©ISTOCKPHOTO.COM/LIGHTCOME

Autonomous Vehicle Safety: An Interdisciplinary Challenge

Philip Koopman
Carnegie Mellon University
5463 Fair Oaks St., Pittsburgh PA 15217,
mobile: +1 412 260 5955
E-mail: koopman@cmu.edu

Michael Wagner
Edge Case Research LLC

Digital Object Identifier 10.1109/MITS.2016.2583491
Date of publication: 19 January 2017

***Abstract*–**Ensuring the safety of fully autonomous vehicles requires a multi-disciplinary approach across all the levels of functional hierarchy, from hardware fault tolerance, to resilient machine learning, to cooperating with humans driving conventional vehicles, to validating systems for operation in highly unstructured environments, to appropriate regulatory approaches. Significant open technical challenges include validating inductive learning in the face of novel environmental inputs and achieving the very high levels of dependability required for full-scale fleet deployment. However, the biggest challenge may be in creating an end-to-end design and deployment process that integrates the safety concerns of a myriad of technical specialties into a unified approach.

1939-1390/17©2017IEEE

Introduction

A typical prediction of the future of autonomous vehicles includes people being relieved from the stress of daily commute driving, perhaps even taking a nap on the way to work. This is expected to be accompanied by a dramatic reduction in driving fatalities due to replacing imperfect human drivers with (presumably) better computerized autopilots. City governments apparently believe this will happen within 10 years (Boston Consulting Group 2015). But, how to get such fully autonomous vehicles to actually be safe is no simple matter (Luettel 2012, Gomes 2014). We outline a number of areas which present significant challenges to creating acceptably safe, fully autonomous vehicles compared to the vehicles of even a few years ago, with an emphasis on the difficulty of validating autonomy at the scale of a full-size vehicle fleet.

The question is not whether autonomous vehicles will be perfect (they won't). The question is when we be able to deploy a fleet of fully autonomous driving systems that are actually safe enough to leave humans completely out of the driving loop. The challenges are significant, and span a range of technical and social issues for both acceptance and deployment (Rupp 2010, Bengler 2014, Learner 2015). A holistic solution will be needed, and must of necessity include a broad appreciation for the range of challenges (and potential solutions) by all the relevant stakeholders and disciplines involved.

Our work in building safety arguments and run-time safety mechanisms for autonomous ground vehicles has taught us that even understanding what "safe" really means for autonomous vehicles is not so simple. "Safe" means at least correctly implementing vehicle-level behaviors such as obeying traffic laws (which can vary depending upon location) and dealing with non-routine road hazards such as downed power lines and flooding. But it also means things such as fail-over mission planning, finding a way to validate inductive-based learning strategies, providing resilience in the face of likely gaps in early-deployed system requirements, and having an appropriate safety certification strategy to demonstrate that a sufficient level of safety has actually been achieved.

Thus, achieving a safe autonomous vehicle is not something that can be solved with a single technological silver bullet. Rather, it as a coupled set of problems that must be solved in a coordinated, cross-domain manner. The remainder of this paper describes some general problem areas and some of the interactions among them. As everyone gains more experience with the technology, no doubt a few more high-level problems and many more detailed issues will emerge, but this is a starting point for understanding the bigger picture.

Safety Engineering

Let's start by assuming that we already have small-scale deployment of fully autonomous Level 4 (NHTSA 2013) vehicles on the road that are generally well behaved. Thus, we start with an expectation that most vehicles will work well most of the time in everyday on-road environmental conditions. Now we want to deploy at scale. The challenge becomes managing failures that are very infrequent for any single vehicle, but will nonetheless happen too often to be acceptable as exposure increases to millions of vehicles in a fleet.

There is a well-developed body of knowledge about how to make computer-based automotive systems safe, and an even longer history of creating safety critical computer-based systems for trains, chemical process components, aircraft, and so on. We'll first explore challenges from this safety engineering point of view, and then revisit things from the point of view of other disciplines.

