ISSN 2574-0741

SAE INTERNATIONAL®

PREVIEW ISSUE

2018

SAE INTERNATIONAL JOURNAL OF

CONNECTED AND AUTOMATED VEHICLES

SAE International Journal of Connected and Automated Vehicles

Preview Issue

SAE International
400 Commonwealth Drive, Warrendale, PA 15096-0001
Phone: 724-776-4841 Fax: 724-776-5760
www.sae.org

400 Commonwealth Drive
Warrendale, PA 15096-0001 USA
E-mail: CustomerService@sae.org
Phone: 877-606-7323 (inside USA and Canada)
724-776-4970 (outside USA)
Fax: 724-776-0790

Printed in USA

ISSN 2574-0741
e-ISSN 2574-075X

For purchase inquiries, please contact: SAE Customer Sales

E-mail: CustomerSales@sae.org
Phone: 888-875-3976 *(inside USA and Canada)*
724-772-4086 *(outside USA)*
Fax: 724-776-0790

Visit the SAE International Journals at www.sae.org/publications/journals

SAE International Journal of Connected and Automated Vehicles

About the Journal

Aims and Scope

SAE International Journal of Connected and Automated Vehicles furthers the state of the art of engineering research by promoting high-quality theoretical and applied investigations in the arena of connected and autonomous vehicles (CAVs) in on-road, off-road, and aerial operational environments. The enormous growth in numbers, diversity, and complexity of CAVs has been driven by: (i) enhancements of fundamental scientific understanding; (ii) technological convergence of computing, communication, and miniaturization; and (iii) increased scale and complexity of tangible embodiments and engineering implementations at the component-, subsystem-, and system-levels.

The Journal seeks to further these goals by publishing peer-reviewed scientific papers that showcase strong theoretical and empirical contributions and findings to the body of engineering knowledge surrounding various facets of the lifecycle treatment (design, modeling, controlling, testing, demonstration, and experimentation) of connected and automated vehicles with an emphasis on the system perspective.

Topics of interest include, but are not limited to, the following within the overall CAV system context:

Technologies

- Active perception architectures and implementations (radar, vision, lidar) for CAVs
- Sensors, sensor fusion (infrastructure and vehicle-based) for CAVs
- Vehicle design, analysis, and control enhancements for CAVs
- Electrification/vehicle electronics architectures and implementations for CAVs
- Communication architectures and implementations (V2x) for CAVs
- Real-time computational paradigms and architectures (AI, model-based) for CAVs
- Novel actuation paradigms (structural control, reconfigurable systems) for CAVs

Cyber-enabled System Capabilities

- Big data analysis, cloud computing architectures
- Vehicle navigation and situational awareness
- Fault detection and diagnosis, fault tolerant control
- Cybersecurity and cyber-enhanced security
- Active- and semi-active connected and automated vehicle control (adaptive, fuzzy, cooperative, neuro, emergent paradigms)
- Hybrid simulation- and empirical-testing paradigms (model-in-the-loop, hardware-in-the-loop)

Human-in-the-loop Element

- Active vehicle safety architectures (occupant, pedestrian)
- Human machine Interaction design (driver- and controller-interfaces)
- Varying grades of driver-assistance systems
- Psycho-social facets of shared control (trust, variability)

Subsystem and System Engineering Frameworks

- Automated Guided Vehicles (AGVs)
- Multi-vehicle cooperation, connected vehicles, platooning
- Platooning and fleet management
- Reproducible testing and validation architectures and paradigms
- Noise, network failure, faults, reliability analysis
- Application use cases (warehousing, x-docking, mining, agriculture, military)

contents

Previously Published SAE International Journal Articles:

Journal article lead editors are underlined and contributing editors are in italics.

The following are previously published SAE International journal articles that serve as examples of the type of papers you will see in upcoming issues of the *SAE International Journal of Connected and Automated Vehicles*

SAE INTERNATIONAL

ARTICLE INFO
Article ID: 2016-01-0145

doi:10.4271/2016-01-0145

Situation Awareness, Scenarios, and Secondary Tasks: Measuring Driver Performance and Safety Margins in Highly Automated Vehicles

Madeleine Gibson, University of Wisconsin

John Lee, Vindhya Venkatraman, Morgan Price, Jeffrey Lewis, Olivia Montgomery, and Bilge Mutlu, *University of Wisconsin*

Joshua Domeyer and James Foley, *Toyota Technical Center USA, Inc.*

Abstract

The rapid increase in the sophistication of vehicle automation demands development of evaluation protocols tuned to understanding driver-automation interaction. Driving simulators provide a safe and cost-efficient tool for studying driver-automation interaction, and this paper outlines general considerations for simulator-based evaluation protocols. Several challenges confront automation evaluation, including the limited utility of standard measures of driver performance (e.g., standard deviation of lane position), and the need to quantify underlying mental processes associated with situation awareness and trust. Implicitly or explicitly vehicle automation encourages drivers to disengage from driving and engage in other activities. Thus secondary tasks play an important role in both creating representative situations for automation use and misuse, as well as providing embedded measures of driver engagement. Latent hazards-hazards that exist in the road environment and merit driver attention, but do not materialize to require a driver response-have been used with great success for understanding the vulnerability of novice drivers. Latent hazards might provide a similarly useful index of driver attention to the road during periods where the automation is vulnerable to failure. With highly automated vehicles, latent hazards include potential roadway threats that might not be sensed by the automation and would require driver attention. This paper describes driving simulator scenarios used to operationalize automation-relevant latent hazards, secondary tasks tuned to index driver disengagement from the driving task, and measures that reflect safety margins rather than driving performance, such as drivers' trust, situation awareness, and expected time to transition to manual control.

History

Received: 20 May 2016
Published: 05 Apr 2016

Citation

Gibson, M., Lee, J., Venkatraman, V., Price, M. et al., "Situation Awareness, Scenarios, and Secondary Tasks: Measuring Driver Performance and Safety Margins in Highly Automated Vehicles," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 9(1):2016, doi:10.4271/2016-01-0145.

ISSN: 1946-4614
e-ISSN: 1946-4622

Introduction

Sophisticated technology is already active in vehicle control. Driver assistance systems support lane keeping, parking, speed maintenance, blind spot monitoring, and they also enhance night vision and detect driver impairment. In general, the potential safety benefits of these safety systems and automated vehicles are promising; however, realizing this promise depends on carefully coordinating driver and vehicle behavior. As vehicles become more capable, this coordination can break down if the drivers' role is not clear.

Types and Levels of Automation

The types and levels taxonomy of automation [1,2] describes how different sets of activities can be allocated to the automation and to what degree. For each stage of information processing from perception to control, levels of automation can range from none to complete. Taxonomies that guide design of vehicle automation [3,4] share some similarities with the types and levels taxonomy: at one extreme the driver does everything and at the other the automation takes full control. The National Highway Traffic Safety Administration (NHTSA) and the Society of Automotive Engineers (SAE) define Level 1 vehicle automation as those that perform one primary vehicle control function - either steering or speed maintenance, while the driver performs the others, monitors the roadway, and remains fully responsible for safe vehicle control. Such automation is available in many vehicles today, for example, where the driver steers manually and engages cruise control. In NHTSA and SAE Level 2 automation, the automation performs at least two primary controls, for example both steering and speed maintenance. The driver is responsible for monitoring the roadway and should be ready to take-over control of the vehicle at any time [3]. Level 3 automation, from both NHTSA and SAE definitions, assigns higher capability to the automation, where the vehicle automation monitors the roadway and performs the primary control tasks; however, the driver may be expected to occasionally take control of the vehicle when the roadway demands exceed the capacity of the automation. The crucial concern for Levels 2 and 3 automation is the potential driver confusion regarding whether the driver or the automation is primarily responsible for driving. NHTSA Level 4 and SAE Level 4 and Level 5 represent automation that can handle all driving situations and requires very limited control input from the driver.

Both Level 2 and Level 3 automation require some amount of driver attention to the roadway, either continuously, at critical moments, or during transitions between levels of automation. Attention to the road for monitoring, compared to attention for control, leads to longer response times and less effective responses [5, 6, 7]. Perception for control involves visual, cognitive, and proprioceptive engagement that is coupled to drivers control inputs and the expected outcomes. Activating vehicle automation can sever the perception-action loop and transform the driver from a controller to a monitor. As a monitor, a driver is *on* or *out* of the control loop rather than *in* the control loop [8,9]. Being *on* the control loop implies the driver is not perceiving the vehicle and roadway state to control the vehicle, but is actively monitoring the vehicle and roadway state to ensure the automation is controlling effectively. With automation that assumes control of steering and speed, drivers might easily disengage from monitoring and slip from on the loop to out of the loop [10].

Drivers might even think of driving as a distraction from other activities [11], and vehicle automation might be seen as a means of disengaging from driving, even when doing so violates the capabilities of the automation and compromises safety. Therefore, a major consequence of increasing automation might be the drivers' willingness to switch attention to non-driving tasks.

The evolving role of drivers in the context of automated driving and the additional activities drivers engage in present new challenges for evaluating how drivers work with the automation. Specifically, there are no standard scenarios and measures to evaluate joint control of the vehicle. This paper outlines a driving simulator evaluation protocol that captures driver engagement and disengagement in the driving task, and measures the safety margins achieved by drivers and automation.

Failure Modes of Driver-Automation Interaction

Central to assessing driver interaction with highly automated vehicles is the need to anticipate and test for likely failure modes. Risk analysis typically identifies failure modes associated with the mechanical, electronic, and software elements of systems and works to assure those risks remain below an acceptable level. Such risk estimates assume that drivers provide an additional safety margin, compensating for failures of the technology and other sources of unanticipated variability. This assumption is not always justified. Just as technology has failure modes, so do drivers and these failure modes should be assessed as part of a simulator-based evaluation.

Although highly automated vehicles lack operational exposure that might reveal prototypical driver-automation failure modes, experience in other domains suggests failure modes that might occur with vehicle automation. A very likely failure mode concerns drivers confusing Level 2 and Level 3 automation, where a critical distinction between these levels involves whether the driver has primary responsibility for monitoring the vehicle. *Responsibility diffusion* regarding whether the driver or the automation is primarily responsible for vehicle monitoring and control is likely a prominent failure mode. People are poorly suited to the role of monitoring automation and are prone to over trusting and neglecting reliable automation[12,13]. *Mode confusion* associated with whether automatic control is engaged and if so what mode of control has been engaged is a prominent failure mode with

automation in other domains and will likely affect vehicles equipped with combinations of Level 1, 2, and 3 automation [12,14, 15, 16, 17]. *Operating envelope awareness,* similar to the more degraded situation awareness that can accompany automation introduction, concerns awareness of the intended operating envelope and proximity to safety boundaries of this operating envelope. This failure mode will affect vehicle automation that is not intended to be used on all road types and in all road conditions [18, 19, 20]. *Ineffective transfer of control* represents a general failure mode that might occur during planned, unplanned, and unwarranted transfers of control from the automation to the person [17,21,22]. Unwarranted transfers of control are those situations where the automation sees no need for driver intervention, but the driver initiates a steering or braking maneuver that might conflict with the automation. These automation failure modes define the requirements for assessing vehicle automation.

No single assessment protocol will likely address all the possible driver-automation failure modes. Several previous automated driving studies have focused primarily on the failure mode of ineffective transfer of control. The objective of these studies was to determine how quickly drivers recognize and respond to safety critical events. An alert or warning was used to prompt drivers to regain control of the vehicle when the automation fails. The method for how and when drivers were informed of needed actions was implemented differently across studies. In some cases the warning time was varied across conditions [23,24]. Another study investigated driver response to one and two step processes to indicate a needed take-over [25]. This study also manipulated how drivers were alerted to the automation failure. The primary measure for these studies was the reaction time to begin the take-over process. Other driving performance metrics included minimum headway to lead vehicle, standard deviation of steering wheel position, and standard deviation of road offset (distance from the centerline of the road) for evaluating performance once the driver gained control.

Although many studies have focused on how and when control is traded, other automated vehicle studies have considered mode confusion failures and responsibility failure modes. These studies have examined driver behavior across different levels of automation. For example, manipulating the levels of automation (manual/fully automated) and traffic density (high/low) and drivers' willingness to overtake slow moving vehicles, management of car following, and secondary task engagement [26]. Similarly, drivers' response to automation was compared to an initial manual baseline condition [27]. When drivers experienced increasing levels of automation (i.e., lateral or longitudinal control followed by full automation) they were more willing to focus attention on secondary tasks as automation capability increased.

The following sections outline a protocol for addressing failure modes of responsibility diffusion, mode confusion, and operating envelope awareness. Another protocol is needed to address ineffective transfer of control. We describe latent hazards as an important element of driving simulator scenarios, secondary tasks as a central component of driving highly automated vehicles, and measures of safety margin, rather than driver performance, as a critical indicators of resilience in the face of driver-automation failure modes.

Driving Simulator Protocol

Driving simulators are usually composed of the following elements: cabs, computers and electronics, vehicle dynamics, scenario, and task environment [28]. Driving simulators have increasingly become a widely used and accepted tool for transportation human factors research due to the several advantages. One major advantage is the safe environment driving simulators provide to understand basic human limitations and driver behavior in safety critical events. For example, in distracted driving studies, experimenters can evaluate invehicle systems through driver engagement in secondary tasks in situations that would be dangerous on the road. Similarly, design of vehicle automation and in-vehicle technologies can be evaluated without the risk inherent in on-road and test track evaluations. Another advantage of driving simulators is the controlled environment. Each participant is exposed to identical driving scenarios, eliminating confounding variables such as weather or traffic found in naturalistic driving environments. Roadway conditions and other vehicle behaviors can be specified for the duration of the experiment. Lastly, driving simulators allow for drivers to experience many test conditions in a short time [29]. During a single study session, drivers can experience many road situations that might take hours or months to occur in naturalistic driving.

Although choosing the correct methods for conducting research is important, it is equally, important is choosing driving performance metrics that are sensitive to automation failure modes. Measures used to assess driver distraction with manual driving have included speed, vehicle following (headway), lane keeping, steering wheel metrics, event detection, response times, and subjective ratings. However, these measures do not consider joint performance of the driver and automation, such as when the driver is no longer in control of the vehicle's primary functions. With automated driving, the driver's role changes from being directly engaged in control to that of a monitor.

Drivers with automation need to be considered part of a joint cognitive system, with the unit of analysis moving from that of the driver to the driver-vehicle combination. Therefore, measures of driver performance should not focus simply on the driver, but on the joint performance of the automation and the driver. Furthermore, because vehicle automation can achieve very high levels of driving performance (e.g., maintain a fixed speed precisely) safety margins are more relevant. A major challenge is to measure safety margins that are maintained by the driver across different levels of automation and in response to a range of roadway situations. More specifically, this involves measuring driver awareness of the automation capability and driver adaptive capacity relative to automation limits.

Situation awareness and trust in automation are important indicators of drivers' adaptive capacity. Situation awareness is defined as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the future" [30]. The first component of situation awareness (SA) has been measured as attention allocation [31], and can be estimated by drivers' glance behavior to road situations that might challenge the automation. Trust is an attitude that the automation will achieve the drivers' goals [32], and could be indicated by the degree to which drivers engage in non-driving tasks, neglect the roadway, and keep hands and feet away from the controls.

Latent Hazards to Assess Situation Awareness

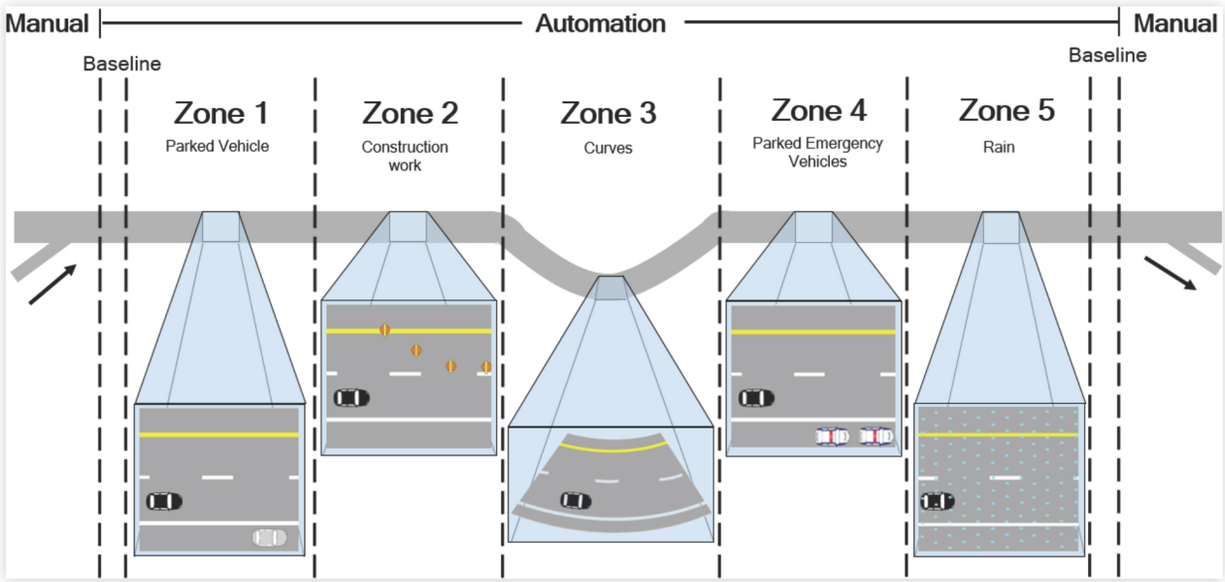
Latent hazards offer a promising measure of situation awareness of automation. Latent hazards are possible roadway threats that do not necessarily develop into hazards that require immediate action [33]. Latent hazards have been used for understanding the limits of novice drivers. Novice drivers' attention to potential threats differs from that of experienced drivers [34,35]. Attention to latent hazards is often measured using glance metrics. Drivers glances can be used as an indicator of what the driver is attention and what information the driver is processing. Whether drivers gaze towards latent hazards can indicate whether they will anticipate the potential threat and are ready to act if needed [36]. Latent hazards might provide a similarly useful measure of driver attention to the road situations where automation is less capable and when the driver needs to resume control.

A critical consideration in using latent hazards to evaluate situation awareness of vehicle automation concerns drivers being part of a joint cognitive system consisting of the driver and the automation. Previous use of latent hazards has focused on challenges that drivers must accommodate to maintain safety. When automation acts with drivers to control the vehicle, latent hazards need to be defined in terms of situations that challenge the automation and might require the driver to intervene.

We have designed scenarios to reflect situations that challenge automation. These scenarios were implemented as latent hazards. Such latent hazards involve potential safety conflict situations that do not develop into active threats. However, drivers need to pay attention to the possibility of such hazards. Ideally, drivers are expected to recognize when latent hazards are present and be prepared to intervene to ensure safe operation of the vehicle. The hazards in our scenario do not require any control actions from the drivers. However, drivers can intervene at any point.

Figure 1 shows a drive composed of several scenarios. At the start of the drive, drivers manually operate the vehicle to merge onto the highway. When instructed, the drivers engage the automation, followed by one minute of baseline driving with the automation. A one-minute baseline period also occurs at the end of the scenario before switching back to manual control and exiting the highway. The rest of the drive includes five different zones. The zones are equal in distance and each includes one latent hazard. Latent hazards include: stopped vehicle on the side of the road, construction work in the adjacent lane, curves, emergency vehicles on the side of the road, and rain. The duration of the latent hazards is small relative to the period where the vehicle can easily accommodate roadway demands. For example, the construction work is 30 seconds in a zone of several minutes.

FIGURE 1 Scenario layout showing a possible ordering of latent hazards over a drive.



Measures for assessing drivers' response to the latent hazards focus on glance behavior: total glance duration to latent hazard, frequency of glance to hazard, and time of first glance to hazard [33,36]. Beyond these measures, trust in automation might be reflected in behaviors that include moving hands to steering wheel, control inputs to the steering wheel, and foot movement towards the brake. The specific latent hazards considered in an evaluation depend on the design features of the automation. Latent hazards should represent situations at the edges of the operating envelope where relying on the automation is not appropriate or situations where drivers are expected to intervene in the event of an automation failure.

Secondary Task Engagement

A major motivation for developing vehicle automation and a major motivation for drivers engaging vehicle automation is the freedom such automation affords in performing secondary tasks. Such secondary tasks include email, social network interactions, and audio and video entertainment. Given the current engagement in secondary tasks while driving, automation evaluation should consider drivers interacting with a relatively engaging secondary task. In addition, secondary task engagement reflects drivers' trust in the automation and their willingness to neglect the monitoring of the road and automation. Attention to latent hazards measure the engagement in the driving task, secondary task usage measures disengagement from the driving task.

When selecting a secondary task to use in an automated driving study, it is important to choose one representative of the experience drivers will enjoy when using automated driving, such as a self-paced interaction with an information system. Many instances of self-paced tasks have been used in previous automated driving studies. These tasks include interaction with the in-vehicle entertainment system, eating, reading magazines, playing hand-held games, watching movies or TV shows, listening to the radio, performing grooming tasks, or completing word puzzles [24,26,27].

The secondary task protocol we developed consists of sorting emails into three categories, 1) Work, 2) Friends and Family, or 3) Trash, as described in Table 1. In the task, the driver sees a series of email subject lines displayed in a touchscreen application that models some basic functions of a common email client on a mobile device. Subject lines from the categories are chosen at random in a distribution of 1:1:4 respectively.

To complete the e-mail task, drivers need to press the edit button in the top right corner. This button prompts checkboxes to appear next to the subject lines. Drivers can then select one or more e-mails to sort into the appropriate category. If the driver correctly sorts the email, the count on the top of the homepage is updated to reflect the total number of emails sorted for each category. If the email is incorrectly sorted, the count remains the same. On the sorting page, a cancel button returns that task to the homepage.

Because each subtask is comprised of several discrete actions that require input from the driver, the secondary task can serve as a surrogate for eye glance and driver engagement in the secondary task. To further ensure driver attention is directed to the task, a pop-up message appears if the task is inactive for more than five seconds. The message directs drivers to touch the screen to resume. Data are collected for each button press, and so the task precisely indexes driver engagement. Overall, this task provides an activity representative of what drivers might do in a highly automated vehicle and one that precisely records drivers' task engagement.

TABLE 1 Categories and examples of email messages.

Email category	Example
Work	Emails relate to work, mentioning deliverables, and distinctly work related meetings or events. • Senior Developer Project Updates • Expense Reports Deadline Tuesday • August Departmental Schedule Request
Family and Friends	Emails relate to family affairs and social gatherings. • Family Dinner on Sunday at 4 • Grandma Birthday Celebration • Get Together 3/21 Downtown?
Trash	Emails are spam, featuring promotions, deals, and products. • Google TV, iPad, MacBook Air • Today Only: Save 20% on Holiday Gifts • Celebrate Summer with a Weekend Cruise

FIGURE 2 The main display shows the four emails available for sorting, a header with counts of the number of emails correctly sorted into each category, and the edit button.

Measures of Driver Performance and Safety Margins

Driving simulator studies often measure driver performance in responding to events or maintaining vehicle control. These measures become less relevant in understanding driver interaction with highly automated vehicles. With measures

of driving performance, the underlying assumption is that better driving performance corresponds to improved safety, which may not be the case with highly automated vehicles. Automation might maintain perfect control performance before it fails catastrophically. Because continuous measures of vehicle control, such as standard deviation of lane position, will likely fail to indicate the safety achieved by the joint cognitive system of the driver and automation, alternatives are needed.

With highly automated systems the distinction between performance and resilience is critical. Resilience represents the actions, time, and resources that enable a system to accommodate unexpected demands [37]. Automation often performs well, but is brittle, failing in the face of unexpected demands. Humans enhance the resilience of the system through their ability to adapt to the unexpected. A crucial measure is the degree of resilience the automation affords. This resilience can be measured in terms of safety margins. Safety margins reflect the capacity of the automation to respond to road situations and the response time of drivers to compensate for automation limitations. This response time can be estimated as a function of drivers' trust in the automation and situation awareness. Quantifying such safety margins represents an important measurement challenge in assessing vehicle automation.

