The AAAI-17 Workshop on
Artificial Intelligence for Connected and Automated Vehicles
WS-17-03

Preemptive Detection of Unsafe Motion Liable for Hazard

Masataka Nishi
Hitachi Research Laboratory, Hitachi Ltd., Japan.
masataka.nishi.en@hitachi.com

Abstract

Establishing a safety standard for autonomous vehicles operating in open and dynamic environment is a challenge. As collisions are inevitable in over-constrained situations, we focus on deciding the liability for a hazard. Our insight is that hazards caused by malfunctions of autonomous vehicles result from loss of functional integrity. Design defects may leave it unnoticed, or the real-world may make integrity-preserving motion infeasible. Guarantee of functional integrity in an observable way at run-time is indispensable for revealing defects by using formal root-cause analysis, and for supporting safety claims by dismissing unreasonable doubts about design defects. From a practical standpoint, we attempt to formalize a verification problem that consists of a novel criterion for determining liability for hazard, a safety claim comprised of confirmed observable states, and assumptions underlying the safety claim. We propose a run-time scheme of monitoring events that may lead to violations of the assumptions and a precursor to root-causes leading to loss of functional integrity and consequent hazards. We formulate a means of preemptively detecting unsafe motions liable to be hazardous as satisfiability problem within the framework of an adversarial motion planning subject to assumptions on maneuverability of movers. A numerical study shows that the run-time scheme using non-linear programming (NLP) encoding is viable in a real-world setting.

Introduction

A major step toward establishing a safety standard for fully autonomous vehicles is getting a consensus on dividing the liability for hazard among a variety of stakeholders (NHTSA 2013; Anderson 2014; NHTSA 2016); passengers, vehicle manufacturers, the regulator, and auto insurance firms. They have diverse levels of technical expertise in the capability and limitations of autonomous systems, and each one has limited control over situations resulting in hazards. The standard should give a rigorous but simple guideline for consistently dividing each party's liability in a way that one is never imposed an unfair disadvantage or obligation. Allocation of liability will be written in legal terms and conditions for use that the passengers need to agree with, in the functional safety requirements that the manufacturers must comply with, in a certification procedure that the regulator imposes to safeguard the passengers and pedestrians on streets, and in insurance contracts specifying conditions of awards for economic loss or injury. We can hardly state what constitutes safe vehicle motion in a few words of formal specification, as difficult situations are diverse or unknown at an early stage. Yet, we can easily state evidently bad consequences in much fewer words. We should not discuss safety, but rather define a hazard that each party is legally required to avoid responsibly. Safety claims should be supported based on the infrequency of bad consequences. By adding new ones that we experienced and by avoiding it by design, we can incrementally lower the frequency of acknowledged bad consequences more efficiently than fixing a vehicle on a patchwork basis. The frequency per million kilometers of drive is a quantifiable measure of safety (Kalra 2016). Four practical hurdles implies a technical direction.

The first hurdle is that each stakeholder has compartmented access to the data necessary to identify the root-cause of a hazard and to decide which party is liable. Current copyright law supports limiting access to the source code of the control software. It has successfully protected such software from unauthorized reproduction or modification, and yet has maintained competitive advantage, consumer protection, desired vehicle performance, and compliance with safety standards and regulatory requirements. However, combined use of the copyright law and the product liability law (Villasenor 2014) unintentionally puts an injured party in a weaker position who attempts to establish a negligence claim regarding hazards when the software is involved. In principle, the burden of proof lies with an accuser. Thus, the injured party may have to endure an unacceptably costly legal process for getting sufficient access to the source code and to hire a third party auditor on its behalf as an expert witness who can analyze the recorded run-time log and reveal the actual causation of the software and the hazard. If a defect of the control software could be actually the root-cause, then the injured party has limited choices other than depending on a prospective strict liability claim or specific consumer protection statutes that some countries adopt to mitigate the burden of proof by the injured party. Thus, a criterion for deciding liability for hazards should be based on only observable states that the injured party can confirm. Use of signaled intent of action and unobservable internal program states should be dismissed.

