D2Cyber: A Design Automation Tool for Dependable Cybercars

Arslan Munir and Farinaz Koushanfar
Department of Electrical and Computer Engineering
Rice University, Houston, Texas, USA
Email: arslan@rice.edu, farinaz@rice.edu

***Abstract*—Next generation of automobiles (also known as cybercars) will increasingly incorporate electronic control units (ECUs) to implement various safety-critical functions such as x-by-wire (e.g., steer-by-wire (SBW), brake-by-wire). ISO 26262 specifies automotive safety integrity levels (ASILs) to signify the criticality associated with a function. Meeting a design's ASIL requirements at a minimum additional cost is a major challenge in cybercar design. In this paper, we propose D2Cyber—a design automation tool for cybercars that facilitates designers in selecting dependable designs by providing built-in models, easy to specify inputs, and easy to interpret outputs. D2Cyber considers the effects of temperature, electronics quality grade, and design lifetime in cybercar's design space exploration for determining a cost-effective solution and also advises on the attainable ASIL from a given design. We elaborate Markov models that form the basis of D2Cyber using SBW as a case study. We further provide evaluation insights obtained from D2Cyber.**

I. Introduction and Motivation

Electronic embedded systems are continuing to proliferate in modern cars as 90% of innovation in automobiles owes to electronic systems as noted by Daimler Chrysler [1]. The next generation of automobiles (also known as cybercars) will further escalate the integration of electronic control units (ECUs) to implement various distributed control functions. X-by-wire (e.g., steer-by-wire(SBW), brake-by-wire) are amongst the novel cybercar applications where electronic controllers substitute for the traditional mechanical and/or hydraulic systems. The use of electronic controllers in automotive systems has not only improved performance and economy for the customer but has also assisted in complying with government legislations to reduce pollution and increase safety and dependability.

Dependability assimilation in cybercar design is paramount because of product liability legislations, automotive standards, and increasing customer expectations. The *product liability* law holds responsible the manufacturers, distributors, suppliers, and retailers for the injuries caused by those products. According to the law, the manufacturer's product liability is excluded if a failure cannot be detected using the state-of-the-art science and technology at the time of product release. ISO 26262 is currently considered as the state-of-the-art of current automotive science and technology, and will continue to be considered as such until this standard is superseded by a further advanced standard. ISO 26262 classifies risk by automotive safety integrity levels (ASILs) where an ASIL specifies a function's (item's) necessary safety requirements for achieving an acceptable residual risk.

Dependability integration in automotive systems is subjected to stringent cost constraints. Governed by economy, manufacturers capabilities, and implemented quality control procedures; semiconductor vendors manufacture electronic products with various quality grades such as safety/military, industrial, and commercial. Due to the economy of scale, more and more commercial and non-automotive semiconductors are being used in automotive applications. The challenge is to ensure reliability for automotive safety functions using semiconductor components that were not originally designed to be integrated into a system with harsh environments or the components whose anticipated lifetimes are below vehicle lifetime requirements.

978-3-9815370-2-4/DATE14/

A holistic reliability analysis of automotive systems considering electronics quality grade, cost, and temperature has not been explored thoroughly in literature. Most of the contemporary dependability and reliability analysis tools (e.g., ReliaSoft [2], SHARPE [3], OSATE [4]) are general-purpose and do not include built-in models for reliability analysis of cybercars. Furthermore, most of prior reliability software tools require a user with a high degree of expertise in reliability engineering and computer design. Developing reliability models, specifying inputs to these models, and properly interpreting outputs from these tools require considerable resources and skills.

