abbildungsverzeichnis

Abbildung 2.1: Ablauf der PSA Stufe 1	16
Abbildung 2.2: Fehlerbaum-Beispiel.....	17
Abbildung 2.3: Ereignisablaufdiagramm-Beispiel	19
Abbildung 3.1: Badewannenkurve	22
Abbildung 4.1: Beispiel - Phased Mission Methoden.....	28
Abbildung 4.2: Beispiel - Markov	30
Abbildung 4.3: Beispiel – Diskreter dynamischer Ereignisablaufdiagramm.....	32
Abbildung 4.4: Beispiel - Event Sequence Diagramm	38
Abbildung 6.1: "Hold-Up-Tank"-System.....	48
Abbildung 6.2: Analyse des "Hold-Up-Tank"-Systems mit herkömmlichem Ereignisablaufanalyse	53
Abbildung 6.3: Analyse des "Hold-Up-Tank"-Systems mit der MCS	55
Abbildung 6.4: Unfallablauf 1 des "Hold-Up-Tank"-Systems	56
Abbildung 6.5: Unfallablauf 2 des "Hold-Up-Tank"-Systems	56
Abbildung 6.6: Unfallablauf 3 des "Hold-Up-Tank"-Systems	57
Abbildung 6.7: Unfallablauf 4 des "Hold-Up-Tank"-Systems	57
Abbildung 6.8: Verläufe des Wasserfüllstandes innerhalb von 100 Stunden	59
Abbildung 6.9: Verläufe des Wasserfüllstandes innerhalb von 10000 Stunden	59

Abbildung 6.10: Eintrittswahrscheinlichkeit des "DRYOUT"-Zustandes innerhalb von 100 Stunden	60
Abbildung 6.11: Eintrittswahrscheinlichkeit des "DRYOUT"-Zustandes innerhalb von 10000 Stunden	60
Abbildung 6.12: Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 100 Stunden	61
Abbildung 6.13: Eintrittswahrscheinlichkeit des "OVERFLOW"-Zustandes innerhalb 10000 Stunden	61
Abbildung 6.14: Analyse des "Hold-Up-Tank"-Systems mit dem Zweischleifenverfahren	63
Abbildung 6.15: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des "DRYOUT"-Zustandes innerhalb von 100 Stunden	64
Abbildung 6.16: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des "DRYOUT"-Zustandes innerhalb von 10000 Stunden	65
Abbildung 6.17: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 100 Stunden	65
Abbildung 6.18: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 10000 Stunden	66
Abbildung 6.19: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes innerhalb von 100 Stunden	66
Abbildung 6.20: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „DRYOUT“-Zustandes innerhalb von 10000 Stunden	67
Abbildung 6.21: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 100 Stunden	67
Abbildung 6.22: Einfluss der epistemischen Größen auf die Eintrittswahrscheinlichkeit des „OVERFLOW“-Zustandes innerhalb von 10000 Stunden	68

Abbildung 7.1: Beispiel – Ein dynamischer Ereignisablaufdiagramm aus dem MCDET für das "Hold-Up-Tank"-System	73
Abbildung 8.1: Beispiel - ATHLET Simulator	82
Abbildung 8.2: Aufbau des MCS-Systems.....	84
Abbildung 8.3: MCS mit 1000 Läufen im ursprünglichen und standardnormalen Raum	85
Abbildung 8.4: MCS mit 3000 Läufen im ursprünglichen und standardnormalen Raum	85
Abbildung 8.5: MCS mit 5000 Läufen im ursprünglichen und standardnormalen Raum	86
Abbildung 8.6: Aufbau des MCSAIS-Systems.....	89
Abbildung 8.7: 1. Iteration der MCSAIS mit 100 Läufen in ursprünglichem und standardnormalem Raum.....	90
Abbildung 8.8: 2. Iteration der MCSAIS mit 100 Läufen in ursprünglichem und standardnormalem Raum.....	90
Abbildung 8.9: 3. Iteration der MCSIAS mit 100 Läufen in ursprünglichem und standardnormalem Raum.....	91
Abbildung 8.10: 4. Iteration der MCSIAS mit 100 Läufen in ursprünglichem und standarnormalem Raum.....	91
Abbildung 8.11: Standardabweichungen der geschätzten Öffnungswahrscheinlichkeiten.....	93
Abbildung 8.12: 95%-Konfidenzintervalle der geschätzten Öffnungswahrscheinlichkeiten.....	93
Abbildung 8.13: Einbindung der MCSIAS ins herkömmlichen Ereignisablaufdiagramm	94

Tabellenverzeichnis

Tabelle 4.1: Zusätzliche Verknüpfungen des dynamischen Fehlerbaumes	34
Tabelle 4.2: Überblick über die dynamischen PSA-Methoden	39
Tabelle 6.1: Schadenzustände des "Hold-Up-Tank"-Systems	49
Tabelle 6.2: Mögliche Komponentenzustände des "Hold-Up-Tank"-Systems	49
Tabelle 6.3: Systemparameter des "Hold-Up-Tank"-Systems	49
Tabelle 6.4: Anfangszustände der Systemkomponenten	50
Tabelle 6.5: Dynamik des "Hold-Up-Tank"-Systems	50
Tabelle 6.6: Regelung des "Hold-Up-Tank"-Systems	51
Tabelle 7.1: Vergleich zwischen FORM und MCS	72
Tabelle 7.2: Vergleich zwischen dem MCDET und der MCS	75
Tabelle 8.1 Verteilungsfunktion der stochastischen Größen	81
Tabelle 8.2: Ergebnisse aus MCS	86
Tabelle 8.3: Vergleich der Ergebnisse aus der MCS und MCSIAS	92

