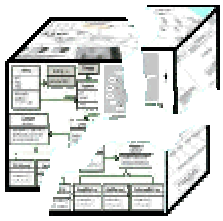
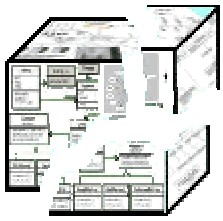


Kurzerklärung Markov-Modellen in der Sicherheitstechnik

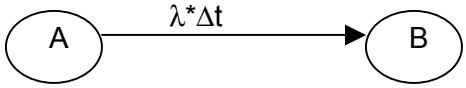
Versions-Nr.:	1.00
Erstellungsdatum:	2003-07-02
Verfasser:	Stefan Kuhn für www.break-it-down.de



Kurzerklärung Markov-Modellen in der Sicherheitstechnik

Markov-Modelle können für vielfältige Einsatzgebiete verwendet werden. Häufig ist der Einsatz bei der Modellierung und Berechnung biologischer, chemischer und physikalischer Prozesse.

Ein Markov-System für sicherheitsrelevante Systeme spiegelt die Hardware in den Markov-Zuständen und die möglichen Fehler in den Zustandsübergängen wieder. Dazu kommen noch Werte wie die Anforderungsrate, die Reparaturrate und die Zeitspanne zwischen den Tests (wie Online-Test und Power-On Test).

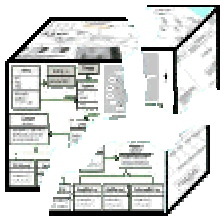


Voraussetzungen:

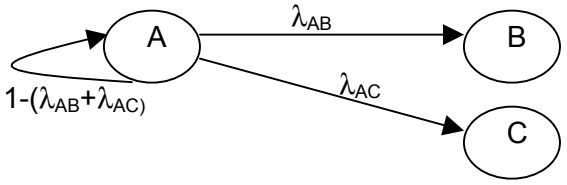
- Die Zustände sind exklusiv. Das Markov-System kann sich zu einem Zeitpunkt immer nur in genau einem Zustand befinden.
- Der Übergang von einem Zustand in einen anderen erfolgt mit fester (unveränderlicher) Wahrscheinlichkeit (λ) in einem festen Zeitraum (Δt). Als Zeitraum wird meist eine Stunde verwendet (Δt=1h), die Fehlerraten sind dann Ausfallwahrscheinlichkeit pro Stunde (z.B. FIT Werte der Bauteile).
- Δt muss für alle Übergänge gleich groß sein. Die Angabe kann somit auch weggelassen werden, jedoch muss generell bekannt sein, wie groß Δt ist.
- Die Zustände werden als Kreise gezeichnet.
- Die Übergänge werden als Pfeile gezeichnet.

Hinweise:

- Im Programm wird unter `Markov-Startparamter` mit `Anzahl Markov-Schritte` festgelegt, wie viele Δt Schritte berechnet werden sollen. Mit Δt=1h ergibt dies die totale Einsatzdauer in Stunden.



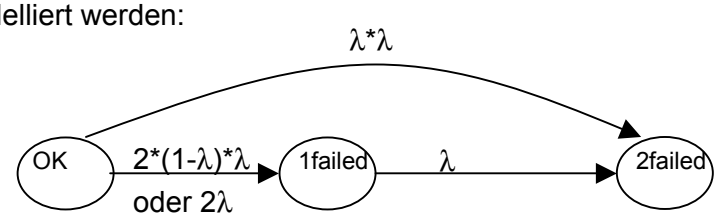
- Ergibt die Summe der Übergangswahrscheinlichkeiten von Zustand A in andere Zustände einen Wert kleiner 1.0, so kann ein Pfeil von A nach A gezeichnet werden, da mit einer gewissen Wahrscheinlichkeit (1.0 – Abgänge) im Zustand A verblieben wird. Diese Berechnung erfolgt automatisch durch das Programm.