The second hurdle is the difficulty of deciding when and why the safety claim became invalid. An accident investigator tries to find the root-cause from the recorded run-time log, reproduce the hazardous situation wherein the safety claim became invalid, and decide liability. Yet, the recorder can store only measurable part of the world state supplied from inherently imperfect onboard sensors. Sensor data can be corrupt or transiently unavailable. It is also difficult to judge whether unmeasurable part of the world state estimated using an imperfect world model is correct or not. The record could represent the world state incorrectly and there is no clue of checking data integrity (Nishi 2016a). If the hazard could result from the loss of data integrity, backward reasoning of causation formulated as MaxSAT (Manu 2011; Griesmayer 2007; Fey 2008) can lead to an ill-posed inverse problem and the root-cause is not uniquely determined. Otherwise, the manufacturer can identify the root-cause, or dismiss an unreasonable claim of design defects based on an evidence that the software cannot produce a program state that is consistent with the record and that would impair an imposed safety claim. The regulator can improve limited observability of the program states by requiring the manufacturer to record essential states that could assist in automated root-cause analysis. But such analysis may fail, if the supposed root-cause ends up to be instead due to unobservables; an unrecorded interplay between activation of embedded mechanisms (ESC and ABS) and the autonomous capability. Without revealing the causation between the root-cause and the consequent hazard, the manufacturer can hardly justify that a software fix could reinstate a safety claim. After all, a safety claim should also consist of confirmed observable states and should not depend on the internal program states. These considerations invalidate a whole class of adaptive techniques that employ a dedicated internal mechanism for supporting safety claims (Jacklin 2008).

The third hurdle is that each stakeholder needs to reserve reasonable control over the situation of imminent hazard. But the capacity-limited regulator has to rely on measures for certification at the pre-market stage and for investigation after a number of incidents have accumulated. The injured party is an evidence of limited control over the situation. Passengers of fully automated vehicles can at most only be responsible for stopping the vehicle. Control software can lose control in severe situations that invalidate assumptions and in which hazards are inevitable. Thus, the vehicle needs to detect a precursor to a hazard at run-time and act preemptively. The detector should be verified at pre-market stage.

The last hurdle is that supporting a valid formal safety claim requires enclosing a subset of all possible situations with sound assumptions in which a certification examiner can look for one satisfying the criterion for hazard. This is mandatory for dismissing an unreasonable doubt regarding design defects by verifying that there exists no such situation in the enclosed subset. The subset may consists of unwarranted assumptions of the real world. Warranted assumptions are invariants describing the environment. Unwarranted assumptions that we inevitably depend on are ones regarding the model of the environment, formal interpretation of traffic rules including how traffic lights control the right-of-way, and a belief that movers comply with the traffic rules. Unfortunately, they are sometimes violated. Also, the urban environment is filled with obstacles. Overlapping spatial constraints can force transient violations of the assumptions. Yet, the regulator needs to endorse the validity of such a dedicated set of assumptions for certification. Initially, the safety claim using such assumptions is incomplete at best since it inevitably leaves residual situations where the safety claim becomes invalid due to the violation. It raises a certification challenge (Rushby 2008). At the very least, the decision of liability for hazard due to assumption violations needs to be compatible with our current practice.

Assumption violation is a new kind of design fault often implicitly acknowledged. If infrequency of assumption violations is based on a certain stochastic process similar to that of faults, then a dedicated claim in an accident insurance contract should be provided to award for loss or injury. Here, the concept of motion prediction (Ziebart 2009) cannot be used, as the mover's intent of action is neither observable nor controllable. The validity of the mover's predicted action is an unwarranted narrow assumption that can be violated at the mover's will and thus cannot be supported on a sound statistical basis. Instead, we need a better assumption to cut out an over-approximation of possible situations.