Current accepted practice for vehicle computer-based system safety is typically based on a functional safety approach (e.g., the automotive-specific ISO 26262 safety standard). A key issue is that ISO 26262 generally gives a system credit for a human driver ultimately being responsible for safety. But, with a fully autonomous vehicle, the human won't be responsible for driving at all. Relying on autonomy to be completely responsible for vehicle safety without driver oversight is a huge change compared to currently deployed advanced driver assistance systems that largely rely upon the driver to be responsible for vehicle safety. One approach to handling lack of human driver oversight is to set ISO 26262's "controllability" aspect to zero for an autonomous system, which could dramatically increase the safety requirements of a variety of automotive functions compared to today's vehicles. Whether this will work, or even if ISO 26262 can be effectively used as-is to validate autonomous vehicles is an interesting, but open question. (Koopman 2016).

A significant safety certification concern is validating any use by autonomous vehicles of self-adaptive system behavior. (de Lemos 2013) Unconstrained adaptation such as real time learning of new behaviors (Rupp 2010) means that a vehicle can be expected to have a different behavior during operation than was displayed during testing and certification. Current certification approaches are essentially unable to handle that situation, because they required considering all possible system behaviors up-front in the design and validation process. Unless limits are somehow put on adaptation and fully explored during system design, it may be impossible to ensure the safety of such a system at design time because the system being tested won't have the same behavior as an adapted system that is deployed. Formal method approaches may be able to prove properties about adaptive systems, but such proofs come with assumptions which are not necessarily provable or testable, and such approaches currently don't scale well to full-size software systems. Note that by "adaptive," we mean that the system essentially changes its behaviors depending upon its

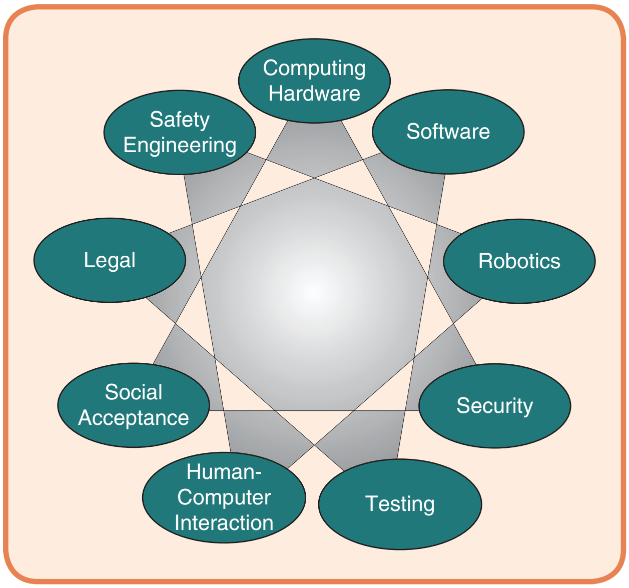

FIG 1 Many different areas require a coordinated, inter-disciplinary approach to ensure safety.

operational history by, for example, using on-line machine learning. This is a much more dynamic range of behaviors than seen in more traditional controls-based systems such as adaptive cruise control, which can be validated using more traditional methods. (Kianfar 2013).

A commonly mentioned approach to hedge system-level safety for highly autonomous systems is to re-engage the driver when there is an equipment failure, providing a human safety net for automation. For example, a human might be taking a nap, and will need time to gain enough situational awareness to take over responsibility for driving. To bridge the human inattention gap, the vehicle will need have to have some sort of fail-operational autonomy capability to carry through until a human can regain control.

Fortunately, cars can typically achieve a safe state in seconds (pull to the side of the road), compared to hours for airplanes (fly to a diversion airport). Thus, an effective safety strategy might be for vehicles to change operational modes to a short duration “safing mission” when a critical primary component fails. The strategy here is that an autonomy system that just has to be smart enough to pull a vehicle over to the side of the road over the course of a few seconds might be designed in a less complex way than a full driving autonomy system.

