Submitted: IEEE ISSRE, Annapolis, November 2002

Predicting Reliability of an Embedded Vehicle System by modeling Coincident Failures and Usage-Profiles

Frederick T. Sheldon[1] and Kshamta Jerath

Software Engineering for Dependable Systems Laboratory©

School of EECS, Washington State University

Pullman, Washington 99164-2752, USA

sheldon@acm.org | kjerath@eecs.wsu.edu

RUNNING TITLE: Realistic Prediction of Component Reliability

AUTHORS' BIOGRAPHIES

Frederick T. Sheldon is an Assistant Professor at the Washington State University teaching and conducting research in the area of software engineering. His research is concerned with developing and validating methods and supporting tools for the creation of safe and correct software. Recent studies conducted at the SEDS laboratory (Software Engineering for Dependable Systems) have focused on verification and validation of systems using modeling and analysis of both logical and stochastic properties. The research has also investigated software evolution in the area of extensibility (maintainability and understandability).

Dr. Sheldon received his Ph.D. at the University of Texas at Arlington (UTA) and has worked at NASA Langley and Ames Research Centers in various capacities since 1993. Prior to that, he worked as a Software Engineer in the area of avionics and diagnostics software development for the YF-22, F-16 and Tornado aircraft programs at General Dynamics and Texas Instruments. He is a member of the IEEE Computer and Reliability Societies, ACM, AIAA, and The Tau Beta Pi and Upsilon Pi Epsilon.

---

[1] Sheldon (+49 711 174 1339 Office | +49 179 6675 9316 Handy) is currently on leave at DaimlerChrysler Research and Technology in System Safety, Stuttgart. This research was partially supported through a small grant from DaimlerChrysler (FT3/AS).

Kshamta Jerath is a Graduate Student at the Washington State University pursuing a master's degree in Computer Science. Her research interests include Software Engineering and Formal Methods (Stochastic Petri Nets) used in safety and reliability analysis. She is currently working as a Teaching Assistant for the Software Engineering course at WSU.

Ms. Jerath holds a bachelor's degree in Computer Engineering from Delhi College of Engineering, India and is a Sun Certified Java 2 Programmer. She was working as a software engineer with IBM India till December 2000 in the area of web application development and three-tier client server applications. She headed a testing team of developers carrying out regression testing on an e-commerce application.

Abstract

The increasingly ubiquitous use of software systems has increased the need to determine their reliability and the extent to which they can be depended upon. Structured models of systems allow us to do this, yet there are numerous challenges that need to be overcome to obtain meaningful results. This paper is an experiment to model and analyze the Anti-lock Braking System of a passenger vehicle using Stochastic Petri Nets. Special emphasis is laid on modeling extra-functional characteristics like coincident failures among components, severity of failure and usage-profiles of the system. Components generally interact with each other during operation, and a faulty component can affect the probability of failure of other components. The severity of a failure also has an impact on the operation of the system, as does the usage profile - failures which occur during active use of the system are the only failures considered (i.e., in reliability calculations). This paper gives emphasis to the importance of the extra-functional properties mentioned above, the challenges incurred in modeling, a detailed description of the models developed, and the results of the analysis carried out for realistically predicting the reliability of system components.

1. Introduction

The increasingly ubiquitous use of software systems has created the need of being able to depend on them more than before; and being able to measure how much one can depend on them. Knowing that the system is *reliable* is absolutely necessary for safety-critical systems, where any kind of failure may result in an unacceptable loss of human life.[2] Reliability is the probability that a system will deliver its intended functionality and quality for a specified period of "time" and under specified conditions, given that the system was functioning properly at the start of this "time" period (Vouk, 2000).

Structured models of reliability allow the reliability of a system to be derived from the reliabilities of its components, which are often easier to estimate or known before the system is even built (Littlewood and Strigini, 2000). Markov Models have been used successfully in numerous instances to specify and evaluate the reliability of systems. However, practical issues that stand in the way of developing such models include: (1) obtaining reliability data of components, (2) a simple model can capture limited interactions among components, (3) the need to estimate fault correlation between components, and (4) reliability depends on how the system is used, thus usage information is an important part of reliability evaluation.

1.1 Motivation

A complex system (like an embedded vehicle system) is composed of numerous components and the probability that the system survives (efficient or acceptable degraded operation) depends directly on each of the constituent components. The reliability analysis of a vehicle system can provide an understanding about the likelihood of failures occurring in the system and an increased insight to manufacturers about inherent "weaknesses." (Jerath and Sheldon, 2001)

[2] For example, the PEIT (Powertrain Equipped with Intelligent Technologies, IST-2000-29542) project has recently qualified for funding from the European Commission. The "X-by-wire" project objectives are to set up new technologies for powertrains to create a nearly "collision free" vehicle. Such a vehicle's powertrain will not only reactively cope with dangerous situations it will also be able to predict such a situation and thus prevent an accident (including failsafe intelligent energy management system for electric energy supply). See http://www.cordis.lu/ist/ka1/trans_tourism/projects/projects2.htm

If a system does not contain any redundancy – that is, if every component must function properly for the system to work – and if component failures are statistically independent, then the system reliability is simply the product of the component reliabilities. Furthermore, the failure rate of the system is the sum of the failure rates of the individual components (Siewiorek and Swarz, 1992). The assumption that failures occur independently (in a statistical sense) in hardware components is a widely used and often successful model for predicting the reliability of hardware devices. However, components generally interact with each other during operation, and a faulty component can affect the probability of failure of other components too (Balbo, 2000). Such failures are not **coincident** in the sense that they occur simultaneously, but in the fact that failure of one increases the probability of the failure of another.