To assist the design space exploration of cybercars, we have developed D2Cyber (Design of Dependable Cybercars)—a tool that provides a systematic approach for modeling and prediction of reliability, mean time to failure (MTTF), and their tradeoffs under cost, temperature, and electronics quality grade constraints, all in a unified manner. D2Cyber facilitates cybercar designers' job by providing built-in models, easy to specify inputs, and easy to interpret outputs. Our main technical contributions are as follows:

- A design automation tool for design space exploration of dependable cybercars (D2Cyber) with SBW application as a case study.
- Cost-efficient cybercar design to meet various reliability and lifetime requirements using D2Cyber.
- Markov models formulation that provides basis for D2Cyber with SBW application as a case study.
- Comprehensive reliability analysis of cybercar designs for various lifetimes, operating temperature profiles, and electronics quality grades.

The remainder of this paper is organized as follows. Section II provides a brief summary of related work. Section III elucidates optimization objective function and design methodology of D2Cyber. Markov models formulation for D2Cyber is presented in Section IV. Evaluation results and insights obtained from D2Cyber are presented in Section V. Finally, conclusions are summarized in Section VI.

II. Related Work

Previous work explored dependability and safety of automotive systems. Baleani et al. [5] discussed various fault-tolerant (FT) architectures for automotive applications including lock-step dual processor architecture, loosely-synchronized dual processor architecture, and triple modular redundant architecture. Although, the authors compared various FT architectures' costs based on the area estimates, the study did not quantify the architectures' reliability or MTTF. Glaß et al. [6] studied symbolic techniques to analyze and optimize reliable systems with automotive as a case study. The

work, however, did not evaluate the reliability of implemented safety functions and the amount of redundancy required to attain a given ASIL.

Several earlier works elaborated dependability related tools and methodologies. Johnson et al. [7] discussed the purpose and type of models used for various reliability, availability, and serviceability modeling tools. Papadopoulos et al. [8] proposed a design approach to integrate semi-automatic reliability and safety analysis with optimization techniques to assist in automotive designs. The proposed approach built fault trees by traversing Matlab Simulink models of the system and evaluated local failure expressions encountered during the traversal. The work, however, did not consider the effects of temperature and electronics quality grade in the analysis. Furthermore, the work did not provide the details of fault tree models or insights obtained from the proposed approach. Lambert [9] discussed the use of fault tree analysis to assess failure modes within automotive systems using a car starting system as a case study. The work only conducted a qualitative evaluation and did not present quantitative (probabilistic) insights on the effects of failure in automotive systems.

III. D2Cyber—Design Automation Tool

D2Cyber is a design automation tool for cybercars that provides a systematic approach for modeling and prediction of reliability and MTTF, and their tradeoffs under cost, temperature, and electronics quality grade constraints, all in a unified manner. This section elucidates optimization objective function, inputs and outputs, and design methodology for D2Cyber.

A. Optimization Objective Function

D2Cyber solves an optimization problem to determine a cost-effective cybercar design under various constraints. The optimization problem can be formulated as:

$$\begin{array}{lll} \min & cost_d & \\ s.t. & \mathfrak{R} \geq a, & a \in \mathbb{R}_{[0,1]} \\ & \text{MTTF} \geq b, & b \in \mathbb{R}_{\geq 0} \\ & \text{ASIL} \geq y, & y \in \{\text{A,B,C,D}\} \\ & Q = q, & q \in \{\mathcal{S}, \mathcal{B}, \mathcal{C}\} \\ & T_p = \tau_x, & x \in \mathbb{N} \\ & \mathcal{L} = t, & t \in \mathbb{R}_{\geq 0} \end{array} \quad (1)$$

where $cost_d$ denotes design cost, $\mathfrak{R}$ denotes reliability, T_p denotes cybercar's operating temperature profile (estimated), $\mathcal{L}$ denotes the desired lifetime for the designed cybercar (for warranty period specification), and Q denotes the electronics quality grade. Q can be either $\mathcal{S}$ (safety-grade product that passes manufacturer's and/or electronic standards top quality check), $\mathcal{B}$ (industrial-grade product that did not pass the manufacturer's and/or electronic standards top quality check), or $\mathcal{C}$ (commercial-grade product whose quality checks are unknown) [10]. $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{[0,1]}$ denote the set of real numbers greater than or equal to 0, and the set of real numbers between 0 and 1, respectively. $\mathbb{N}$ denotes the set of natural numbers.