Anhang A - Analyse des „Hold-Up-Tank“-Systems mit der Monte Carlo Simulation

```
printf("\n \n \n \n");
printf(" \n ++++++");
printf(" \n ++++++ +++++");
printf(" \n ++++++ Monte Carlo Simulation +++++");
printf(" \n ++++++ „Hold-Up-Tank“-System +++++");
printf(" \n ++++++ +++++");
printf(" \n ++++++");
printf("\n \n \n \n");

$Jetztwert = time();
$Jetztzeit = localtime($Jetztwert);
printf("\n \nSimulation beginnt: %s\n \n", $Jetztzeit );
printf("\n \n");

$LAUF = 100;
$Simulationszeit = 10000;

$Verzeichnis = ">E:/Perl-HoldUpTankBeispiel/Ergebnis/";
$Endung = ".txt";
$Maple = "_Maple";

#Speicherung der Simulationsergebnisse
open(DAT2, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/Ergebnis.txt");
open(DAT5, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/DRYOUT-MAPLE.txt");
printf(DAT5 " data1:=");
open(DAT7, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/DRYOUT.txt");
open(DAT6, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/OVERFLOW-MAPLE.txt");
printf(DAT6 " data1:=");
open(DAT8, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/OVERFLOW.txt");

#Eintrittswahrscheinlichkeiten der Schadenzustände innerhalb von einer Simulationszeit
open(DAT9, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/DFDryout.txt");
open(DAT10, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/DFOverflow.txt");
open(DAT11, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/VFDryout.txt");
open(DAT12, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/VFOverflow.txt");

#Verlauf des Wasserfüllstandes
open(DAT13, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/WasserWasserfüllstand-Spalte.txt");

#Initialisierung
for ($n = 1; $n <= $LAUF; $n++) {
    for ($m = 1; $m <= $Simulationszeit; $m++) {
        $Wasserfuellstand[$n][$m] = -1;
    }
}

open(DAT14, ">E:/Perl-HoldUpTankBeispiel/Ergebnis/WasserWasserfüllstand-Zeile.txt");

#Zaehler fuer den Dryout-Zustand und Overflow-Zustand
$DZ = 0;
$OZ = 0;

#Dichtefunktion und Verteilungsfunktion für den Dryout-Zustand und Overflow-Zustand
for ($n = 1; $n <= $Simulationszeit; $n++) {
    $DDryout[$n] = 0;
    $VDryout[$n] = 0;
    $DOverflow[$n] = 0;
    $VOverflow[$n] = 0;
}

for ($lauf = 1; $lauf <= $LAUF; $lauf++) {

#Ergebnis speichern
$FileName = join(' ', $Verzeichnis, $lauf, $Endung);
$MapleName = join(' ', $Verzeichnis, $lauf, $Maple, $Endung);
open(DAT3, $FileName);
open(DAT4, $MapleName);
```

```

#Festlegung der Systemparameter
$a = 0;
$a1 = 3;
$a2 = 7;
$b = 10;
$FH = 5;

#Festlegung des Anfangszustandes, des Zeitschrittes und der Simulationszeit
$t = 0;
$dt = 1;
$st = $Simulationszeit;

#Generierung der zufälligen Ausfallzeitpunkte der Systemkomponenten
printf("\n \n \n \n");
system("cmaple9 ZufallZeitVSN-Maple.txt");
printf("\n \n \n \n");

$n = 0;
open(DAT1,"E:/Perl-HoldUpTankBeispiel/Zufallzeit.txt");
while (<DAT1>) {
    $n = $n + 1;
    eval ($_);
    printf(DAT4 $_);
    chop;
    ($Variable,$Zufallzeit) = split(/:/, $_, 2);
    if ($n == 1) {
        $Tsv = $Zufallzeit;
    }
    if ($n == 2) {
        $Tsp1 = $Zufallzeit;
    }
    if ($n == 3) {
        $Tsp2 = $Zufallzeit;
    }
}

#Festlegung der Anfangszustände der Systemkomponenten
#Ventil 0)auf 1)zu 2)oeffnet nicht 3)schliesst nicht
#Pumpe 1/2 0)an 1)aus 2)startet nicht 3)stoppt nicht

$sv = 0;
$sp1 = 2;
$sp2 = 1;

printf("\n");
printf("\nVentil = auf");
printf("\nPumpe 1 = statet nicht");
printf("\nPumpe 2 = aus");
printf("\nFH = %s",$FH );
printf("\nT = %d",$t );
printf("\n");

#zufällige Zustandsänderungen der Systemkomponenten
while ($t < $st && $FH > 0 && $FH < 10) {

    $t = $t + $dt;

    if ($t == $Tsv) {

        if ($sv == 0) {
            $sv = 2;
        }
        elsif ($sv == 1) {
            $sv = 3;
        }
    }
    if ($t == $Tsp1) {

        if ($sp1 == 0) {
            $sp1 = 2;
        }
        elsif ($sp1 == 1) {