Another aspect of modeling failures occurring in the system is their **severity**. Severity of a failure is the impact it has on the operation of the system. It is closely related to the threat the problem poses, in functional terms, to the correct operation of the system (Vouk, 2000). Severity is an important candidate to weight the data used in reliability calculations and must be incorporated into the model to determine the probability that the system survives, including efficient or acceptable degraded operation.

The reliability of a system also depends on its **usage profile** – users interact with the system in an intermittent fashion, resulting in operational workload profiles that alternate between periods of "Active" and "Passive" use. Reliability is concerned with the service that is actually delivered by the system as opposed to a system's *capacity* to deliver such service (Meyer, 2000). Specifically, while considering usage profiles, faults need not necessarily cause failures since they can be repaired; failures occurring during "active" use of the system *only* should contribute to reliability calculations.

In (Sheldon et al., 2000), the authors presented Stochastic Petri Net (SPN) models of a vehicle dynamic driving regulation (DDR) system. Subsystem representations of the Anti-lock Braking system (ABS), the Electronic Steering Assistance (ESA), the traction control (TC) and a

combined model were developed and analyzed for critical failures. In this paper, we focus on the Anti-lock Braking system and develop Stochastic Petri Net models to model the coincident failures of components, severity of failures and usage-profiles. Naturally this is but one component of the total system and the issue of scalability of this approach is a subject for future work.

1.2 Organization of Paper

This paper focuses on understanding and modeling the likelihood of a failure in the Anti-lock braking system of a passenger vehicle. Section 2 briefly describes the structural and functional aspects of an Anti-lock Braking System (ABS) and the Petri Net approach to modeling. The challenges faced in modeling, and the tools and environment used for modeling and analysis are also described briefly.

Section 3 presents the assumptions, SPN models and results for the Petri-nets modeling coincident failures and severity of failures in the ABS. The assumptions, SPN models and results for Petri-nets incorporating usage-profiles are presented in Section 4. Finally, the challenges faced in this study and the scope for future work are discussed in Section 5.

2. System Description and Modeling Approach

In this section, we briefly examine the structural composition of an Anti-lock Braking System and its functionality. Stochastic Petri Nets (SPNs) were used to model the system and the Stochastic Petri Net Package (SPNP) to analyze the models. The modeling and analysis approach is discussed later in this section.

2.1 Anti-lock Braking System

Anti-lock Braking System is an integrated part of the total braking system in a vehicle. Applying excessive pressure on the brake pedal, or panic slamming the brake pedal, can cause wheels to lock up and possibly send the vehicle careening into a terrifying skid. Excessive brake pedal pressure often occurs in an emergency or adverse situations, such as wet or icy roads (Kolsky,

1997). The ABS prevents wheel lockup during an emergency stop by modulating the brake pressure and permits the driver to maintain steering control while braking.

The ABS consists of the following major components (Nice, 2001):

- Wheel Speed Sensors: These measure wheel-speed and transmit information to an electronic control unit.
- Electronic Control Unit (Controller): This receives information from the sensors, determines when a wheel is about to lock up and controls the hydraulic control unit.
- Hydraulic Control Unit (Hydraulic Pump): This controls the pressure in the brake lines of the vehicle.
- Valves: Valves are present in the brake line of each brake and are controlled by the hydraulic control unit to regulate the pressure in the brake lines.

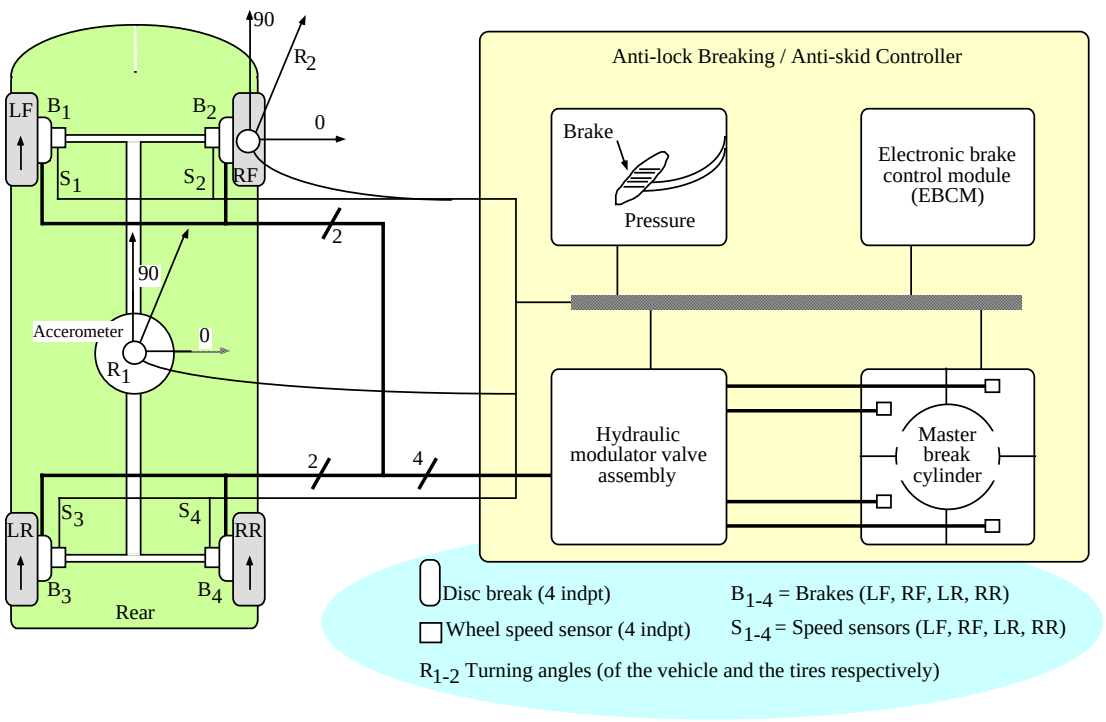

Figure 1: Top-level system schematic shows sensors, processing and actuators.