$cost_d$ is modeled in D2Cyber as:

$$cost_d = C_r + C_Q + C_b \quad (2)$$

where C_r denotes a cost factor due to design redundancy for improving dependability, C_Q denotes a cost factor due to the costs associated with electronics quality grade assurance procedures, and C_b denotes the design base cost using commercial grade electronics without any redundancy.

τ_x in Eq.1 denotes the temperature profile of the designed cybercar's expected operating environment. D2Cyber permits cybercar designers to specify different temperature profiles. τ_x can be specified as:

$$\begin{array}{rcl} \tau_x & = & ((p_1\%, t_1), (p_2\%, t_2), \cdots, (p_n\%, t_n)) \\ & s.t. & \sum_{i=1}^{n} p_i = 100\% \end{array} \quad (3)$$

where $t_i \quad i \in \mathbb{N}$ denotes the ambient temperature and $p_i \quad i \in \mathbb{N}$ denotes the overall time spent (expressed as a percentage) in the temperature t_i. We define six temperature profiles for illustration, which we will also use in our tool evaluation results (Section V):

$$\begin{array}{rcl} \tau_1 & = & ((5\%, 15\,^\circ\text{C}), (20\%, 20\,^\circ\text{C}), (30\%, 30\,^\circ\text{C}), \\ & & (20\%, 45\,^\circ\text{C}), (15\%, 20\,^\circ\text{C}), (10\%, 100\,^\circ\text{C})) \end{array} \quad (4)$$

Similarly, we define other temperature profiles τ_2, τ_3, τ_4, τ_5, and τ_6 (details of these profiles are omitted for brevity). Since a cybercar is expected to operate in many different temperatures, D2Cyber permits designers to verify the attainable ASIL from a design for various temperature profiles.

B. D2Cyber Inputs, Outputs, and Design Methodology

D2Cyber takes input from the designer on desired reliability, MTTF, ASIL, temperature profile, electronics quality grade, and lifetime, and outputs a minimum cost design that satisfies the design constraints. D2Cyber is built on top of existing reliability and modeling softwares namely SHARPE (Symbolic Hierarchical Automated Reliability and Performance Evaluator) [3] and the MIL-217 module of ITEM toolkit [11]. D2Cyber provides built-in Markov models for cybercar design focusing on a SBW application (Section IV). D2Cyber Markov models are specified in SHARPE, which requires failure rates specification of ECUs involved in the design.

For determination of ECU failure rates, D2Cyber leverages the MIL-217 module of the ITEM tookit [12]. We model the processor cores, caches, and main memory of ECUs in the ITEM toolkit. The parameters we specify in the ITEM toolkit include process technology (e.g., CMOS), microelectronics quality grade ($\mathcal{S}$, $\mathcal{B}$, or $\mathcal{C}$), package type, operating environment (e.g., ground and mobile in case of cybercars), ambient temperature, and power dissipation. D2Cyber determines failure rates for each of the operating temperature specified in the temperature profile τ_x (Eq.3) using the ITEM toolkit, and then estimates an overall failure rate corresponding to τ_x. D2Cyber then uses this overall failure rate in the developed Markov models to determine a design's reliability and MTTF. D2Cyber leverages a greedy approach to solve the optimization problem for design space exploration (Eq.1). D2Cyber explores the design space starting from the lowest cost design (normally the design with least redundancy and commercial grade electronics unless restricted by a constraint) and determines the design's reliability and MTTF. D2Cyber returns the design configuration as soon as the reliability and MTTF of the explored design meets the specified requirements.

IV. Markov Models Formulation for Permanent Faults

In this section, we discuss Markov models formulation for permanent faults, which forms the basis of D2Cyber. Since studies indicate that failure in computer systems is dominated by hardware-related failures with actual failure percentage ranging from 30% to more than 60% [13], we focus on permanent faults in this work.