For example, a safing subsystem might stay in the current lane while coming to a stop, and thereby dispense with sensors and control systems needed for lane change maneuvers. Moreover, a short mission time (seconds, not hours) would be likely to ease reliability and redundancy requirements on the safing subsystem itself. As an added benefit, designing a safe shutdown mission capability may relax the safety requirements on primary vehicle autonomy. If a safing mission is always available, primary autonomy need not be fail operational. Instead, it might be sufficient to ensure that a safing mission is invoked whenever there is a failure of the primary autonomy system, permitting a less-than-perfect primary autonomy system so long as failures are detected quickly enough to invoke a safing mission. Relaxing the safety requirements on primary autonomy (while keeping the vehicle as a whole safe) could potentially offer a dramatic reduction in overall system cost and complexity. (Koopman 2016).

Ultra-Dependable Robots

Making autonomous systems (which are robots) work in a wide variety of everyday driving situations as has been done on current prototypes is a truly significant and impressive achievement. However, making them work well enough to achieve the safety levels required for a fleet of fully autonomous vehicles will take another significant set of achievements. For example, consider a possible goal of making fully autonomous cars as safe per operating hour as aircraft. This would require a safety level of about 1 billion operating hours per catastrophic event. (FAA 1988) Let’s call such a safety target “ultra-dependability.”

A number of challenges in achieving ultra-dependable autonomous vehicles will arise, starting with improving system robustness for messy environmental situations (e.g., dealing with debris, clutter, and sensor noise). In general, it seems implausible to design a system that can handle every possible environmental situation perfectly, especially in the initial stages of deploying a fleet. Thus, it seems desirable to ensure that the systems are not brittle, and in particular have some way of knowing when they are not working properly. In other words, such systems need to be able to self-monitor their confidence in their own proper operation, and be very good at knowing when they don’t know what’s going on. Achieving reliable detection of system degradation will be difficult. A high false-negative rate will lead to vehicles unintentionally operating in an unsafe way. But a high false-positive rate will leave too many cars stranded at the side of the road due to false alarm cyber-angst (hopefully after having performed a successful safing mission in response to the autonomy failure).

Another significant challenge is that machine learning techniques (Domingos 2012) such as classifiers that are widely used in fully autonomous vehicles (e.g., Aeberhard 2015) tend to be based on inductive training approaches rather than a more traditional requirements-based design process. Validating inductive reasoning has long been known to be inherently difficult (Hume 1748), and there does not seem to be a way to make ultra-dependable levels of guarantees as to how such a system will behave when it encounters data not in the training set nor test data set. Autonomous vehicles operate with highly dimensional data, with high-rate data flows from video, LIDAR and radar. It is certain that they will be exposed to real-world data that differs somehow from training and validation data. Machine-learning results often involve decision rules that are generally inscrutable to human reviewers. (Dosovitskiy 2012) Thus, it is difficult to reason about the correctness of the machine learning system's behavior in the face of novel data.

Software

Software safety is a long-standing research topic. (Leveson 1986) Current software safety approaches such as the ISO 26262 "V" process typically assume that high-quality requirements are refined into an implementation. This ultimately produces a chain of evidence that couples final test results back to the safety-relevant system requirements.

With adaptive and machine learning systems, it can be challenging to articulate system requirements in a way that supports the V or other traditional system engineering processes. (Koopman 2016) For example, consider a pedestrian classifier that has been created based on a set of training data. Saying that the system is safe because its accuracy on a validation set is sufficiently high begs the question of whether the system will actually work as it needs to when confronted with messiness of the real world.

What really matters is that the machine learning validation set has to be comprehensive enough to make sure that there are no gaps in system behavior. In terms of the "V" model, the training set is the closest we have to system requirements, and the validation set is the closest we have to a testing plan. But, knowing that the training and validation sets are good enough is not so easy. How can we be sure that the edge cases and subtle behavioral interactions that are likely to affect safety are actually learned by the system?