Figure 1 displays the top-level schematic of the system showing the interconnections between the components. Under braking, the electronic control unit (ECU) “reads” signals from electronic sensors monitoring wheel rotation. If a wheel’s rate of rotation suddenly decreases, the

ECU orders the hydraulic control unit (HCU) to reduce the line pressure to that wheel's brake. Once the wheel resumes normal operation, the controls restore pressure to its brake. Depending on the system, this cycle of "pumping" can occur at up to 15 times per second. The result is that the tire slows down at the same rate as the car, with the brakes keeping the tires very near the point at which they will start to lock up. This gives the system the highest steering capability.

Anti-lock braking systems use different schemes depending on the type of brake in use (Bosch, 1993): (1) Four channel, four sensor ABS – There is a speed sensor on all four wheels and a separate valve for all four wheels; (2) Three channel, three sensor ABS – There is a speed sensor and a valve for each of the front wheels with one speed sensor and valve for both rear wheels; (3) Two channel, two sensor ABS – There are two speed sensors and valves for each of the two rear wheels. In the model developed we assume a four channel four sensor ABS. The model can be easily modified to represent other ABS schemes.

2.2 Modeling and Analysis using SPNs

A powerful tool for modeling systems composed of several processes (such as a failure process and a repair process) is the Markov Model. Markov Models are a basic tool for both reliability and availability modeling. The two central concepts of this model are state and state transitions. The state of a system represents all that must be known to describe the system at that instant. For reliability models, each state represents a distinct combination of working and failed components. As time passes, the system goes from state to state as components fail and are repaired. These changes are called state transitions (Siewiorek and Swarz, 1992).

Stochastic Petri Nets (SPN) can be used to generate the (large) underlying Markov chain automatically starting from a concise description of the system. In such cases the SPN provides a high level interface for the specification of the underlying Markov model. Petri Nets are a powerful tool for the description and the analysis of systems that exhibit concurrency, synchronization and conflicts. Stochastic Petri Nets in which the basic model is augmented with time specifications are commonly used to evaluate the performance and reliability of complex

systems (Balbo, 2001). Stochastic Reward nets (SRNs) are SPNs augmented with the ability to specify output measures as reward-based functions, for the evaluation of reliability for complex systems (Muppala et al., 1994).

The graphical nature of SPNs lends itself to a more intuitive understanding of the system's inner workings and allows one to understand dependencies better. This enables one to identify conflicts and address localities where the overall system performance is more significantly affected. However, there are many challenges that need to be overcome in order to develop a meaningful model.

2.2.1 Challenges in modeling

Since the system we study here is very complex, this prevents us from making a direct analysis. A series of abstraction steps are needed to obtain system measures from the real system. Initially the system model is created at an abstract level and the data collected from system measurements are used to parameterize the abstract model. In the second abstraction step the computational model is created which allows an easier and more efficient system analysis (Sheldon and Greiner, 1999). The key element therefore in our modeling approach was to identify the essential components of the system, the different ways in which they interact and introduce various assumptions. The details of the models developed and the assumptions made are discussed in Sections 3 and 4.

Two distinct problems that arise while using SPNs are largeness and stiffness (Popstojanova and Trivedi, 2000). The size of a Markov Model for the evaluation of a system grows exponentially with the number of components in the system. If there are n components, the Markov Model may have up to 2^n states. This causes the analysis to take a great deal of time. Stiffness is due to the different orders of magnitude between the rates of failure-related events in different components. An approximate solution can be obtained by decomposing the original model into smaller sub-models, solving the sub-models in isolation and then combining the solutions into the solution of the original model. This doesn't work in our case, since we are

trying to model coincident failures and the original model cannot be decomposed into independent sub-models.

2.2.2 Tool and Environment

A number of tools for specification and analysis/simulation of stochastic processes exist today. Some of them are listed in Table 1. We described the models in CSPL (C-based Stochastic Petri net Language) and the stochastic analysis was carried out using SPNP (Stochastic Petri Net Package). SPNP is a versatile modeling tool which allows the specification of SPN reward models, the computation of steady state, transient, cumulative, time-averaged and "up-to-absorption" measures and the sensitivities of these measures (Ciardo et al., 1993). SPNP allows the prediction of the Mean Time to Failure (MTTF) of a system. The MTTF of a system is the expected time of the first system failure given successful startup at time zero.

Table 1: Overview of Stochastic Analysis/Simulation Tools

Tool	Description	Features	Environments
Möbius	A tool for building performance and dependability models of stochastic, discrete-event systems.	Graphical Editor, Atomic and Composite Model, Analytic Solvers, Discrete Event Simulator, Multiple Modeling Formalisms	Unix, MS Windows
Moses	An integrated, extendable tool suite for specifying concurrent systems with a range of modeling formalisms. High level Petri Nets, Stochastic Petri Nets and Petri Nets with time are supported	Graphical Editor, Token Game Animation, Fast Simulation, User-extendable	Sun Linux MS Windows Java
PACE	A widely used object-oriented simulator-development system based on high-level Petri nets with time modeling.	Graphical Editor, Token Game Animation, Fast Simulation, Net Reductions, Fuzzy Modeling	Sun MS Windows
PEP	A tool to model, simulate, analyze and verify parallel systems by combining Petri nets and Process algebras.	Graphical Editor, Token Game Animation, Condensed State Spaces, Net Reductions, Structural Analysis, Model checking, Petri Net Generators	Sun Linux
SPNP	A Petri Net tool based on GSPN-like formalism and Markov Reward Model.	Reachability Graph Construction, Transient and Steady-state performance and performability analysis	Unix
UltraSAN	A software package for model-based evaluation of systems represented as Stochastic Activity Networks.	Graphical Editor, Steady-state and transient simulation, Reduced Base Model Construction, Analytical solution	Sun, Unix

The transient analysis duration of the models developed was deliberately conservative. The period covered 50,000 hours even though the average life span of a passenger vehicle ranges

from 3000 – 9000 hours.[3] The models were solved using Version 6 of SPNP installed on a Sun Ultra 10 (400Mhz) with 500MB of memory (dedicated to solving the models). The models took approximately 5 days of continuous execution before converging to solution. This time may have been drastically reduced we believe had the Multi-level solution method been available within the SPNP package (Greiner and Horton, 1996).