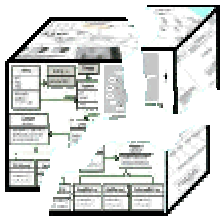
- Ergibt sich ein Wert größer 1.0 so ist das Markov-Modell nicht gültig, da die Übergangswahrscheinlichkeiten größer als 100% sind. Im Programm wird dies erkannt und angezeigt.
- Viele Markov-Modelle pendeln sich über die Zeit ein, d.h. diese bleiben nach einer bestimmten Anzahl Zyklen im Rahmen der Rechengenauigkeit unverändert. Auch dies wird durch das Programm angezeigt.

System Modellierung für Redundanz

- Redundante Systeme mit gemeinsamen Ausfällen (common cause) können wie folgt modelliert werden:

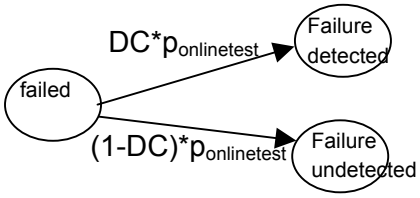


  Hier ist λ die Wahrscheinlichkeit, dass eine Komponente ausfällt und damit λ*λ die Wahrscheinlichkeit, dass beide Komponenten gleichzeitig ausfallen. Mit 2*(1-λ)*λ wird die Wahrscheinlichkeit berechnet, dass eine der Komponenten ausfällt und die andere nicht. Dies wird doppelt gewertet, da ja zwei Komponenten vorhanden sind.
- Der Term 2*(1-λ)*λ = 2λ - λ*λ kann vereinfacht werden, da λ meist sehr klein und somit λ*λ vernachlässigbar. Zudem ergibt dies eine WorstCase Berechnung, womit man auf der „sicheren Seite“ ist.

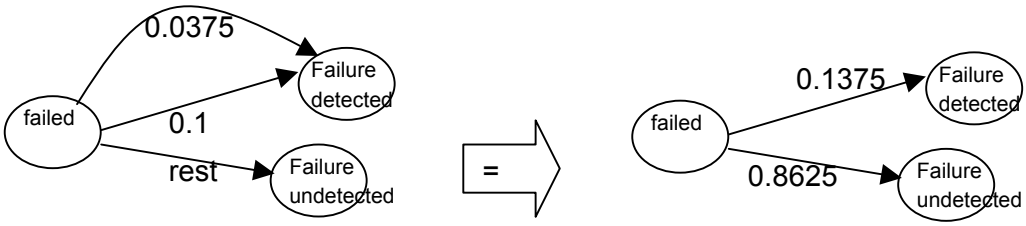


System Modellierung für Online-Tests

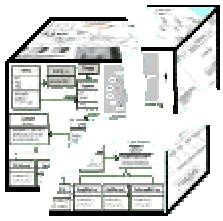
- Einige Komponenten des Systems werden vermutlich einem periodischem Online-Test unterzogen. Mittels Markov-Systemen kann die notwendige Häufigkeit der Tests und der notwendige Fehleraufdeckungsgrad der Tests (DC=diagnostic coverage) ermittelt werden.
- Die Wahrscheinlichkeit, dass im Zeitintervall Δt=1h ein Online-Test durchgeführt wird, kann berechnet werden mit: $p_{onlinetest}$=1 / Testintervall_in_Stunden. Für kleine Testintervalle (<1h) gilt $p_{onlinetest}$=1.0.
- Ob der Online-Test den Fehler erkennen kann oder nicht, wird über den Fehleraufdeckungsgrad DC angegeben: $p_{erkannt}$ =DC * $p_{onlinetest}$



- Auch Power-On Selbsttests können modelliert werden, falls diese zyklisch aufgerufen werden. Wenn das System z.B. alle 24h einen Selbsttest durchführen muss ist $p_{powerontest}$=1 / 24 = 0.042 .
  Im Bild dargestellt ist ein zyklischer OnlineTest, welcher mehrmals pro Stunde durchlaufen wird mit einem Fehleraufdeckungsgrad von 10% (1.0*0.10=0.10), sowie ein Power-On Test, welcher alle 24h durchlaufen wird mit einem DC von 90% (1/24 * 0.9 = 0.0375).