```

```

    $sp1 = 3;
  }
}

if ($t == $Tsp2) {

  if ($sp2 == 0) {
    $sp2 = 2;
  }
  elseif ($sp2 == 1) {
    $sp2 = 3;
  }
}

#Modellierung der Systemdynamik
#Systementwicklung (z.B. Ventil auf == Ventil schliesst nicht)
#Systementwicklung (z.B. Ventil zu == Ventil oeffnet nicht)
if ($sv == 0 || $sv == 3) {
  $Lsv = 0
}

if ($sv == 1 || $sv == 2) {
  $Lsv = 1
}

if ($sp1 == 0 || $sp1 == 3) {
  $Lsp1 = 0
}

if ($sp1 == 1 || $sp1 == 2) {
  $Lsp1 = 1
}

if ($sp2 == 0 || $sp2 == 3) {
  $Lsp2 = 0
}

if ($sp2 == 1 || $sp2 == 2) {
  $Lsp2 = 1
}

if ($Lsv == 0 && $Lsp1 == 1 && $Lsp2 == 1) {
  $FH = $FH - 0.1 * 5 * $dt;
  printf("\n");
  printf(DAT3 "\n");
  if ($sv == 0) {
    printf("\nVentil = auf");
    printf(DAT3 "\nVentil = auf");
  }
  elseif ($sv == 3) {
    printf("\nVentil = schliesst nicht");
    printf(DAT3 "\nVentil = schliesst nicht");
  }
  if ($sp1 == 1) {
    printf("\nPumpe 1 = aus");
    printf(DAT3 "\nPumpe 1 = aus");
  }
  elseif ($sp1 == 2) {
    printf("\nPumpe 1 = startet nicht");
    printf(DAT3 "\nPumpe 1 = startet nicht");
  }
  if ($sp2 == 1) {
    printf("\nPumpe 2 = aus");
    printf(DAT3 "\nPumpe 2 = aus");
  }
  elseif ($sp2 == 2) {
    printf("\nPumpe 2 = startet nicht");
    printf(DAT3 "\nPumpe 2 = startet nicht");
  }
  printf("\nFH = %s", $FH );
  printf("\nT = %s", $t );
  printf("\n");

  printf(DAT3 "\nFH = %s", $FH );
  printf(DAT3 "\nT = %s", $t );
  printf(DAT3 "\n");
}

```

```

}
if ($Lsv == 0 && $Lsp1 == 0 && $Lsp2 == 1) {
    $FH = $FH;
    printf("\n");
    printf(DAT3 "\n");
    if ($sv == 0) {
        printf("\nVentil = auf");
        printf(DAT3 "\nVentil = auf");
    }
    elseif ($sv == 3) {
        printf("\nVentil = schliesst nicht");
        printf(DAT3 "\nVentil = schliesst nicht");
    }
    if ($sp1 == 0) {
        printf("\nPumpe 1 = an");
        printf(DAT3 "\nPumpe 1 = an");
    }
    elseif ($sp1 == 3) {
        printf("\nPumpe 1 = stoppt nicht");
        printf(DAT3 "\nPumpe 1 = stoppt nicht");
    }
    if ($sp2 == 1) {
        printf("\nPumpe 2 = aus");
        printf(DAT3 "\nPumpe 2 = aus");
    }
    elseif ($sp2 == 2) {
        printf("\nPumpe 2 = startet nicht");
        printf(DAT3 "\nPumpe 2 = startet nicht");
    }
    }
    printf("\nFH = %s", $FH );
    printf("\nT = %s", $t );
    printf("\n");

    printf(DAT3 "\nFH = %s", $FH );
    printf(DAT3 "\nT = %s", $t );
    printf(DAT3 "\n");
}
if ($Lsv == 0 && $Lsp1 == 1 && $Lsp2 == 0) {
    $FH = $FH - 0.05 * 5 * $dt;
    printf("\n");
    printf(DAT3 "\n");
    if ($sv == 0) {
        printf("\nVentil = auf");
        printf(DAT3 "\nVentil = auf");
    }
    elseif ($sv == 3) {
        printf("\nVentil = schliesst nicht");
        printf(DAT3 "\nVentil = schliesst nicht");
    }
    if ($sp1 == 1) {
        printf("\nPumpe 1 = aus");
        printf(DAT3 "\nPumpe 1 = aus");
    }
    elseif ($sp1 == 2) {
        printf("\nPumpe 1 = startet nicht");
        printf(DAT3 "\nPumpe 1 = startet nicht");
    }
    if ($sp2 == 0) {
        printf("\nPumpe 2 = an");
        printf(DAT3 "\nPumpe 2 = an");
    }
    elseif ($sp2 == 3) {
        printf("\nPumpe 2 = stoppt nicht");
        printf(DAT3 "\nPumpe 2 = stoppt nicht");
    }
    }
    printf("\nFH = %s", $FH );
    printf("\nT = %s", $t );
    printf("\n");

    printf(DAT3 "\nFH = %s", $FH );
    printf(DAT3 "\nT = %s", $t );
    printf(DAT3 "\n");
}
if ($Lsv == 0 && $Lsp1 == 0 && $Lsp2 == 0) {
    $FH = $FH + 0.05 * 5 * $dt;
    printf("\n");
    printf(DAT3 "\n");
}