3. Modeling Coincident Failures and Severity

The assumption that failures occur independently is a widely used and often successful model for predicting the reliability of hardware devices. However, components generally interact with each other during operation, and a faulty component can affect the probability of failure of other components too (Balbo, 2000). Severity of a failure is the impact it has on the operation of the system and is an important candidate to weight the data used in reliability calculations. In this section, we describe the Petri net models developed to model coincident failures and severity of failures for the Anti-lock Braking System.

3.1 Assumptions

In order to allow a Markov chain analysis, the time to failure of all components is assumed to have an exponential distribution. This signifies that the distribution of the remaining life of a component does not depend on how long the component has been operating. The component does not "age" or it forgets how long it has been operating, and its eventual breakdown is the result of some suddenly appearing failure, not of gradual deterioration (Trivedi, 1982). While this might be true for electronic components, the failure of other mechanical parts like valves might occur due to gradual deterioration. However, mechanical parts are generally replaced at regular intervals and essentially can be assumed not to age for our purposes. Hence, the assumption of an exponential distribution of failures for all components is justified. This assumption carries over to the models representing Usage-Profiles as well, as discussed in Section 4.1.

[3] Essentially the average hours of operation for a passenger vehicle per year range from 300-600 hours/year and the average lifetime is 10-15 years.

To consider the severity of failures, every component is assumed to operate in three modes: normal operation, degraded operation or causing loss of stability. The system is assumed to ***fail*** when more than five components function in a degraded state or, more than three components cause loss of stability; or the failure of an important component causes the loss of the vehicle. A component operating in a degraded condition causes its failure rate to increase by two orders of magnitude, while a component causing loss of stability causes the failure rate to increase by four orders of magnitude. The correlation between failure rates of two "related" components (to model coincident failures) is consistent with the above scheme.

Since the model is an abstraction of a real world problem, predictions based on the model must be validated against actual measurements collected from the real phenomena. A poor validation may suggest modifications to the original model (Trivedi, 1982).

3.2 Model

The ABS is represented as a combination of all the important components it consists of, as shown in Figure 2. It represents the operation of the ABS under normal, degraded and lost stability conditions. Loss of vehicle, extreme degraded operation and extreme loss of stability signify critical failures and determine the halting condition for the model. The model is instantiated with a single token in the *start* place. When the *central_op* and the *axle_op* transitions fire, a token is deposited in each place that represents a component of the ABS. The operation of each component is now independent of every other component (except where

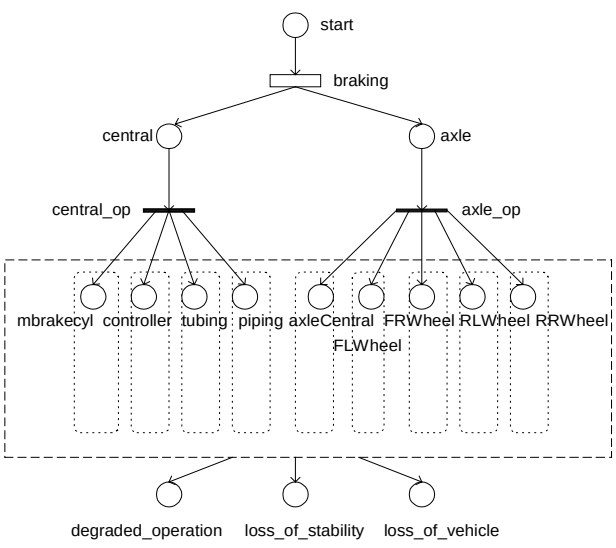

Figure 2: The ABS Model

coincident failures are modeled explicitly). The model of a component of the ABS is shown in Figure 3.

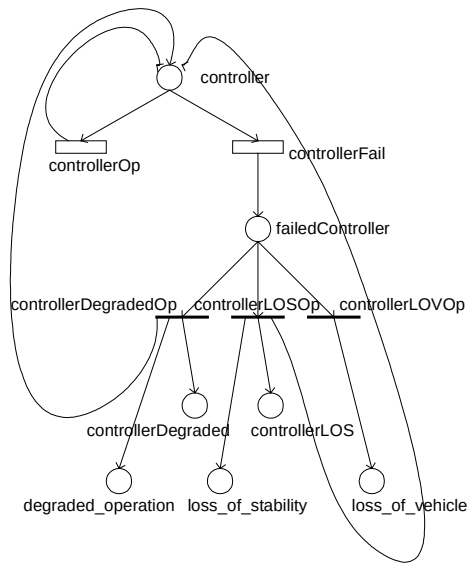

Figure 3: The SPN Model of a component

The component depicted here is the controller. Every component either functions "normally" as shown by the *controllerOp* transition or "fails" as shown by the *controllerFail* transition. A failed component may either cause degraded operation, loss of stability or loss of vehicle. The probability of any one of these three transitions occurring is different for each component. When the failure causes either degraded operation or loss of stability, the component continues to operate, though the failure rate increases by two and four orders of magnitude respectively.