A. Steer-by-Wire System

An SBW system replaces a mechanical steering system with ECUs, sensors, and actuators, which interact via a communication bus such as controller area network (CAN) or FlexRay. Our SBW case study architecture leverages multi-core ECUs to provide dependability. The architecture consists of two multi-core (dual-core or triple-core) hand wheel ECUs (HW ECU1 and HW ECU2) and two multi-core (dual-core or triple-core) front axle actuator ECUs (FAA ECU1 and FAA ECU2). Each of the ECUs is connected to the CAN bus. Our SBW architecture consists of three hand wheel sensors (hws1, hws2, and hws3) that are placed near the hand wheel to measure the driver's requests in terms of hand wheel angle, hand wheel torque, and the hand wheel speed. Similarly, three front axle sensors (fas1, fas2, and fas3) measure the front axle position. Both the hand wheel sensors (the front axle sensors) are connected to the HW ECUs (FAA ECUs) by point-to-point links. Two front axle actuator (FAA) motors (FAA motor 1 and FAA motor 2) operate in active redundancy on the front axle while two hand wheel (HW) motors (HW motor 1 and HW motor 2) operate in active redundancy on the hand wheel.

An SBW system aims to provide two main services [14]: 1) front axle control (FAC) that controls the wheel direction in accordance with the driver's request, and 2) hand wheel force feedback (HWF) that provides a mechanical-like force feedback to the hand wheel. In our SBW architecture, the FAC function implementation utilizes HW ECU1 and FAA ECU1 whereas the HWF function implementation utilizes HW ECU2 and FAA ECU2.

B. Markov Models

Cybercar systems can be modeled by Markov chains consisting of various states. When a failure occurs in a given state, the system's next operating state is determined completely by its current state regardless of how the system entered the current state. For illustration purposes, this section discusses Markov modeling formulation of an SBW system as a case study. Our Markov modeling illustration focuses on ECUs to reduce the state space assuming that redundancy for sensors and actuators can be provided cost-effectively due to relatively low-cost of sensors and actuators than ECUs.

Consider an SBW system consisting of a triple-core HW ECU1 and a triple-core FAA ECU1 that implement FAC function (we denote this system as SBW_{FAC}^{TMR} for conciseness where TMR stands for triple modular redundancy). The FAC function is operational as long as there is at least one HW ECU1 processor core and one FAA ECU1 processor core alive. We assume that the HW ECU1 and FAA ECU1 processor cores failure times are exponentially distributed with failure rates λ_{he} and λ_{fe}, respectively.

Fig. 1 depicts Markov model for permanent faults in SBW_{FAC}^{TMR}. Each state (i, j) represents the state when i processor cores of HW ECU1 and j processor cores of FAA ECU1 are functioning. The solid arrows represent state changes due to HW ECU1 and FAA ECU1 processor cores failure. The initial state is $(3, 3)$ in which the three processor cores of both HW ECU1 and FAA ECU1 are functioning. The differential equations describing this Markov model are given as:

$$\begin{aligned} P'_{(3,3)}(t) &= -3\lambda_{he}P_{(3,3)}(t) - 3\lambda_{fe}P_{(3,3)}(t) \\ P'_{(2,3)}(t) &= 3\lambda_{he}P_{(3,3)}(t) - 2\lambda_{he}P_{(2,3)}(t) \\ &\quad -3\lambda_{fe}P_{(2,3)}(t) \\ &\vdots \\ P'_{(1,0)}(t) &= \lambda_{fe}P_{(1,1)}(t) \end{aligned} \tag{5}$$