```

```

if ($sv == 0) {
    printf("\nVentil = auf");
    printf(DAT3 "\nVentil = auf");
}
elseif ($sv == 3) {
    printf("\nVentil = schliesst nicht");
    printf(DAT3 "\nVentil = schliesst nicht");
}
if ($spl == 0) {
    printf("\nPumpe 1 = an");
    printf(DAT3 "\nPumpe 1 = an");
}
elseif ($spl == 3) {
    printf("\nPumpe 1 = stoppt nicht");
    printf(DAT3 "\nPumpe 1 = stoppt nicht");
}
if ($sp2 == 0) {
    printf("\nPumpe 2 = an");
    printf(DAT3 "\nPumpe 2 = an");
}
elseif ($sp2 == 3) {
    printf("\nPumpe 2 = stoppt nicht");
    printf(DAT3 "\nPumpe 2 = stoppt nicht");
}
printf("\nFH = %s", $FH );
printf("\nT = %s", $t );
printf("\n");

printf(DAT3 "\nFH = %s", $FH );
printf(DAT3 "\nT = %s", $t );
printf(DAT3 "\n");
}
if ($Lsv == 1 && $Lspl == 1 && $Lsp2 == 1) {
    $FH = $FH;
    printf("\n");
    printf(DAT3 "\n");
    if ($sv == 1) {
        printf("\nVentil = zu");
        printf(DAT3 "\nVentil = zu");
    }
    elseif ($sv == 2) {
        printf("\nVentil = oeffnet nicht");
        printf(DAT3 "\nVentil = oeffnet nicht");
    }
    if ($spl == 1) {
        printf("\nPumpe 1 = aus");
        printf(DAT3 "\nPumpe 1 = aus");
    }
    elseif ($spl == 2) {
        printf("\nPumpe 1 = startet nicht");
        printf(DAT3 "\nPumpe 1 = startet nicht");
    }
    if ($sp2 == 1) {
        printf("\nPumpe 2 = aus");
        printf(DAT3 "\nPumpe 2 = aus");
    }
    elseif ($sp2 == 2) {
        printf("\nPumpe 2 = startet nicht");
        printf(DAT3 "\nPumpe 2 = startet nicht");
    }
    printf("\nFH = %s", $FH );
    printf("\nT = %s", $t );
    printf("\n");

    printf(DAT3 "\nFH = %s", $FH );
    printf(DAT3 "\nT = %s", $t );
    printf(DAT3 "\n");
}
if ($Lsv == 1 && $Lspl == 0 && $Lsp2 == 1) {
    $FH = $FH + 0.1 * 5 * $dt;
    printf("\n");
    printf(DAT3 "\n");
    if ($sv == 1) {
        printf("\nVentil = zu");
        printf(DAT3 "\nVentil = zu");
    }
    elseif ($sv == 2) {

```

```

        printf("\nVentil = oeffnet nicht");
        printf(DAT3 "\nVentil = oeffnet nicht");
    }
    if ($sp1 == 0) {
        printf("\nPumpe 1 = an");
        printf(DAT3 "\nPumpe 1 = an");
    }
    elseif ($sp1 == 3) {
        printf("\nPumpe 1 = stoppt nicht");
        printf(DAT3 "\nPumpe 1 = stoppt nicht");
    }
    if ($sp2 == 1) {
        printf("\nPumpe 2 = aus");
        printf(DAT3 "\nPumpe 2 = aus");
    }
    elseif ($sp2 == 2) {
        printf("\nPumpe 2 = startet nicht");
        printf(DAT3 "\nPumpe 2 = startet nicht");
    }
    printf("\nFH = %s", $FH );
    printf("\nT = %s", $t );
    printf("\n");

    printf(DAT3 "\nFH = %s", $FH );
    printf(DAT3 "\nT = %s", $t );
    printf(DAT3 "\n");
}
if ($Lsv == 1 && $Lsp1 == 1 && $Lsp2 == 0) {
    $FH = $FH + 0.05 * 5 * $dt;
    printf("\n");
    printf(DAT3 "\n");
    if ($sv == 1) {
        printf("\nVentil = zu");
        printf(DAT3 "\nVentil = zu");
    }
    elseif ($sv == 2) {
        printf("\nVentil = oeffnet nicht");
        printf(DAT3 "\nVentil = oeffnet nicht");
    }
    if ($sp1 == 1) {
        printf("\nPumpe 1 = aus");
        printf(DAT3 "\nPumpe 1 = aus");
    }
    elseif ($sp1 == 2) {
        printf("\nPumpe 1 = startet nicht");
        printf(DAT3 "\nPumpe 1 = startet nicht");
    }
    if ($sp2 == 0) {
        printf("\nPumpe 2 = an");
        printf(DAT3 "\nPumpe 2 = an");
    }
    elseif ($sp2 == 3) {
        printf("\nPumpe 2 = stoppt nicht");
        printf(DAT3 "\nPumpe 2 = stoppt nicht");
    }
    printf("\nFH = %s", $FH );
    printf("\nT = %s", $t );
    printf("\n");

    printf(DAT3 "\nFH = %s", $FH );
    printf(DAT3 "\nT = %s", $t );
    printf(DAT3 "\n");
}
if ($Lsv == 1 && $Lsp1 == 0 && $Lsp2 == 0) {
    $FH = $FH + 0.15 * 5 * $dt;
    if ($sv == 1) {
        printf("\nVentil = zu");
        printf(DAT3 "\nVentil = zu");
    }
    elseif ($sv == 2) {
        printf("\nVentil = oeffnet nicht");
        printf(DAT3 "\nVentil = oeffnet nicht");
    }
    if ($sp1 == 0) {
        printf("\nPumpe 1 = an");
        printf(DAT3 "\nPumpe 1 = an");
    }
}

```

```

elseif ($sp1 == 3) {
    printf("\nPumpe 1 = stoppt nicht");
    printf(DAT3 "\nPumpe 1 = stoppt nicht");
}
if ($sp2 == 0) {
    printf("\nPumpe 2 = an");
    printf(DAT3 "\nPumpe 2 = an");
}
elseif ($sp2 == 3) {
    printf("\nPumpe 2 = stoppt nicht");
    printf(DAT3 "\nPumpe 2 = stoppt nicht");
}
printf("\nFH = %s", $FH );
printf("\nT = %s", $t );
printf("\n");

printf(DAT3 "\nFH = %s", $FH );
printf(DAT3 "\nT = %s", $t );
printf(DAT3 "\n");
}

#Modellierung des Regelsystems
#Maple 0) nicht erfolgreich 1) erfolgreich

if ($FH > $a2 && $sv == 1) {
    printf("\n \n \n \n");
    system("cmaple9 VentilOeffnen-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1, "E:/Perl-HoldUpTankBeispiel/VentilOeffnen.txt");
    while (<DAT1>) {
        eval ($_);
    }
    chop;
    ($Variable, $Erfolgreich) = split(/:/, $_, 2);
}
printf("\n");
printf(DAT3 "\n");
printf(DAT3 "\n");
if ($Erfolgreich == 0) {
    $sv = 2;
    printf("\nErfolgreich = %s", $Erfolgreich);
    printf("\nVentil wird nicht erfolgreich geöffnet");
    printf("\nVentil = oeffnet nicht");

    printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
    printf(DAT3 "\nVentil wird nicht erfolgreich geöffnet");
    printf(DAT3 "\nVentil = oeffnet nicht");

}
elseif ($Erfolgreich == 1) {
    $sv = 0;
    printf("\nErfolgreich = %s", $Erfolgreich);
    printf("\nVentil wird erfolgreich geöffnet");
    printf("\nVentil = auf");

    printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
    printf(DAT3 "\nVentil wird erfolgreich geöffnet");
    printf(DAT3 "\nVentil = auf");

}
printf("\nFH = %s", $FH );
printf("\nT = %s", $t );
printf("\n");

printf(DAT3 "\nFH = %s", $FH );
printf(DAT3 "\nT = %s", $t );
printf(DAT3 "\n");
}

if ($FH > $a2 && $sp1 == 0) {
    printf("\n \n \n \n");
    system("cmaple9 PumpeStoppen-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1, "E:/Perl-HoldUpTankBeispiel/PumpeStoppen.txt");
}