Coincident failures are modeled in a similar manner. The rule for calculating failure rates is shown in Figure 4. The failure of a component A to a degraded mode causes the failure rate of a "related" component B to increase by two orders of magnitude. The failure of component A to a lost stability mode causes the failure rate of a "related" component B to increase by four orders of magnitude.

```
function failureRateForB()
{
        // other calculations for severity of failure

        // coincident failures
         if failedA(degraded) then
                        failureB = failureB * 100;
         else if failedA(loss of stability) then
                        failureB = failureB * 10000;
}
```

Figure 4: Rule for failure rates

The function that calculates the failure rate of the transition *controllerFail* is shown in Figure 5. It is assumed that tubing malfunction affects the operation of the controller. Hence, while calculating the failure rate of the controller, the normal rate is increased by two orders of magnitude if the tubing has failed causing degraded operation (indicated by a token in the *tubingDegraded* place).

```
double controllerRate()
{
    double controller_rate = 0.0000006;

    if (mark("controllerLOS") > 0) return controller_rate * 10000;
    if ((mark("controllerDegraded") > 0) || (mark("tubingDegraded") > 0))
        return controller_rate * 100;
    return controller_rate;
}
```

Figure 5: Variable rate to model coincident failures

Only a few coincident failures have been represented in the model. However, coincident failures between other components can be easily modeled by suitably modifying the failure rate function of the component in question using the rule shown in Figure 4. The model is easily extensible to include other components deemed relevant to the ABS.

3.3 Results and Discussion

The Stochastic Petri Net Package (SPNP) allows the computation of steady state, transient, cumulative, time-averaged, "up-to-absorption" measures and sensitivities of these measures. Steady-state analysis of SRNs is often adequate to study the performance of a system, but time-dependent behavior is sometimes of greater interest: instantaneous availability, interval availability, reliability, response time distribution, and computational availability. The reliability of the system at time *t* is computed as the expected instantaneous reward rate at time *t* (Muppala et al., 1994).

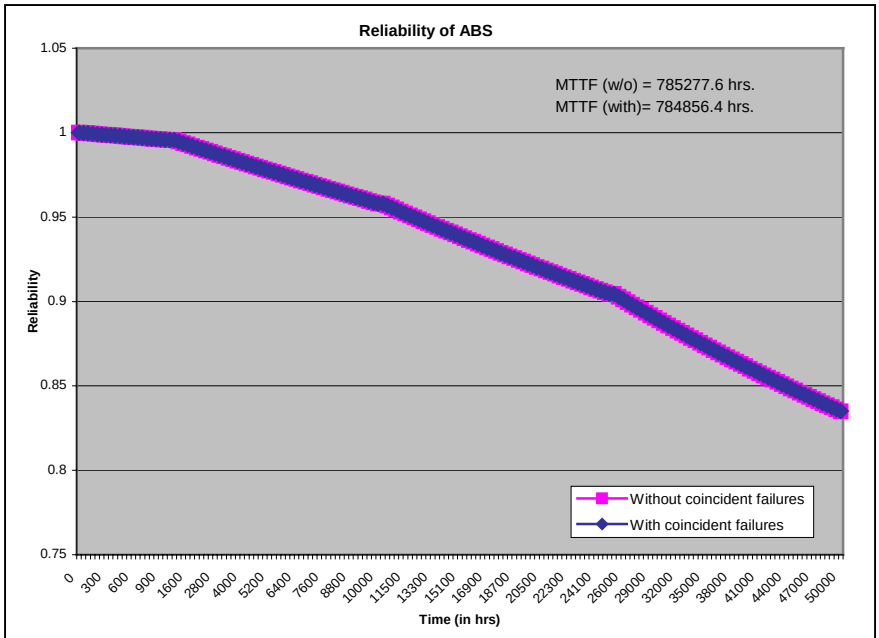

Figure 6: Reliability analysis results for coincident failures

Transient analysis of the ABS model was carried out and the reliability was measured between 0 and 50K hours. The expected values of reliability at various time instances were determined and plotted as a function of time. The measure was predicted at 169 different

points along the range. The interval between the points did not remain constant along the entire time range; instead the time range was divided into four segments. Each of these segments has a different time interval.

In Figure 6, the Y-axis gives the measure of interest - the reliability; while the time range (0 to 50K hours) is shown along the X-axis. The shape of the curve is not a property of the system but of how the data was collected from the Petri net model. As expected, the reliability steadily decreases with time. The blue line indicates the reliability function when coincident failures are modeled and the pink line indicates the reliability function when coincident failures are not modeled. For the limited number of coincident failures that were modeled, it is clear that the Mean Time to Failure (MTTF) for the model with coincident failures (784,856.4 hrs) is approximately 421 hours less than the model without coincident failures (785,277.6 hrs).

Figure 7 displays the difference between the two reliability functions more subtly. The reliability functions diverge starting around 350 hours of operation, and the difference becomes discernible after around 13K hours of operation. The difference continues to increase with time. It is significant to note that the difference in Mean Time To Failure between the two cases becomes marked only beyond the average lifetime of the vehicle. For the limited number of coincident failures that have