Hopefully the training set and validation set are extremely comprehensive, and have covered all conceivable operating scenarios. But, as with more traditional systems, there is still the issue of inconceivable operating scenarios that might still happen (those famous "unknown unknowns"). Moreover, there is the possibility that some new operating scenario would seem to be ordinary to a person (and thus not included in the test data set), but in fact is exceptional in some way to the machine learning algorithm, potentially causing unanticipated system behaviors.

Basing a safety argument on the sufficiency of training and validation data also potentially makes the system that collects this data safety critical. After all, safety for such systems ultimately hinges on the accuracy of the training and validation data collection. This might, for example, lead to a need for the data collection system to be developed according to safety critical software standards, with attention paid to reducing hazards such as unintended bias or distortion in the collected data.

A potential solution to the problem of validating machine learning is to separately and independently define what "safe" operation means. This separate set of safety requirements could be imposed as a set of independently monitored behavioral requirements on the autonomous vehicle's autonomy. (Kane 2015) Such monitoring can be used during validation, on-road testing, and perhaps deployment to ensure that the vehicle does not exhibit unsafe behaviors, even if there are gaps or glitches in machine learning systems.

Computing Hardware

Even with the use of a safing mission strategy, an obvious hardware challenge is creating ultra-low cost hardware with safe failure behaviors. This requires creating a combined hardware/software architecture that employs redundancy properly. (Hammett 2001) As an example of progress in this area, chip-makers have introduced computing chips with dual cores that can provide at least partial redundancy for computations. However, a more thorough approach to ensuring sufficient redundancy and fault tolerance will likely be required. There is a long history of designing such systems for aerospace and other safety critical applications, providing rich background knowledge. (Randell 1978).

A more subtle, but critical, hardware challenge is the issue of latent fault detection. Completely fault-free redundancy is typically assumed at the beginning of each mission when performing reliability calculations. Any undetected fault undermines the benefits of redundancy dramatically. Even a percent or two of gaps in self-diagnosis has dramatic implications for achievable reliability. For example, achieving only 95% test coverage can reduce achievable reliability of redundant automotive system by orders of magnitude. (Honeywell 1995) The reason for this is that undiagnosed failures can accumulate for the entire working life of the vehicle, so the probability of experiencing multiple independent undiagnosable failures during the life of a vehicle is quite high compared to the probability of multiple failures occurring during a single driving mission for diagnosed parts of the system. Thus, it will be important to create chips that can be self-tested before each driving cycle with an extremely high level of diagnostic coverage.

Testing

Traditional, pre-computer, vehicle safety practices and regulations have emphasized vehicle-level testing. More recently, autonomous vehicle prototype projects have typically emphasized the importance of on-road testing. (e.g., Urmson 2008, Levinson 2011, Broggi 2013, Ziegler 2014, Aeberhard 2015, SAE J3018) However, it is well known that a testing-only approach is insufficient to ensure the safety of even non-autonomous software-based critical systems. (Butler 1993) More than just testing is required for full scale deployment of any safety-critical automotive software. Nonetheless, thorough testing is still required.

Rigorous testing of autonomy runs into numerous issues. Primary among them is that in the "V" development model, testing compares a rigorously defined design document against a system to determine if the system matches its design and requirements. For probabilistic systems such as planners (Geraerts 2002), the behavior of the system is expected to differ on each test run even for essentially identical initial conditions. Moreover, small changes in initial conditions can result in large changes in system behavior. Therefore, the testing oracle (something that predicts what a correct test result would be) will need to support abstract test result analysis rather than the more traditional technique of feeding some specific values into a piece of software and expecting a particular, exact result from a computation. (Feather 2001).

Inductive-learning systems are even more challenging to test, because there is no design as such, and therefore no starting point for building a test oracle. Rather, as discussed, there is a set of training data and a set of validation data. However, even with comprehensive sets of data, it is unclear how to ensure that the machine learning system has trained on the essential characteristics of the training data instead of coincidental correlations.