```

```

while (<DAT1>) {
    eval ($_);
chop;
($Variable,$Erfolgreich) = split(/:/, $_, 2);
}
printf("\n");
printf(DAT3 "\n");
if ($Erfolgreich == 0) {
    $sp1 = 3;
    printf("\nErfolgreich = %s",$Erfolgreich);
    printf("\nPumpe 1 wird nicht erfolgreich ausgeschaltet");
    printf("\nPumpe 1 = stoppt nicht");

    printf(DAT3 "\nErfolgreich = %s",$Erfolgreich);
    printf(DAT3 "\nPumpe 1 wird nicht erfolgreich ausgeschaltet");
    printf(DAT3 "\nPumpe 1 = stoppt nicht");

}
elseif ($Erfolgreich == 1) {
    $sp1 = 1;
    printf("\nErfolgreich = %s",$Erfolgreich);
    printf("\nPumpe 1 wird erfolgreich ausgeschaltet");
    printf("\nPumpe 1 = aus");

    printf(DAT3 "\nErfolgreich = %s",$Erfolgreich);
    printf(DAT3 "\nPumpe 1 wird erfolgreich ausgeschaltet");
    printf(DAT3 "\nPumpe 1 = aus");
}
printf("\nFH = %s",$FH );
printf("\nT = %s",$t );
printf("\n");

printf(DAT3 "\nFH = %s",$FH );
printf(DAT3 "\nT = %s",$t );
printf(DAT3 "\n");
}

if ($FH > $a2 && $sp2 == 0) {
    printf("\n \n \n \n");
    system("cmaple9 PumpeStoppen-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1,"E:/Perl-HoldUpTankBeispiel/PumpeStoppen.txt");
    while (<DAT1>) {
        eval ($_);
chop;
($Variable,$Erfolgreich) = split(/:/, $_, 2);
}
printf("\n");
printf(DAT3 "\n");
if ($Erfolgreich == 0) {
    $sp2 = 3;
    printf("\nErfolgreich = %s",$Erfolgreich);
    printf("\nPumpe 2 wird nicht erfolgreich ausgeschaltet");
    printf("\nPumpe 2 = stoppt nicht");

    printf(DAT3 "\nErfolgreich = %s",$Erfolgreich);
    printf(DAT3 "\nPumpe 2 wird nicht erfolgreich ausgeschaltet");
    printf(DAT3 "\nPumpe 2 = stoppt nicht");
}
elseif ($Erfolgreich == 1) {
    $sp2 = 1;
    printf("\nErfolgreich = %s",$Erfolgreich);
    printf("\nPumpe 2 wird erfolgreich ausgeschaltet");
    printf("\nPumpe 2 = aus");

    printf(DAT3 "\nErfolgreich = %s",$Erfolgreich);
    printf(DAT3 "\nPumpe 2 wird erfolgreich ausgeschaltet");
    printf(DAT3 "\nPumpe 2 = aus");
}
printf("\nFH = %s",$FH );
printf("\nT = %s",$t );
printf("\n");

printf(DAT3 "\nFH = %s",$FH );
printf(DAT3 "\nT = %s",$t );

```

```

    printf(DAT3 "\n");
}

if ($FH < $a1 && $sv == 0) {
    printf("\n \n \n \n");
    system("cmaple9 VentilSchliessen-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1,"E:/Perl-HoldUpTankBeispiel/VentilSchliessen.txt");
    while (<DAT1>) {
        eval ($_);
    }
    chop;
    ($Variable,$Erfolgreich) = split(/:/, $_, 2);
}
printf("\n");
printf(DAT3 "\n");
if ($Erfolgreich == 0) {
    $sv = 3;
    printf("\nErfolgreich = %s",$Erfolgreich);
    printf("\nVentil wird nicht erfolgreich zugemacht");
    printf("\nventil = schliesst nicht");

    printf(DAT3 "\nErfolgreich = %s",$Erfolgreich);
    printf(DAT3 "\nVentil wird nicht erfolgreich zugemacht");
    printf(DAT3 "\nVentil = schliesst nicht");
}
elseif ($Erfolgreich == 1) {
    $sv = 1;
    printf("\nErfolgreich = %s",$Erfolgreich);
    printf("\nVentil wird erfolgreich zugemacht");
    printf("\nventil = zu");

    printf(DAT3 "\nErfolgreich = %s",$Erfolgreich);
    printf(DAT3 "\nVentil wird erfolgreich zugemacht");
    printf(DAT3 "\nVentil = zu");
}
printf("\nFH = %s",$FH );
printf("\nT = %s",$t );
printf("\n");

printf(DAT3 "\nFH = %s",$FH );
printf(DAT3 "\nT = %s",$t );
printf(DAT3 "\n");
}

if ($FH < $a1 && $spl == 1) {
    printf("\n \n \n \n");
    system("cmaple9 PumpeStarten-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1,"E:/Perl-HoldUpTankBeispiel/PumpeStarten.txt");
    while (<DAT1>) {
        eval ($_);
    }
    chop;
    ($Variable,$Erfolgreich) = split(/:/, $_, 2);
}
printf("\n");
printf(DAT3 "\n");
if ($Erfolgreich == 0) {
    $spl = 2;
    printf("\nErfolgreich = %s",$Erfolgreich);
    printf("\nPumpe 1 wird nicht erfolgreich angeschaltet");
    printf("\nPumpe 1 = startet nicht");

    printf(DAT3 "\nErfolgreich = %s",$Erfolgreich);
    printf(DAT3 "\nPumpe 1 wird nicht erfolgreich angeschaltet");
    printf(DAT3 "\nPumpe 1 = startet nicht");
}
elseif ($Erfolgreich == 1) {
    $spl = 0;
    printf("\nErfolgreich = %s",$Erfolgreich);
    printf("\nPumpe 1 wird erfolgreich angeschaltet");
    printf("\nPumpe 1 = an");

    printf(DAT3 "\nErfolgreich = %s",$Erfolgreich);
    printf(DAT3 "\nPumpe 1 wird erfolgreich angeschaltet");
}