  Da die Auswirkung beider Tests gleich ist (Failure detected) werden die Übergänge im rechten Bild zusammengefasst. Zu erkennen ist, dass diese Konstellation zu einem insgesamt schlechten Fehlererkennungsgrad führt und vermutlich Ansatz für Verbesserung bietet.
  Anzumerken ist hier noch die Annahme, dass wenn ein Fehler durch den Test nicht erkannt wurde, dieser Fehler durch den gleichen Test auch in späteren Durchläufen nicht erkannt wird.

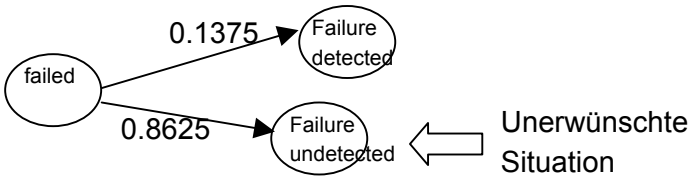


System Modellierung der Anforderung

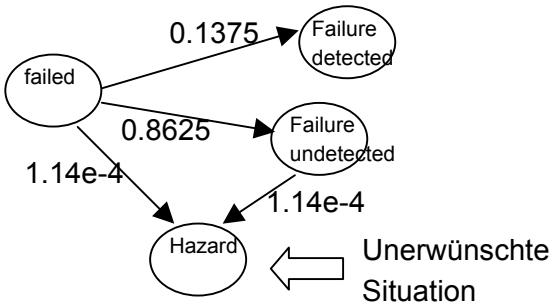
- Die IEC61508 spricht von zwei zu unterscheidenden Betriebsmodi, dem Low-Demand und dem High-Demand/Continuous Mode. In Markov-Modellen wird modelliert:

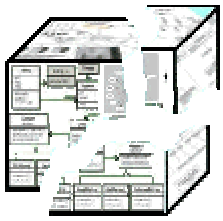
das System ist in einem unsicherem Zustand (z.B. failed der failure undetected)	+	eine Anforderung an das System erfolgt	=	Die unerwünschte/ gefährliche Situation tritt auf.

- Für Systeme mit Continuous-Mode ist die Anforderungsrate unendlich, das System erzeugt also im Fehlerfalle immer eine gefährliche Situation.



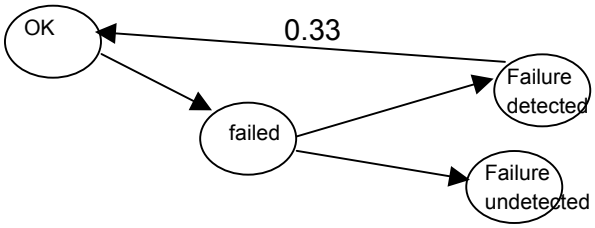
- Wird die Sicherheitsfunktion des Systems in einem Markov-Intervall (Δt) mehr als einmal angefordert, so gilt das obere Modell. Ist jedoch die Anforderungsrate viel kleiner so kann das Modell erweitert werden.
  Im Bild ist ein Low-Demand Notaus-System mit einer Anforderungsrate von einmal pro Jahr dargestellt (=1 / (365*24) = 1.14e-4).



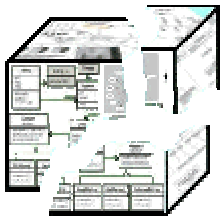


System Modellierung der Reparatur

- Das Markov-Modell sollte um die Reparaturrate erweitert werden. Falls das System einen Fehler erkennt und sicher darauf reagiert, wird das System einer Reparatur unterzogen. (Ist das nicht der Fall, so bleibt das System für den Rest der Einsatzdauer Außerbetrieb, was meist nicht der Fall ist).
  Im Beispiel wird das System im Falle eines Ausfalls innerhalb 3 Stunden repariert oder ausgetauscht $p_{reparatur} = 1 / 3 = 0.33$.