This paper focused on simulator-based methods for evaluating vehicle automation. Equally important are analytic methods to evaluate automation and assess potential failure modes. Several promising techniques have emerged to support formal evaluation of human-automation interaction [38, 39, 40]. These complement simulator-based methods because the can uncover failure modes that might occur too rarely to be detected in a simulator evaluation and yet these failure modes might substantially undermine vehicle safety.

Conclusion

Highly automated vehicles will dramatically change the role of the driver. To ensure such changes enhance rather than degrade driving safety, an evaluation of design assumptions and driver-automation failure modes is needed. Driving simulators offer a promising approach to addressing these failure modes, but only if the scenarios, secondary tasks, and measures are tuned to the particular demands of vehicle automation assessment. Drivers and automation should be considered as part of a joint cognitive system that is vulnerable to new failure modes, and measures of safety margin rather than driving performance are most appropriate to ensuring these failure modes do not compromise driving safety.

Contact Information

John D. Lee can be reached at
john.d.lee@wisc.edu

Acknowledgments

Funding for this work was provided by Toyota Collaborative Safety Research Center (CSRC).

References

1. Parasuraman, R., Sheridan, T.B., and Wickens, C.D., "A model for types and levels of human interaction with automation," *IEEE Trans. Syst. Man Cybern. -Part A Syst. Humans* 30(3):286-297, 2000.
2. Sheridan, T.B. and Verplank, W., "Human and computer control of undersea teleoperations," 1978.
3. NHTSA, "Preliminary statement of policy concerning automated vehicles," 2013.
4. SAE International Surface Vehicle Information Report, "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems (J3016)," SAE Standard J3016, Rev. Jan. 2014.
5. Eprath, A.R. and Young, L.R., "Monitoring vs. man-in-theloop detection of aircraft control failures," in: Rasmussen, J. and Rouse, W. B., eds., *Human Detection and Diagnosis of System Failures*, Plenum Press, New York: 143-154, 1981.
6. Gibson, J.J., "Observations on active touch," *Psychol. Rev.* 69:477-491, 1962.
7. Walker, G.H., Stanton, N., and Young, M.S., "The ironies of vehicle feedback in car design," *Ergonomics* 49(2):161-179, 2006.
8. Bainbridge, L., "Ironies of automation," *Automatica* 19(6):775-779, 1983, doi:10.1016/0005-1098(83)90046-8.
9. Wickens, C.D. and Kessel, C., "The Effects of Participatory Mode and Task Workload on the Detection of Dynamic System Failures," *IEEE Trans. Syst. Man. Cybern.* 9(1):24-34, 1979, doi:10.1109/TSMC.1979.4310070.
10. Merat, N. and Lee, J.D., "Preface to the Special Section on Human Factors and Automation in Vehicles: Designing Highly Automated Vehicles With the Driver in Mind," *Hum. Factors J. Hum. Factors Ergon. Soc.* 54(5):681-686, 2012.
11. Hancock, P.A., "Driven to Distraction and Back Again," in: Regan, M. A., Lee, J. D., and Victor, T. W., eds., *DRiver Distraction and Inattention : Advances in Research and Countermeasures*, 1st ed., Ashgate Publication Limited, Surrey, England: 440, 2013.
12. Bainbridge, L., "Ironies of automation," *Automatica* 19(6):775-779, 1983, doi:10.1016/0005-1098(83)90046-8.
13. Strand, N., Nilsson, J., Karlsson, I.C.M., and Nilsson, L., "Semi-automated versus highly automated driving in critical situations caused by automation failures," *Transp. Res. Part F Traffic Psychol. Behav.* 27:218-228, 2014, doi:10.1016/j.trf.2014.04.005.
14. Sarter, N.B., Woods, D.D., and Billings, C.E., "Automation surprises," in: Salvendy, G., ed., *Handbook of Human Factors and Ergonomics*, 2nd ed., Wiley, New York: 1926-1943, 1997.

15. Sarter, N.B. and Woods, D.D., "How in the world did we ever get into that mode? Mode error and awareness in supervisory control," *Hum. Factors* 37(1):5-19, 1995.
16. Degani, A., Shafto, M., and Kirlik, A., "Modes in automated cockpits: Problems, data analysis, and a modeling framework," *Proceedings of the 36th Israel Annual Conference on Aerospace Science*, Haifa, Israel, 1995.
17. Merat, N. and Lee, J.D., "Preface to the Special Section on Human Factors and Automation in Vehicles: Designing highly automated vehicles with the driver in mind," *Hum. Factors J. Hum. Factors Ergon. Soc.* 54(5):681-686, 2012, doi:10.1177/0018720812461374.
18. Pritchett, A.R., "Aviation Automation: General Perspectives and Specific Guidance for the Design of Modes and Alerts," *Rev. Hum. Factors Ergon.* 5(1):82-113, 2009, doi:10.1518/155723409X448026.
19. Young, M.S., Stanton, N.A., and Harris, D., "Driving automation: Learning from aviation about design philosophies," *Internaltional J. Veh. Des.* 45(3):323-338, 2007.
20. Rasmussen, J., "Risk Management in a Dynamic Society: a modeling problem," *Saf. Sci.* 27:183-213, 1997, doi:10.1016/S0925-7535(97)00052-0.
21. Merat, N., Jamson, A.H., Lai, F.C.H., Daly, M., and Carsten, O.M.J., "Transition to manual: Driver behaviour when resuming control from a highly automated vehicle," *Transp. Res. Part F Traffic Psychol. Behav.* 26(PART A):1-9, 2014, doi:10.1016/j.trf.2014.05.006.
22. Parasuraman, R. and Sheridan, T.B., "A model for types and levels of human interaction with automation," *Syst. Man …* 30(3):286-297, 2000, doi:10.1109/3468.844354.
23. Gold, C., Dambock, D., Lorenz, L., and Bengler, K., "'Take over!' How long does it take to get the driver back into the loop?," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* 57(1):1938-1942, 2013, doi:10.1177/1541931213571433.
24. Mok, B., Johns, M., Lee, K.J., Miller, D., Sirkin, D., Ive, P., and Ju, W., "Emergency , Automation Off?: Unstructured Transition Timing for Distracted Drivers of Automated Vehicles," *Proc. 2015 IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015.
25. Miller, D., Sun, A., Johns, M., Ive, H., Sirkin, D., Aich, S., and Ju, W., "Distraction becomes engagement in automated driving," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.* 2-6, 2015.
26. Jamson, A.H., Merat, N., Carsten, O.M.J., and Lai, F.C.H., "Behavioural changes in drivers experiencing highly-automated vehicle control in varying traffic conditions," *Transp. Res. Part C Emerg. Technol.* 30:116-125, 2013, doi:10.1016/j.trc.2013.02.008.
27. Carsten, O., Lai, F.C.H., Barnard, Y., Jamson, H., and Merat, N., "Control task substitution in semi-automated driving: Does it matter what aspects are automated?," *Hum. Factors* 54(5):747-761, 2012.
28. Allen, R.W., Rosenthal, T., J., and Cook, M., "A short history of driving simulation," *Handbook of driving simulation for engineering, medicine, and psychology*, 2-3, 2011.
29. Young, K.L., Regan, M.A., and Lee, J.D., "Measuring the effects of driver distraction: direct driving performance methods and measures," *Driver Dirstraction: Theory, Effects, and Mitigation*, 86-102, 2009.
30. Endsley, M.R., "Toward a Theory of Situation Awareness in Dynamic Systems," *Hum. Factors J. Hum. Factors Ergon. Soc.* 37(1):32-64, 1995, doi:10.1518/001872095779049543.
31. Gugerty, L., "Situation awareness in driving," *Handbook of driving simulation for engineering, medicine, and psychology*, 1-8, 2011.
32. Lee, J.D. and See, K.A., "Trust in computer technology and the implications for design and evaluation," in: Miller, C., ed., Etiquette for Human-Computer Work: Technical Report FS-02-02, American Association for Artificial Intelligence, Menlo Park, CA: 20-25, 2002.
33. Vlakveld, W., Romoser, M.R.E., Mehranian, H., Diete, F., Pollatsek, A., and Fisher, D.L., "Do Crashes and Near Crashes in Simulator-Based Training Enhance Novice Drivers' Visual Search for Latent Hazards?," *Transp. Res. Rec. J. Transp. Res. Board* 153-160, 2011, doi:10.1016/j.biotechadv.2011.08.021.Secreted.
34. Garay-Vega, L., Fisher, D.L., and Pollatsek, A., "Hazard Anticipation of Novice and Experienced Drivers: Empirical Evaluation on a Driving Simulator in Daytime and Nighttime Conditions," *Transp. Res. Rec.* 2009(1):1-7, 2007, doi:10.3141/2009-01.
35. Underwood, G., Crundall, D., and Chapman, P., "Driving simulator validation with hazard perception," *Transp. Res. Part F Traffic Psychol. Behav.* 14(6):435-446, 2011, doi:10.1016/j.trf.2011.04.008.
36. Taylor, T., Pradhan, A.K., Divekar, G., Romoser, M., Muttart, J.W., Gomez, R., Pollatsek, A., and Fisher, D.L., "The view from the road: The contribution of on-road glance-monitoring technologies to understanding driver behavior.," *Accid. Anal. Prev.* 58(null):175-86, 2013, doi:10.1016/j.aap.2013.02.008.
37. Woods, D.D. and Branlat, M., "Hollnagel's test: Being 'in control' of highly interdependent multi-layered networked systems," *Cogn. Technol. Work* 12(2):95-101, 2010, doi:10.1007/s10111-010-0144-5.
38. Degani, A. and Heymann, M., "Formal verification of human-automation interaction.," *Hum. Factors* 44(1):28-43, 2002.
39. Bolton, M.L., Bass, E.J., Member, S., and Siminiceanu, R.I., "Using formal verification to evaluate Human-automation interaction: A review," 43(3):488-503, 2013.
40. Leveson, N.G., "A systems-theoretic approach to safety in software-intensive systems," *IEEE Trans. Dependable Secur. Comput.* 1(1):66-86, 2004, doi:10.1109/TDSC.2004.1.

SAE INTERNATIONAL

ARTICLE INFO

Article ID: 2017-01-0052

doi:10.4271/2017-01-0052

Markov Chain-based Reliability Analysis for Automotive Fail-Operational Systems

Andre Kohn, AUDI AG
Rolf Schneider, *AUDI AG*
Antonio Vilela and Udo Dannebaum, *Infineon Technologies AG*
Andreas Herkersdorf, *Technical University of Munich*

Abstract

A main challenge when developing next generation architectures for automated driving ECUs is to guarantee reliable functionality. Today's fail safe systems will not be able to handle electronic failures due to the missing "mechanical" fallback or the intervening driver. This means, fail operational based on redundancy is an essential part for improving the functional safety, especially in safety-related braking and steering systems. The 2-out-of-2 Diagnostic Fail Safe (2oo2DFS) system is a promising approach to realize redundancy with manageable costs. In this contribution, we evaluate the reliability of this concept for a symmetric and an asymmetric Electronic Power Steering (EPS) ECU. For this, we use a Markov chain model as a typical method for analyzing the reliability and Mean Time To Failure (MTTF) in majority redundancy approaches. As a basis, the failure rates of the used components and the microcontroller are considered. The comparison to a non-redundant system shows a significantly higher reliability and MTTF of the redundant approaches.

History

Received: 05 Apr 2017
Published: 28 Mar 2017

Citation

Kohn, A., Schneider, R., Vilela, A., Dannebaum, U. et al., "Markov Chain-based Reliability Analysis for Automotive Fail-Operational Systems," *SAE Int. J. Trans. Safety* 5(1):2017, doi:10.4271/2017-01-0052.

ISSN: 2327-5626
e-ISSN: 2327-5634

Introduction

One of the main aspects when developing safety-related automotive systems is fulfilling the requirements of the ISO 26262 standard for functional safety. Today, automotive Electronic Control Units (ECUs) are able to detect a misbehavior and as a consequence shutdown or reboot the device. The corresponding fail-safe architectures are realized in current single-core or multi-core control units.

Due to the more complex algorithms, Driving Assistance Systems (DAS) can already overtake several tasks which are up to now handled by the driver. This will be much more intensified in the future by extensive automation. But the system shutdown of a fail-safe architecture in a fully-automated vehicle will cause an uncontrollable behavior of the car. Hence, a redesign of the safety architecture, including the replacement of fail-safe systems by fail operational approaches, is an important step for coming E/E architectures.

In our contribution we present a system design approach for a fail operational Electronic Power Steering (EPS) ECU which uses two subsystems based on a dual-core lockstep CPU each. We define a corresponding Markov chain and identify several failure use cases to derive the reliability of the EPS ECU. Furthermore, we compare the results to a non-fail operational system with respect to the general automotive life cycle.

Related Work

Fail-operational is up to now not widely-spread in automotive industry although it is not a completely new topic. An overview about today's microcontroller safety architectures is given in previous publications [1], [2]. Furthermore, the authors present the 2-out-of-2 Diagnostic Fail Safe (2oo2DFS) architecture as an approach for fail-operational in future automotive multicore systems. These contributions are the basis for our work in this paper.

In this context, researchers from Tsinghua University of Beijing analyzed different M out of N decisions concerning safety and reliability [3]. However, this work is a more generic approach and does not consider automotive systems.

A comparison of a lock-step pair, the traditional Triple Modular Redundancy (TMR)-spare scheme and a Duplication with Temporary TMR and Reconfiguration (DTTR) scheme can be found in [4]. The results show a reliability improvement of their DTTR scheme for multicore ECUs.

The focus of another related work was an approach for a flexible microcontroller architecture which can be applied for fail-safe and also fail-operational. This concept addresses semiconductor vendors for developing new hardware architectures considering functional safety issues [5].

A review of fault-tolerant architectures focusing on industrial applications is given in [6]. The authors propose a cost effective lockstep platform implemented as a fail-operational or two fail-silent systems.

The challenges when replacing fail-safe by fail-operational in automotive ECUs is also addressed in [7]. For this, common fail-operational approaches such as TMR, a duo-duplex and a hybrid concept are presented.

Reliability Basics

Generally, a technical system consists of several units which must not fail for a certain time to improve the reliability. The reliability defines the unit's trustworthiness relating to a continuous functionality for a specified time. Hence, the reliability R of a technical unit defines the probability of functionality within the interval (0, t]. Another value in this context is the failure rate λ. This means the time-dependent reliability $R(t)$ is defined as:

$$R(t) = e^{-\int_0^t \lambda(\tau)d\tau}$$

Electronic components mostly have an exponential distribution for the fault rate which is represented by the so called *Bathtub Curve*.

This curve consists of three phases while the first phase the fault rate is much higher. The reason for this are early failures in the start-up at t = 0 due to manufacturing errors. The second phase represents the useful life of the component with a constant failure rate while in the end the probability for a failure is again higher due to component aging (Figure 1).

The assumptions of a system analysis often includes a faultless system at start-up which means $R(0) = 1$. With this prerequisite the following reliability function can be applied:

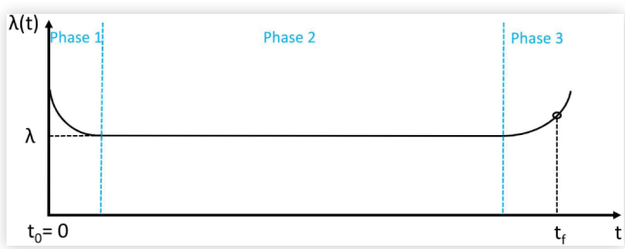
$$R(t) = e^{-\lambda t}$$

Generally, the reliability $R(t)$ is a degree for the continuous functionality of an irreparable technical unit. Hence, in the failure case it cannot be moved to an operating state.

A further factor for reliability analysis is the mean durability of a system which is represented by the *Mean Time To Failure (MTTF)* and which is defined by the integral on the reliability:

$$MTTF = \int_0^\infty R(t)dt$$

FIGURE 1 Typical course for the failure rate of a technical unit.



Safety-related Automotive Systems

The reliability of safety-related system is an essential part when developing ECU architectures. This means, the car must always be controllable to avoid hazards which might injure the passengers or traffic participants. Hence, the availability of basic vehicle functions which control the vehicle, namely braking and steering, must be guaranteed. But the trend towards more intelligent Driving Assistance Systems (DAS) and automated driving complicate this concern due to the fact that more driving tasks are handled by the vehicle instead of the driver. This is also confirmed by the SAE standard J3016 which defines fives levels of driving automation [14]. The highest level implies a fully automated car which provides a full-time performance by an automated driving system even without driver. Hence, a mechanical fallback for braking and steering, e.g. the driver, is not available. As a consequence, these levels require a redesign of the automotive E/E and ECU architectures to achieve the fault tolerance required for realizing a fail-operational function. Typical use cases are systems as steer-by-wire and brake-by-wire which use a second control electronic instance so that functionality can temporally by guaranteed even in case of a failure.

Generally, safety-related automotive systems must comply with the requirements of the ISO 26262 which defines four risk classes, namely ASIL-A, ASIL-B, ASIL-C and ASIL-D. Moreover, the standard uses the Failures in Time (FIT) as a measure of the failure rate and as a measure of the probability of failure per hour. This includes the calculation of the failure amount in 10^9 hours. In this context, the ASILs require different FIT rates which are listed in Table 1.

Homogeneous and Diverse Redundancy

Fail operational systems are the main challenge of future E/E and ECU architectures to comply with safety requirements in automated cars. A key aspect for realizing fail operational is the implementation of either homogeneous or diverse redundancy.

Homogeneous redundancy means an application of two identical calculation instances. The advantage is a reduced development effort due to the same components. But the usage of this approach is often limited by the complexity to control dependent failures, especially the ones originating from systematic faults. This means, a systematic error affects all parts of a homogeneous system.

Diverse redundancy contains two or more heterogeneous components which do either the same or different calculations to achieve equivalent functionality. Hence, different algorithms or sub-architectures can be applied. Typically, in avionics there is homogeneous redundancy to improve the reliability and diversity to achieve functional safety. An advantage of this approach is that components can be developed by different suppliers. Furthermore, diverse hardware with different failure rates helps to reduce common cause faults which lead to system failure.

TABLE 1 ASIL FIT rate requirements [7].

ASIL	Failure Rate
D	$< 10^{-8}h^{-1} = 10$ *FIT*
C	$< 10^{-7}h^{-1} = 100$ *FIT*
B	$< 10^{-7}h^{-1} = 100$ *FIT*
A	$< 10^{-6}h^{-1} = 1000$ *FIT*

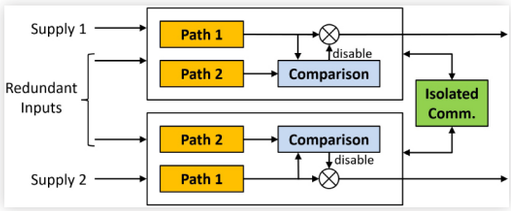
Majority Redundancy

A popular approach in redundant circuits is the so called Majority Redundancy which is based on an M out of N (MooN) decision. This means the functionality of a system with *N* components can be guaranteed if *M* components are faultless. A voter as a central unit decides which calculating units provide correct values. Generally, all elements are implemented to guarantee the same functionality so that each of them has the same failure rate at runtime (*hot redundancy*). A special case of a MooN redundancy is a series connection (M = N) so that all elements must be faultless. In a system with two components this also means a higher failure rate and hence a lower availability. However, compared to a one-component system, the second communication channel improves the system's integrity. A disadvantage is that in case of different outputs if one channel fails, the correct value cannot be identified and the system must be shut down or transferred to the safe state.

An alternative approach for a fail operational architecture in automotive systems is the 2-out-of-2 Diagnostic Fail Safe (2oo2DFS) architecture realized by two 1-out-of-1 (1oo1) channels with diagnostics (1oo1D). The monitoring function of each 1oo1D channel is capable to detect a misbehavior and shut down the channel thus achieving a fail silent property. The 2oo2DFS can be enhanced with a cross-communication between the two channels enabling one of the channels to diagnose the state of the other channel and by that takeover some of the functionality of the failing channel. This is of interest in the presence of an asymmetric software architecture where one channel comprises safety-critical functions and comfort functions and the second channel safety-critical functions and partial comfort functions in a hot standby mode. After a detected failure of the first channel, the second channel can take over partial functionality of the comfort functions. The same principle could be applied to safety critical functions at the cost of a more complex software architecture. An example for the 2oo2DFS architecture can be found in Figure 2.

Safety-related microcontrollers of the automotive industry often use Lockstep cores for redundant calculations in order to achieve the necessary level safety integrity. This feature allows realizing the fail safe part of the 2oo2DFS architecture.

FIGURE 2 Example for a 2oo2DFS architecture.



In our contribution we apply the 2oo2DFS to a close-to-production automotive function while we analyze the reliability of a symmetric and an asymmetric 2oo2DFS system.

Redundancy as a Basis for Fail-Operational in Cars

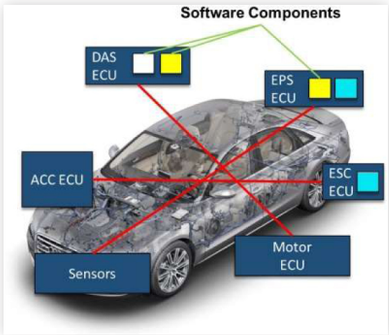
Generally, redundancy respectively MooN decisions require a safety analysis on different implementation levels. Distributed redundancy on E/E architecture level means an implementation of software components on different ECUs which can be located in different vehicle domains. For example, a steering control function can be computed in an Electronic Power Steering (EPS) ECU and in a central DAS ECU in parallel. This means, the ECUs provide redundant outputs from different domains.

An implementation example for diverse redundancy on E/E level is illustrated in Figure 3.

As an alternative, homogeneous redundancy can be realized by using two identical ECUs, either in different domains or within the same domain. While diverse redundancy can apply existing ECUs, the homogeneous approach doubles the ECU costs, packaging, weight and necessary bandwidth of the communication bus.

A higher level of integration for fail operational consists on the implementation of the redundant sub-systems inside one ECU. In many domains like EPS this alternative is gaining momentum. Hence, an EPS ECU is an appropriate candidate for a concept evaluation of an ECU internal fail-operational system.

FIGURE 3 Example for distributed redundancy on E/E architecture level.



2oo2DFS Architecture for Vehicles

In this contribution, we apply the 2oo2DFS approach with two separate microcontrollers providing two homogeneous cores each. Furthermore, the hardware architecture uses two redundant torque angle sensors for signal capturing and forwarding them to the CPU. In case of a symmetric architecture, the dual-core lockstep CPUs are identical and use a safety power supply with an integrated watchdog each. This allows a safe and uninterruptable supply for an ASIL-D implementation. A three-phase driver IC with seven MOSFETs is used to control the motor while a rotor position sensor captures the required values for the microcontroller. The controller applies a CAN transceiver to communicate over the vehicle high-speed CAN bus (Figure 4).

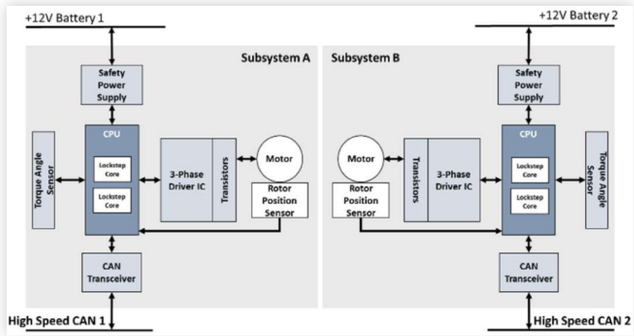
The integrated fail-operational approach could be a basis for such implementations in future domain control units (DCUs). For a reliability evaluation of this 2oo2DFS system, the functions *R* and *MTTF* have to be calculated.