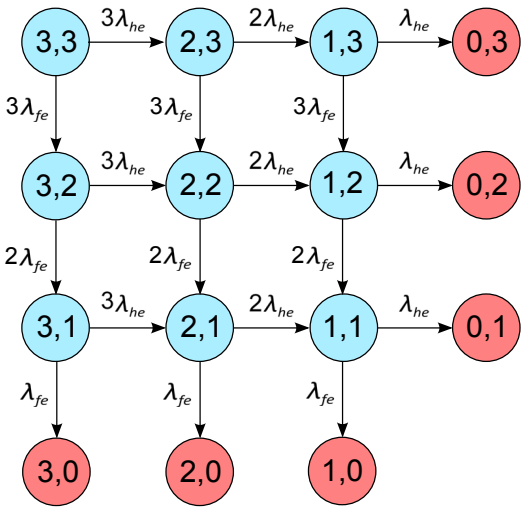

Fig. 1: Markov model for permanent faults in an SBW system with triple-core HW ECU1 and FAA ECU1.

where $P_{(i,j)}(t)$ denotes the probability that the SBW system will be in state (i, j) at time t and $P'_{(i,j)}(t)$ represents the first order derivative of $P_{(i,j)}(t)$. Solving the differential equations (Eq.5) with initial conditions $P_{(3,3)}(0) = 1$ and $P_{(i,j)}(0) = 0, \forall\, i \neq 3, j \neq 3$ yields the solution for $P_{(i,j)}(t), \forall\, i, j \in \{0, 1, 2, 3\}$.

The hardware reliability of SBW_{FAC}^{TMR}, $R_{SBW\_FAC}^{TMR}(t)$, is given by:

$$\begin{aligned} R_{SBW\_FAC}^{TMR}(t) &= 1 - P_{(0,3)}(t) - P_{(0,2)}(t) - P_{(0,1)}(t) \\ &\quad -P_{(3,0)}(t) - P_{(2,0)}(t) - P_{(1,0)}(t) \end{aligned} \tag{6}$$

The MTTF of the SBW FAC function with triple-core HW ECU1 and FAA ECU1, $\text{MTTF}_{SBW\_FAC}^{TMR}$, is:

$$\text{MTTF}_{SBW\_FAC}^{TMR} = \int_0^\infty R_{SBW\_FAC}^{TMR}(t)\, dt \tag{7}$$

Markov models for permanent faults in an SBW system with a dual-core HW ECU1 and a dual-core FAA ECU1 (we denote this system as SBW_{FAC}^{DMR} for conciseness where DMR stands for dual modular redundancy) and for an SBW system with single-core HW ECU1 and single-core FAA ECU1 (we denote this system as SBW_{FAC} for conciseness) are a subset of Fig. 1 and details of these models are omitted for brevity.

V. Evaluation Results and Insights

In this section, we present reliability evaluation results and insights obtained from D2Cyber on the effects of design lifetime, operating temperature profile, and electronics quality grade on reliability (MTTF evaluations are not presented for brevity). D2Cyber translates the designer specified ASIL requirements to representative reliability values depending on the specified lifetime assuming an exponential distribution for failure rate. For failure rate determination, we model processor core, cache, and memory as close as possible to Freescale's MPC5746M ECU [15] in the ITEM toolkit. The MPC5746M is a dual-core processor consisting of e200Z4 processor cores where each processor core contains 8 KB of instruction cache, 4 KB of data cache, 16 KB of instruction random access memory (RAM), and 64 KB of data RAM. For TMR modeling, we add another e200Z4 core with similar capabilities.

Effect of Design Lifetime on Reliability: A cybercar system is designed to meet reliability requirements for a given lifetime (for

TABLE I: Reliability for SBW designs for different design lifetimes t when electronics quality grade is $\mathcal{S}$ and temperature profile is τ_1.