In this paper, we explore a viable way of supporting safety claims subject to four hurdles described above. We argue that in reality collision events are inevitable in over-constrained situations, despite that a majority of studies presume collision avoidance as the right criterion(Ziebart 2009; Montemerlo 2008; Urmson 2008; Wongpiromsarn 2009). First, we propose a novel criterion for deciding liability for hazard that takes contribution to hazard into account. Second, we propose a safety claim that functional integrity and loss of it must be detected immediately. It should depend only on confirmed observable information, and an endorsed set of assumptions. Third, we propose a run-time scheme that monitors for both violations of assumptions and a risk of liability for hazard. The latter sort of monitoring is a problem of deciding if the situation could potentially evolve adversely and the criterion becomes satisfiable. Fourth, we examine the risk of a hazard by determining the satisfiability of an adversarial motion planning problem from the mover's standpoint subject to an assumption on the bounded maneuverability of the mover. If the problem is satisfiable, then it points to the risk of getting into a hazardous situation if the mover attempts computed adversarial motion. Otherwise, any doubt that the vehicle attempted an unsafe motion is dismissed. The scheme is compatible with existing negligence scheme (Villasenor 2014) that instructs a driver to foresee the risk of a hazard and responsibly prevent it. We propose dedicated encoding techniques using a non-linear programming (NLP) solver IPOPT. The scheme is viable as we can solve the problem in a real-world situation in 100 ms.

Background

Functional Integrity

Motion of an autonomous vehicle is specified with four classes of constraints on a temporal series of the vehicle

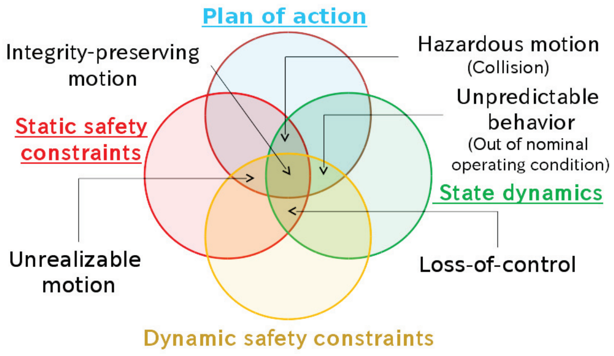
Plan of action
Integrity-preserving motion
Hazardous motion (Collision)
Unpredictable behavior (Out of nominal operating condition)
Static safety constraints
State dynamics
Unrealizable motion
Loss-of-control
Dynamic safety constraints

Figure 1: Side effects of loss of functional integrity.

states; a plan of action, static and dynamic safety constraints, and temporal constraints originating from state dynamics within a foreseeable time horizon. Functional integrity is the status of the vehicle such that all of the constraints are satisfiable (Nishi 2014). Integrity-preserving motion is an instance that satisfies all of them. Figure 1 shows a variety of silent hazards resulting from loss of functional integrity.

The plan of action encodes a desirable motion formulated as constraints on a temporal series of states of the vehicle. If the plan is unsatisfiable, the vehicle goes out of control. The static safety constraints typically represent nominal operating conditions that the manufacturer provides a warranty. If unsatisfiable, then the vehicle is forced to operate beyond the warranty and there is a risk of hazard due to an unpredictable behavior that the manufacturer is not liable for. Constraints regarding state dynamics encode limits on the motion that originates from the mechanical constraints of the vehicle. Attempting any motion that violates this constraint is unrealizable by design and never occurs in the real-world. Any violation immediately implies data corruption and a claim of functional integrity can no longer be supported. Dynamic safety constraints typically represent negation of the hazard criterion. Spatial separation between the vehicle and nearby movers encodes the notion of collision avoidance. If unsatisfiable, then a hazard occurs. We also develop a model of the environment that puts assumptions on possible situations that includes reasonable motions obeying existing traffic rules. As validity of the assumptions is controlled by the environment, functional integrity can be impaired in an adverse situation where some of the assumptions are violated. Even if at least one of the assumptions becomes unsatisfiable, the risk of unintended consequence is exposed by design. A mechanism of detecting a conflicting subset of constraints would remove the risk of unintended consequence that would otherwise go unnoticed.