System Reliability of the 2oo2 Approach using Markov Chains

For determining the system reliability, literature describes several methods while we use a Markov Chain-based model which is recommended and applied for many MooN decisions [1],[9]. Due to redundant components, the functionality of the fail-operational is still given even in case of an electronic failure. Hence, we only consider in this context irreparable subsystems and permanent faults. The calculation use the Reliability data handbook of the IEC 62380 standard and the Siemens norm SN29500 as a basis [11],[12]. This conceptual work focuses on the failure rates of the used hardware components while possible failures based on wiring, soldering or temperature dependencies are not considered here.

The 2oo2DFS in Figure 4 consists of two symmetric or asymmetric subsystems with redundant hardware and

FIGURE 4 Example for a fail-operational EPS system.



software components. For simplification, we first consider a single subsystem with the following failure rates:

- λ_0: One core of the dual-core lockstep microcontroller
- λ_1: 3-phase driver IC for motor control
- λ_2: Safety power supply
- λ_3: Torque angle sensor
- λ_4: CAN transceiver
- λ_5: 3-phase bridge
- λ_6: Rotor position sensor
- λ_7: Motor
- λ_8: Phase splitter

The failure rate of the complete subsystem S is the sum of the component's failure rates:

$$\lambda_S = \sum_{i=0}^{8} \lambda_i = \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8$$

The corresponding fault tree shows the series connection of the components so that the corresponding failure rates are summed (Figure 5).

In the following calculations, we differ between the failure rate of a microcontroller core (λ_C) and the summed remaining components (λ_K). With these prerequisites the Markov chain model for the subsystem S can be derived (Figure 6).

The starting point is the initial state Z_0 which represents a faultless subsystem S. The system is down if exactly one component does not work and hence the microcontroller cannot read or provide correct values anymore. Furthermore, both cores of the microcontroller can be broken or first one core fails before one component does not work (absorbing state Z_D). In case of a faultless subsystem, the probability for a core failure is $2\lambda_C$ which means either Core 1 or Core 2 fails (state Z_C). We assume an identical failure rate for the cores so that we can summate them. If one core already failed, the failure probability for the faultless core is λ_C ($Z_C \rightarrow Z_D$). The Markov model in is the basis for the reliability function and MTTF of the subsystem:

$$R_S(t) = e^{-(\lambda_K + 2\lambda_C(\lambda_C + \lambda_K))t}$$

$$MTTF_S = \frac{1}{\lambda_K + 2\lambda_C(\lambda_C + \lambda_K)}$$

If we consider the complete system, we have to enhance the Markov model and the fault tree by a second subsystem. Hence, we differ between subsystem A and subsystem B which form a parallel connection with equal constraints. This results in a combined parallel and series connections (Figure 7).

If the system is in the initial state Z_0 and one of the components fails, it changes with failure rate λ_{AK} to the state Z_{AD}, respectively with λ_{BK} to the state Z_{BD}. In this case, we assume that nor sensor signals can be read neither actors can be controlled anymore and one of the subsystems is down.

A further change of state from Z_0 to Z_{AC} or Z_{BC} is applied if one of the cores fail with the probability $2\lambda_{AC}$ respectively $2\lambda_{BC}$. The remaining cores as well as the components of the subsystems A and B are faultless in this case. If the other core or one of the components then fail, the corresponding subsystem is down as well (state Z_{AD} or Z_{BD}). We assume that the state is now in state Z_{AD}. The probability for a complete broken system is now the same as the probability for changing state from Z_0 to Z_{BD} (λ_{GesA}). The corresponding fault tree for the two-component system is illustrated in Figure 8.

The reliability $R(t)$ of this system is the sum of the state probabilities $P(t)$ excluding the "down state" Z_D.

$$R(t) = P_{Z0}(t) + P_{ZAC}(t) + P_{ZBC}(t) + P_{ZAD}(t) + P_{ZBD}(t)$$

Furthermore, the MTTF can be calculated as the integral of the reliability function:

$$MTTF = \int_0^\infty P_{Z0}(t) + P_{ZAC}(t) + P_{ZBC}(t) + P_{ZAD}(t) + P_{ZBD}(t)dt$$

The state probabilities are calculated in the following section.

FIGURE 5 Fault tree for subsystem S.

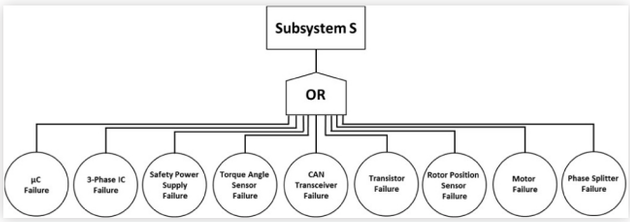


FIGURE 6 Markov chain for subsystem S.

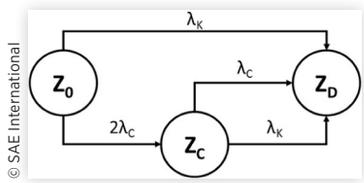


FIGURE 7 Markov chain for two-component system.

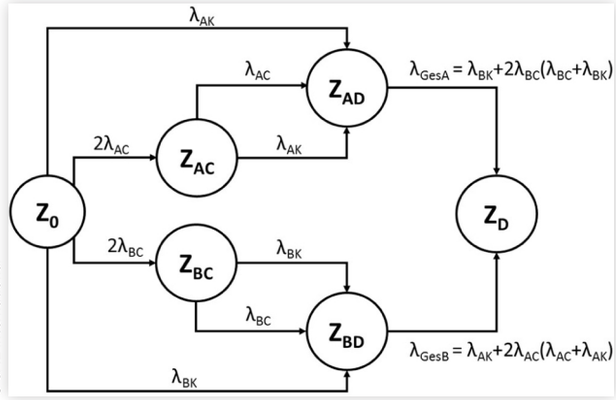
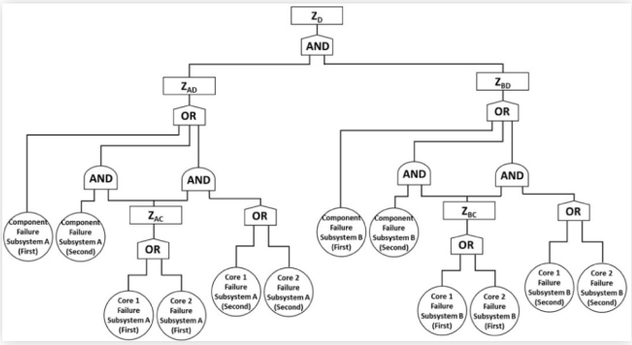


FIGURE 8 Fault tree for two-component system.



State Probabilites, Reliability and Mean Time To Failure

The series and parallel connection requires a combination of corresponding calculation methods for Markov chain analysis [13]. The model in Figure 7 is the basis for the transition matrix in the appendix. With the help of this matrix, the following differential equations can be derived:

$$\dot{P}_{Z0} = -(2\lambda_{AC} + 2\lambda_{BC} + \lambda_{AK} + \lambda_{BK})P_{Z0}$$
$$\dot{P}_{ZAC} = 2\lambda_{AC}P_{Z0} - (\lambda_{AC} + \lambda_{AK})P_{ZAC}$$
$$\dot{P}_{ZAD} = \lambda_{AK}P_{Z0} + (\lambda_{AC} + \lambda_{AK})P_{ZAC} - (\lambda_{BK} + 2\lambda_{BC}(\lambda_{BC} + \lambda_{BK}))P_{ZAD}$$
$$\dot{P}_{ZBC} = 2\lambda_{BC}P_{Z0} - (\lambda_{BC} + \lambda_{BK})P_{ZBC}$$
$$\dot{P}_{ZBD} = \lambda_{BK}P_{Z0} + (\lambda_{BC} + \lambda_{BK})P_{ZAD} - (\lambda_{AK} + 2\lambda_{AC}(\lambda_{AC} + \lambda_{AK}))P_{ZBD}$$

Furthermore, we assume a faultless system with the initial state Z_0 so that the following prerequisites are considered:

$$P_{Z0}(t=0) = 1$$
$$P_{ZAC}(t=0) = P_{ZAD}(t=0) = P_{ZBC}(t=0) = P_{ZBD}(t=0) = P_{ZD}(t=0) = 0$$

These conditions allow to calculate the state probabilities and the resulting reliability functions for the asymmetric ($R_{asym}(t)$) and the symmetric ($R_{sym}(t)$) system. In a next step the MTTF for both approaches can be calculated. In contrast to the asymmetric approach, the subsystems do not differ in the symmetric approach so that we can simplify the equation with the constraints:

$$\lambda_C = \lambda_{AC} = \lambda_{BC}$$
$$\lambda_K = \lambda_{AK} = \lambda_{BK}$$

All these equations can be found in the appendix of this contribution.

Constraints of the Analysis

A representative analysis requires the consideration of the component and microcontroller FIT rates. The goal is to achieve a system's FIT rate which meets the requirements from Table 1. However, these values are based on the sum of single point faults and residual faults, meaning on failures which are not completely covered. Hence, the diagnostic coverage as a measure for failure detection by testing is < 100 %. The complete failure rate can be optimized by mechanisms to reduce the raw failure rate.

Generally, we identified the FIT rates by a Failure Modes, Effects and Diagnostic Analysis (FMEDA) based on the SN29500 standard [12]. An FMEDA allows a quantitative analysis of all electronic components with respect to failure rate, failure type and effects of the safety function. Another goal is a systematic analysis to detect hazard-free failures respectively Safe Failure Fraction (SFF) and the coverage. Integrated safety mechanisms of microcontrollers can also be considered when configuring the FMEDA.

Due to the fact that cars are common in all climes, the operating temperature influences the reliability of an automotive system. Hence, the industry claims a reliability in the range from -40°C to 125°C. Moreover, the component's life time of E/E system is expected to last 15 years, 8000 hours of operation and a 300.000 km. These constraints are used for determine the CPU FIT rate by the FMEDA. In our contribution, use the failure rates of the hardware components which we identified by a CPU FMEDA. For this, we appoint to internal research projects and to information from semiconductor vendors. In our work, we focus on hard errors because soft errors can often be corrected and we assume identical components for both subsystems. This means, in the asymmetric approach, the subsystems only differ in the applied microcontroller. The following component FIT rates are used:

- λ_1: 140 FIT
- λ_2: 200 FIT
- λ_3: 40 FIT
- λ_4: 5 FIT
- λ_5: 80 FIT
- λ_6: 40 FIT
- λ_7: 5 FIT
- λ_8: 70 FIT

Based on these values, the component FIT rate λ_K for one subsystem can be calculated. Due to a potential parallel component failure, the FIT rates are summed:

$$\lambda_K = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 = 580\,\text{FIT}$$

In our work, we analyzed three different use cases. The first step applies a single subsystem with the described components which represents a microcontroller with a single lockstep core. The FIT rate of a core (λ_C = 0.367 FIT) could be determined with the help of FMEDA table provided by a semiconductor vendor. In our case, this value does not include any connection to the peripheral or memory. If we also consider the interfaces to sensors and actors, our FMEDA table calculates a FIT rate of λ_C = 47.11 FIT which serves as a basis for our calculation.

For our symmetric approach, both microcontrollers apply the failure rate λ_0 = 47.11 FIT while we assume two different controllers in the asymmetric system. In this case we chose λ_{0A} = 47.11 FIT and a variable FIT rate for subsystem B (λ_{0B} = 0.367 FIT and λ_{0B} = 150 FIT).

Results

These values allow to calculate the reliability functions for a subsystem (*RS(t)*), the symmetric system ($R_{sym}(t)$) and the two asymmetric systems ($R_{asym1}(t)$ and $R_{asym2}(t)$). The corresponding analysis is illustrated in Figure 9.

The graphs show an exponentially decreasing reliability for all approaches while the 2oo2DFS systems are the significantly better choice for reliable systems. Furthermore, the difference between the symmetric and asymmetric graphs are minimal.

The graphs in Figure 10 show the reliability functions after 15 years of operation while leap years are not considered. After this time, a single using a core FIT rate of λ_C = 47.11 FIT has a reliability of *RS(t*= subsystem *15 yrs)* = 0.9266 (92.66 %). In contrast, the probability for a faultless symmetric 2oo2DFS using core FIT rates of λ_C = 47.11 FIT is $R_{sym}(t=15\ yrs)$ = 0.9946 (99.46 %). In this case, the values of asymmetric systems with λ_{0B} = 0.367 FIT and λ_{0B} = 100 FIT do not differ from the symmetric approach.

In a further step, we analyzed the MTTF as a function of the core FIT rates. For this, we incremented the failure rate from 1 FIT to 150 FIT. This means, for the symmetric system both core FIT rates are incremented while for the asymmetric approach only subsystem B is concerned. Subsystem A has

FIGURE 9 Reliability functions for different approaches.

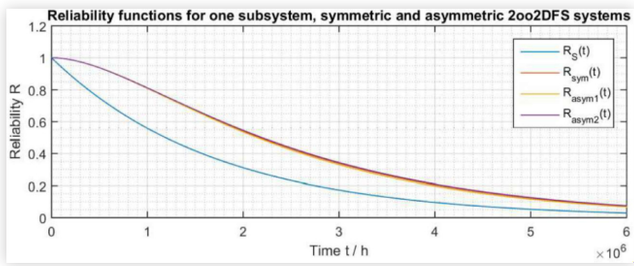


FIGURE 10 Reliability functions after 15 years of operation.

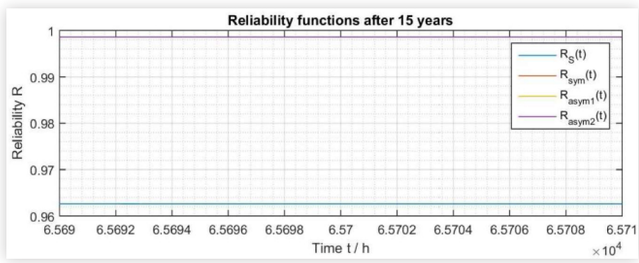
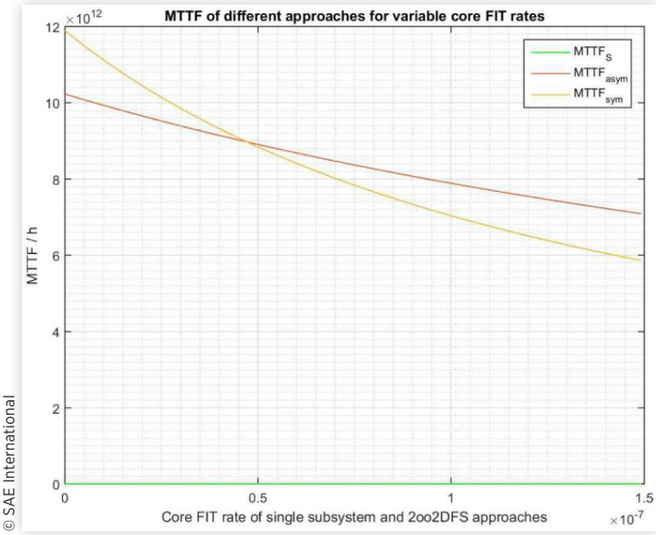


FIGURE 11 MTTFs for single subsystem, symmetric and asymmetric 2oo2DFS systems.



a constant value (λ_{0A} = 47.11 FIT). Due to the scaling, the $MTTF_S$ of the single subsystem seems nearly 0 h. However, we measured values about $1.72 \cdot 10^6$ h (= 196 yrs) for the 1-FIT core and $1.60 \cdot 10^6$ h (= 183 yrs) for the 150-FIT core.

In contrast to the single subsystem, the MTTFs of the 2oo2DFS approaches decrease significantly for higher FIT rates. Nevertheless, these systems show much higher MTTFs meaning for the asymmetric implementation $10.23 \cdot 10^{12}$ h (= $1.17 \cdot 10^9$ yrs) for the 1-FIT core and $7.09 \cdot 10^{12}$ h (= $809.61 \cdot 10^6$ yrs) for 150-FIT core.

Finally, we calculated $11.88 \cdot 10^{12}$ h (= $1.36 \cdot 10^9$ yrs) for symmetric 1-FIT cores and $586.24 \cdot 10^6$ h (= $669.23 \cdot 10^6$ yrs) for 150-FIT cores. In the beginning, the symmetric approach provides the best MTTF (Figure 11). For FIT rates higher than 49 FIT, the asymmetric 2oo2DFS show the highest MTTF.

Conclusion

In our contribution, we evaluated the reliability of the symmetric and asymmetric 2oo2DFS approach with the help of a Markov chain. Our use case was an automotive EPS ECU based on either a single or two dualcore microcontrollers. Furthermore, we assumed constant FIT rates for the components and applied a core FIT rate of 47.11 FIT as a basis for each system. For the second subsystem of the asymmetric EPS ECU and also for the symmetric approach, we calculated the reliability and MTTF for different core FIT rates. As expected, the redundant systems are much more reliable and provide multiply higher MTTFs than the single system. Generally, in our case an asymmetric approach is recommended for core FIT rates higher than 49 FIT while for more reliable cores, the symmetric approach is the better choice.

As a conclusion, we can say that today all three approaches can be applied to the described scenario. After 15 yrs of operation, each system provides a reliability of more than 92 %. However, as future ECU architectures require fail operational approaches, a single subsystem will not be sufficient. In general, this approach is not sufficient for considering systematic faults which have to be handled by further mechanisms.

Contact Information

André Kohn, M.Sc.
AUDI AG
85045 Ingolstadt, Germany
andre.kohn@audi.de

Dipl.-Inform. (FH) **Rolf Schneider**
AUDI AG
85045 Ingolstadt, Germany
rolf.schneider@audi.de

Antonio Vilela, M.Sc.
Infineon Technologies AG
85579 Neubiberg, Germany
antonio.vilela@infineon.com

Dipl.-Ing. (FH) **Udo Dannebaum**
Infineon Technologies AG
85579 Neubiberg, Germany
udo.dannebaum@infineon.com

Prof. Dr. sc. techn. **Andreas Herkersdorf**
Institute for Integrated Systems
Technical University of Munich
80290 München, Germany
herkersdorf@tum.de

Definitions/Abbreviations

2oo2 - Two-out-of-two

ASIL - Automotive Safety Integrity Level

DAS - Driver Assistance System

DCU - Domain Control Unit

DFS - Diagnostic Fail Safe

DTTR - Duplication with Temporary TMR and Reconfiguration

ECU - Electronic Control Unit

E/E - Electrical/Electronic

EPS - Electronic Power Steering

ESC - Electronic Stability Control

FIT - Failure In Time

FMEDA - Failure Modes, Effects and Diagnostic Analysis

IC - Integrated Circuit

MTTF - Mean Time To Failure

MooN - M out of N

MOSFET - Metal Oxide Semiconductor Field Effect Transistor

TMR - Triple Modular Redundancy

References

1. Kohn, A., Kaeßmeyer, M. et.al., "Fail-Operational in Safety-Related Automotive Multi-Core Systems", *10th IEEE International Symposium on Industrial Embedded Systems (SIES)*, IEEE, Siegen, 2015
2. Kohn, A., Schneider, R., Vilela, A., Roger, A. et al., "Architectural Concepts for Fail-Operational Automotive Systems," SAE Technical Paper 2016-01-0131, 2016, doi:10.4271/2016-01-0131.
3. Dai, X., Wei, D. and Xinya, S., "Reliability and safety analysis of M out of N system based on Markov Process." *IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, Chongqing, 2016
4. Imai, M and Yoneda, T., "Comparing Permanent and Transient Fautl Tolerance of Multiple-core based Dependable ECUs", *CARS 2015 - Critical Automotive Applications: Robustness 6 Safety*, Paris, 2015
5. Mariani, R., Kuschel, T. and Shigehara, H., "A flexible microcontroller architecture for fail-safe and fail-operational systems", *2nd HiPEAC Workshop on Design for Reliability (DFR'10)*, Pisa, 2010
6. Baleani, M., Ferrari et.al., "Fault-Tolerant Platforms for Automotive Safety-Critical Applications", In: *Proceedings of the 2003 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, ACM, pp. 170-177
7. Temple, C. and Vilela, A., "*Fehlertolerante Systeme im Fahrzeug - Von Fail Safe zu Fail Operational*", Elektroniknet, 2014
8. Börcsök, J., "Elektronische Sicherheitssysteme: Hardwarekonzepte, Modelle und Berechnung.", VDE, 2007.
9. Eberlin, S. and Hoch, B., "*Zuverlässigkeit und Verfügbarkeit technischer Systeme*.", Springer Vieweg, 2014.
10. ISO 26262 - Part 5: Product development at the hardware level, ISO, 2011
11. IEC TR 62380, IEC, 2004
12. Siemens AG, SN92500 1-16
13. Eberlin, S. and Hock, B, "*Zuverlässigkeit und Verfügbarkeit technischer Systeme*", Springer Vieweg, 2014
14. SAE J3016: Taxonomy and Definitions for Terms Related to On-Boars Motor Vehicle Automated Driving Systems, SAE, 2016

Appendix

Transition Matrix

$$\begin{pmatrix} \dot{P}_{Z0} \\ \dot{P}_{ZAC} \\ \dot{P}_{ZAD} \\ \dot{P}_{ZBC} \\ \dot{P}_{ZBD} \\ \dot{P}_{ZD} \end{pmatrix} = \begin{pmatrix} -(2\lambda_{AC}+2\lambda_{BC}+\lambda_{AK}+\lambda_{BK}) & 0 & 0 & 0 & 0 & 0 \\ 2\lambda_{AC} & -(\lambda_{AC}+\lambda_{AK}) & 0 & 0 & 0 & 0 \\ \lambda_{AK} & \lambda_{AC}+\lambda_{AK} & -(\lambda_{BK}+2\lambda_{BC}(\lambda_{BC}+\lambda_{BK})) & 0 & 0 & 0 \\ 2\lambda_{BC} & 0 & 0 & -(\lambda_{BC}+\lambda_{BK}) & 0 & 0 \\ \lambda_{BK} & 0 & 0 & \lambda_{BC}+\lambda_{BK} & -(\lambda_{AK}+2\lambda_{AC}(\lambda_{AC}+\lambda_{AK})) & 0 \\ 0 & 0 & \lambda_{BK}+2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}) & 0 & \lambda_{AK}+2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}) & 0 \end{pmatrix} \begin{pmatrix} P_{Z0} \\ P_{ZAC} \\ P_{ZAD} \\ P_{ZBC} \\ P_{ZBD} \\ P_{ZD} \end{pmatrix}$$