Machine learning techniques are quite sophisticated, but a typical argument for correctness ends up being statistical in nature. Although some might assume that a system's quoted accuracy statistics hold over any conceivable input, they do in fact only measure performance on the test data, and may be wildly different on different data sets that they encounter in the wild (e.g., Nguyen 2015). Any claims of safety have to argue that there are no safety-relevant situations missing from the training and testing data sets. This can clearly work well to attain moderate dependability via brute force approaches (e.g., 99.9% accuracy). However, it is unclear how to ensure ultra-dependability for machine learning algorithms.

Consider an example system that performs 10,000 operations per hour for vehicle control (about 3 per second), and a fleet of one million vehicles. Testing to validate a particular failure rate requires processing more test cases than the desired failure rate. Thus, ensuring less than one catastrophic failure per hour for this fleet would require passing significantly more than 10 billion *representative* test cases. This estimate additionally demands satisfying the optimistic assumption that failures are independent and that the data set is actually representative of everything a fleet of vehicles will encounter. It also might be too modest a safety target, because if each vehicle in the fleet is driven only one hour per day, that would still permit a daily catastrophic vehicle failure.

In other words, developers would have to test for more hours of exposure than they claim as the fleet failure rate. Collecting that much test data will clearly be a significant challenge, as would be validating that amount of synthetic data as being realistic in all respects. (Kalra 2016) If the goal is an aviation-style one of making sure your entire fleet will not experience a catastrophic software failure during its operational life (FAA 1988), then it is difficult to see how this can be done via testing alone.

Another dimension of testing is fault injection and failure management. With full-scale vehicle deployment will come daily instances of vehicle equipment failure simply due to the huge numbers of vehicles in the fleet. If vehicle controllability is fully the responsibility of an autonomy system, it will be necessary to characterize what happens when the autonomy system has to deal with a vehicle experiencing a tire blow-out, sensor failure, actuator failure, and even an autonomy algorithm failure across the full spectrum of operational conditions.

Security

Automotive computing security has been receiving increased attention, and shows no signs of becoming an easy problem. Clearly, autonomous vehicles will have to deal with security too. (SAE J3061).

In addition to attacks on specific vehicles, security measures will need to encompass system-level attacks and failures. In particular, it may be problematic to blindly trust in the security of other vehicles or even roadside infrastructure when performing optimized autonomous maneuvers such as free-flowing intersection traffic. For example, encrypting vehicle-to-vehicle communications may help with the security of inter-vehicle coordination messaging. But what if the vehicle you are securely communicating with has been subverted and is providing maliciously incorrect information? What if someone has physically broken into a roadside infrastructure computer and reprogrammed it, or kills power to a set of roadside infrastructure support systems in a coordinated attack?

At the very least, it seems prudent to ensure that every stand-alone vehicle has the ability to realize when it is being fed incorrect or malicious external information, detect that an attack is occurring, and perform a safing mission if it cannot continue full operation in the face of the attack.

Human-Computer Interaction

As autonomous vehicles supplant human drivers, automation's ability to communicate and cooperate with people will become more important. The risks of human supervisor inattention in systems with nearly—but not quite—full autonomy should be self-evident. But even fully autonomous vehicles will at least need to make sure that the occupants feel that the vehicle's behavior is safe if they are to build customer trust, and will need to learn how to anticipate the behaviors of other vehicles as well. (Gindele 2015) While some might say that customer trust is not strictly speaking a safety issue, it is a vital issue for technology adoption, and thus indirectly safety if autonomous vehicles fulfill their promise of saving lives on the road. Other human-computer interaction issues form significant challenges as well.

Autonomous vehicles will have to interact with the human drivers of other vehicles. A car that is too polite or too rude will disrupt traffic flow at the very least, and perhaps indirectly cause more significant safety problems. Cutting human drivers out of the picture is likely to take many years while market penetration of fully autonomous technology ramps up. Even when the day comes that all cars are fully autonomous, the road will still be home to human drivers of bicycles, motor scooters, horses, farm equipment, and so on. Many of these human road users will be reluctant or unable to follow normal traffic rules and expectations for passenger vehicles. Even if there are dedicated lanes initially to ease deployment (Shladover 2009), over time it seems likely that there will be tremendous public pressure to spread autonomy to mixed autonomous/human driven vehicle scenarios. Thus, it seems likely that mixed traffic scenarios will have to be dealt with eventually.