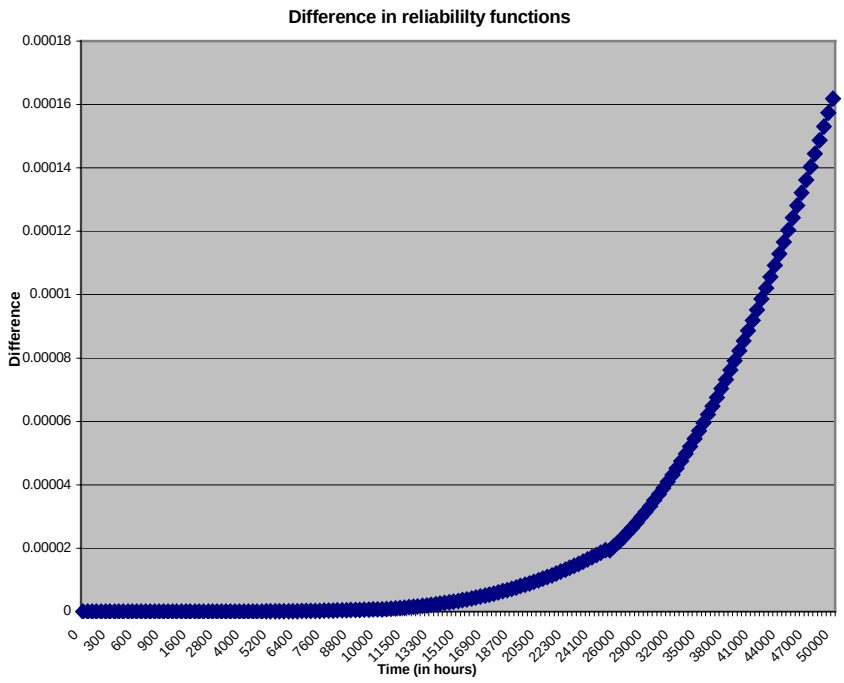

Figure 7: Difference in reliability functions

been modeled, the difference of 421 hours in the two cases is considered well within the confidence interval. However, it is evident that the model representing the coincident failures predicts the system reliability closer to the real picture.

4. Modeling Usage-Profiles

A software-based product's reliability depends on just how a customer will use it. The operational profile – quantitative characterization of how a system will be used – is essential in software reliability engineering (Musa, 1993). The same basic concept can be extended and applied for predicting the system reliability. We extend the idea of operational profiles – considering the use of a software system during testing; into usage profiles – the usage of the system (hardware and software) for modeling and reliability analysis. Reliability is concerned with the service that is *actually* delivered by the system as opposed to the system's *capacity* to deliver such service. The usage profile considers the intermittent use of a system – alternate periods of active and passive use. Such intermittent use influences the mean time to failure and reliability of the system (Meyer, 2000). In this section, we describe the Petri net models developed to model usage-profiles for the Anti-lock Braking System.

4.1 Assumptions

Unlike traditional reliability models where repair of components is not considered, when considering intermittent use it is important to note that faults need not necessarily cause failures. Faults occurring only during the active use cause failures while those occurring during passive use can be repaired. Hence repair can affect reliability calculations. For simplicity, we assume an infinite repair rate of all components.

Further, in order to comprehend the significance of intermittent use on reliability, we assume two usage-profiles exceedingly different in degree. The first profile models sparse use of the Anti-lock Braking System e.g. a driver who is extremely cautious while driving the vehicle (longer periods of passive use). The second usage profile models dense use of the anti-lock

braking system e.g. a driver in perilous conditions like driving over ice (frequent active use periods).

Again, for simplicity and to allow Markovian analysis, the active period duration is assumed to be exponentially distributed, as are the failure rates of the components. The second usage-profile is assumed to have a rate two orders of magnitude greater than the first usage profile. In order to work around the stiffness problem in Petri nets caused by the difference in magnitude between the failure rates of the components and the active period duration distribution rates, the duration distribution rates are assumed to be factored by the failure rates of individual components.

4.2 Model

In order to incorporate the usage-profiles scenario in the ABS model, the model of each individual component as depicted in Figure 3 could be extended as shown in Figure 8. The figure again shows the controller component with the additions to the model marked in red. In case of a failure (*failedController*) one determines whether the system was in active use or not. The parameter 1/*mu* indicates the mean duration of active use while the parameter 1/*alpha* indicates the mean duration of passive use.

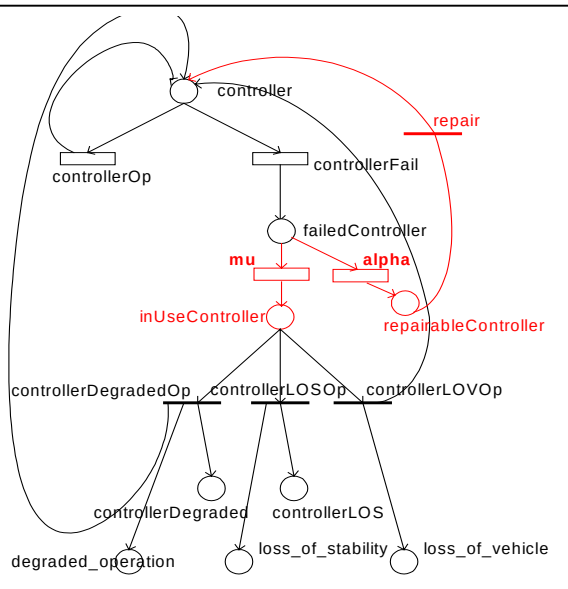

Figure 8: SPN model with Usage parameters

In case the failure occurs during the active period (*inUseController*), the system either continues to operate in the degraded (*controllerDegradedOp*) or lost stability mode (*controllerLOSOp*) or causes loss of vehicle (*controllerLOVOp*) – the severity of failure as described in Section 3. In case the failure occurs during passive use of the system

(*repairableController*), the fault can be repaired and an infinite repair rate is assumed. The system continues to operate as if no failure had occurred.

```
double controllerRate()
{
        double controller_rate = 0.0000006;

        // usage parameter
        controller_rate += controller_rate * mu();

        if (mark("controllerLOS") > 0) return controller_rate * 100;
        if ((mark("controllerDegraded") > 0) || (mark("tubingDegraded") > 0))
                return controller_rate * 100;
        return controller_rate;
}
```

Figure 9: Variable rate to model usage parameter

To work around the state explosion problem that occurred due the apparent increase in the number of states in the model as shown in Figure 9, the model was simplified to incorporate the usage parameters while calculating the failure rate itself for each component. The modified function for calculating the failure rate in light of the usage-profile is shown in Figure 9. The value of *mu* was assumed to be 2.5 for infrequent active use periods and 250 for frequent active use periods. As stated in the assumptions and shown in Figure 9, the value of these usage distributions was factored by the actual failure rate of the component to avoid stiffness in the model.