State Probabilities

$$P_{Z0}(t) = e^{-(2\lambda_{AC}+2\lambda_{BC}+\lambda_{BK}+\lambda_{AK})t}$$

$$P_{ZAC}(t) = \frac{2\lambda_{AC}}{\lambda_{AC}+2\lambda_{BC}+\lambda_{BK}} e^{-(\lambda_{AC}+\lambda_{AK})t} - \frac{2\lambda_{AC}}{\lambda_{AC}+2\lambda_{BC}+\lambda_{BK}} e^{-(2\lambda_{AC}+2\lambda_{BC}+\lambda_{BK}+\lambda_{AK})t}$$

$$P_{ZAD}(t) = \left[\frac{(\lambda_{AC}+\lambda_{AK})\cdot 2\lambda_{AC}}{(\lambda_{AC}+\lambda_{AK}-\lambda_{BK}-2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})} + \frac{\lambda_{AK}}{(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})}\right] e^{-(\lambda_{BK}+2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))t} + \frac{\lambda_{AC}(2\lambda_{AC}+\lambda_{AK})-\lambda_{AK}(2\lambda_{BC}+\lambda_{BK})}{(\lambda_{AC}+2\lambda_{BC}+\lambda_{BK})(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})} e^{-(2\lambda_{AC}+2\lambda_{BC}+\lambda_{BK}+\lambda_{AK})t} - \frac{(\lambda_{AC}+\lambda_{AK})\cdot 2\lambda_{AC}}{(\lambda_{AC}+2\lambda_{BC}+\lambda_{BK})(\lambda_{AC}+\lambda_{AK}-\lambda_{BK}-2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))} e^{-(\lambda_{AC}+\lambda_{AK})t}$$

$$P_{ZBC}(t) = \frac{2\lambda_{BC}}{\lambda_{BC}+2\lambda_{AC}+\lambda_{AK}} e^{-(\lambda_{BC}+\lambda_{BK})t} - \frac{2\lambda_{BC}}{\lambda_{BC}+2\lambda_{BC}+\lambda_{AK}} e^{-(2\lambda_{BC}+2\lambda_{AC}+\lambda_{AK}+\lambda_{BK})t}$$

$$P_{ZBD}(t) = \left[\frac{(\lambda_{BC}+\lambda_{BK})\cdot 2\lambda_{BC}}{(\lambda_{BC}+\lambda_{BK}-\lambda_{AK}-2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})} + \frac{\lambda_{BK}}{(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})}\right] e^{-(\lambda_{AK}+2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))t} + \frac{\lambda_{BC}(2\lambda_{BC}+\lambda_{BK})-\lambda_{BK}(2\lambda_{AC}+\lambda_{AK})}{(\lambda_{BC}+2\lambda_{AC}+\lambda_{AK})(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})} e^{-(2\lambda_{BC}+2\lambda_{AC}+\lambda_{AK}+\lambda_{BK})t} - \frac{(\lambda_{BC}+\lambda_{BK})\cdot 2\lambda_{BC}}{(\lambda_{BC}+2\lambda_{AC}+\lambda_{AK})(\lambda_{BC}+\lambda_{BK}-\lambda_{AK}-2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))} e^{-(\lambda_{BC}+\lambda_{BK})t}$$

Reliability (R) Functions

$$R_{asym}(t) = \left[\frac{2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}-\lambda_{BK}-2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))-(\lambda_{AC}+\lambda_{AK})\cdot 2\lambda_{AC}}{(\lambda_{AC}+2\lambda_{BC}+\lambda_{BK})(\lambda_{AC}+\lambda_{AK}-\lambda_{BK}-2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))}\right] e^{-(\lambda_{AC}+\lambda_{AK})t} + \left[\frac{2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}-\lambda_{AK}-2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))-(\lambda_{BC}+\lambda_{BK})\cdot 2\lambda_{BC}}{(\lambda_{BC}+2\lambda_{AC}+\lambda_{AK})(\lambda_{BC}+\lambda_{BK}-\lambda_{AK}-2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))}\right] e^{-(\lambda_{BC}+\lambda_{BK})t} + \left[\frac{(\lambda_{BC}+\lambda_{BK})\cdot 2\lambda_{BC}}{(\lambda_{BC}+\lambda_{BK}-\lambda_{AK}-2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})} + \frac{\lambda_{BK}}{(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})}\right] e^{-(\lambda_{AK}+2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))t} + \left[\frac{(\lambda_{AC}+\lambda_{AK})\cdot 2\lambda_{AC}}{(\lambda_{AC}+\lambda_{AK}-\lambda_{BK}-2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})} + \frac{\lambda_{AK}}{(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})}\right] e^{-(\lambda_{BK}+2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))t} + \left[1 + \frac{\lambda_{AC}(2\lambda_{AC}+\lambda_{AK})-\lambda_{AK}(2\lambda_{BC}+\lambda_{BK})}{(\lambda_{AC}+2\lambda_{BC}+\lambda_{BK})(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})} + \frac{\lambda_{BC}(2\lambda_{BC}+\lambda_{BK})-\lambda_{BK}(2\lambda_{AC}+\lambda_{AK})}{(\lambda_{BC}+2\lambda_{AC}+\lambda_{AK})(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})} - \frac{2\lambda_{AC}}{\lambda_{AC}+2\lambda_{BC}+\lambda_{BK}} - \frac{2\lambda_{BC}}{\lambda_{BC}+2\lambda_{BC}+\lambda_{AK}}\right] e^{-(2\lambda_{BC}+2\lambda_{AC}+\lambda_{AK}+\lambda_{BK})t}$$

$$R_{sym}(t) = \frac{4\lambda_C(-\lambda_K-2\lambda_C(\lambda_C+\lambda_K))}{(3\lambda_C+\lambda_K)(\lambda_C-2\lambda_C(\lambda_C+\lambda_K))} e^{-(\lambda_C+\lambda_K)t} + \frac{2\lambda_C(\lambda_C+\lambda_K)+\lambda_K(\lambda_C-2\lambda_C(\lambda_C+\lambda_K))}{(\lambda_C-2\lambda_C(\lambda_C+\lambda_K))(2\lambda_C(2-\lambda_C-\lambda_K)+\lambda_K)} 2e^{-(\lambda_K+2\lambda_C(\lambda_C+\lambda_K))t} + \left[1 + 2\frac{2\lambda_C(\lambda_C+\lambda_K)-\lambda_K(3\lambda_C+\lambda_K)}{(3\lambda_C+\lambda_K)(2\lambda_C(2-\lambda_C-\lambda_K)+\lambda_K)} - \frac{4\lambda_C}{3\lambda_C+\lambda_K}\right] e^{-(4\lambda_C+2\lambda_K)t}$$

Mean Time To Failure (MTTF) Functions:

$$MTTF_{asym} = \int_0^\infty R_{asym}(t)dt = \left[\frac{2\lambda_{AC}\left(-\lambda_{BK}-2\lambda_{BC}(\lambda_{BC}+\lambda_{BK})\right)}{(\lambda_{AC}+2\lambda_{BC}+\lambda_{BK})(\lambda_{AC}+\lambda_{AK}-\lambda_{BK}-2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))}\right]\frac{1}{\lambda_{AC}+\lambda_{AK}} + \left[\frac{2\lambda_{BC}\left(-\lambda_{AK}-2\lambda_{AC}(\lambda_{AC}+\lambda_{AK})\right)}{(\lambda_{BC}+2\lambda_{AC}+\lambda_{AK})(\lambda_{BC}+\lambda_{BK}-\lambda_{AK}-2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))}\right]\frac{1}{\lambda_{BC}+\lambda_{BK}} + \left[\frac{(\lambda_{BC}+\lambda_{BK})\cdot 2\lambda_{BC}}{(\lambda_{BC}+\lambda_{BK}-\lambda_{AK}-2\lambda_{AC}(\lambda_{AC}+\lambda_{AK}))(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})} + \frac{\lambda_{BK}}{(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})}\right]\frac{1}{\lambda_{AK}+2\lambda_{AC}(\lambda_{AC}+\lambda_{AK})} + \left[\frac{(\lambda_{AC}+\lambda_{AK})\cdot 2\lambda_{AC}}{(\lambda_{AC}+\lambda_{AK}-\lambda_{BK}-2\lambda_{BC}(\lambda_{BC}+\lambda_{BK}))(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})} + \frac{\lambda_{AK}}{(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})}\right]\frac{1}{\lambda_{BK}+2\lambda_{BC}(\lambda_{BC}+\lambda_{BK})} + \left[1 + \frac{\lambda_{AC}(2\lambda_{AC}+\lambda_{AK})-\lambda_{AK}(2\lambda_{BC}+\lambda_{BK})}{(\lambda_{AC}+2\lambda_{BC}+\lambda_{BK})(2\lambda_{AC}+2\lambda_{BC}(1-\lambda_{BC}-\lambda_{BK})+\lambda_{AK})} + \frac{\lambda_{BC}(2\lambda_{BC}+\lambda_{BK})-\lambda_{BK}(2\lambda_{AC}+\lambda_{AK})}{(\lambda_{BC}+2\lambda_{AC}+\lambda_{AK})(2\lambda_{BC}+2\lambda_{AC}(1-\lambda_{AC}-\lambda_{AK})+\lambda_{BK})} - \frac{2\lambda_{AC}}{\lambda_{AC}+2\lambda_{BC}+\lambda_{BK}} - \frac{2\lambda_{BC}}{\lambda_{BC}+2\lambda_{BC}+\lambda_{AK}}\right]\frac{1}{2\lambda_{BC}+2\lambda_{AC}+\lambda_{AK}+\lambda_{BK}}$$

$$MTTF_{sym} = \int_0^\infty R_{sym}(t)dt = \frac{4\lambda_C\left(-\lambda_K-2\lambda_C(\lambda_C+\lambda_K)\right)}{(3\lambda_C+\lambda_K)(\lambda_C-2\lambda_C(\lambda_C+\lambda_K))}\frac{1}{\lambda_C+\lambda_K} + \frac{2\lambda_C(\lambda_C+\lambda_K)+\lambda_K(\lambda_C-2\lambda_C(\lambda_C+\lambda_K))}{(\lambda_C-2\lambda_C(\lambda_C+\lambda_K))(2\lambda_C(2-\lambda_C-\lambda_K)+\lambda_K)}\frac{2}{\lambda_K+2\lambda_C(\lambda_C+\lambda_K)} + \left[1 + 2\frac{2\lambda_C(\lambda_C+\lambda_K)-\lambda_K(3\lambda_C+\lambda_K)}{(3\lambda_C+\lambda_K)(2\lambda_C(2-\lambda_C-\lambda_K)+\lambda_K)} - \frac{4\lambda_C}{3\lambda_C+\lambda_K}\right]\frac{1}{4\lambda_C+2\lambda_K}$$

SAE INTERNATIONAL

ARTICLE INFO

Article ID: 2017-01-0086

doi:10.4271/2017-01-0086

Potentials for Platooning in U.S. Highway Freight Transport

Matteo Muratori, National Renewable Energy Laboratory

Jacob Holden, Michael Lammert, Adam Duran, Stanley Young, and Jeffrey Gonder,
National Renewable Energy Laboratory

Abstract

Smart technologies enabling connection among vehicles and between vehicles and infrastructure as well as vehicle automation to assist human operators are receiving significant attention as a means for improving road transportation systems by reducing fuel consumption – and related emissions – while also providing additional benefits through improving overall traffic safety and efficiency. For truck applications, which are currently responsible for nearly three-quarters of the total U.S. freight energy use and greenhouse gas (GHG) emissions, platooning has been identified as an early feature for connected and automated vehicles (CAVs) that could provide significant fuel savings and improved traffic safety and efficiency without radical design or technology changes compared to existing vehicles. A statistical analysis was performed based on a large collection of real-world U.S. truck usage data to estimate the fraction of total miles that are technically suitable for platooning. In particular, our analysis focuses on estimating "platoonable" mileage based on overall highway vehicle use and prolonged high-velocity traveling, and established that about 65% of the total miles driven by combination trucks from this data sample could be driven in platoon formation, leading to a 4% reduction in total truck fuel consumption. This technical potential for "platoonable" miles in the United States provides an upper bound for scenario analysis considering fleet willingness and convenience to platoon as an estimate of overall benefits of early adoption of connected and automated vehicle technologies. A benefit analysis is proposed to assess the overall potential for energy savings and emissions mitigation by widespread implementation of highway platooning for trucks.

History

Received: 02 May 2017
Published: 28 Mar 2017

Citation

Muratori, M., Holden, J., Lammert, M., Duran, A. et al., "Potentials for Platooning in U.S. Highway Freight Transport," *SAE Int. J. Commer. Veh.* 10(1):2017, doi:10.4271/2017-01-0086.

ISSN: 1946-391X
e-ISSN: 1946-3928

Introduction

Connected and automated vehicles (CAVs) are receiving significant attention as a technology solution to realize safer, more cost-effective, and efficient operation of several transportation systems [1]. CAVs can also potentially help curb energy consumption and greenhouse gas (GHG) emissions from the transportation sector. In this paper we focus on the role of platooning for combination trucks[1] in the United States, one of the most promising CAV technologies that could experience widespread adoption in the next 5 to 10 years. Platooning is a demonstrated method of groups of vehicles travelling close together actively coordinated in formation at high speed that has the potential to reduce energy consumption resulting from aerodynamic drag [2] [3]. Trucks are ideal applications for platooning due to their technical characteristics and mode of operation (several vehicles driving for long distances along the same route, often concentrated in few corridors).

Combination trucks, currently powered by petroleum-derived fuels, account for the majority of the energy use in the U.S. freight sector (64.9% of freight, and 4.8% of total U.S. energy use in 2013 [4]) and an even larger share of GHG emissions (77.1% of freight, and 7.5% of total U.S. GHG emissions in 2013 [5]). Looking at the future, the importance of trucking on the U.S. energy use and GHG emissions is likely to increase, due mainly to three factors: *a)* freight transport has been growing more rapidly than passenger transport, and the trend is likely to continue in the future [6] [7]; *b)* a continued increase in the share of trucking in total freight activity [8] [9] [10]; *c)* transportation, and freight in particular, is more expensive to decarbonize compared to other sectors, and will experience lower energy and GHG emissions reduction in response to economy-wide climate change mitigation measures [11].

Several studies, reviewed in the Methods section, have been focusing on assessing the potential savings achievable by platooning operations for a group of two or more trucks, as well as extrapolating these savings on a national scale, based on overall miles traveled by trucks. However, a key element has been neglected in the existing literature: what is the "platoonable" fraction of traveled miles during real-world operations? Namely, in a fleet of trucks, what fraction of miles driven is amenable for platooning operation? Clearly not every mile driven can be driven in a platoon formation, and platooning operations at low speeds do not lead to significant fuel saving. However, for large trucks operating extensively on highways over long distances the fraction of platoonable miles at high speed can be significant (in estimating the potential savings related to trucks platooning, MacKenzie *et al.* [12] assume that every mile traveled by trucks is platoonable, leading to significantly different results compared to this study).

We provide an estimate of the platoonable fraction of miles driven by combination trucks in the United States based on a large set of driving data collected by the National Renewable Energy Laboratory (NREL) and others. This data set includes over 3 million miles of driving data across a variety of fleet operators, truck manufacturers, times of operation, and regions. In particular, we assume that a truck could potentially operate in a platoon if it continuously travels at a speed larger than a certain threshold for a significant period of time. A sensitivity analysis shows that the velocity and the time threshold significantly impact the resulting fraction of platoonable miles. These thresholds have been chosen to be 50 mph (80.5 km/h) and 15 minutes for representative operations in the United States.

This estimate represents a technical potential, or upper bound, which does not account for truck and fleet operators' willingness to platoon. This willingness, which will be assessed in future works, reduces the technical potential identified in this paper due to three main factors. First, the economic savings related to platooning operations (value of fuel saved) must outweigh the increased costs, namely the additional drivers' time cost during platoon formation (as most likely some drivers will have to wait for other trucks traveling towards the same destination) and the value of delayed delivery.[2] Second, truck and fleet operators must be willing to cooperate. While this might be easier for large fleets including hundreds of vehicles, smaller operators might not have the required connectivity and willingness to collaborate with direct competitors. Third, uniform and standard technologies are required across vehicle manufacturers and operators to allow for widespread implementation of platooning across fleets.

The remainder of this paper is structured as follows: The Methods section describes the data set and the methodology used to estimate the real-world fraction of platoonable miles for combination trucks in the United States and a review of literature of existing studies on energy savings achievable by operating trucks in platoons. The Results section reports the quantitative results on this analysis, including a sensitivity analysis aimed at understanding the impact of time and velocity thresholds in estimating the fraction of platoonable miles and additional insights for targeted applications (i.e., platoonable miles for vehicles performing only long-distance missions on highways). These insights are used in the National Impact section to calculate an upper-bound estimate of the potential energy savings and GHG emissions reductions related to widespread adoption of platooning for combination trucks. Concluding remarks and proposed future work are reported in the last section.

[1] Combination trucks include Class 7 and Class 8 trucks, as defined by the U.S. Federal Highway Administration. Class 8 trucks, which are the majority of combination trucks, are vehicles with a gross weight rating exceeding 33,001 lbs (14,969 kg). Class 8 includes tractor-trailer tractors as well as single-unit dump trucks. The typical 5-axle tractor-trailer combination, also called a "semi" or "18-wheeler," is a Class 8 vehicle.

[2] Given the U.S. network and the large volume of freight moved on the road, we assumed that trucks will not modify their original route to travel in a platoon. Namely we assume that within a reasonable time a truck will be able to join others and form a platoon heading towards its final destination. This assumption might not be realistic for very early adoption in the United States or other countries.

Methods

In this paper we use a large data set of about 200 real-world Class 8 tractors' driving data, which includes over 3 million miles of data, to estimate the fraction of platoonable miles in a variety of real-world operations in the United States. The data considered have been collected directly by NREL and other partners who have contributed data to NREL's Fleet DNA database using on-board data logging devices or telematics systems [13]. Vocations represented in the data set include line haul truck load, less than truck load, regional parcel movement, port drayage, refrigerated operations, tanker operations, transfer truck operations, and regional food delivery. The data set includes information on vehicle speed (1-second resolution), global positioning system position, road segments (classified as highway, freeway, or collectors and local), and various levels of engine/vehicle parameters such as fuel rate and engine temperatures.

Table 1 summarizes the data set considered, while Figure 1 and Figure 2 show the distribution of all the trips included in the data set based on trip length and duration, respectively. Trips shorter than 0.5 mile and 6 minutes have been excluded to avoid including logging errors and short vehicle movements that do not constitute trips.

From Figure 1 and Figure 2, it appears that very short trips (*i.e.*, less than 25 miles and less than one hour) constitute a significant share of trips included in the data set considered. Nevertheless, these trips account for a small fraction of total miles driven, as shown in Figure 3 and Figure 4, which report the share of driven miles for several classes of trip length and duration. The majority of miles driven by the trucks included in the data set were driven in trips between 50 and 250 miles long that lasted between 2 and 6 hours. Some very long trips (*i.e.*, over 500 miles and 8 hours) are also present in the data set (about 10% of total miles driven), resulting from vehicles being driven by multiple drivers who took turns driving without turning off the engine for prolonged periods of time.

TABLE 1 Summary statistics of the driving data set considered in this study.

Data Set	
Vehicles	194
Days	9,154
Trips	54,583
Hours	60,450
Miles	3,170,079

FIGURE 1 Distribution of trips included in the data set based on trip length.

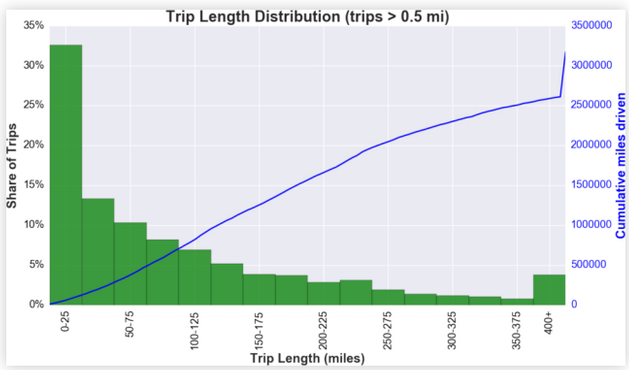


FIGURE 2 Distribution of trips included in the data set based on trip duration.

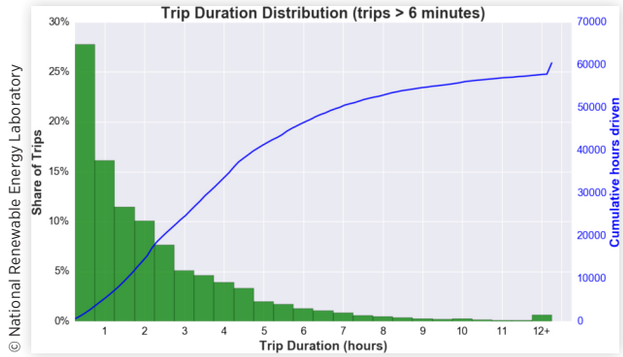


FIGURE 3 Share of miles driven as a function of trip length.

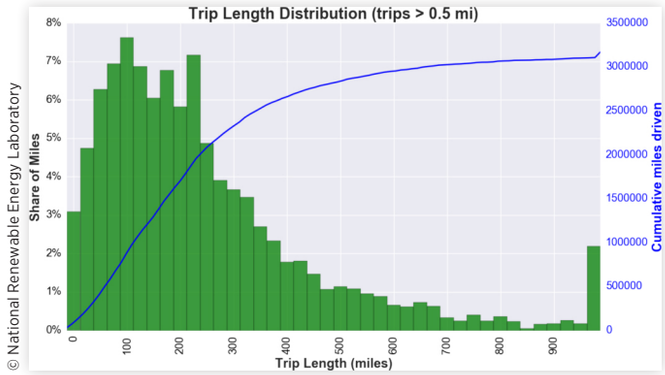
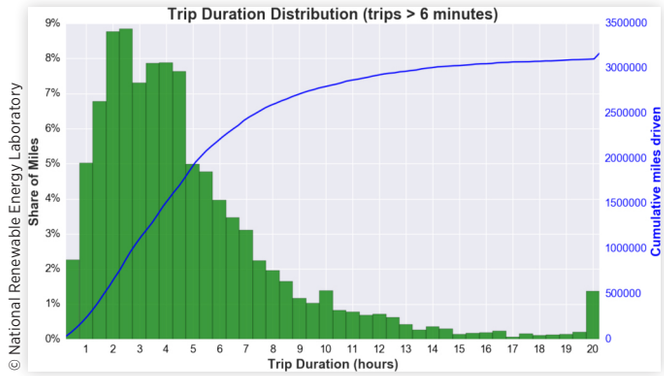


FIGURE 4 Share of miles driven as a function of trip duration.



The extensive and comprehensive data analyzed allow calculating a reasonable estimate of the total fraction of platoonable miles across different U.S. regions and truck applications.

State-of-the-Art for Trucks Energy Savings in Platooning Operations

Several analyses, based both on simulation studies and experiments, have estimated the energy savings during platoon operations of two or more trucks. While platooning opportunities for a variety of applications have been explored for over a decade (*e.g.*, [14], [15]), no consensus has been reached in the open literature on the fuel savings related to platooning operations of more than two vehicles.

Lu and Shladover [16] tested a platoon of three Class 8 tractor-trailers under different driving conditions and following distances, reporting a fuel saving of 4%–5% for the leading truck and 10%–14% for the following trucks. Lammert *et al.* [17] performed ten modified SAE Type II J1321 fuel consumption track tests to evaluate fuel consumption results of two Class 8 tractor-trailer combinations platooned together compared to their standalone fuel consumption, reporting combined "Team" fuel savings ranging from 3.7% to 6.4% (between 2.7% and 5.3% fuel savings for the lead tractor and between 2.8% and 9.7% for the trailing vehicle).

A recent study by the North American Council for Freight Efficiency (NACFE) reviewed the results of ten analyses performed over the last decade directly comparing driving speeds from 43 to 70 mph, conventional and cab-over-engine configurations, and a range of vehicle curb weights, which showed a significant spread among the different test results. While lead vehicle savings had significant correlation across the variables with following distance being most important factor in the 0-9% range of observed fuel savings; for the trailing truck(s) fuel consumption reductions was reported to vary between 3% and 23% and showed much higher dependence on speed, mass and cab configuration variables [18]. Combining results with all the above variables, NACFE estimates the savings to be approximately 4% for the lead truck and 10% for the following truck when trucks are operated on a closed track in a consistent two-truck platooning arrangement. This equates to a 7% fuel efficiency improvement on average between the two trucks versus a truck operating in isolation. Moreover, NACFE identified road congestion and actual platoonable miles as the two most relevant factors influencing real-world fuel economy of trucks operating in platoon formation and offered an estimate of impact from these factors [18]. Significant correlation was observed between multiple track studies, wind tunnel testing, and computational fluid dynamics analyses when compared at the same speed, mass, and aerodynamic class/type over a range of following distances [19].

In this paper we consider a 6.4% potential fuel saving for platooning operations, based on the study by Lammert *et al.* [17], with the best combined result being for 55 mph and a 30-ft following distance. In future applications, platooning fuel savings can be enhanced by addressing barriers to closer platoon formation – such as reduced engine cooling – and by including more vehicles in each platoon [19]. Alam *et al.* suggested that a large-scale cooperative method to enhance safety and efficiency of truck platoons by increasing the level of cooperation between vehicles be used to maximize platooning benefits [20]. Additional benefits of truck platooning, such as road capacity optimization and accidents reduction, as well as additional truck safety and operational considerations have also been explored in previous studies ([21], [22], [23]).