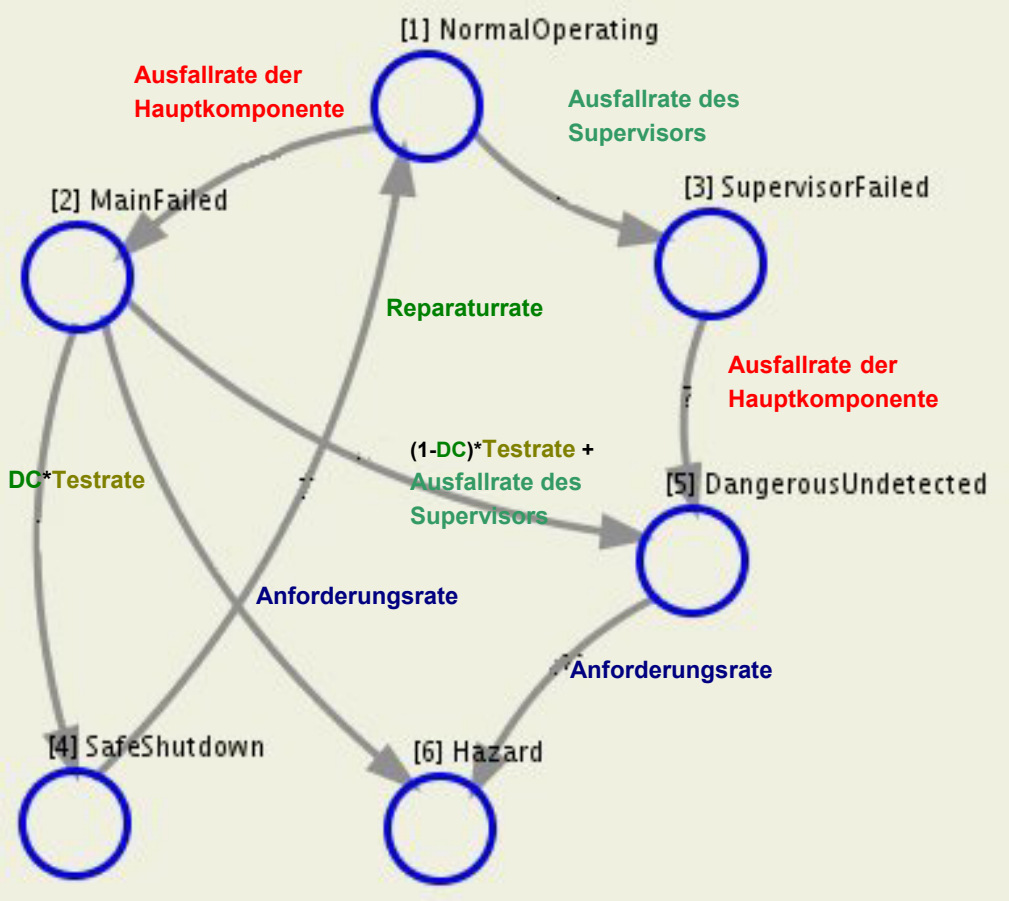


- Unerkannte Fehlerzustände werden nicht repariert. Es kann jedoch modelliert werden, das in Wartungsintervallen das System einem ausführlichem Test unterzogen wird, wo auch unerkannte Fehler zu einem gewissen Grad gefunden werden. Dies erfolgt dann analog zur Modellierung eines Power-On Tests.



Modellierung zur SIL Berechnung

- Hier soll ein einfaches 1oo1D System betrachtet werden. Es besteht aus einer Hauptkomponente und einem Supervisor (Watchdog). Die Fehlererkennungs-wahrscheinlichkeit durch den Watchdog ergibt den Diagnostic Coverage Faktor.



Ein gefährliche Situation (Hazard) tritt auf:

a) wenn die Hauptfunktion ausgefallen ist UND eine Anforderung an das System erfolgt;
b) wenn ein Fehler nicht durch den Supervisor erkannt wird (Zustand DangerousUndetected) UND eine Anforderung an das System erfolgt;
c) Ist die Anforderungsrate <=1h so gilt der Zustand DangerousUndetected als Hazard.