In any urban environment, autonomous vehicles will also have to act with pedestrians, who are unlikely to follow traffic rules at all times. The vehicle will need to react safely to ill-behaved pedestrians, unpredictable children, and pranksters.

An underlying need across many areas will be for the automation to behave in a way that is comprehensible to humans. By this, we mean that it a human should be able to readily perceive what the automation intends to do and why it exhibited some behavior. This will be important for areas such as human vehicle interaction safety (Is the vehicle stopping to let me cross? Or does it not even notice me? How do I make something akin to eye contact with an autonomous vehicle to make sure it won't run me over?); testing coverage (did the vehicle stop because it saw a child in the crosswalk, or because some blowing leaves confused it?); and design comprehension (where is the part of the machine learning system that knows what it means to see a child, and does it cover all children or just the ones in the training set that all happened to be pointing at the fancy-looking data collection vehicle?).

Legal

A significant early issue in deploying these vehicles will be dealing with legal issues of liability. (Marchant 2012) When a fully autonomous vehicle is involved in a mishap, it may well be that the vehicle passenger is justifiably inattentive (perhaps even asleep). Vehicle data logs might be the primary source of information available as to what happened in a mishap. However, data from a vehicle that has malfunctioned cannot be blindly trusted. After all, if the vehicle caused a mishap due to a malfunction, why should we assume that any data from that malfunctioning system is accurate? While one can envision a satisfactory independent data recording system for mishap forensics, such a system has to be intentionally designed in a suitable manner. It would be no surprise if currently deployed Event Data Recording devices need to be rethought to provide adequate data for this purpose.

A key liability issue will be determining who is ultimately responsible for proper vehicle operation. Is it the occupant who got into a rental vehicle even though it had sensor damage that should have been noticeable to a layperson? Is it the vehicle manufacturer who trusted a faulty third party training data set? Is it the mechanic who mistakenly installed a slightly incompatible version of replacement sensor software? Is it the mapping update service that was too slow to record a bridge wash-out? Is it the operating system vendor who didn't deploy a security patch fast enough to prevent a malicious mishap? While some of these issues might be purely legal, resolving many legal issues will require an adequate technological foundation to build upon. While much experience is being gained with pilot deployments (Parent 2013), the legal questions surrounding autonomous vehicles are still very much open, as are a number of more general legal related topics. (Transport Styrelsen 2014).

Social Acceptance

The social acceptance of autonomous vehicles will no doubt be a complex process. (Anderson 2014) A primary incentive to adoption is the expectation that autonomous vehicles will, on the whole, be safer drivers than people. However, it is unrealistic, especially early on, to assume that this will mean zero mishaps. The somewhat straightforward cases to reckon with will be those in which avoiding a collision is physically impossible (e.g., a tree falling essentially on top of a car in a storm). But not all situations will be so easy to judge. We will have to address whether the standard for autonomous safety should be whether it is better than an excellent human driver, or merely a *typical* human driver, and exactly how such a "typical" driver might be characterized. Especially tricky situations will be ones in which an ordinary human driver would have had a good chance of avoiding a mishap (at least in the view of a layperson driver sitting on a jury), but the autonomous vehicle crashed.

Establishing an actuarial basis for insurance purposes is often discussed as a significant hurdle for autonomous vehicles. But in the end perhaps this can resolved by suitable application of monetary reserves. For example, if an autonomous vehicle vendor acts as a reinsurer, then it can set an arbitrarily low re-insurance rate, subsidizing technology adoption, and then fine-tune both the rates and the vehicle design as actual loss rates become apparent.