Results

Based on the data set described in the Methods section, we compute the fraction of miles that are continuously driven above a speed threshold V for at least T minutes, where T is a time threshold. This is intended to capture the fraction of driven miles that are suitable for platooning operations. In principle, V should equal the prescribed speed limit, and T should be a time long enough to offset the tradeoffs due to platoon formation.

Figure 5 shows the share of miles driven in each road segment based on the entire data set summarized in Table 1, as well as the fraction of miles continuously driven above a speed limit for time T for a set of different thresholds. The results show that for a time threshold of $T = 15$ minutes and a speed threshold $V = 50$ mph, 65.6% of vehicle miles are platoonable. The figure also shows how this number changes as different time and speed thresholds are selected.

FIGURE 5 Share of total miles (y-axis) continuously driven above a certain speed threshold (x-axis) for T minutes (different lines) and share of load segments considering the entire data set.

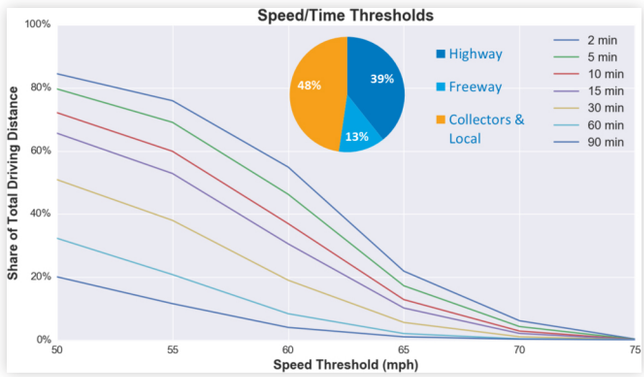
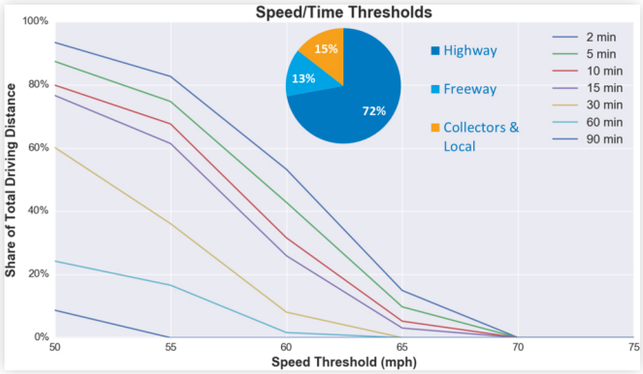


FIGURE 6 Share of total miles (y-axis) continuously driven above a certain speed threshold (x-axis) for T minutes (different lines) and share of load segments for targeted platoonable applications.



Targeted Applications

The same methodology used to estimate the share of platoonable miles for the entire data set is applied to a subset of the data, including about 4,500 miles of mostly-highway long-distance driving to evaluate the fraction of platoonable miles for specific applications that might represent early adopters of this technology.

The results shown in Figure 6 indicate that for a time threshold of T = 15 minutes and a speed threshold V = 50 mph, 76.6% of vehicle miles are platoonable. The vocation represented in Figure 6 is a split-duty combination truck that runs local pickup and delivery trips during the day and regional line-haul operation at night (representing the majority of the miles driven, and making this application ideal for platoon operations).

National Impact

In 2014 169.8 billion miles were driven by combination trucks in the United States [24], consuming a total of 29.1 billion gallons of fuel and emitting approximately 6.9 billion metric tons of carbon dioxide equivalent [25]. Based on the analysis provided in this paper, approximately 65.6% of those miles could potentially be driven in platoon formation. Assuming an energy (and emissions) savings of approximately 6.4% for each team of platooned vehicles (based on efficiency improvements previously published in a platooning benefits study [17]), widespread adoption of platooning operations can potentially reduce trucks energy use by approximately 4.2%.

With these bounding assumptions, the widespread adoption of platooning operations for combination trucks in the United States could lead to a total savings of 1.5 billion gallons of petroleum-derived fuels (equal to 1.1% of the current US import of oil: 2.7 billion barrels in 2015 [26]) and 15.3 million metric tons of CO_2 (a 0.22% emissions reduction).

Conclusions and Future Work

In this paper we estimate a technical potential, or upper bound, for the fraction of platoonable miles for combination trucks in the United States based on an extensive data set of real-world driving data. This study complements existing literature on this subject that neglected to consider that not all miles driven by trucks are suitable for platooning applications.

Our results show that approximately 65.6% of the total miles driven by combination trucks (Class 7 and 8) could be driven in platoon formation, leading to significant energy and emissions savings. For targeted applications, which are likely to be early adopters of connected and automated technologies, this fraction increases to approximately 76.6%. A more comprehensive "Big Data" analysis considering a larger data set that covers multiple years and a wider array of applications is planned to further refine this estimate.

Based on an assumed energy saving of 6.4%, resulting from a review of recent literature, this translates into 2.7% potential energy savings in the U.S. freight sector and a reduction in U.S. GHG emissions on the order of 15.3 million metric tons of carbon dioxide per year.

As discussed, this technical potential study presents an upper bound because in the real world, truck and fleet operators may not be willing to participate to platoon operations under all the conditions considered here (*e.g.*, an operator might not be willing to wait to form a platoon). Therefore, an expert elicitation study involving truck owners and fleet operators will be performed to assess the overall willingness to participate in platooning and the main barriers for the widespread adoption of this technology.

Contact Information

Corresponding Author:
Matteo Muratori
National Renewable Energy Laboratory
15013 Denver West Parkway
Golden, Colorado 80401, USA
Phone: 303-275-2927
matteo.muratori@nrel.gov

Acknowledgments

This work was supported by the U.S. Department of Energy under Contract No. DE-AC36-08GO28308 with the National Renewable Energy Laboratory. Funding was provided by the Vehicle Systems Program within the DOE Energy Efficiency and Renewable Energy's Vehicle Technologies Office. The authors particularly appreciate the support and guidance provided by DOE program managers David Anderson, Lee Slezak and Rachael Nealer. For many years the Vehicle Systems Program has additionally supported the Fleet DNA database of commercial vehicle in-use operating data, which was also

instrumental in the completion of this study. The views and opinions expressed in this paper are those of the authors alone.

NREL is a national laboratory of the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, operated by the Alliance for Sustainable Energy, LLC. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

1. Wadud, Z., MacKenzie, D., and Leiby, P., "Help or hindrance? The travel, energy and carbon impacts of highly automated vehicles," *Transportation Research Part a: Policy and Practice* 86:1–18, 2016, doi:10.1016/j.tra.2015.12.001.
2. Bergenhem, C., Shladover, S., Coelingh, E., Englund, C., and Tsugawa, S., "Overview of platooning systems," 2012.
3. Brown, A., Gonder, J., and Repac, B., "An Analysis of Possible Energy Impacts of Automated Vehicle," *Road Vehicle Automation*, Springer International Publishing, Cham, ISBN 978-3-319-05989-1: 137–153, 2014, doi:10.1007/978-3-319-05990-7_13.
4. Davis, S., Diegel, S., and Boundy, R., "*Transportation Energy Data Book*," 2015.
5. U.S. Department of Transportation, "Freight Facts and Figures" 2015.
6. *Energy Information Administration (EIA)*, "International energy outlook,"," 2014.
7. ExxonMobil, "The Outlook for Energy: A view to 2040," Irving, 2013.
8. Kamakaté, F. and Schipper, L., "Trends in truck freight energy use and carbon emissions in selected OECD countries from 1973 to 2005," *Energy Policy* 37(10):3743–3751, 2009, doi:10.1016/j.enpol.2009.07.029.
9. Greening, L.A., Ting, M., and Davis, W.B., "Decomposition of aggregate carbon intensity for freight: trends from 10 OECD countries for the period 1971–1993," *Energy Economics* 21(4):331–361, 1999.
10. Eom, J., Schipper, L., and Thompson, L., "We keep on truckin': Trends in freight energy use and carbon emissions in IEA countries," *Energy Policy* 45:327–341, 2012, doi:10.1016/j.enpol.2012.02.040.
11. Muratori, M., Smith, S.J., Kyle, P., Link, R., Mignone, B., and Kheshgi, H., "The Role of the Freight Sector in Future Climate Change Mitigation Scenarios." *Environmental Science & Technology*, Forthcoming.
12. MacKenzie, D., Wadud, Z., and Leiby, P., "A first order estimate of energy impacts of automated vehicles in the United States," *Poster Presentation at the …*, 2014.
13. Walkowicz, K., Kelly, K., Duran, A., and Burton, E., "Walkowicz: Fleet DNA project data," 2014.
14. Gehring, O. and Fritz, H., "Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication," IEEE, ISBN 0-7803-4269-0: 117–122, 1997, doi:10.1109/ITSC.1997.660461.
15. Robinson, T., Chan, E., and Coelingh, E., "Operating platoons on public motorways: An introduction to the sartre platooning programme," *17th world congress on intelligent transport systems*, 2010.
16. Lu, X.-Y. and Shladover, S.E., "Automated Truck Platoon Control and Field Test," *Road Vehicle Automation*, Springer International Publishing, Cham, ISBN 978-3-319-05989-1: 247–261, 2014, doi:10.1007/978-3-319-05990-7_21.
17. Lammert, M., Duran, A., Diez, J., Burton, K. et al., "Effect of Platooning on Fuel Consumption of Class 8 Vehicles Over a Range of Speeds, Following Distances, and Mass," *SAE Int. J. Commer. Veh.* 7(2):626–639, 2014, doi:10.4271/2014-01-2438.
18. North American Council for Freight Efficiency (NACFE), "Confidence Report on Two-Truck Platooning," 2016.
19. Lammert, M.P., Gonder, J., Kelly, K., Salari, K., and Ortega, J., "Class 8 Tractor Trailer Platooning: Effects, Impacts, and Improvements," 2016.
20. Alam, A., Besselink, B., Turri, V., Martensson, J., and Johansson, K.H., "Heavy-Duty Vehicle Platooning for Sustainable Freight Transportation: A Cooperative Method to Enhance Safety and Efficiency," *IEEE Control Syst.* 35(6):34–56, 2015, doi:10.1109/MCS.2015.2471046.
21. ATA Technology and Maintenance Council Future Truck Program, "Automated Driving and Platooning Issues and Opportunities," 2015.
22. Janssen, R., Zwijnenberg, H., Blankers, I., and de Kruijff, J., "Truck platooning: driving the future of transportation," 2015.
23. Identifying Autonomous Vehicle Technology Impacts on the Trucking Industry, "Identifying Autonomous Vehicle Technology Impacts on the Trucking Industry," 2016.
24. U.S. Federal Highway Administration, "Highway Statistics *-Annual Vehicle Distance Traveled in Miles and Related Data*."
25. U.S. Environmental Protection Agency (EPA), "Greenhouse Gas Inventory Explorer," https://www3.epa.gov/climatechange/ghgemissions/inventoryexplorer/#transportation/allgas/source/all, Oct. 2016.
26. U.S. Energy Information Administration (EIA), "US Imports of Crude Oil."

SAE INTERNATIONAL

ARTICLE INFO

Article ID: 2017-01-1662

doi:10.4271/2017-01-1662

A Balanced Approach for Securing the OBD-II Port

Tom R. Markham, Honeywell
Alex Chernoguzov, *Honeywell*

Abstract

The On-Board Diagnostics II (OBD-II) port began as a means of extracting diagnostic information and supporting the right to repair. Self-driving vehicles and cellular dongles plugged into the OBD-II port were not anticipated. Researchers have shown that the cellular modem on an OBD-II dongle may be hacked, allowing the attacker to tamper with the vehicle brakes. ADAS, self-driving features and other vehicle functions may be vulnerable as well. The industry must balance the interests of multiple stakeholders including Original Equipment Manufacturers (OEMs) who are required to provide OBD function, repair shops which have a legitimate need to access the OBD functions, dongle providers and drivers. OEMs need the ability to protect drivers and manage liability by limiting how a device or software application may modify the operation of a vehicle. This paper outlines a technical approach based upon cryptographic authentication and granular access control policy which addresses the needs of stakeholders. This allows the OEM to protect the security of the vehicle by carefully controlling the functions a particular device plugged into the OBD-II port is able to perform. This allows device makers (diagnostic tools, insurance dongles, etc.) to have their products certified to work with the OEM's vehicles. The result is the OEMs can protect driver safety and maintain the right to repair.

History

Received: 01 Aug 2017
Published: 28 Mar 2017

Citation

Markham, T. and Chernoguzov, A., "A Balanced Approach for Securing the OBD-II Port," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 10(2):2017, doi: 10.4271/2017-01-1662.

ISSN: 1946-4614
e-ISSN: 1946-4622

1. Introduction

Motivation

There is a real and growing threat to automotive cyber security and, in turn, safety. In particular, the OBD-II port has become a greater risk [19]. The number of vehicles with brakes, steering and acceleration influenced by messages on the CAN bus is increasing due to self-parking, ADAS and the move toward autonomous driving. The OBD-II port has transitioned from a physical port only accessible from inside the cabin of the vehicle to one that is wirelessly accessible via cellular modems, Bluetooth and Wi-Fi dongles [9, 1]. There is evidence that interest in the cyber security of vehicles is migrating from white hat researchers to hackers with hostile intent. Taken together, these facts call for taking a fresh look at the OBD-II port and identifying an up-to-date approach for protecting vehicle security and safety.

Related Work

There has been a growing list of researchers who have demonstrated that gaining access to the CAN bus via the OBD-II port can allow a hacker to influence control of a vehicle [8]. Those researchers who have used wireless access via an OBD-II device to compromise a vehicle [12, 3] are of particular interest.

Prior Society of Automotive Engineers (SAE) standards work in the form of SAE J2186 - E/E Data Link Security [14] addressed access to ECUs via the Data Link Connector (DLC). (Throughout this document "OBD" is used as a shorthand for OBD-II and the DLC.) In some sense, J2186 may have been ahead of its time because when it was revised in 2005 the threat environment and motivation to secure vehicles was very different from today. The OBD-related threat has recently gained the attention of Congress which in a Sep 12, 2016 letter to NHTSA [13] wrote "… stakeholders cited the ports and the direct connection they provide to the vehicle internal networks as one of the fundamental sources of cyber security risk in the modern vehicle ecosystem."

Within the broader cyber security community there has been a great deal of work done on cryptographic protocols [11], public key cryptography [16], public key infrastructure and role based access control [17, 10, and 2]. These techniques and lessons learned can accelerate efforts to protect vehicles from attacks through the OBD port.

The remainder of this paper is organized as follows:

2. Brief history of vehicle network technologies, providing background on how we got here.
3. Requirements, informally summarizes the cyber security challenges this paper is addressing. There are many other cyber security items to address beyond the limited scope of this paper.
4. Technical approach, provides one technical approach for meeting the needs of various stakeholders in a manner which is scalable, cost effective and secure.
5. Operation, outlines how the technical approach would impact device certification and use.
6. Design considerations, discusses certification, implementation cost and technology options.
7. Summary/Conclusions, touches upon standardization of the OBD security concept and application to the telematics control unit (TCU).

2. Brief History of Vehicle Network Technologies

CAN Bus

What was the original environment for the CAN bus? Historically the CAN bus was an island. An air-gapped network contained within the sheet metal envelope of the vehicle. Everything on the network was integrated by the OEM and trusted to preserve the cyber security of the vehicle. Anyone attempting to send CAN bus messages on the bus was assumed to be inside the vehicle and therefore trusted.

This level of trust and openness on the CAN bus was reasonable back in the day when brakes, steering and transmission were controlled by physical linkages, not a network. Today the lack of source authentication on the CAN bus is a security and safety issue.

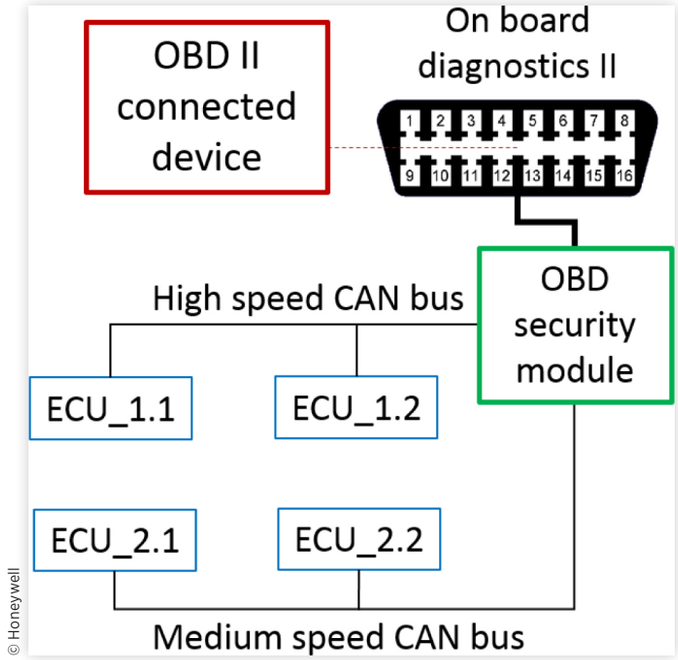
OBD

What was the original purpose of the data link connector and OBD? It was designed to allow repair shops to connect with engine controls to perform diagnostics and gather emissions related information. Few envisioned the OBD port as a means of remotely injecting commands which could affect brakes, steering or acceleration. When J1979 was first issued in 1991, who imagined an insecure cellular modem connected to the vehicle OBD port controlling vehicle operation?

This paper and SAE J2186 both address aspects of OBD security. However, SAE J2186 was focused on protecting the integrity of a single Electronic Control Unit (ECU). i.e. J2186 could be used to prevent a device from changing firmware or calibration parameters within a specific ECU unless the device at the OBD port could authenticate to the ECU as a device authorized to make changes to that ECU. While this proposal can also prevent an OBD connected device from making changes to an ECU it can also prevent the OBD connected device from injecting CAN messages which trigger ECU actions. e.g., a message injected via the OBD triggers an ECU to unlock the doors.

The CAN bus does not perform source authentication of messages placed on the CAN bus. Thus, an attacker who is

FIGURE 1 Placement of the OBD security module within the vehicle.



able to inject messages into the OBD port can send messages which appear to come from one of the ECUs on the bus. Referring to Figure 1, if the OBD security module was not present, the OBD connected device would be able to place messages on the CAN bus which would normally be sent by ECU_1.x or ECU_2.x. Thus, anything these ECUs would normally do could be done by an attacker via the OBD. A few of the many things the attacker could do via the OBD port include:

- Flash the headlights
- Unlock the doors
- Turn the air conditioning on/off
- Activate anti-lock brake functions
- Turn the front wheels (on vehicle with automatic parking capability)
- Change the speed shown on the speedometer

Some of these actions may be categorized as nuisance aimed at distracting the driver. Other actions directly affect vehicle operation taking control away from the driver or interfering with driver actions. Either can result in a safety incident.

OBD Devices

The OBD-II port is a source of data that can be used and manipulated for insurance tracking, monitoring vehicle performance and other "diagnostic" functions. Low cost (less than $20) wireless OBD devices are commercially available [1]. These can provide access via cellular, Wi-Fi or Bluetooth. The security of these devices is questionable. The web description for one model makes the claim

Easy to Use: Plug the device in your car's OBD2 port, turn on your car, enable Bluetooth on your Android phone or tablet, search for "OBDII" and pair with it (pin 1234), run the download App with simple settings and wait until it connects your car's ECU successfully

The hackers now know the pairing code for all devices of this model. What other security flaws could be lurking in these devices?

Hacking Culture

The world of hacking has also changed over the last 20 years. We have witnessed a progression from hacking a PC via a virus on floppy disc to hacking a website to get bragging rights and more recently hacking of industrial control systems. The Stuxnet attack on an industrial control system [7] was a wake-up call: hacking could remotely damage or destroy physical systems. This, of course, has been followed by multiple demonstration hacks of vehicles as well as research addressing vehicle security [18]. What is more concerning is the change in who is doing the hacking and their motivation. Years ago, bored college students hacked to show their skills. Today organized crime hacks for money. Ransomware attacks on home PCs and even businesses are common. When will we see ransomware against vehicles?

3. Requirements

The OBD environment has been examined to identify security requirements necessary to protect vehicle safety. It is obvious that a one-size-fits-all access control policy is not appropriate. The requirements to secure the OBD port and associated components are outlined below:

- Support diagnostics. The OEM is required by law to provide an OBD-II port to support diagnostics. Independent repair shops and vehicle owners have a right to repair which includes access to many vehicle functions via the OBD-II. Some "secured Functions" are unlocked per SAE J2186 but many functions are not locked.
- Provide third parties (e.g. insurance companies) read access to vehicle data such as Vehicle Identification Number (VIN), speed and braking via OBD.
- Enable OBD device providers the ability to offer vehicle-related services via the OBD port. The industry must balance access with vehicle security and safety concerns.
- Provide the vehicle owner the ability to use mobile apps (Torque, Dash, OBD Auto Doctor, etc.) to retrieve information from their vehicle.
- Deliver vehicles which are safe. The safety of modern automobiles requires that they have cyber security to protect them from malicious commands/ data injected via the OBD port. This implies these derived requirements.
  - Provide a means to test and certify the CAN messages injected by an OBD connected device (e.g., Insurance dongle, diagnostic tool).
  - Provide a means to authenticate a device plugged into an OBD port. i.e., determine the manufacturer of the device and the model number.
  - Determine and enforce the OEM approved access rights for the authenticated device relative to the OBD port. e.g., the device can read all data but only write to the body control modules.

4. Technical Approach

This section provides an introduction to the technologies which can be used to solve the OBD security issue. It introduces key technologies and outlines the system.

Defensive Technologies

Fortunately, there are defensive technologies which can be employed to enhance security. Public key cryptography and access control mechanisms could be combined to provide an authenticated access control function for the OBD port.

Public Key Public key (aka asymmetric key) cryptography [16] technology in the form of RSA and elliptic curve makes it possible to provide digital signatures and create public key infrastructure tailored to the needs of protecting vehicles. This is a mature technology already in use in other industries.

Public key cryptography uses two keys which are mathematically-related. The private key must be protected from unauthorized access. It is typically stored in some form of protected hardware and only accessed by the cryptographic engine within the hardware. Thus, the private key is used by the cryptographic engine to sign and encrypt/decrypt data but the key cannot be extracted from the cryptographic engine. The corresponding public key may be freely distributed. It does not require any confidentiality protection. The approach described in this paper makes use of two cryptographic functions.

A *digital signature* is an operation in which the creator digitally signs a piece of data by using the private key. The signature is a string of bits which practically speaking, could only be created by the entity holding the private key. Anyone holding the corresponding public key is then able to use that public key to perform a cryptographic operation and confirm that the piece of data in question was indeed signed by the entity holding the private key. As an example, if an OEM wanted to sign a firmware file such that any vehicle could verify that the firmware came from the OEM, the OEM would:

1. Create the file containing the firmware.
2. Compute a hash function over the firmware
3. Use the OEM private key to encrypt the hash result, thus creating a digital signature.
4. Send the original firmware file and the encrypted hash result together to any device which is to use the firmware.

The receiving entity, which has obtained the OEM public key through a trusted channel (e.g. loaded into the vehicle at the time of manufacture) then performs the following:

1. Compute the hash function over the firmware
2. Use the OEM public key to decrypt the encrypted hash result distributed with the firmware file.
3. Compare the locally computed hash result to the result of the decrypted hash from step 2.

If the decrypted hash and locally computed hash match, then the vehicle knows the firmware came from the OEM. If the decrypted hash does not match the locally computed hash then the vehicle rejects the firmware file. In the context of the OBD security solution being described here the "firmware file" would be security policy data which the OBD security module will apply to messages from the device connected to the OBD port. More on this later.