4.3 Results and Discussion

Transient analysis of the ABS model developed was carried out and the reliability was measured between 0 and 50K hours. The expected values of reliability at various time instances and different usage profiles was determined and plotted as a function of time. Again, the measure was predicted at 169 different points along the range. The interval between the points did not remain constant along the entire time range; instead the time range was divided into four segments. Each of these segments has a different time interval. The results are depicted in Figure 10.

In Figure 10, the Y-axis gives the measure of interest - the reliability; while the time range (0 to 50K hours) is shown along the X-axis. The shape of the curve is not a property of the system but of how the data was collected from the Petri net model. As expected, the reliability steadily decreases with time. The blue line indicates the reliability function when the usage of the system is infrequent and the pink line indicates the reliability function when the usage of the

system is frequent. Interestingly, the reliability of the system with heavy usage decreases alarmingly within the first 1K hours of operation, while the reliability of the system with not so heavy usage decreases perceptibly only after 2.5K hours of operation and then steadily afterwards. Also, the mean time to failure (MTTF) for the high usage case is 771022.9 hours as opposed to 775111.7 hours for the low usage case, a difference of approximately 4089 hours.

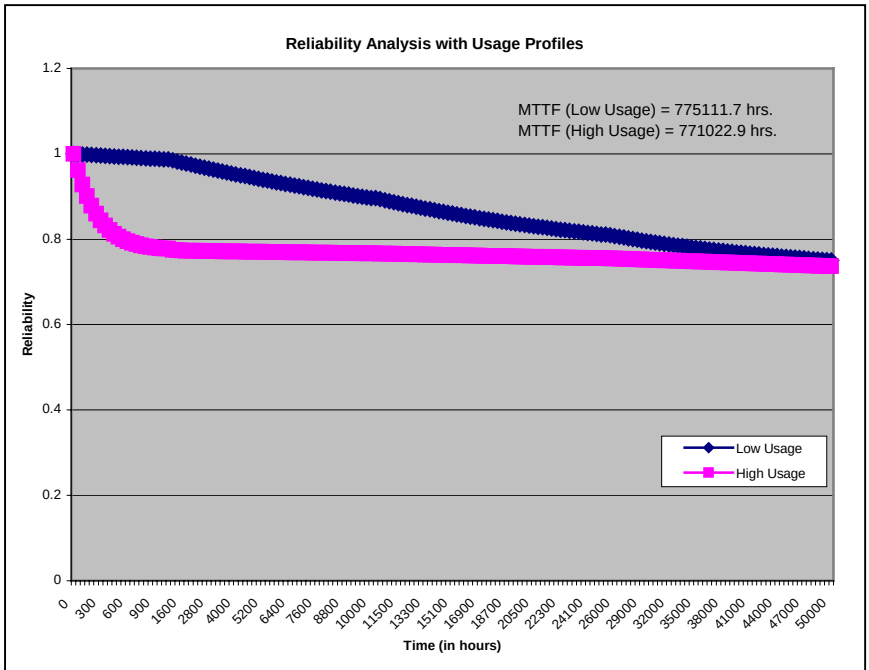

Figure 10: Reliability analysis results for usage profiles

An important fact to consider is that some components are used only for a few minutes during the entire lifetime of the vehicle (10-15 years) while other components like the tubing are used all of the time during that period. Hence, the usage of different components is different even within a given usage profile and might affect the actual reliability. However, what is important is the approach we used and the results clearly indicate that it is important to consider the usage profiles while determining the reliability for any given system.

5. Conclusion and Future Work

In this paper, we have shown how to model coincident failures, severity and usage-profiles in the Anti-lock Braking system of a passenger vehicle using Stochastic Reward Nets. To specify and analyze the system, we made some simplifying assumptions in order to manage the complexity of the system being modeled apart from handling the general challenges in the modeling like state explosion and stiffness. The Stochastic Petri Net models were developed for a four channel four

sensor ABS. The model, however, is easily extensible to model other schemes of ABS. Other coincident failures between components can be easily modeled by suitably modifying the failure rate function of the component in question. Similarly, other profiles with different usage parameters can be easily incorporated and analyzed. SPNP was used to specify the system and carry out the reliability analysis.

Modifications to the model can be carried out with the goal of predicting the behavior of the system. Parts of the model can be removed or changed in an effort to investigate the cause and effects of proposed enhancements or adaptations. Refining the system model can reveal trade-offs in design alternatives such as deciding what features of the system should be changed to improve the system's reliability or validating certain assumptions with respect to various performance goals. Once a system is validated, it may be used to perform sensitivity analysis, which can be used to support or discredit the modeling assumptions and analysis conclusions (Sheldon et al., 2002).