Criteria of Liability for Hazard

According to a currently accepted practice in situations involving inevitable collisions that the vehicle is not liable for, liability for hazard depends on which party violated an assumption that the traffic rules instruct one to follow. As a decision that a hazardous event occurred is a function of a situation at a time, the criterion of liability should not depend on a temporal series of situations, but depend on the other mover's state at the time. This proposition is reasonable because we need to impose a rule in a way that the vehicle can find an integrity-preserved motion at any situation at any time for supporting a formal safety claim. This means that selection of the rule itself should not permit an unreasonable claim of a design defect on the basis of an argument that the mover can attempt a counter-strategy that keeps the vehicle in a situation where the safety claim is unconditionally unsatisfiable. At the time of a collision when the relative distance between the vehicle's position $\mathbf{r}^s(t)$ and the mover's position $\mathbf{r}^m(t)$ is closer than ϵ, we can determine if a criterion $haz(\mathbf{r}^s(t), \mathbf{r}^m(t))$ in (1) is satisfied by using a threshold on the relative velocity $\delta > 0$.

$$(\mathbf{r}^s(t) - \mathbf{r}^m(t))^2 \leq \epsilon^2 \wedge \frac{d\mathbf{r}^s(t)}{dt} \frac{(\mathbf{r}^m(t) - \mathbf{r}^s(t))}{\|\mathbf{r}^m(t) - \mathbf{r}^s(t)\|} \geq \delta \quad (1)$$

The first term in (1) is the one that is currently used. The second one represents a contribution of the vehicle's unsafe motion to the collision. The vehicle needs to perform a motion $M^s \equiv \{\mathbf{r}^s(t + \alpha T_p) | 0 \leq \alpha \leq 1\}$ within a temporal horizon T_p such that (1) is unsatisfiable at any time subject to an assumption on the mover's motion $M^m \equiv \{\mathbf{r}^m(t + \alpha T_p) | 0 \leq \alpha \leq 1\}$. Criterion (1) states that an unsafe motion that we judge to be at least partially liable for hazard is that the vehicle did not try evasive move at the time of contact and let the second term of (1) get larger than δ. This criterion covers the first situation in Section 2.2. To cover the second one, we need to extend (1) and combine it with a spatial constraint on $\mathbf{r}^s(t)$ and $\mathbf{r}^m(t)$ regarding the right of way. Granting the right to a relevant roadway is controlled by a traffic signal. Because the traffic light changes colors abruptly, the extended part of the criteria could be transiently violated. To reasonably handle a transient violation, the spatial constraint should not be a predicate added to (1) but a guard condition of (1).

Assumptions Regarding Environment

To conclude that criterion (1) is unsatisfiable in a real-world setting, we need to assume a reasonable and verifiable bound on the mover's motion M^m and other states relevant to a situation. A variety of movers such as cars, bikes and pedestrians have their physical limits on their maneuverability and thus warranted. A car cannot accelerate beyond a kinetic coefficient of friction of 0.7 g on a dry roadway, and a mechanical design of the car constrains steering maneuver. A pedestrian cannot move beyond Mach 3. Aside from the limit, our driving practice depends on more moderate assumptions on their maneuverability. Cars rarely accelerate beyond 0.2 g for comfortable drive, and this assumption is statistically supported. The assumptions based on physical limits are never violated but are pessimistic, while the moderate one can be violated if the mover could attempt a statistically unlikely move. Assumptions originating from the traffic rules are unwarranted and thus moderate. Any decision scheme using unwarranted assumptions needs to pass a risk-utility test, and all stakeholders need to reach a consensus when justifying the scheme and exempting from a subject of potential design defects. If the unwarranted assumptions are validated, economic loss or injury arising from an assumption violation should be covered by an insurance contract. Also, we need to state the risks of the hazard that the manufacturer is not liable for, to warn the passenger and nearby movers, and to equip vehicles with devices to protect the passenger responsibly.