```

```

        printf(DAT3 "\nPumpe 1 = an");

    }
    printf("\nFH = %s", $FH );
    printf("\nT = %s", $t );
    printf("\n");

    printf(DAT3 "\nFH = %s", $FH );
    printf(DAT3 "\nT = %s", $t );
    printf(DAT3 "\n");
}

if ($FH < $a1 && $sp2 == 1) {
    printf("\n \n \n \n");
    system("cmaple9 PumpeStarten-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1, "E:/Perl-HoldUpTankBeispiel/PumpeStarten.txt");
    while (<DAT1>) {
        eval ($_);
    }
    chop;
    ($Variable, $Erfolgreich) = split(/:/, $_, 2);
    }
    printf("\n");
    printf(DAT3 "\n");
    if ($Erfolgreich == 0) {
        $sp2 = 2;
        printf("\nErfolgreich = %s", $Erfolgreich);
        printf("\nPumpe 2 wird nicht erfolgreich angeschaltet");
        printf("\nPumpe 2 = startet nicht");

        printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
        printf(DAT3 "\nPumpe 2 wird nicht erfolgreich angeschaltet");
        printf(DAT3 "\nPumpe 2 = startet nicht");
    }
    elsif ($Erfolgreich == 1) {
        $sp2 = 0;
        printf("\nErfolgreich = %s", $Erfolgreich);
        printf("\nPumpe 2 wird erfolgreich angeschaltet");
        printf("\nPumpe 2 = an");

        printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
        printf(DAT3 "\nPumpe 2 wird erfolgreich angeschaltet");
        printf(DAT3 "\nPumpe 2 = an");
    }
    }
    printf("\nFH = %s", $FH );
    printf("\nT = %s", $t );
    printf("\n");

    printf(DAT3 "\nFH = %s", $FH );
    printf(DAT3 "\nT = %s", $t );
    printf(DAT3 "\n");
}

if ($FH >= $a1 && $FH <= $a2 && $sv == 0 && $spl == 1) {
    printf("\n \n \n \n");
    system("cmaple9 PumpeStarten-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1, "E:/Perl-HoldUpTankBeispiel/PumpeStarten.txt");
    while (<DAT1>) {
        eval ($_);
    }
    chop;
    ($Variable, $Erfolgreich) = split(/:/, $_, 2);
    }
    printf("\n");
    printf(DAT3 "\n");
    if ($Erfolgreich == 0) {
        $spl = 2;
        printf("\nErfolgreich = %s", $Erfolgreich);
        printf("\nPumpe 1 wird nicht erfolgreich angeschaltet");
        printf("\nPumpe 1 = startet nicht");

        printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
        printf(DAT3 "\nPumpe 1 wird nicht erfolgreich angeschaltet");
        printf(DAT3 "\nPumpe 1 = startet nicht");
    }
}

```

```

elseif ($Erfolgreich == 1) {
    $spl = 0;
    printf("\nErfolgreich = %s", $Erfolgreich);
    printf("\nPumpe 1 wird erfolgreich angeschaltet");
    printf("\nPumpe 1 = an");

    printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
    printf(DAT3 "\nPumpe 1 wird erfolgreich angeschaltet");
    printf(DAT3 "\nPumpe 1 = an");
}
printf("\nFH = %s", $FH );
printf("\nT = %s", $t );
printf("\n");

printf(DAT3 "\nFH = %s", $FH );
printf(DAT3 "\nT = %s", $t );
printf(DAT3 "\n");
}

if ($FH >= $a1 && $FH <= $a2 && $sv == 1 && $spl == 0) {
    printf("\n \n \n \n");
    system("cmaple9 VentilOeffnen-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1, "E:/Perl-HoldUpTankBeispiel/VentilOeffnen.txt");
    while (<DAT1>) {
        eval ($_);
    }
    chop;
    ($Variable, $Erfolgreich) = split(/:/, $_, 2);
}
printf("\n");
printf(DAT3 "\n");
if ($Erfolgreich == 0) {
    $sv = 2;
    printf("\nErfolgreich = %s", $Erfolgreich);
    printf("\nVentil wird nicht erfolgreich geöffnet");
    printf("\nVentil = öffnet nicht");

    printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
    printf(DAT3 "\nVentil wird nicht erfolgreich geöffnet");
    printf(DAT3 "\nVentil = öffnet nicht");
}
elseif ($Erfolgreich == 1) {
    $sv = 0;
    printf("\nErfolgreich = %s", $Erfolgreich);
    printf("\nVentil wird erfolgreich geöffnet");
    printf("\nVentil = auf");

    printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
    printf(DAT3 "\nVentil wird erfolgreich geöffnet");
    printf(DAT3 "\nVentil = auf");
}
printf("\nFH = %s", $FH );
printf("\nT = %s", $t );
printf("\n");

printf(DAT3 "\nFH = %s", $FH );
printf(DAT3 "\nT = %s", $t );
printf(DAT3 "\n");
}

if ($FH >= $a1 && $FH <= $a2 && $spl == 0 && $sp2 == 0) {
    printf("\n \n \n \n");
    system("cmaple9 PumpeStoppen-Maple.txt");
    sleep(1);
    printf("\n \n \n \n");
    open(DAT1, "E:/Perl-HoldUpTankBeispiel/PumpeStoppen.txt");
    while (<DAT1>) {
        eval ($_);
    }
    chop;
    ($Variable, $Erfolgreich) = split(/:/, $_, 2);
}
printf("\n");
printf(DAT3 "\n");
if ($Erfolgreich == 0) {
    $sp2 = 3;