The second function of the public key cryptography is to perform a *key exchange* between the OBD security module and the device connected to the OBD port. The purpose of the key exchange is to authenticate the OBD device to the OBD

security module and establish a secure session for exchanging data between them. A similar type of key exchange is typically performed when your web browser connects to your bank website. The public and private keys contained within the OBD device are used as follows.

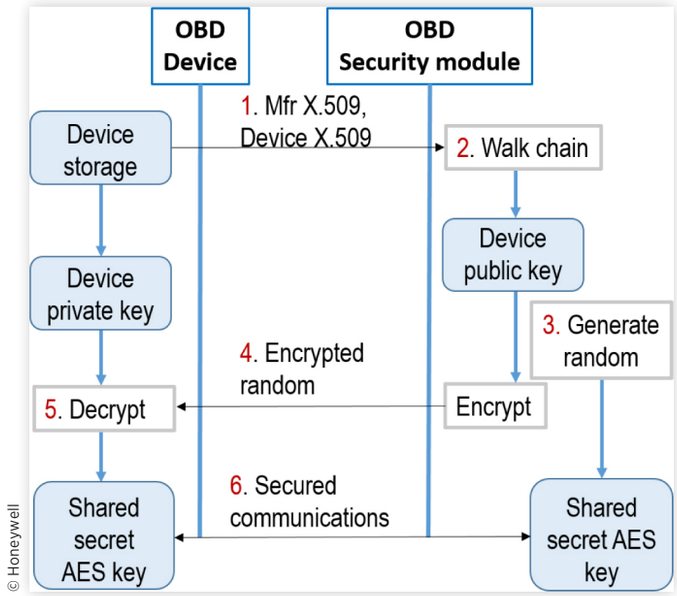
The OBD device (e.g. a diagnostic tool) passes its public key to the OBD security module. This exchange uses an X.509 certificate [5] structure. This structure provides fields which may be used to carry data identifying the OBD device type (e.g. Acme Tool Company, OBD scanner, model 100). The X.509 certificate, when combined with a successful key exchange;

- Proves to the vehicle that the vehicle is connected to a device certified by the OEM.
- Authenticates the identity of the OBD device
- Binds the OBD device to the access rights assigned by the OEM.

The key exchange (aka key agreement) is illustrated in Figure 2 and is performed as follows.

1. The OBD device sends the OBD security module within the vehicle the OBD device X.509 certificate and the X.509 certificate for the device manufacturer. This device manufacturer X.509 certificate has been signed by the OEM using the OEM private key. (Certification is addressed below in the subsection - Certification process.)
2. The OBD security module verifies the digital signature on the X.509 certificate confirming it was issued (perhaps indirectly) by the OEM. The OBD security module "walks the certificate chain" to establish trust. First, the OBD security module uses the OEM public key (embedded in the security module when the vehicle was manufactured) to verify the digital signature on the manufacturer X.509 certificate. The OBD security module then extracts the manufacturer public key from the freshly verified certificate. The OBD security module repeats the certificate verification process, this time using the device manufacture's public key to verify the certificate bound to the OBD device. The vehicle OBD security module then extracts the OBD device public key from the freshly verified OBD device X.509 certificate.
3. The OBD security module generates a random number which will be used as a key for a symmetric cryptographic algorithm (e.g. Advanced Encryption Standard (AES)). The OBD security module protects the confidentiality of this random number.
4. The OBD security module uses the public key of the OBD device to encrypt the random number. Public key cryptography ensures that only the OBD device holding the private key corresponding to the public key extracted from the X.509 certificate can decrypt the message and recover the original random number. The message containing the encrypted random number is then sent across the OBD port to the OBD device.
5. The OBD device uses its private key to decrypt the random number and recover the original plaintext version of the random number. Now the OBD security module and the OBD device both have the shared secret random number. No other entity has this number because only the OBD device with the right private key can perform the decryption.
6. The OBD security module and OBD device use the shared secret in a key derivation process to create the shared symmetric key which will be used to encrypt and authenticate messages sent across the OBD port.

FIGURE 2 Key exchange between the OBD device and the OBD security module.



Successful authentication of the OBD device allows the OBD security module to apply the access control specified by the OEM.

Role-Based Access Control The OBD security module is responsible for enforcing an access control policy on traffic flowing through the OBD port. The policy may restrict the type of messages (if any) the OBD device is authorized to send into the vehicle. The policy could also restrict CAN messages which are allowed to flow from the vehicle to the OBD device.

A typical vehicle may have 50 - 100 ECUs capable of sending thousands of types of CAN messages. The OEM could create the access control policy for each OBD device by stepping through the entire list of messages and selecting the access rights for each message type individually.

- Null: The device is not able to read or write the message type.
- Read only: The device is able to read (listen for) the message type but not write the message type.
- Write: The device is allowed to write messages into the vehicle. Typically writes would be restricted to a white list specifying the list of arbitration IDs and associated message content e.g. In case of ISO 15765 diagnostic messages, the white list of allowed service identifiers (SIDs) and parameter identifiers (PIDs) needs to be specified in the access control list.

This specification of the access policy becomes a burdensome task when it is repeated for each make, model and year for each OBD device type manufactured by an OBD device vendor. Role based access control (RBAC) allows the OEM to create a structured model for assigning access rights. This technique has been applied to industrial control systems controlling plants with thousands of control points [2]. A similar model will be used to allow OEMs to efficiently assign access rights to an OBD device based upon the role associated with the device (e.g. insurance tracking vs. diagnostic tool).

System Level Entities

The entities within the proposed system include:

- **OBD security module**: This module sits between the OBD port and the vehicle CAN bus as shown in Figure 1 and enforces security. It could be embedded into the OBD port, it could be an inline module between the OBD and CAN bus or it could be implemented within a gateway device if the vehicle uses a gateway to bridge CAN busses. The OBD security module authenticates the OBD connected device then applies the corresponding security policy as specified by the OEM.
- **OEMs**: The OEMs are responsible for establishing policies to be enforced by the OBD security module. The OEM provides a process by which ODB device manufacturers may request security policies for their devices. The OEM also serves as the public key Certification Authority (CA) for the vehicles it manufacturers. The OEM holds the private key and loads the corresponding public key into each OBD security module. The OEM uses its private key to digitally sign certificates for OBD device manufacturers.
- **OBD device manufacturers**: The manufactures receive an X.509 certificate from the OEM and act as a certification authority for the devices they produce. They also produce a private/public key pair for each device manufactured. They use their private key to sign certificates which are bound to their product. These certificates contain the public key of the manufactured device as well as the security policy the OEM approved for the product.
- **OBD device and certificate**: The device which will plug into the OBD port contains a public/private key pair unique to that device. The public key is also contained in the device certificate signed by the device manufacturer. The device uses trusted storage (e.g. a secure microcontroller) to protect its private key.
- **Policy change authorization token**: The OEM creates a token and digitally signs it. This token flows from the OEM, to the OBD device manufacturer, into an OBD device and finally into the OBD security module. The OBD security module is able to verify the digital signature applied by the OEM.

These entities work together during the certification process and operation as described below.

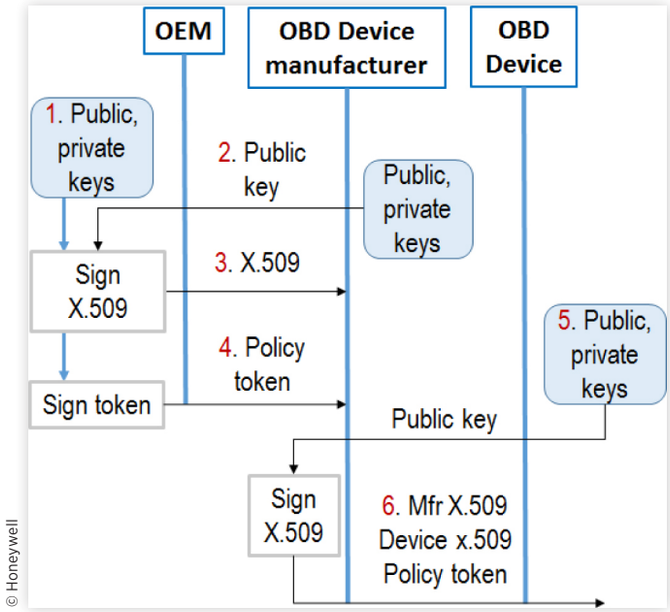
Certification Process

The OEM controls the certification process for OBD devices which will interact with the OEM's vehicles. The method the OEM uses to certify a manufacturer and their devices is outside the scope of this paper. It is expected to include a statement by the manufacturer regarding the access rights their device needs in order to perform its intended function. The OEM would typically perform a risk assessment which takes into account the level of confidence placed in the OBD device and the potential for introducing a security or safety risk to the vehicle. At the end of the risk analysis the OEM and OBD device manufacturer agree upon the security policy the vehicle will apply to the device. The remainder of this subsection describes the process for binding the OEM specified security policy to the OBD device.

The public key material and a corresponding public key infrastructure to allow the vehicle to verify the OEM specified security policy to be applied to an OBD device is outlined below and shown in Figure 3.

1. The OEM creates a public/private key pair. The private key is protected in a hardware security module and backed up by the OEM.

FIGURE 3 Creation of OBD device credentials.



2. The OBD device manufacturer creates a public/private key pair and sends the public key to the OEM along with the manufacturer identification data using a certificate signing request protocol (e.g., Certificate Management Protocol (CMP) [6]).
3. The OEM uses the OEM certification authority private key to digitally sign a certificate (e.g. X.509v3) containing the OBD device manufacturer identity and public key.
4. The OEM creates a policy change authorization token which includes the changes to the policy and the manufacturer, make and model of the set of devices associated with this authorization token. The authorization token is signed using the OEM private key and sent to the OBD device manufacturer.
5. A public/private key pair are created for the OBD device.
6. The OBD device manufacturer uses the OBD device manufacturer private key to sign a certificate for the OBD device containing the device identification data (manufacturer, make and model) and the device public key. The OBD device manufacturer also loads a copy of the OBD device manufacturer certificate into the OBD device.

At the end of the process above the OBD device contains

- Device manufacturer certificate - Signed by the OEM
- Device certificate - Signed by the device manufacturer
- Policy change authorization token specific to the OBD device type (manufacturer, model) - signed by the OEM

During the vehicle manufacturing process the OEM loads the OEM public key into the OBD security module.

Sample Policies

The sample policies below illustrate the application of security policy within the OBD security module to enhance vehicle security and safety.

- **Safety**: Assume that CAN message ID 5 with content 0000 0000 0000 1111 tells the brake system to vent pressure from the wheel cylinders. This is typically done to bleed air out of the hydraulic brake system. The OBD security module default safety policy would block message ID = 5, content 0000 0000 0000 1111 from being injected via the OBD port. (The policy would likely block a broad set of messages.) This would prevent a compromised OBD device (e.g. insurance dongle with a modem) from disabling the brakes on a vehicle [12]. A tool certified by the OEM to be used by a service technician to bleed the brakes would have an authorization token which turns off this policy, thus allowing the service technician to bleed the brakes. If another device (e.g. a compromised insurance dongle) was plugged into the OBD port after the technician serviced the vehicle, the safety policy would prevent the compromised dongle from issuing the bleed brakes command to the vehicle. The authorization token is only in effect while the device linked to the authorization token is plugged into the OBD port and using the randomly generated cryptographic key from the OBD security module.
- **Security**: The keyless entry system on a vehicle is able to send a CAN bus message to unlock the doors. However, a compromised OBD device could also send an unlock command, allowing a thief into the vehicle. Therefore, a security policy in the vehicle security module would be to block "unlock door" messages sent into the vehicle via the OBD port unless the OBD device has provided a policy change authorization token signed by the OEM.

One can imagine many more threats and policies which would block those threats. In general, the OBD security module policy should be to block messages from the OBD device unless the device has a policy change authorization token opening up the policy. A few of the many types of policy changes which could be requested for an OBD device include:

- Allow the device to limit the speed of a vehicle. This may be used in fleet management applications.
- Allow the device to load new firmware on an ECU.
- Allow the device to read the GPS coordinates of a vehicle so the OBD device can provide vehicle tracking
- Allow a device to remotely start the engine of a vehicle
- Allow the device to send a command to change the emissions control setting on a vehicle

The OEM is responsible for the policy specification in the OBD security module as well as the format used within the policy change authorization token. The authorization token could take many forms depending upon the structure used for specifying policy within the OBD security module. Data fields which would be typical in the authorization token include:

- Applicable vehicles: A device may be allowed to change the policy on vehicles produced since 2016 but not before. The device may only be licensed with the OEM to be used on certain models and therefore the authorization token would identify the set of licensed vehicles.
- Policy/rule identifier: Depending upon how the OEM manages policy/rules in the OBD security module, the token may specify the number of the rule which is to be modified or turned off, e.g. "turn off rule 3."
- Allowable message ID list: Most firewalls and OBD security modules operate using a "deny unless explicitly allowed" model. Thus, if the policy does not explicitly allow a message ID, that message ID would be blocked. An "allowable message ID" field could specify IDs which are allowed to pass across the OBD port.
- Specific message ID/value pairs: Frequently there are multiple parameters within a message. e.g. one byte could control the throttle position while another byte in the same message controls the amount of fuel delivered. The default policy could allow message ID x but only allow a specific set of parameters. e.g. only allow byte specifying the amount of fuel to be between 10 and 100. The policy could allow an authorized device to change these parameter limits.

5. Operation

The OBD security module would typically ship with a default security policy. This policy specifies the types of messages allowed to flow to and/or from the vehicle to a device plugged into the OBD port. The policy change authorization token specifies changes to be applied to the policy when the device associated with the authorization token is plugged in.

Up to this point the focus has been on creating the infrastructure and the process of creating policy. What actually happens in the field when a device is plugged into the OBD port of a vehicle equipped with an OBD security module? Two cases of interest are outlined below.

Non-Certified Devices. No Extra Privilege

When a device which has not been certified by the OEM (e.g. a legacy insurance dongle) is plugged into a vehicle, the OBD security module will attempt to authenticate the device and obtain the policy change authorization token. Obviously, a legacy OBD device will not complete the authentication. The OBD security module will simply enforce the default policy established by the OEM.

The default policy could allow the OBD device to observe all CAN traffic but not allow the device to inject any messages into the vehicle. A more practical and fair policy is to allow a set of diagnostic SIDs required for emissions compliance testing that are known to be benign. Allowing diagnostics SIDs does not put vehicle in danger since they only read information and do not modify anything. A recommended default policy would also allow reading Diagnostic Trouble Codes (DTCs) and clearing them. This semi-open default policy is still secure while preserving all legitimate legacy use cases.

Certified with Privilege

When a certified OBD device is plugged into a vehicle, the OBD security module and OBD device will perform the protocol exchange described above in subsection Public Key and Figure 2. This protocol exchange and cryptographic processing within the OBD security module accomplishes the following:

- Allows the OBD security module to confirm the identity of the device connected to the OBD port
- Allows the OBD security module to confirm that the policy change authorization token is bound to the device
- Allows the OBD security module to confirm that the policy change authorization token was authorized by the vehicle OEM.
- Allows the OBD security module and OBD device to establish a cryptographically protected communications channel.

The OBD device will be able to perform all of the functions it has been certified for. There is no need for additional manual operations by the vehicle owner or technician to enable the OEM approved security policy modification.

6. Design Considerations

There are many possible variations to the approach described above. Some of the considerations to consider when designing this system are discussed below.

Implementation Cost

How much will it cost the OEMs and OBD device manufactures to implement such a system?

Certification Authority It is not necessary for the policy change authorization token to interoperate with vehicles from other OEMs or any government entities. Thus each OEM is free to serve as the certification authority for their own vehicles. This eliminates the costs associated with a 3rd party certification authority.

Certification Cost The certification cost are expected to roughly correspond to the level of risk associated with a device. Here, risk is driven by the range of authorized CAN messages (commands) and the type of access required to cause the OBD device to send data.

A device which only reads data viewable at the OBD port but never writes data into the port essentially has no privilege and thus would not require any certification of authentication. The policy implemented in the OBD security module would block the device from sending any unauthorized messages into the vehicle.

A device which only requests diagnostic trouble codes would require little or no certification. Typically the OEM default policy in the OBD security module would allow any device to make benign requests.

A device which performs only limited functions (e.g. limit the speed of the vehicle using a well-defined set of CAN messages) but does not have authorization to manipulate safety critical systems (steering, brakes) would require minimal review to certify.

A device which performed safety related diagnostics (manipulate ABS) or triggers events related to physical security (unlock doors) would require a more rigorous certification effort to ensure that the device could not be easily misused. If such a device had no external access (e.g. the device is only controlled by a technician pressing buttons) it would be low risk and require a minimal certification. In contrast a similar OBD device which is wirelessly connected to a repair shop network or cellular network would require a more stringent certification effort.

Key Storage A concern in environments containing cryptography is the protection of keying material [15]. The approach being described here requires the OEM to protect their private key material in a hardware security module. The number of hardware security modules required by an OEM is small so cost is not a problem.

The OBD security module stores only public key material so the integrity of the public key must be protected. The OBD security module does not require hardware to store a private key. Thus, OEMs are free to use a range of microcontrollers available from multiple competitors.

The devices which connect to the OBD port will be required to store private key and thus must have trusted storage. Fortunately, new microcontrollers are available with trusted storage technology which can store cryptographic keying material in low cost hardware to prevent tampering and thus improve trust in the overall public key environment. Key material in this trusted store may be used by the local cryptographic engine but the keys cannot be extracted.

The bottom line is that the cost of protecting the cryptographic keys is not a significant issue.

OBD Security Module Many OEMs already include some type of gateway module within their bus architecture to route traffic to/from the OBD port and CAN busses. The security function described here may be implemented within a gateway module with modest processing and storage reserves.

Potential Costs of not Adding Security

OEMs are forced to make a decision regarding the costs of adding security as outlined above versus the potential costs of not providing security. The following factors conspire to make the analysis difficult.

- Lack of historical data: The connected car is relatively new and to date most of the attacks have been carried out by researchers, not attackers with hostile intent. We will not know the frequency or severity of attacks for several years. Thus, OEMs are forced to look at other industries and extrapolate the risk to their market.
- Deterrent effect: Security features (both cyber and physical) tend to deter illegal activity. Thus, if OEMs do implement security and few attacks occur it raises questions about how many attacks would have occurred if security was not implemented.
- Financial impact: The financial impact of an event can depend upon public perception, the court system and the opinion of Government authorities. Will the OEM be viewed as negligent or will all responsibility be placed upon the attacker?

The following scenario is given as a framework for discussing the potential costs of not addressing OBD security. It is very speculative for the reasons listed above.

There are currently about 4M vehicles with OBD devices containing cellular modems (e.g. Insurance company provided devices for monitoring driving behavior). The majority of these contain security vulnerabilities. Assume that the targeted OEM has 100,000 vehicles using an insecure cellular dongle.

Attackers have demonstrated the tendency to attack large targets. This allows them to develop the attack once and use it many times. Windows PCs represent an example of a large target. Thus, an OEM with a large fleet makes for a large target.

Assume that attackers are able to compromise 1% of the OEMs 100,000 vehicles with cellular devices. What could the attackers do?

- Hold the vehicles for ransom by preventing operation of the vehicle
- Attempt to involve the vehicle in an accident by interfering with acceleration, braking or steering.

What would be the costs of the compromise? OEMs could look at historical data involving a similar number of vehicles with a common defect as a calibration point. (Faulty ignition switches, SUVs with tire problems prone to rollover, unintended acceleration) They could then sum the costs of the following.

- Class action lawsuit
- Government fines
- Mandated recall
- Damage to brand image and perceived lower value of vehicles

What is the proper decision in the face of such uncertainty?

Technology Options

The system description above is intentionally generic to allow an OEM and their tool partners to select specific technologies and processes suitable for their business. Variations in the process described above that would produce a similar result include:

- Cryptographic algorithm: The public key algorithm could be RSA or elliptic curve.
- Cryptographic key length: The various public key pairs (OEM, OBD device, and manufacturer) could use different key lengths appropriate to the perceived threat.
- Certificate structure: The certificates could be X.509 v3 certificates, IEEE 1609.2 [4] or a proprietary format. However, there is a large base of mature software available for X.509 certificates.
- Public Key infrastructure (PKI): The PKI could be a simple structure in which the OEM serves as its own root of trust. (aka Root certificate authority). The system could use a larger and more complete PKI in which the root of trust is above the OEM. This type of structure could allow one OBD device to be recognized by OBD security modules associated with multiple OEMs.
- Key exchange protocol: Many different protocols exist to establish the secure communications between the OBD security module and the OBD device. The TLS protocol provides a worked example which has been thoroughly reviewed by the security community.
- Policy change authorization token distribution: The authorization token could be distributed by embedding it within the OBD device such that the OBD device is the transport mechanism. The authorization token could be embedded in the OBD security module firmware and activated when an authorized OBD device authenticates itself. The authorization token could be fetched in real time by either the vehicle or the OBD device.
- Authorization token structure: The structure of the policy override could be a complete replacement of the default OBD security module policy. Alternatively the structure could simply identify changes to the default policy.

7. Summary/Conclusions

This paper has presented an approach which provides the industry with an alternative to the one-size-fits-all security policy for the OBD port.

The approach provides OEMs with the flexibility to change the security and/or safety policy of the OBD security module in the field using a cryptographically protected authorization token which is cryptographically bound to the OBD connected device.

This same policy change concept may be applied to devices or services accessing the vehicle via any of the TCU interfaces (Cellular, Wi-Fi, Bluetooth or even USB). The device or service attempting to access the vehicle would be required to perform a cryptographic handshake with an access control function within the vehicle. The device or service requesting access would pass its X.509 certificate containing identification data along with policy change authorization data to the access control function within the vehicle. The vehicle could then enforce access rights established by the OEM specifically for the device or service requesting access.

The industry should determine if there is a need to protect vehicle from unauthorized message injection via the OBD port. Assuming there is a need, SAE could serve as the leader in coordinating the development of a standard to move from concept to fielded solution.