Limiting temporal bound with respect to the velocity of the vehicle is also reasonable. On-board sensors have a limited line-of-sight and a scope at distance. We cannot justify setting a trajectory beyond the scope. We want to reserve an interval for an initial responsive move to avoid the hazard.

Supporting Safety Claims

Linking one root-cause with each responsible stakeholder serves to separating a boundary of liability. We propose a safety claim that the vehicle preserves functional integrity subject to posed assumption set and loss of it is detected correctly and quickly. As loss of functional integrity is a silent event, we need to monitor the stated root-causes and violation of the assumption that impair the claim.

Monitoring Assumptions A run-time monitor (Barringer 2004) detects violations of the static safety constraints and the posed assumption set. To remove an uncontrollable risk, a capability of detecting violations should be tested at the pre-market stage when the warranty is finalized. Limits on observability due to poor visibility and inaccurate sensors are static constraints. The consequences of an assumption violation vary. The assumption on maneuverability is an over-approximation of possible motions M^m that depend on the mover's decision. Thus, violations of this kind of assumption may not immediately result in loss of functional integrity, unless the mover could actually attempt an adversarial motion that violates the assumption. Yet, the assumption on spatial separation enforced by traffic rules is a belief as to the benefit of promoting smooth traffic. Violations of this kind can result in an imminent risk of hazard. Moreover, as they are a result of the mover's unobservable decision, the situation poses an uncontrollable risk that is not stochastic in nature. Thus, at least, we need to prove no inevitable hazard in a subset of situations within the assumption set and clarify a limit of warranty on the functional integrity. Further, we should develop a preemptive method to deal with assumption violations. In practice, sensor data are often corrupted and cause spurious violations of assumptions. As the platform hardware can hardly detect such a violation by itself, assumption monitoring is needed to safeguard the vehicle against a malfunction caused by the broken sensor data.

Preemptive Detection of Infeasible Plans Second, we need to detect when the plan of action is unsatisfiable. While this is the same idea as testing feasibility of a conventional motion planning problem, tailored control logic is needed to withdraw any infeasible motion. It can be part of a motion planner that computes a series of input vectors from the posed plan of action (Nishi 2014).

Preemptive Detection of Unsafe Motions Third, a vehicle need to detect preemptively when it attempts a potentially unsafe motion in an over-constrained situation wherein dynamic safety constraints become unsatisfiable and the hazard becomes inevitable. This decision capability is formulated as adversarial motion planning problem. We compute a temporal series of the mover's states and actions subject to the posed assumption on maneuverability and check if a mover's motion that satisfies criterion (1) is feasible. The capability should be built into a run-time scheme, as hazard risk must be detected instantly and reserve an interval to figure out a preemptive action for retaining functional integrity.

Preemptive Response The vehicle must assure an on-board passenger at any time that it reserves an alternative liability-free plan of action which avoids any risk of a mover taking an action that would render criterion (1) unconditionally satisfiable. Also, the vehicle needs to withdraw the plan of action once an assumption violation is detected but the vehicle has no way to recover from such a situation wherein the safety claim becomes invalid.