```

```

printf("\nErfolgreich = %s", $Erfolgreich);
printf("\nPumpe 2 wird nicht erfolgreich ausgeschaltet");
printf("\nPumpe 2 = stoppt nicht");

printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
printf(DAT3 "\nPumpe 2 wird nicht erfolgreich ausgeschaltet");
printf(DAT3 "\nPumpe 2 = stoppt nicht");
}
elseif ($Erfolgreich == 1) {
    $sp2 = 1;
    printf("\nErfolgreich = %s", $Erfolgreich);
    printf("\nPumpe 2 wird erfolgreich ausgeschaltet");
    printf("\nPumpe 2 = aus");

    printf(DAT3 "\nErfolgreich = %s", $Erfolgreich);
    printf(DAT3 "\nPumpe 2 wird erfolgreich ausgeschaltet");
    printf(DAT3 "\nPumpe 2 = aus");
}
printf("\nFH = %s", $FH );
printf("\nT = %s", $t );
printf("\n");

printf(DAT3 "\nFH = %s", $FH );
printf(DAT3 "\nT = %s", $t );
printf(DAT3 "\n");
}

$Wasserfuellstand[$lauf][$t] = $FH;
printf(DAT14 "%s ", $Wasserfuellstand[$lauf][$t] );
}

printf(DAT14 "\n");

printf(DAT2 "\nLauf = %s", $lauf );
printf(DAT2 "\nFH = %s", $FH );
printf(DAT2 "\nT = %s", $t );
printf(DAT2 "\n");

#Simulationsergebnisse für den DRYOUT-Zustand
if ($FH <= 0 ) {
    $DZ = $DZ + 1;
    $DDryout[$t] = $DDryout[$t] + 1;
    printf(DAT5 "%s", $t );
    printf(DAT7 "\nLauf = %s", $lauf );
    printf(DAT7 "\nFH = %s", $FH );
    printf(DAT7 "\nT = %s", $t );
    printf(DAT7 "\n");
}

#Simulationsergebnisse für den OVERFLOW-Zustand
if ($FH >= 10 ) {
    $OZ = $OZ + 1;
    $DOverflow[$t] = $DOverflow[$t] + 1;
    printf(DAT6 "%s", $t );
    printf(DAT8 "\nLauf = %s", $lauf );
    printf(DAT8 "\nFH = %s", $FH );
    printf(DAT8 "\nT = %s", $t );
    printf(DAT8 "\n");
}
}

printf(DAT8 "\n");
printf("\nZ(Dryout) = %s", $DZ );
printf("\nP(Dryout) = %s", $DZ/$LAUF );
printf("\n");
printf("\nZ(Overflow) = %s", $OZ );
printf("\nP(Overflow) = %s", $OZ/$LAUF );
printf("\n");

printf(DAT8 "\n");
printf(DAT7 "\nZ(Dryout) = %s", $DZ );
printf(DAT7 "\nP(Dryout) = %s", $DZ/$LAUF );
printf(DAT7 "\n");

```

```

printf(DAT8 "\nZ(Overflow) = %s", $OZ );
printf(DAT8 "\nP(Overflow) = %s", $OZ/$LAUF );
printf(DAT8 "\n");

#Eintrittswahrscheinlichkeit des Dryout-Zustands und des Overflow-Zustandes innerhalb
von der Simulationszeit
$DSpeicher = 0;
for ($n = 1; $n <= $Simulationszeit; $n++) {
    if ($DZ != 0) {
        $DSpeicher = $DSpeicher + $DDryout[$n];
        printf(DAT9 "\n%d %e", $n, $DDryout[$n]/$LAUF);
        printf(DAT11 "\n%d %e", $n, $DSpeicher/$LAUF);
    }
}
$OSpeicher = 0;
for ($n = 1; $n <= $Simulationszeit; $n++) {
    if ($OZ != 0) {
        $OSpeicher = $OSpeicher + $DOverflow[$n];
        printf(DAT10 "\n%d %e", $n, $DOverflow[$n]/$LAUF);
        printf(DAT12 "\n%d %e", $n, $OSpeicher/$LAUF);
    }
}

#Wasserfüllstand innerhalb von der Simulationszeit
for ($m = 1; $m <= $Simulationszeit; $m++) {
    for ($n = 1; $n <= $LAUF; $n++) {

printf(DAT13 "%s ", $Wasserfuellstand[$n][$m] );

    }
    printf(DAT13 "\n" );
}

#Rechenzeit
$Jetztwert = time();
$Jetztzeit = localtime($Jetztwert);
printf("\n \nSimulation endet: %s\n \n", $Jetztzeit );
printf("\n \n");