Contact Information

Tom Markham, Engineering Fellow
Honeywell ACST,MN10-211A
1985 Douglas Dr.
Golden Valley, MN 55422-3935
Office: 1 763 954-6840
tom.markham@honeywell.com

Definitions/Abbreviations

ABS - Anti-lock braking system

ADAS - Advanced driver assistance systems

AES - Advanced Encryption Standard. NIST FIPS 197

CA - Certification authority

CAN - Controller area network

CANFD - CAN with flexible data-rate

CMP - Certificate Management Protocol

DLC - Diagnostic link connector

ECU - Electronic control unit

FIPS - Federal information processing standards

GPS - Global positioning system

NHTSA - National Highway Traffic Safety Administration

NIST - National Institute of Standards and Technology

OBD, OBD-II - On-board diagnostics (~1988) and On-board diagnostics version 2 (~1997)

OEM - Original equipment manufacturer

PID - Parameter identifiers. See SAE J1939

PKI - Public key infrastructure

RBAC - Role based access control

RSA - Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman public key cryptographic system

SCEP - Simple certificate enrollment protocol

SID - Service identifier. See SAE J1939

TCU - Telematics control unit

TLS - Transport layer security

X.509 - International Telegraph Union standard for public key certificates

References

1. Bluetooth and Wi-Fi/cellular OBD-II devices https://www.amazon.com/Panlong-Bluetooth-Diagnostic-ScannerAndroid/dp/B00PJPHEBO?ie=UTF8&psc=1&redirect=true&ref_=oh_aui_detailpage_o02_s00&tag=androidcentralb-20, accessed Sep. 2016.
2. Department of Energy, Office of Scientific and Technical Information, “RBAC Driven Least Privilege Architecture for Control Systems,” http://www.osti.gov/scitech/servlets/purl/1124080 , Accessed Sep. 2016.
3. Ian, Foster, Prudhomme Andrew, Koscher Karl, and Savage Stefan. "Fast and vulnerable: a story of telematic failures." In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. 2015.
4. IEEE Standard for Wireless Access in Vehicular Environments “Security Services for Applications and Management Messages”, 1609.2-2016.
5. Internet Engineering Task Force (IETF), “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” IETF RFC 5280, May 2008.
6. Internet Engineering Task Force (IETF), “Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)),” IETF RFC 6712, Sep 2012.
7. Karnouskos S., "Stuxnet worm impact on industrial cyber-physical system security," *IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society*, Melbourne, VIC, 2011, pp. 4490-4494. doi: 10.1109/IECON.2011.6120048
8. Miller, C., and Valasek, C., “Advanced CAN Injection Techniques for Vehicle Networks,” Presentation at BlackHat USA 2016
9. Nathanson, M., "Vehicle Intelligence and Remote Wireless OBD," SAE Technical Paper 2000-01-3506, 2000, doi:10.4271/2000-01-3506.
10. NIST Computer Security Research Center, “HL7 Role-Based Access Control (RBAC) Role Engineering Process,” http://csrc.nist.gov/groups/SNS/rbac/documents/hl7_role-based_access_control_(rbac).pdf accessed Sep. 2016.
11. Canetti, Ran, and Herzog Jonathan. "Universally composable symbolic analysis of mutual authentication and key-exchange protocols." In *Theory of Cryptography Conference*, pp. 380-403. Springer Berlin Heidelberg, 2006.
12. Cnet.com, Researchers hack a Corvette's brakes via insurance black box,” http://www.cnet.com/roadshow/news/researchers-hack-a-corvettes-brakes-via-insurance-black-box/, Accessed Sep 2016
13. Rosekind, M. (NHTSA), Congress of the United States, Committee on Energy and Commerce, Sep 12, 2016.
14. SAE Vehicle E E System Diagnostic Standards Committee, “E/E Data Link Security,” SAE Standard J2186 Rev. June 2005.
15. SAE Vehicle Electrical System Security Committee, “Requirements for Hardware-Protected Security for Ground Vehicle Applications”, SAE Standard J3101, Jun. 2015.
16. Wikipedia, “Public-key cryptography,” https://en.wikipedia.org/wiki/Public-key_cryptography, accessed Sep 2016.
17. Wikipedia, “Role-based access control,” https://en.wikipedia.org/wiki/Role-based_access_control , accessed Sep 2016.
18. Woo S., Jo H. J. and Lee D. H., "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993-1006, April 2015. doi:10.1109/TITS.2014.2351612
19. Yada1, A. et al, “Security, Vulnerability and Protection of Vehicular On-board Diagnostics” *International Journal of Security and Its Applications* Vol. 10, No. 4 (2016), pp.405-422 http://dx.doi.org/10.14257/ijsia.2016.10.4.36 ISSN: 1738-9976 IJSIA

SAE INTERNATIONAL

ARTICLE INFO
Article ID: 2017-01-1649

doi:10.4271/2017-01-1649

SmartDeviceLink as an Open Innovation Platform for Connected Car Features and Mobility Applications

Jeffrey Yeung, Ford Motor Company
Omar Makke, Perry MacNeille, and Oleg Gusikhin, *Ford Motor Company*

Abstract

SmartDeviceLink (SDL) is open-source software that connects the vehicle's infotainment system to mobile applications. SDL includes an open-source software development kit (SDK) that enables a smart-device to connect to the vehicle's human-machine interface (HMI), read vehicle data, and control vehicle sub-systems such as the audio and climate systems. It is extensible, so other convenience sub-systems or brought-in aftermarket modules can be added. Consequently, it provides a platform for cyber-physical systems that can integrate wearables, consumer sensors and cloud data into an intelligent vehicle control system. As an Open Innovation Platform, new features can be rapidly developed and deployed to the market, bypassing the longer vehicle development cycles. This facilitates a channel for rapid prototyping and innovation that is not constrained by the traditional process of automotive parts development, but is rather on the timeline of software development. This allows parties other than the OEM, including third party developers, universities, and startups, to quickly integrate their applications into the car. By digitizing the controls into a programmable application program interface (API), we can incorporate machine learning and ambient intelligence to learn user preferences and needs over time and intelligently pre-set them automatically. This can disrupt the traditional model of the human machine interaction. This is particularly important as transportation moves towards autonomous vehicles. In this paper, we discuss some specific examples using SDL to enhance the transportation experience, including intelligent air quality and climate control, enhanced radio interface, mobility service applications, and innovative intelligent vehicle features.

History

Received: 02 May 2017
Published: 28 Mar 2017

Citation

Yeung, J., Makke, O., MacNeille, P., and Gusikhin, O., "SmartDeviceLink as an Open Innovation Platform for Connected Car Features and Mobility Applications," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 10(1):2017, doi:10.4271/2017-01-1649.

ISSN: 1946-4614
e-ISSN: 1946-4622

Introduction

The automotive industry is experiencing an era of expanded technology opportunity, particularly in the areas of vehicle electronics and connectivity. The industry has developed methods of innovation and product development suited to the production of very robust commodity products in a highly optimized, global manufacturing system. In recent years, vehicle infotainment systems have been combined with internet services via the wireless cellular data network to create new opportunities. This combination disrupts the traditional development process significantly. New methods of innovation and development are needed for this era of rapid development.

Open innovation is a topic within the field of innovation management. The term has been promoted by Henry Chesbrough from the Center for Open Innovation at the University of California. He has written books and papers on the subject [1, 2], although the idea has been discussed as far back as 1960s [3]. The fundamental premise is that properly managed inflows and outflows of knowledge can improve innovation within a firm while creating new markets of the technology outside the firm.

In a new business model, a good understanding of the value proposition, target market, value chain, revenue mechanism(s), value network, and competitive strategy is needed to extract value from the technology. Developing this understanding often involves a process of trial and error and can take considerable time. A firm with a new technology needs to develop the business model very quickly. Most of the time, firms and industries operate with a closed innovation strategy where incremental technological improvements drive product differentiation, meet new government regulations, market changes, etc. These firms frequently compete by being lean and robust, but lack the resources to consistently create value from rapidly changing technology.

The value network, or ecosystem, is developed through an evolutionary process that begins pretty chaotically but ends in business model(s) that add value. Early in the process, members of the ecosystem have varied characteristics, motivations, and contributions to make. As the ecosystem matures through trial and error, the necessary operational components become clearer, and some members may drop out while others may commit more resources. There can be mergers, spinoffs and selloffs until a stable business is established, commoditization sets in and a closed innovation model is once again preferred.

In software development, open source is a key enabler for an open innovation platform. The ecosystem consists of member firms working together on a common software platform that allows innovators to develop technology and business models utilizing road vehicle entertainment systems for live testing for their concepts. In the following sections, the authors introduce their open innovation platform and several use cases, hoping to encourage technology entrepreneurs to join their ecosystem.

Open Innovation Platform

SDL is the focus of an open innovation ecosystem intended to discover new businesses arising from recent technology advancements. These new businesses may be based on unexpected combinations of hardware, software, and ecosystem members that are admittedly difficult to predict at this point. Therefore, the adaptive nature of open innovation approach is necessary.

The SDL Open Innovation ecosystem is intended to involve members from large and small firms that may develop software, hardware or both. Currently the SDK is targeted to mobile devices that support iOS and Android operating systems but is open to new hardware devices if they support Bluetooth, WiFi and/or USB.

The ecosystem also accommodates members from other automotive OEMs, suppliers and after-market companies that wish to implement the core application (Figure 1) on their infotainment platforms. The openness of the platform makes possible many different alliances of development partners, OEMs, suppliers and aftermarket companies. The nature of the software interfaces is such that members can have very different business priorities, development cycles, revenue models and technologies.

SDL allows mobile apps to read vehicle data and control vehicle sub-systems such as the audio and climate systems. It is extensible, other convenience sub-systems or even brought-in aftermarket modules can be added. Figure 2 shows how SDL Core interfaces with the rest of the vehicular sub-systems

FIGURE 1 Innovation ecosystems.

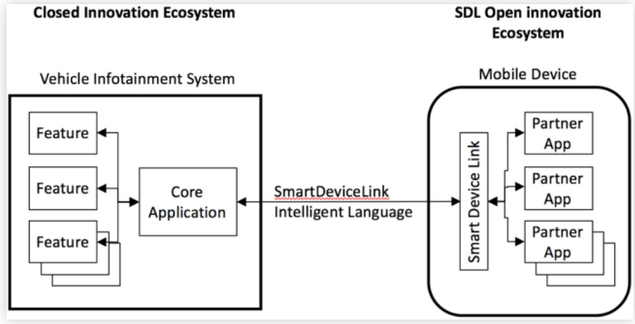
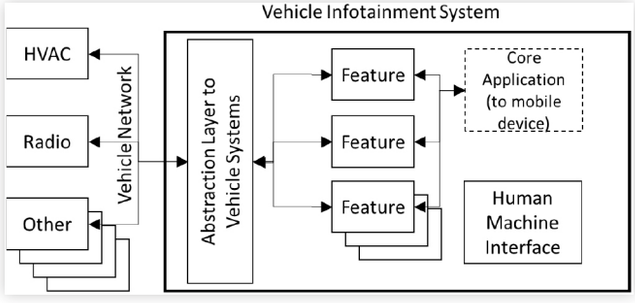


FIGURE 2 Vehicle systems flowchart.



and features. These features include navigation systems, climate control, GPS, geo-fencing, speech recognition, radio, audio equalizer, etc.

SDL facilitates a channel for rapid prototyping and innovation that is not constrained by the traditional process of automobile parts development but is rather on the timeline of software development. This allows parties other than the OEM, including third party developers, universities, and startups, to quickly integrate their applications into the car using mobile device apps. By digitizing the controls into an open source programmable application program interface (API), it is possible to incorporate machine learning and ambient intelligence to learn user preferences over time and intelligently pre-set them automatically. This can disrupt the traditional model of the human machine interaction.

This paper describes a system to connect the rider to the vehicle. It first begins with an introduction into SmartDeviceLink and its capabilities to show how it can be used as an Open Innovation Platform for the vehicle. This is followed by examples of innovative features for the Connected Car using SDL.

The first use case is an intelligent climate control system, integrating brought-in sensors and wearable devices for automated climate and air quality control. This utilizes air-quality sensors to detect both the interior and exterior air quality and adjust the fan speed and recirculation vent appropriately to maintain clean air within the vehicle cabin. This has particular use in high-density cities high air pollution. Furthermore, we discuss the usage of wearable devices with cloud data to create an intelligent climate control system based on real-time feedback from the vehicle occupants to improve the user experience.

The second example is for a hybrid radio system, which combines internet content providers with broadcast radio services. We will demonstrate an example of hybrid radio implementation with a mobile app called NextRadio. NextRadio receives the FM radio in the car and combines it with metadata received on the smartphone to enhance the radio experience in the vehicle. Album art, song information, and music genre are added to the display. Additionally, it enables selection of radio stations within range of the vehicle by genre or current song, none of which is possible with just FM radio.

The third example is a mobility application, specifically ride-sharing. In this case, SDL acts as the medium to enable not the driver but the passenger to control the vehicle subsystems from their phone, either in a taxi or ride-share or in a large passenger vehicle. This allows for dynamic control of their preferences from their phone based on various metadata, which we can learn about them over time. This also has great implications for autonomous vehicles, which may lack a familiar console interface to control these options.

We discuss the implications of this system as a whole and how it fits into the current trends in Mobility. Finally, we also touch on some other examples including audio equalizer and auto-high beam.

SmartDeviceLink Overview

SmartDeviceLink (SDL) is an open source project pioneered by Ford Motor Company that connects in-vehicle infotainment systems to smartphone applications allowing automakers the opportunity to provide customers with highly integrated connected experiences, and app developers with new and exciting ways of connecting with their customers (Figure 3). It is managed by the open source community, SmartDeviceLink Consortium Inc. (SDLC), which includes companies such as Ford Motor Company and Toyota Motor Company. It is open to OEMs, suppliers and app developers who are integrating with SDL or have plans to integrate with SDL in the future [4].

SDL comprises of head unit software, known as SDL core, and mobile SDKs for Android, iOS, and cloud configuration. SmartDeviceLink can be treated as a specification which can be implemented by arbitrary developers. Nevertheless, open source implementation is provided. SDL core's implementation is provided in C++. It supports several transport protocols: Bluetooth, WiFi and USB. SmartDeviceLink supports both media and non-media apps. Media apps are dedicated to audio streaming and provide an alternative user interface (UI) to the native media UI, which usually include FM/AM/XM, and CD. Non-media apps extend the built-in applications, and normally read vehicle data and provide added functionality to the driver. The head unit defines four states called HMI_LEVEL for each connected mobile app: FOREGROUND, LIMITED, BACKGROUND, and NONE.

When the app is selected from the head unit, it opens a predefined UI templates and is put in FOREGROUND state, which gives the app all of its allowed permissions. Once opened, the driver may switch to a different screen on the head unit, such as the navigation screen. If the driver navigates to a different screen on the head unit, the app enters BACKGROUND state where most of its permissions are lost. A LIMITED mode is defined for media apps. This mode is entered if the HMI is displaying the predefined UI template, but does not have propriety. This can happen if an app is streaming music and the driver receives a phone call for example.

Mobile applications can communicate with SDL through the SDL software development kit (SDK), which is available for Android and iOS platforms [5]. The SDK makes the app discoverable by the vehicle's head unit. It exposes a set of

FIGURE 3 SDL architecture.

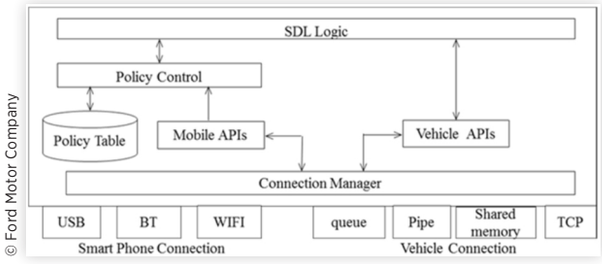


TABLE 1 SDL remote control APIs.

API	Parameters	Description
getInteriorVehicleCapabilities()	Zone	Returns supported modules the vehicle is equipped with (Radio, Climate Unit...).
getInteriorVehicleData()	Zone Module	Reads module data. Modules are obtained from getInteriorVenicleCapabilities.
setIInteriorVehicleData()	Zone Module, Data	Control API. Sets the data of the Module for the specified zone

remote procedure calls (RPCs) through a defined set of application programming interface (API). In brief, the app instantiates an instance of the SDL proxy class, provided by the SDK, which handles the communication between the app and the vehicle. The RPCs are methods of the proxy class. Moreover, the proxy class receives the vehicle's notifications and makes them available for the mobile app.

Different OEM's can implement SDL and provide different capabilities. Once the proxy starts within the app and connects with the vehicle, basic information about vehicle capabilities such as vehicle's language, audio capabilities, etc. are immediately made available to the app. The app can further query for specific capabilities through the proxy APIs. Although the RPC implementation is not directly exposed to the app developer, it is worth noting that the RPC protocols are implemented as JSON strings.

A remote control extension for SDL is created and is available in the public repository. The extension consists of additions to SDL core inside the vehicle and to the mobile SDK for Android and iOS. Three major RPC's responsible for remote control, and their corresponding APIs are shown in Table 1. These APIs provide enough abstraction for mobile apps to control vehicle modules with different capabilities inside vehicles by different OEMs.

Security Considerations

As with any open source platform and wireless applications, a major consideration for any system is security. In the case of SmartDeviceLink, the base security mechanism is predefined, but the OEM can extend it to add additional layers of security. Apps during testing phases can also have more privileges than during release.

The core of SmartDeviceLink security is the policy table, which is hosted in the cloud by the OEM. Each SmartDeviceLink App requires an App ID to be generated by the OEM, which identifies the app in the policy table. Each App ID is associated with an explicit set of RPCs which the app can execute in each of the four HMI_LEVEL states of the app. If the App ID is not present in the table or if the app attempts to execute an RPC not listed in its HMI state, it will be denied. In this way, the OEM has central control over all app permissions and modes. Usage of a new app prompts an update of the policy table from the cloud. Updates can also be set to be done periodically. Vehicles are shipped with a default policy table at the outset.

Security starts by requiring the application developers to request an App ID from the OEM and specifying what APIs are required. The OEM can choose to group the APIs into logical groups which are referred to as functional groups. Each OEM can group the APIs differently, or not group them at all for more control.

Once App ID is approved, the OEM registers that App ID on a server and specifies which function groups can be used, and in what HMI state (FOREGROUND, LIMITED, BACKGROUND, NONE). Whenever the mobile application establishes connection, it tries to download a policy table automatically from the Cloud which contains the app's privileges. The policy table implementation and security is left to the OEM, and it is highly recommended that this policy table be encrypted. The mobile application will then be able to call APIs as dictated in the policy table.

If the OEM wants to revoke App ID permissions, the OEM has to modify the entries for the App ID in the cloud. As soon as any arbitrary mobile application downloads a new policy table to the vehicle, the specified App ID gets revoked.

Without any enhancements, the default security model for SmartDeviceLink protects against average user misuse only. The App ID can be reverse engineered in the current model, and updates to the policy table do not occur if there is no internet connection. This can be enhanced by the OEM in several ways. For example, it would be possible for the OEM to not keep any persistent policy table inside the vehicle, and hence severely limit the capability if there is no internet connection available. It is also possible to force the permissions for a given App ID to expire after a given time as long as the vehicle has a tamper-proof clock available.

To protect the App ID from reverse engineering, the mobile application could obtain an encrypted App ID from the third-party server and decrypt it in the head unit. This would make it possible to use asymmetric encryption, similar to how the HTTPS protocol works. An API can be added to SmartDeviceLink and be configured such that all mobile apps have access to it at any time. Each mobile application would act as a relay between the head unit and the third party server, and then the third party server sends the App ID for the mobile application securely to the vehicle. For this to work, the customer would have to provide credentials to login to the server in order for the server to respond back to the head unit. The login mechanism in this case would most likely be the weakest link since a human is involved. If an attacker stole a username and password, the attacker could then use the third-party server to personalize an App.

Regardless of the security model used, it is highly recommended that the OEM give the ultimate control to the vehicle owner. Whenever a mobile application attempts to call a remote control API for the first time in an ignition cycle, a pop-up screen with timeout timer can be displayed to the driver to grant remote control permission to the mobile application. Moreover, at all times there must be a notification on

the head unit display that there is a mobile application running capable of remotely controlling the vehicle. At any time, the driver can navigate to the proper menus to disable remote control ability for a specific application or for all applications simultaneously.

Other security considerations arise depending on the implementation specifications. For example, Denial of Service attacks can be possible if the mobile application keeps causing remote control permission pop-up to appear. A mobile application may increase the load of the CAN bus if it can keep requesting parameters to change at a fast rate. If the mobile application can cause a menu on the head unit to change, another form of Denial of Service is possible. Solving these challenges is left to the OEM because they depend on the OEM's architecture and specification. Finally, the OEM is strongly advised against adding API's which affect the vehicle's torque.

FIGURE 5 Air quality diagram and image of ChemiSense sensor.

© ChemiSense

Climate Control and Air Quality

In this section, we discuss how SDL-RC can be used to create innovative models for enhanced climate control that allows integration of brought-in sensors and cloud information (Figure 4).

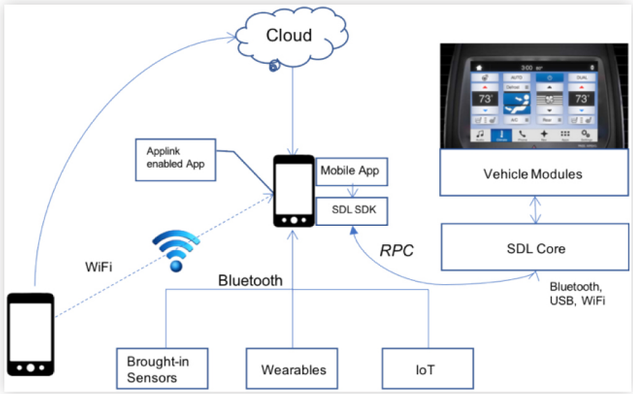
Climate comfort in the vehicle involves a system of integrated heating, ventilation and air conditioning (HVAC), either controlled manually or automatically [6]. Automotive HVAC systems vary in capabilities, from a basic system that just maintains the set temperature to one that adjusts based on temperature, humidity and sun-load sensors. High-end vehicles feature multi-zone automatic climate control that differentiates driver, front passenger and rear passenger zones. Some even have infrared sensors that monitor the occupants' surface temperature.

Further advancement includes the addition of air quality management. This is an increasing concern as urban areas continue to grow. It is characterized by an Air Quality Index (AQI), which is an indicator of the health impact of the air as measured by the concentration of harmful gases and particulates in the air. Cabin air quality can be improved through proper management of the climate control system [7].

As shown in Figure 5, a sensor (powered by ChemiSense, www.chemisense.com) installed in the vehicle measure different pollutants in the cabin air, such as PM2.5 (particulate matter less than 2.5 micron in diameter), carbon monoxide (CO), and hydrocarbons (HC). The sensor communicates with the mobile app and sends the measurements of the different pollutants periodically. The app communicates with the climate control system through SDL, modifying the parameters shown in Table 2. A threshold is defined in the mobile app, such that when PM2.5 or other important chemical measurements exceed that threshold, a clean cycle is initiated. The recirculation door is closed if it is not already closed, and the blower speed is increased, causing the cabin air to flow through the air filter continuously. This filters the cabin air.

Also, the system in Figure 5 leverages cloud information about air quality. If there is a spike of pollution in the external air, the system can switch to recirculation while the vehicle is passing through the polluted area. This architecture has been piloted in China and demonstrated effectiveness of the proposed solution [8].

FIGURE 4 Diagram of SDL connections.



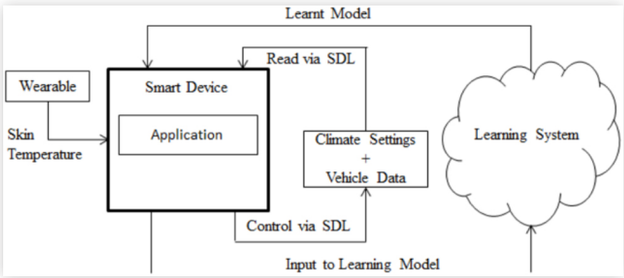
© Ford Motor Company

TABLE 2 SDL parameters for climate control.

Parameter	Description
acEnable	Toggles AC ON/OFF
desiredTemp	User input (the set temperature in the head unit).
fanspeed	Blower speed in %
currentTemp	Outside Ambient Temperature
temperatureUnit	Unit of the temperatures
circulateAirEnable	Air recirculation ON/OFF
autoModeEnable	Auto mode value ON/OFF
defrostZone	Front, Rear defrost ON/OFF
dualModeEnable	Dual mode is ON/OFF for units supporting zone control

© Ford Motor Company

FIGURE 6 Architecture of intelligent climate control.



© Ford Motor Company

As seen in the example, using SDL, the intelligence in the system is implemented on the smartphone. The smartphone can connect to brought-in air-quality sensors of the customer's choice. This circumvents the need for automotive-grade sensors to be developed and greatly expedites the process of getting this feature into the vehicle, given that automotive-grade air quality sensors in the market are limited.

Furthermore, this adds the ability to incorporate intelligence into the algorithm to automatically make the necessary adjustments to maintain both cabin comfort and air quality [9]. For example, if the system learns that a particular rider consistently adjusts the temperature higher than average, then it can remember this and apply it to future rides, even in climates distinct from where it observed this preference. With this information, the vehicle can even pre-condition the cabin to its best guess at climate preferences prior to picking up the rider. If he or she adjusts the settings, it can record this and further improve its understanding of the rider's preferences.

Wearables and Automated Climate Control

To further the ambient intelligence system, we have incorporated wearable devices that measure the wearer's skin temperature via a wrist-worn skin contact sensor. If the wearer is cold, the system can automatically increase the cabin temperature, and vice versa. This can be facilitated by a learning system that tracks trends and potential causes of the wearer's discomfort. The architecture of such a system is shown in Figure 6.