A major obstacle in modeling using SPNs was the persistent state explosion problem. This caused the programs to abort due to insufficient memory while solving the Markov chains. Stochastic Activity Networks (SANs) (Sanders and Meyer, 2001) are a stochastic extension to SPNs and are used for performability evaluation. Composed models in SANs exploit symmetries in the model to reduce the number of reachable states. Since, SANs are a more expressive tool for modeling systems, the goal is to develop SAN models for the Anti-lock braking system. The models can be specified and analyzed using UltraSAN, a software tool for model-based performance, dependability and performability evaluation of computer, communication and other systems (Sanders, 1994-95). The goal of future work is to specify SAN models for the Anti-lock braking system, analyze them using UltraSAN and compare the results obtained for SPN models.

Further, the Anti-lock Braking system is a small part of the DDR (Dynamic Driving Regulation) system. Figure 11 shows the Finite State Machine representation of the DDR system which consists of subsystems like the Anti-lock Braking system (ABS), the Electronic Steering Assistance (ESA), the traction control (TC) (Sheldon et al., 2000). Another goal is to develop a model that scales well for the combined system with emphasis on representing coincident failures, severity of failures and usage-profiles and analyze it for critical failures.

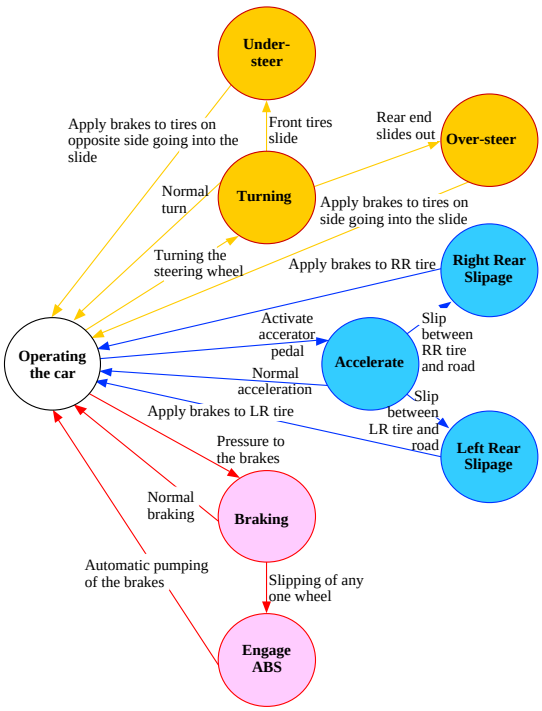

Figure 11: FSM of the DDR system

References

Balbo, G., 2000. Professor, Universita di Torino, Italy. Personal Communication at EEF-Summerschool Formal Methods and Performance Analysis, Netherlands, July 2000.

Balbo, G., 2001. Introduction to Stochastic Petri Nets, Lecture Notes in Computer Science, 2090, 84-155.

Bosch, R., 1993. Automotive Handbook, Bentley Pubs.

Ciardo, G., Muppala, J. and Trivedi, K., 1993. SPNP: Stochastic Petri Net Package, 1st Intl. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, San Diego, California.

Greiner, S. and Horton, G., 1996. Analysis of Stiff Markov Chains with the Multi-level Method, Proc. European Simulation Symposium, ESS '96.

Jerath, K. and Sheldon, F.T., 2001. Reliability Analysis of an Anti-lock Braking System using Stochastic Petri Nets, PMCCS5, Erlangen, Germany, Springer Verlag.

Kolsky, M., 1997. ABS: Understanding Anti-Lock Brakes, http://www.abrn.com/archives/0797tech.htm, 5th June, 2001.

Littlewood, B. and Strigini, L., 2000. Software reliability and dependability: a roadmap, International Conference on Software Engineering, Limerick, Ireland, ACM Press.

Meyer, J., 2000. Professor, University of Michigan, Ann Arbor, MI. Personal Communication at PMCCS5, Erlangen, Germany, September 2001.

Muppala, J.K., Ciardo, G. and Trivedi, K., 1994. Stochastic Reward Nets for Reliability Prediction, Communications in Reliability, Maintainability and Serviceability, 1, 9-20.

Musa, J.D., 1993. Operational Profiles in Software-Reliability Engineering, IEEE Software, 10, 14-32.

Nice, K., 2001. How Anti-Lock Brakes Work, http://www.howstuffworks.com/anti-lock-brake.htm, 4th June, 2001.

Popstojanova, K.G. and Trivedi, K., 2000. Stochastic Modeling Formalisms for Dependability, Performance and Performability, Lecture Notes in Computer Science, 1769, 403-422.

Sanders, W.H., 1994-95. UltraSAN User's Manual version 3.0, http://www.crhc.uiuc.edu/PERFORM/Papers/USAN_papers/manual_v3.0_all.pdf.

Sanders, W.H. and Meyer, J., 2001. Stochastic Activity Networks: Formal Definitions and Concepts, Lecture Notes in Computer Science, 2090, 315-343.

Sheldon, F.T. and Greiner, S., 1999. Composing, Analyzing and Validating Software Models to Assess the Performability of Competing Design Candidates, Annals of Software Engineering -Special Volume on Software Reliability, Testing and Maturity, 8, 49.

Sheldon, F.T., Greiner, S. and Benzinger, M., 2000. Specification, Safety and Reliability Analysis Using Stochastic Petri Net Models, Tenth International Workshop on Software Specification and Design, San Diego, California, IEEE Computer Society.

Sheldon, F.T., Xie, G., Pilskalns, O., et al., 2002. A Review of Some Rigorous Software Design and Analysis Tools, Software Focus Journal.

Siewiorek, D.P. and Swarz, R.S., 1992. Reliable Computer Systems: Design and Evaluation, Digital Press.

Trivedi, K., 1982. Probability and Statistics with Reliability, Queuing and Computer Science Applications, Prentice-Hall .

Vouk, M.A., 2000. Software Reliability Engineering, 2000 Annual RELIABILITY and MAINTAINABILITY Symposium, Los Angeles, CA, IEEE.