```

Anhang B - Modellierung der zufälligen Komponentenausfälle im „Hold-Up-Tank“-Systems

```
with(stats):
Seed:=readlib(randomize)():

#zufaellige Zeit der Zustandsaenderung
Ventil:=statevalf[icdf,exponential[0.005]]:
#plot(Ventil(x),x=0.0..1.0);
#Zufallszahl:=random[uniform[0,1]](1);
TVentil:=Ventil(Zufallszahl);
TVentil:=ceil(TVentil);

Pumpe1:=statevalf[icdf,exponential[0.001]]:
#plot(Pumpe1(x),x=0.0..1.0);
#Zufallszahl:=random[uniform[0,1]](1);
TPumpe1:=Pumpe1(Zufallszahl);
TPumpe1:=ceil(TPumpe1);
TPumpe1:=10000;

Pumpe2:=statevalf[icdf,exponential[0.001]]:
#plot(Pumpe2(x),x=0.0..1.0);
#Zufallszahl:=random[uniform[0,1]](1);
TPumpe2:=Pumpe2(Zufallszahl);
TPumpe2:=ceil(TPumpe2);

DATEN1:=fopen("E:/Perl- HoldUpTankBeispiel/ZufallZeit.txt",WRITE,BINARY):
fprintf(DATEN1,"Ventil:%d \n",TVentil):
fprintf(DATEN1,"Pumpe1:%d \n",TPumpe1):
fprintf(DATEN1,"Pumpe2:%d \n",TPumpe2):
fclose(DATEN1):
```

Anhang C - Modellierung des zufälligen Ausfalls des Regelungssystems im „Hold-Up-Tank“-System

```
with(stats):
Seed:=readlib(randomize)():

#Wahrscheinlichkeit, dass Pumpe1/Pumpe2 erfolgreich angeschaltet werden kann, betraegt
0.95
#Wahrscheinlichkeit, dass Pumpe1/Pumpe2 nicht erfolgreich angeschaltet werden kann,
betraegt 0.05
#0)Pumpe kann nicht erfolgreich angeschaltet werden
#1)Pumpe kann erfolgreich angeschaltet werden
f:=x->piecewise(x=0,0,x>0 and x<0.05,0,x=0.05,1, x>0.05 and x<1,1,x=1, 1);
#plot(f(x),x=0..1);
Zufallszahl:=random[uniform[0,1]](1);
Erfolgreich:=f(Zufallszahl);

DATEN1:=fopen("E:/Perl-HoldUpTankBeispiel/PumpeStarten.txt",WRITE,BINARY):
fprintf(DATEN1,"Erfolgreich:%d \n",Erfolgreich):
fclose(DATEN1):

with(stats):
Seed:=readlib(randomize)():

#Wahrscheinlichkeit, dass Pumpe1/Pumpe2 erfolgreich ausgeschaltet werden kann, betraegt
0.97
#Wahrscheinlichkeit, dass Pumpe1/Pumpe2 nicht erfolgreich ausgeschaltet werden kann,
betraegt 0.03
#0)Pumpe kann nicht erfolgreich ausgeschaltet werden
#1)Pumpe kann erfolgreich ausgeschaltet werden
f:=x->piecewise(x=0,0,x>0 and x<0.03,0,x=0.03,1, x>0.03 and x<1,1,x=1, 1);
#plot(f(x),x=0..1);
Zufallszahl:=random[uniform[0,1]](1);
Erfolgreich:=f(Zufallszahl);

DATEN1:=fopen("E:/Perl-HoldUpTankBeispiel/PumpeStoppen.txt",WRITE,BINARY):
fprintf(DATEN1,"Erfolgreich:%d \n",Erfolgreich):
fclose(DATEN1):

with(stats):
Seed:=readlib(randomize)():

#Wahrscheinlichkeit, dass das Ventil erfolgreich geöffnet werden kann, betraegt 0.92
#Wahrscheinlichkeit, dass das Ventil nicht erfolgreich geöffnet werden kann, betraegt
0.08
#0)Ventil kann nicht erfolgreich geöffnet werden
#1)Ventil kann erfolgreich geöffnet werden
f:=x->piecewise(x=0,0,x>0 and x<0.08,0,x=0.08,1, x>0.08 and x<1,1,x=1, 1);
#plot(f(x),x=0..1);
Zufallszahl:=random[uniform[0,1]](1);
Erfolgreich:=f(Zufallszahl);

DATEN1:=fopen("E:/Perl-HoldUpTankBeispiel/VentilOeffnen.txt",WRITE,BINARY):
fprintf(DATEN1,"Erfolgreich:%d \n",Erfolgreich):
fclose(DATEN1):

with(stats):
Seed:=readlib(randomize)():

#Wahrscheinlichkeit, dass das Ventil erfolgreich zugemacht werden kann, betraegt 0.95
#Wahrscheinlichkeit, dass das Ventil nicht erfolgreich zugemacht werden kann, betraegt
0.05
#0)Ventil kann nicht erfolgreich zugemacht werden
#1)Ventil kann erfolgreich zugemacht werden
f:=x->piecewise(x=0,0,x>0 and x<0.05,0,x=0.05,1, x>0.05 and x<1,1,x=1, 1);
#plot(f(x),x=0..1);
```

```
Zufallszahl:=random[uniform[0,1]](1);  
Erfolgreich:=f(Zufallszahl);
```

```
DATEN1:=fopen("E:/Perl-HoldUpTankBeispiel/VentilSchliessen.txt",WRITE,BINARY):  
fprintf(DATEN1,"Erfolgreich:%d \n",Erfolgreich):  
fclose(DATEN1):
```