Conclusion

In the end, there will have to be a safety certification strategy of some sort for fully autonomous vehicles. This strategy must address the cross-disciplinary concerns of safety engineering, hardware reliability, software validation, robotics, security, testing, human-computer interaction, social acceptance, and a viable legal framework. In each of these areas there will be edge cases and subtle tradeoffs to be resolved, and likely significant cross-coupling tradeoffs between areas. Some of these tradeoffs are already being explored with real-world prototype deployments, while others will only become pressing issues when the deployed fleet scales up. This paper in particular points out the challenge with validating machine-learning based systems to the ultra-dependable levels required for autonomous vehicle fleets, and how that challenge relates to a number of other areas. A long-term task before us is updating accepted practices to create an end-to-end design and validation process that addresses all these safety concerns in a way that is acceptable in terms of cost, risk, and ethical considerations.

About the Authors

Philip Koopman has a Ph.D. in Computer Engineering from Carnegie Mellon University. As an Electrical and Computer Engineering faculty member at Carnegie Mellon University he specializes in software safety and dependable system design. He has affiliations with the Robotics Institute and the Institute for Software Research. He is a Senior Member of IEEE and ACM. E-mail: koopman@cmu.edu. Postal Mail: Prof. Philip Koopman, CMU/ECE HH A-308, 5000 Forbes Ave., Pittsburgh, PA 15213.

Michael Wagner has an M.S. Degree in Electrical and Computer Engineering from Carnegie Mellon University. He is the CEO and co-founder of Edge Case Research, LLC, which specializes in software robustness testing and high quality software for autonomous vehicles, robots, and embedded systems. He also has an affiliation with the National Robotics Engineering Center. E-mail: mwagner@edge-case-research.com Postal Mail: Michael Wagner, Edge Case Research LLC, 100 43rd Street, Suite 208, Pittsburgh, PA 15201.

References

[1] M. Aeberhard, et al., “Experience, results and lessons learned from automated driving on Germany’s highways,” *IEEE Intell. Transp. Syst. Mag.*, pp. 42–57, Spring 2015.

[2] J. Anderson, et al., *Autonomous Vehicle Technology: A Guide for Policymakers. Santa Monica CA*: RAND, 2014.

[3] K. Bengler, et al., “Three decades of driver assistance systems,” *IEEE Intell. Transp. Syst. Mag.*, pp. 6–22, Winter 2014.

[4] Boston Consulting Group. “*Self-driving vehicles in an urban context press briefing*,” World Economic Forum, Nov. 2015.

[5] Broggi, et al., “Extensive tests of autonomous driving technologies,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1403–1415, Sept. 2013.

[6] F. Butler, “The infeasibility of experimental quantification of life-critical software reliability,” *IEEE Trans. SW Engr.*, vol. 19, no. 1, pp. 3–12, Jan. 1993.

[7] R. de Lemos, et al., “Software engineering for self-adaptive systems: A second research roadmap,” *LNCS*, vol. 7475, pp. 1–32, 2013.

[8] P. Domingos, “A few useful things to know about machine learning,” *CACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012.

[9] A. Dosovitskiy and T. Brox, “Inverting convolutional networks with convolutional networks,” *CoRR*, vol. abs/1506.02753, 2015.

[10] “FAA, System Design and Analysis,” AC 25.1309-1A, June 21, 1988.

[11] M. Feather and B. Smith, “Test oracle automation for V&V of an autonomous spacecraft’s planner,” AAAI Tech. Report SS-01-04, 2001.

[12] R. Geraerts and M. H. Overmars, “A comparative study of probabilistic roadmap planners,” in *Proc. Workshop Algorithmic Foundations Robotics*, 2002, pp. 43–57.

[13] T. Gindele, S. Brechtel, and R. Dillmann, “Learning driver behavior models from traffic observations for decision making and planning,” *IEEE Intell. Transp. Syst. Mag.*, pp. 69–79, Spring 2015.

[14] L. Gomes, “Hidden obstacles for Google’s self-driving cars,” *MIT Technol. Rev.*, Aug. 28, 2014.