Hybrid Radio

Recent years have seen rapid growth of novel vehicle infotainment options that leverage connectivity platforms between phone apps and the vehicle HMI through, for example, Pandora, Spotify, and TuneIn radio. Although the popularity of these services challenges the radio industry, traditional AM/FM radio is still the most prevalent in-car infotainment option. AM/FM radio has a number of advantages, especially as an in-vehicle entertainment option. It delivers local and national content that is free, convenient, and is generally available in places where data coverage is fragmented. It is governed by well-established government authorities with regulations and international treaties that provide defined standards across political boundaries. AM/FM Radio can also take advantage of vehicle connectivity platforms to introduce innovative ways of enhancing the experience of traditional broadcast.

Hybrid radio is an emerging concept that enhances traditional broadcast radio with internet connectivity to provide a richer user experience and potential enterprise value. An example of a hybrid radio application that can be demonstrated today is NextRadio. The NextRadio mobile application is currently available on many popular cellphones sold in the United States. It takes advantage of the FM radio built into the cellphone to receive audio content. It also uses a cloud based platform, called TagStation (Figure 7), to create an interactive real-time experience. TagStation enables the radio industry to manage metadata associated with the broadcasted content. TagStation is unique in that the cloud-service interfaces directly with the radio station's live on-air system, which allows for event by event metadata management. NextRadio synchronizes the broadcast signal with the backend meta-data provided by a TagStation server to supplement the audio content with additional information such album art, listener feedback, song tagging, and social integration.

NextRadio can be naturally extended to the in-vehicle environment leveraging SDL, while taking advantage of existing car radio equipment. The vehicle has more electrical energy capacity than the cellphone. Consequently, the location system in the vehicle is more robust than in most cellphones and using the vehicle's GPS instead of the mobile phone's saves power on the cellphone battery. Similarly, the audio system in the car is better powered, larger and is tuned for the vehicle cabin compared to the mobile device. In integrating SDL with NextRadio we can leverage the radio parameters available for radio control presented in Table 3.

In order to obtain the meta-data for the station, the current station needs to be identified. To do this, the radio band (AM/FM) and currently tuned frequency from the vehicle are sent to TagStation from the NextRadio app. For vehicles with HD Radio, additional data includes HD Radio status, and the currently tuned HD Radio subchannel.

FIGURE 7 NextRadio and TagStation.

© NextRadio

TABLE 3 SDL parameters for radio control.

Parameters	Description
Frequency Integer	The integral part of the frequency
Frequency Fraction	The decimal part of the frequency
band	AM/FM/XM
rdsData	RDS Data received for the station (read only)
availableHDs	Number of available HD channels (read only)
signalStrength	Signal strength in percentage
signalChangeThreshold	Threshold for "available" in percentage
radioEnable	Radio is ON or OFF

Global Positioning System (GPS) location data supplied by either the vehicle or the cellphone are also used to confirm the identity of the currently tuned radio station.

After the station is identified, TagStation sends back the meta-data for the specified station. Then, using SDL functions, the NextRadio App provides supplemental content and soft buttons and meta-data with associated voice commands for the possible actions related with the given broadcast. The sequence diagram is presented in Figure 8.

NextRadio facilitates customer engagement with the content through enhanced user interactions, such as book-marking a song (Figure 9a), calling a phone number for a radio station (Figure 9b), saving a coupon from an advertisement (Figure 9c), finding advertised POI in the embedded navigation (Figure 9d), or requesting more information with an e-mail or SMS message. All actions work with a single button press or a voice trigger.

NextRadio also provides enhanced station discovery compare to the standard car radio HMI shown in Figure 10a. Figure 10b show the car interface that lists genres of the stations available at given GPS location. Selecting a given genre as shown in Figure 10c will open the list of the stations with currently played content shown in Figure 10d. Selecting a station will switch the radio tuner to this station.

FIGURE 8 NextRadio station discovery sequence diagram.

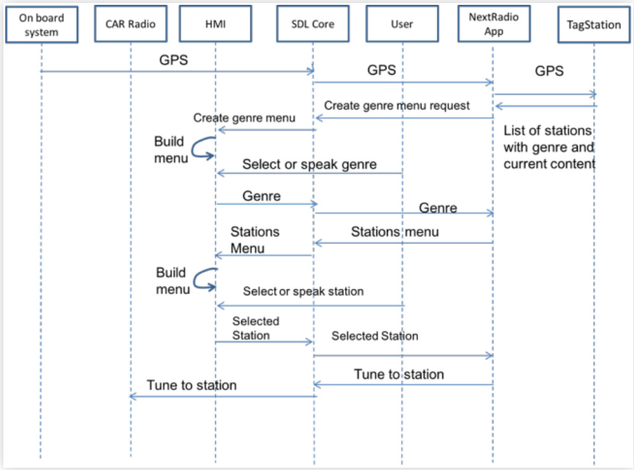
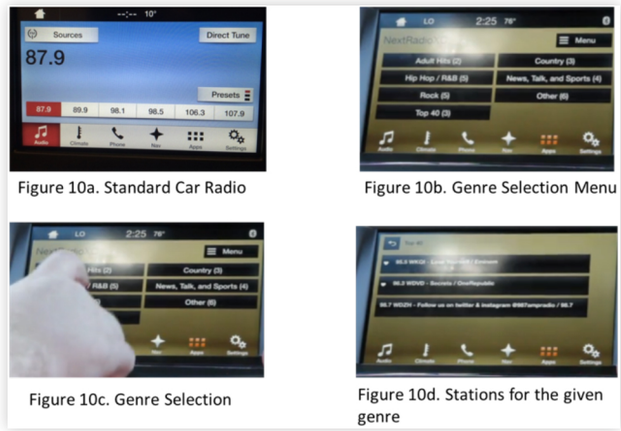


FIGURE 9 NextRadio interactive experiences.

FIGURE 10 Standard car radio interface vs SDL integrated NextRadio interface.



This use case demonstrates the possibility and benefits of integrating the NextRadio app over SDL with the vehicle AM/FM RDS radio with its large antenna, powerful receiver and integration into the vehicle sound system. SmartDeviceLink also supports integration with the vehicle HMI that provides conceptual integrity between the vehicle and NextRadio. Through this integration it is possible to provide NextRadio features in a robust, high-quality vehicle environment.

This capability enables intelligent searching for broadcast content to find stations with specific content, for example: a regional hockey game; a list of stations within a particular genre or format such as local news; stations that play jazz, classical music, etc. The system can be further extended to include autonomous recommendations for station selection based on the current broadcast options, historical listening habits, and connection to the social networks. Additional advantages of using the hybrid system include using broadcast

radio when streaming audio is not available and vice-versa, and using broadcast radio instead of streaming to reduce data usage.

The integration of hybrid radio can also provide significant enterprise value. The ability to capture and analyze customer listening patterns can provide substantially more detailed analytics for the station programming and advertisement positioning.

Mobility

As transportation moves away from personal car ownership and towards ride-sharing, vehicle and subsystems design will likely need to change. The user experience will be different, and user interaction models will also shift given that riders will no longer be in their own personal vehicles with their preferences already set and their personal belongings in place. The vehicles driven in may be unfamiliar to the rider as well, and travelers will be seated in the rear or the passenger seat. If traveling, they may not speak the local language, either to communicate with the driver to change settings or to read the lettering on the vehicle controls. In any event, setting personal preferences becomes a frequent, sometimes cumbersome, task on almost every trip.

In the mobility space, SDL enables these vehicle subsystem controls to be extended to other passengers in the vehicle who may either be seated in the rear of the vehicle or otherwise unable to configure the climate in their zones. For vehicles with multiple zones, such as a larger family van or a shuttle bus, this can extend control of the individual zone climate to the passengers themselves.

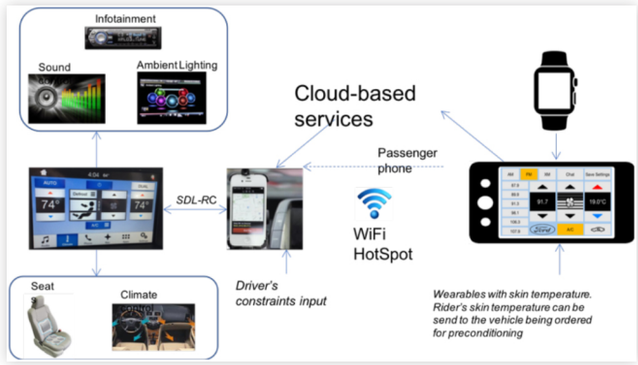
Additionally, since SDL allows digital control of analog features, these controls can be initiated dynamically with a computer algorithm that remembers the rider's preferences. For example, upon booking a ride, the vehicle can be preconditioned to the rider's preferences (set on a previous ride) given the current weather conditions and location as parameters. When the vehicle arrives, it will already be set to the rider's preferences, inclusive of seat positioning, climate, and audio entertainment.

In theory, with SDL, even a remote coordinator could now adjust the vehicle settings. For example, in the case of a ride-sharing fleet, a traffic controller could remotely send pickup and destination routing instructions to the fleet directly to their navigation units. Vehicle data can also be remotely read and tracked in real-time in the same fashion.

Ride-Sharing

Using SDL, we prototyped a ride-sharing application to demonstrate this use case. Specifically, we created a pair of rider and driver mobile apps to simulate the typical user experience in booking a ride-share with any of the major ride-sharing providers in the market.

FIGURE 11 Ride-sharing flowchart.



The layout and flow of these interactions is shown in Figure 11.

Essentially this exposes the vehicle sub-systems as APIs to the cloud to be controlled as a secure service. The driver app acts as a relay station between the vehicle and the rider app, which will control the vehicle's subsystems.

The driver app is paired to the vehicle through Bluetooth to the head unit, as it typically is. The professional driver uses this to facilitate Bluetooth phone calls, music, and running their ride-sharing application to receive rides, run the map, and receive payments. When a driver is ready to embark and receive passengers, he/she marks as such in the app and then can receive requests for rides.

The rider app can be run on any phone with no prior connections to the vehicle. Through a cloud connection, the rider requests a ride and is specifically paired to a vehicle and a driver app. Once the rider is in the vehicle, the driver can start the ride and so lock out other riders from taking the vehicle. By doing so, this creates a secure connection between the two phones to coordinate the ride.

In a similar fashion, our driver app also listens for requests for rides from the rider app. However, in this case, our app also adds a layer to facilitate commands to SDL to programmatically control radio and climate control settings. These commands can be triggered from the rider app, which are then passed to the cloud. The driver app is listening for these commands from the cloud and relays them to SDL to control the vehicle. This is all done through the same secure pairing of the driver and rider apps already done with ride-sharing applications today.

For our prototype, we used an Android driver application, both an iOS and a web-based rider app, and Google Firebase as the cloud service intermediary. While we only implemented climate and radio control, this is extensible to any other functions in the vehicle, both for monitoring data as well as controlling the sub-systems.

Automatic Audio Equalizer

Beyond this, we can also add a feature for ride-sharing to adjust the audio equalizer depending on the occupancy of the vehicle. To optimize the audio, most vehicles have the capability to adjust where the audio system focuses the sound. That is, like a home audio system, audio can be focused in the front, middle, or rear seats.

However, most car owners do not even know this feature exists, let alone take the time to adjust it. While it may seem a trivial setting, in the case of ride-sharing, it can be useful to focus the music towards either the driver or the passengers, particularly with our capability to let the passengers choose their own music. When using the translation app, of course, the sound system should focus toward the driver so they can hear.

Given these examples, there is a use case for setting focus in the audio system in ride-sharing (Figure 12). Using SDL, we can intelligently set this through the app itself whenever it is necessary. For example, when using the translation application, the audio can be focused in the front on the driver. If the passenger requests control of the music, then the audio focus will target the rear where the passenger is seated. In all other cases, we will leave it at the discretion of the driver with the original settings on the vehicle.

Taking this a step further now, with SDL, we also can control the sound equalizer. This is also a setting that few know about or set themselves. With SDL, we can set this programmatically through our mobile app. With metadata about the song being played, whether through NextRadio, streaming radio service providers, or metadata on the local file itself, we can know the genre of the song currently being played. Together with genre-specific presets, the equalizer can dynamically be set from song to song. For example, if listening to music from a local library of favorite songs, the equalizer will set itself automatically to match the current song. This enhances the experience and is a seamless process that can actually be done invisibly to the user (Figure 13).

FIGURE 12 Audio settings for passenger zones.

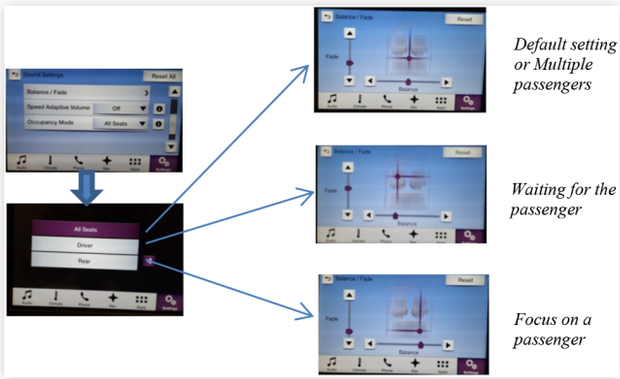
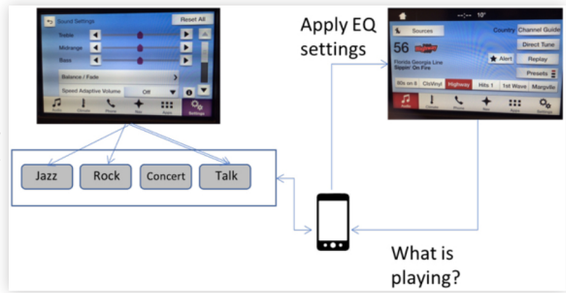


FIGURE 13 Dynamic equalizer flowchart.



Auto High Beam

In luxury vehicles, there is a feature called auto high beam, which automatically toggles the high beam at night. Typically, the driver is supposed to turn off the high beam at night when there is oncoming traffic or vehicles nearby in front to avoid blinding the other drivers. However, when there are no vehicles in front, one can re-engage the high-beam. Sometimes this interaction may take place many times on a given stretch of road.

Auto high beam controls this toggling at speeds of over 25 mph at night using a front-facing camera that detects lights from other vehicles or street lighting. When other vehicles are detected or speeds reduce below a threshold, it turns off the high beam. After a short while, it will re-engage the high beam.

On certain roads with short range line of sight due to curvature or changes in altitude, the brief delay before re-engaging the high beam may be long enough to encounter another oncoming vehicle and thus irritate the other driver. Also in subdivisions, the feature may engage the high beam even when typically, drivers would switch it off. In these instances, drivers will have to manually disengage auto high beam to disable it.

With SDL, the GPS location of these areas where drivers are disabling this feature can be recorded and intelligently managed. With this crowdsourced information, the feature can be automatically disabled in those areas while enabled in other areas.

Summary

This paper summarizes the SmartDeviceLink platform and its capabilities as an Open Innovation Platform to bring novel Connected Car features and services. It is open source, so any automaker who wishes to adopt it as their system will be able to benefit from the work done by other contributors. For developers, features developed for SDL become portable across different vehicles which implement SDL.

We have presented several use cases for this technology, including intelligent climate control, enhanced hybrid radio, and mobility solutions. This is only intended to exhibit some

possibilities. There are many other use cases, from dynamically controlling the audio equalizer to the seats positions, heating or cooling, and even massage seats.

The addition of more connected sensors like the air quality sensors will enable the vehicle to gain more data and insight. With the popularity of wearable devices like the Apple Watch, data like the user's real-time skin-temperature can be used to personalize climate settings using cloud services and machine learning algorithms. Besides learning from sensors, the user can also set preferences to guide the system. These can be implemented through neural networks. Neural networks have been shown to be an efficient and effective approach to implement non-linear models for personalized ambient experience such as climate control [10, 11].

Finally, as vehicles move towards autonomous driving, this type of system becomes a necessity to improve the daily user experience. The main differentiator in a rideshare will be the personalization of the trip, given that there will be no car ownership in the process. This provides an extensible open innovation platform to facilitate implementation of novel features in the areas of infotainment and passenger comfort systems in the vehicle.

Contact Information

Jeffrey Yeung, M.S.
Ford Motor Company
Dearborn, MI
Email: jyeung5@ford.com

Acknowledgments

The authors wish to thank Ben Hussmann from NextRadio for kindly reviewing this paper.

Definitions/Abbreviations

SDL - SmartDeviceLink

HMI - human machine interface

API - application program interface

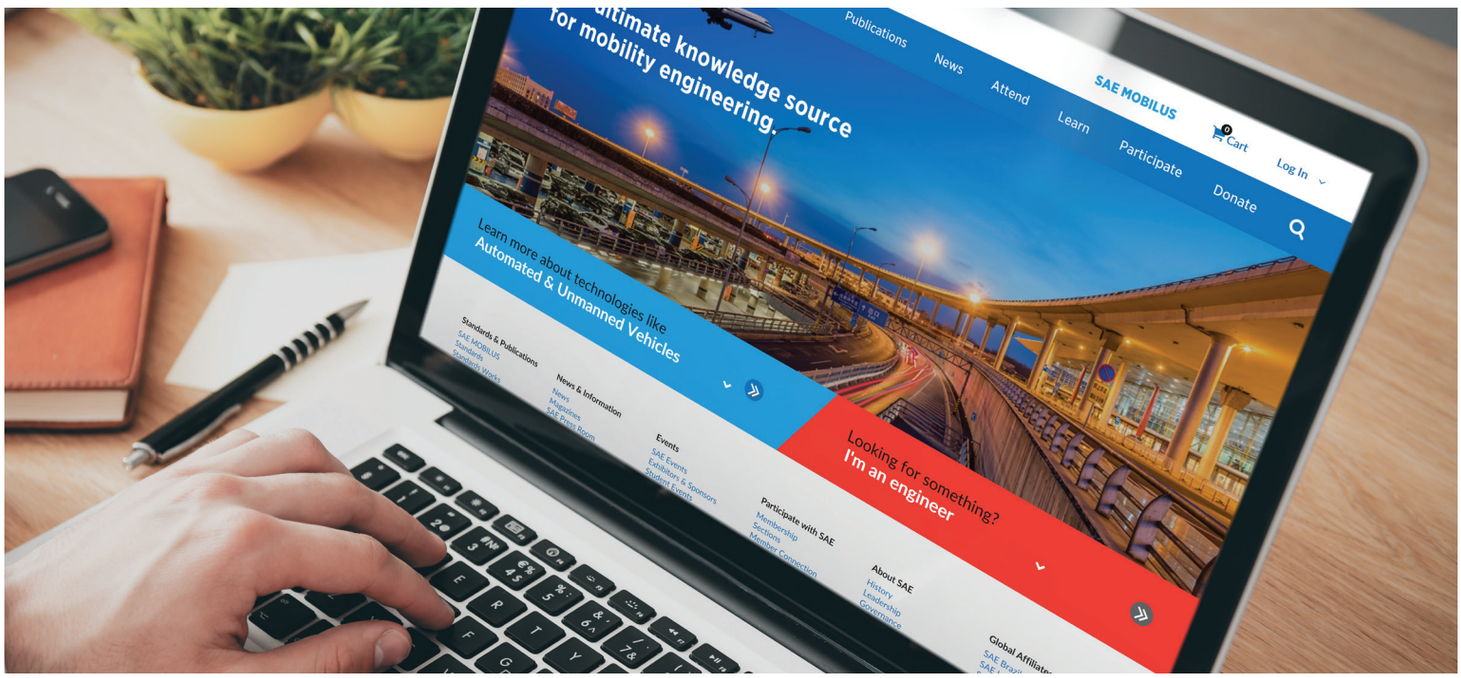
CAN - controller area network

References

1. Chesbrough, H.W., *Open Innovation: The New Imperative for Creating and Profiting from Technology*, (Boston, MA, Harvard Business School Press, 2003).
2. Chesbrough, H.W., *Open Innovation: The New Imperative for Creating and Profiting from Technology*, (Boston, MA, Harvard Business School Publishing Corporation, 2005).
3. Hartmann, P. and Trott, P., "Why Open Innovation is Old Wine in New Bottles," *International Journal of Innovation Management*, 13:715-736, 2009. http://www.hamafarini.com/images/EditorUpload/1.pdf, retrieved Oct. 13, 2016.
4. SmartDeviceLink Consortium. *SmartDeviceLink*, 2017. [Online] Available at: https://smartdevicelink.com/consortium/, accessed Jan. 20, 2017.
5. Ford, *Developer Documentation*, 2016. [Online] Available at: https://developer.ford.com, accessed Oct.12, 2016.
6. Daly, S., *Automotive Air Conditioning and Climate Control Systems*, (Butterworth-Heinemann, 2011), 432.
7. Müller, D., Klingelhöfer, D., Uibel, S., and Groneberg, D.A., "Car Indoor Air Pollution - Analysis of Potential Sources," *Journal of Occupational Medicine and Toxicology* 6:33, 2011.
8. Yang, J., Chen Y., Liu Y., Makke O., Yeung J., Gusikhin O., and MacNeille, P., "The Effectiveness of Cloud-Based Smart In- vehicle Air Quality Management," in *Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference*, October 3-5, 2016, Xi'an, China, 325-329.
9. Gusikhin, O., Makke, O., Yeung, J. and MacNeille, P. "SmartDeviceLink Application to Intelligent Climate Control," in *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Vol. 1, 2016, 234-240, ISBN: 978-989-758-198-4.
10. Thomas, B. and Soleimani-Mohseni, M., "Artificial Neural Network Models for Indoor Temperature Prediction: Investigations in two Buildings," *Neural Computing and Applications* 16(1):81-89, Jan 2007.
11. Kajino, Y., Sugi, H., Kawai, T., Ito, Y., Tateishi, M., Samukawa, K., "Development of Automatic Climate Control with Neural Control," SAE Technical Paper 2000-01-0978, 2000, 1-6.

SAE INTERNATIONAL®

THE NEW SAE.ORG FULLY REDESIGNED WITH YOU IN MIND



USER-DRIVEN
Enjoy a better, more personalized user experience, powered by extensive research, testing, and critical feedback from users like you.

MOBILE-RESPONSIVE
Access the new SAE.org from any device, anywhere.

ORGANIZED FOR EFFICIENCY
We've simplified our site architecture and navigation, making it easier for you to use your current resources and discover new ones.

Not only did we listen to your feedback — we delivered results.

SAE.org has been completely redesigned and enhanced to make it easier for you to get the maximum value from your SAE International relationship.

Launching a new and improved **SAE.org** is just the first step on a roadmap of continued user experience improvement.

P18191191

SAE MOBILUS™ TECHNICAL RESOURCE PLATFORM

Your critical advantage to develop the future of mobility engineering

SAE MOBILUS™ is your destination for mobility engineering resources with instant access to explore, discover, and share more than 226,000 of SAE's current and historical standards, technical papers, eBooks, magazines, and more.

Developed with extensive user feedback, the SAE MOBILUS™ platform features intuitive, easy search and navigation so engineers and students can focus on solving essential problems facing the mobility industry.

THE FEATURES YOUR PEERS REQUESTED

- The customizable dashboard keeps pertinent materials accessible by allowing saved searches, personal document annotations, and custom folder creation
- Dynamic redlining visually represents revision tracking for standards, eliminating a tedious manual process
- Improved, intuitive site search returns focused results with content snippets so you can preview the resource before you download
- COUNTER 4 reporting provides administrators with accurate, timely content usage data to enable informed subscription decisions
- SAE MOBILUS™ Knowledge Hubs provide deep exploration of emerging topics including cybersecurity and advanced manufacturing

For more information

+1.888.875.3976
(U.S. and Canada only)
+1.724.772.4086
(Outside U.S. and Canada)

Visit **sae.org/mobilus**

P18200473