t (hours)	$SBW_{FAC}(t)$	$SBW^{DMR}_{FAC}(t)$	$SBW^{TMR}_{FAC}(t)$
4,380	0.9987081314	0.9999991650	0.9999999995
8,760	0.9974179317	0.9999966622	0.9999999957
17,520	0.9948425306	0.9999866659	0.9999999656
35,040	0.9897116606	0.9999468017	0.9999997256
52,560	0.9846072530	0.9998806146	0.9999990776
70,080	0.9795291712	0.9997883114	0.9999978220
78,840	0.9769999600	0.9997324308	0.9999969048
87,600	0.9744772794	0.9996700986	0.9999957625

warranty period specification). A system that is designed to meet reliability requirements for a given lifetime may not be able to meet the reliability requirements for another lifetime specification. Table I presents reliability evaluations obtained from D2Cyber for various SBW designs for different design lifetimes $\mathcal{L} = t$ (ranging from six months to ten years) when operating temperature profile is τ_1 and electronics quality grade is $\mathcal{S}$. We point out that the lifetime of 4,380 hours $\approx$ six months and the lifetime of 87,600 hours $\approx$ 10 years.

Results validate that the reliability of an SBW system with redundancy is higher than an SBW system without redundancy for all lifetime specifications. For example, reliability of SBW^{TMR}_{FAC} is 0.03% and 2.6% greater than that of SBW^{DMR}_{FAC} and SBW_{FAC}, respectively, when $\mathcal{L}$ = 10 years. Similarly, reliability of SBW^{DMR}_{FAC} is 2.6% greater than that of SBW_{FAC} when $\mathcal{L}$ = 10 years. Results reveal that reliability decreases as the time elapses for a given design. For example, reliability decreases by 0.129%, 2.5×10^{-4}%, and 3.8×10^{-7}% for SBW, SBW^{DMR}_{FAC}, and SBW^{TMR}_{FAC}, respectively, as the SBW electronics time after installation elapses from six months to one year ($\approx$ 8,760 hours). These results indicate that reliability decreases more sharply for designs with fewer redundancy as the time elapses.

Effect of Temperature on Reliability: Our reliability evaluation experiments with D2Cyber reveal that a cybercar's operating temperature impacts reliability. For example, reliability decreases by 0.038% as the temperature increases from 15 °C to 20 °C whereas reliability decreases by 0.142% as the temperature increases from 15 °C to 30 °C for SBW_{FAC} when $Q = \mathcal{S}$ and $\mathcal{L}$ = 43,800 hours $\approx$ 5 years. Results divulge that the temperature impact on reliability becomes more prominent at high temperatures. Furthermore, the temperature effect on reliability is more pronounced on designs with fewer or no redundancy. For example, decrease in reliability as temperature increases from 15 °C to 30 °C is 0.142%, 7.3×10^{-4}%, and 2.8×10^{-6}% for SBW_{FAC}, SBW^{DMR}_{FAC}, and SBW^{TMR}_{FAC}, respectively, when $Q = \mathcal{S}$ and $\mathcal{L}$ = 5 years.

Effect of Electronics Quality Grade on Reliability: Electronics quality grade used in the design of cybercar systems significantly impacts reliability. Results reveal that reliability deteriorates as the electronics quality grade degrades for all operating temperature profiles. For instance, reliability decreases by 2.1% as Q is replaced from $\mathcal{S}$ to $\mathcal{B}$ and the reliability decreases by 30.7% as Q degrades from $\mathcal{S}$ to $\mathcal{C}$ for SBW_{FAC} when $T_p = \tau_2$. Electronics quality grade impact on reliability is manifested significantly on designs with fewer or no redundancy. For example, reliability decreases by 30.7% for SBW_{FAC} whereas reliability decreases by only 3.4% and 0.42% for SBW^{DMR}_{FAC} and SBW^{TMR}_{FAC}, respectively, as Q degrades from $\mathcal{S}$ to $\mathcal{C}$ and $T_p = \tau_2$. This decrease in reliability due to electronics quality grade has ramifications on attainable ASIL from a design. For instance, for an SBW design with $\mathcal{L}$ = 70,080 hours $\approx$ 8 years and $T_p = \tau_3$, ASIL-D is attainable from DMR when $Q = \mathcal{S}$ whereas only ASIL-A is attainable from TMR when Q degrades to $\mathcal{C}$.