[15] Hammett. “Design by extrapolation: An evaluation of fault-tolerant avionics,” in *Proc. IEEE 20th Conf. Digital Avionics Systems*, 2001.

[16] Honeywell. “Malfunction management activity area report for AHS health management,” AHS Precursor Task E report, DoT FHA Publication, FHWA-RD-95-047, Nov. 1995.

[17] D. Hume, *An Enquiry Concerning Human Understanding. New York*: Collier, 1910.

[18] *Road Vehicles: Functional Safety*, ISO Standard 26262, 2011.

[19] N. Kalra and S. M. Paddock. (2016). *Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? Santa Monica, CA*: RAND [Online]. Available: http://www.rand.org/pubs/research_reports/RR1478.html

[20] K. Datta, and Koopman, “A case study on runtime monitoring of an autonomous research vehicle (ARV) system,” *RV*, 2015.

[21] R. Kianfar, P. Falcone, and J. Fredriksson, “Safety verification of automated driving systems,” *IEEE Intell. Transp. Syst. Mag.*, pp. 73–86, Winter 2013.

[22] Koopman and Wagner, “Challenges in autonomous vehicle testing and validation,” in *Proc. SAE World Congress*, Apr. 2016.

[23] P. Learner, “The hurdles facing autonomous vehicles,” *Automobile*, June 22, 2015.

[24] Leveson, “Software safety: Why, what, how,” *ACM Comput. Surv.*, pp. 125–163, June 1986.

[25] Levinson et al., “Towards fully autonomous driving: systems and algorithms,” in *Proc. IEEE Intelligent Vehicles Symp.*, June 5–9, 2011, pp. 163–168.

[26] T. Luettel, M. Himmelsbach, and H. -J. Weunsche, “Autonomous ground vehicle: Concepts and a path to the future,” Proc. IEEE, pp. 1831–1839, May 2012.

[27] G. Marchant and R. Lindor, “The coming collision between autonomous vehicles and the liability system,” *Santa Clara Law Rev.*, vol. 52, no. 4, pp. 1321–1340, 2012.

[28] A. Nguyen, J. Yosinski and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *Proc. IEEE Computer Vision and Pattern Recognition*, 2015.

[29] NHTSA. (2013, May). Preliminary statement of policy concerning automated vehicles [Online]. Available: http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated_Vehicles_Policy.pdf

[30] M. Parent, et al., “Legal issues and certification of the fully automated vehicles: Best practices and lessons learned,” CityMobil2 Rep., June 11, 2013.

[31] Randell and L. Treleaven, “Reliability issues in computer system design,” *ACM Comput. Surv.*, pp. 123–165, June 1978.

[32] D. Rupp and A. King, “Autonomous driving: A practical roadmap,” SAE 2010-01-2335.

[33] “Guidelines for safe on-road testing of SAE Level 3, 4, and 5 prototype Automated Driving Systems (ADS),” SAE J3018, Mar. 2015.

[34] “Surface vehicle recommended practice: Cybersecurity guidebook for cyber-physical vehicle systems,” SAE J3061, Jan. 2016.

[35] S. Shladover, “Cooperative (rather than autonomous) vehicle-highway automation systems,” *IEEE Intell. Transp. Syst. Mag.*, pp. 10–19, Spring 2009.

[36] D. Silver, J. Bagnell, and A. Stentz, “Active learning from demonstration for robust autonomous navigation,” in *Proc. IEEE Conf. Robotics and Automation*, May, 2012.

[37] Transport Styrelsen. “Autonomous driving pilot study,” Swedish Transport Agency, Tech. Rep. TSG 2014-1316, 2014.

[38] C. Urmson, et al., “Autonomous driving in urban environments: Boss and the urban challenge,” *J. Field Robot.*, pp. 425–466, 2008.

[39] J. Ziegler, et al., “Making bertha drive: An autonomous journey on a historic route,” *IEEE Intell. Transp. Syst. Mag.*, pp. 8–20, Summer 2014.

ITS