VI. Conclusions

In this paper, we propose D2Cyber—a design automation tool for design space exploration of dependable cybercars. D2Cyber explores the cybercar design space and outputs a design with minimum cost that meets reliability, MTTF, ASIL, and lifetime requirements for a specified operating temperature profile and electronics quality grade. We formulate Markov models that provide basis for D2Cyber with steer-by-wire (SBW) system as a case study. We evaluate reliability of various designs for different lifetimes, temperature profiles, and electronics quality grades. Evaluation results reveal that reliability decreases slightly with increasing temperature and more noticeably with degrading electronics quality grade. This decrease in reliability due to electronics quality grade has ramifications on attainable ASIL from a design. Results divulge that designs that leverage redundancy are more immune to the effect of changes in temperature and electronics quality grade on reliability.

Acknowledgments

This work was supported by the Office of Naval Research (ONR R17460) and SRC GRC Freescale grant (R65000). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONR and the SRC GRC.

References

[1] F. Simonot-Lion, "The Design of Safe Automotive Electronic Systems: Some Problems, Solutions, and Open Issues," in *Proc. of IEEE IES*, October 2006.

[2] ReliaSoft, "ReliaSoft—Empowering the Reliability Professional," 2013. [Online]. Available: http://www.reliasoft.com/

[3] SHARPE, "The SHARPE Tool & the Interface (GUI)," 2013. [Online]. Available: http://people.ee.duke.edu/~chirel/IRISA/sharpeGui.html

[4] OSATE, "OSATE—Open Source Tools," in *Software Engineering Institute, Carnegie Mellon University*, 2013. [Online]. Available: http://www.sei.cmu.edu/dependability/tools/osate/index.cfm

[5] M. Baleani, A. Ferrari, L. Mangeruca, A. Sangiovanni-Vincentelli, M. Peri, and S. Pezzini, "Fault-Tolerant Platforms for Automotive Safety-Critical Applications," in *Proc. of ACM CASES*, October-November 2003.

[6] M. Glaß, M. Lukasiewycz, F. Reimann, C. Haubelt, and J. Teich, "Symbolic Reliability Analysis and Optimization of ECU Networks," in *Proc. of IEEE/ACM DATE*, March 2008, pp. 158–163.

[7] A. M. Johnson and M. Malek, "Survey of Software Tools for Evaluating Reliability, Availability, and Serviceability," *ACM Computing Surveys*, vol. 20, no. 4, pp. 227–269, December 1988.

[8] Y. Papadopoulos and C. Grante, "Techniques and Tools for Automated Safety Analysis & Decision Support for Redundancy Allocation in Automotive Systems," in *Proc. of IEEE COMPSAC*, November 2003, pp. 105–110.

[9] H. Lambert, "Use of Fault Tree Analysis for Automotive Reliability and Safety Analysis," in *SAE 2004*, March 2004.

[10] 3DPlus, "Quality Grades and Reliability," 2013. [Online]. Available: http://www.3d-plus.com/quality-grades-reliability-2.php

[11] ITEM, "ITEM ToolKit—Reliability Analysis Software," 2013. [Online]. Available: http://www.itemtoolkit.com/

[12] MIL-HDBK-217F, "Military Handbook: Reliability Prediction of Electronic Equipment," Washington, D.C., 1991. [Online]. Available: www.sre.org/pubs/Mil-Hdbk-217F.pdf

[13] B. Schroeder and G. A. Gibson, "A Large-Scale Study of Failures in High-Performance Computing Systems," in *Proc. of IEEE DSN*, June 2006.

[14] C. Wilwert, N. Navet, Y.-Q. Song, and F. Simonot-Lion, *Design of Automotive X-by-Wire Systems*. The Industrial Communication Technology Handbook CRC Press, 2005.

[15] Freescale, "Qorivva MPC5746M MCU Fact Sheet," 2013. [Online]. Available: http://cache.freescale.com/files/32bit/doc/fact_sheet/MPC5746MFS.pdf?fpsp=1