Formula for Checking Functional Integrity

Solving Satisfiability Problem by NLP Solver

A NLP solver IPOPT (Wachter 2006) computes an optimal solution of a function $f(\boldsymbol{X})$ subject to constraints (2).

$$\overset{[}{j}= 1]m\bigwedge g_j^L \le g_j(\mathbf{x}) \le g_j^U \wedge \overset{[}{i}= 1]n\bigwedge x_i^L \le x_i \le x_i^U \tag{2}$$

It consists of problem variables $\mathbf{x} \in \mathbb{R}^n$, a twice-differentiable objective function $f(\mathbf{x})$, twice-differentiable constraint functions $\{g_j(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}|\ 1 \le j \le m\}$ with a bound $g_j^L, g_j^U \in \mathbb{R}$ on each one, and a search domain $\{\left(x_i^L, x_i^U\right) | 1 \le i \le n\}$. For our purpose of correctly checking satisfiability of (2), we set $f(\mathbf{x}) = 0$ and use IPOPT in a way that global convergence is guaranteed (Nishi 2016b). As IPOPT can receive a conjunction of predicates, we need to reformulate disjunctions of predicates into conjunctions of them in a way that the Boolean structure is preserved.

Constraints Constituting Functional Integrity

State Dynamics To expresses constraints on maneuverability of the vehicle nearby cars, bikes, and pedestrians, we use a simple non-linear dynamics equation; $\frac{d}{dt}\mathbf{x}_k^s(t) = \{v^s(t)cos\theta^s(t), v^s(t)sin\theta^s(t), a^s(t), \tau^s(t)\}$. The state vector $\mathbf{x}_k^s(t) \equiv \{r^{sx}(t), r^{sy}(t), v^s(t), \theta^s(t)\}$ consists of a position vector $\mathbf{r}^s(t) \equiv \{r^{sx}(t), r^{sy}(t)\}$, forward velocity $v^s(t)$, yaw angle $\theta^s(t)$, and input vector $\mathbf{u}^s(t) \equiv \{a^s(t), \tau^s(t)\}$ regarding acceleration and steering. Here, all state variables are confirmed observable ones. We apply Implicit Euler method and build a constraint function $F_k^s(\mathbf{X}^s, \mathbf{U}^s)$ expressed as (3). We denote a state vector at discrete-time k as $\boldsymbol{x}_k^s \equiv \{r_k^{sx}, r_k^{sy}, v_k^s, \theta_k^s\}$, and input vector at discrete-time k as $\boldsymbol{u}_k^s \equiv \{a_k^s, \tau_k^s\}$. $T_p = T\Delta t$ is the temporal horizon. We denote temporal series of state and input vectors as $\mathbf{X}^s \equiv \{\mathbf{x}_k^s | 0 \le k \le T\}$ and $\mathbf{U}^s \equiv \{\mathbf{u}_k^s | 0 \le k \le T\}$.

$$\bigwedge_{0\leq k<T} \left[\begin{array}{c} -r^{sx}_{k+1} + r^{sx}_{k} + v^{s}_{k+1} cos\theta^{s}_{k+1}\Delta t \\ -r^{sy}_{k+1} + r^{sy}_{k} + v^{s}_{k+1} sin\theta^{s}_{k+1}\Delta t \\ -v^{s}_{k+1} + v^{s}_{k} + a^{s}_{k+1}\Delta t \\ -\theta^{s}_{k+1} + \theta^{s}_{k} + \tau^{s}_{k+1}\Delta t \end{array} \right] = \mathbf{0} \qquad (3)$$

A precise formulation of $\mathbf{F}^s_k(\mathbf{X}^s, \mathbf{U}^s)$ is unnecessary, so long as the state vectors $\boldsymbol{X}^s$ produced using $\{\mathbf{x}^s_0, \mathbf{U}^s\}$ cover the whole reachable set. The impact of extending Δt is negligible, as the reachable set except for truncation error at the boundary is roughly irrelevant to Δt. The idea of over-approximating the reachable set prevents any flawed attempt of precisely modeling an inherently imperfect and uncertain real world. We reuse the same formulation regarding a mover $\mathbf{x}^m(t)$. We discretize the state vector of the mover $\mathbf{x}^m(t)$ as $\mathbf{x}^m_k \equiv \{r^{mx}_k, r^{my}_k, v^m_k, \theta^m_k\}$, and the input vector of the mover as $\mathbf{u}^m_k \equiv \{a^m_k, \tau^m_k\}$. We reuse the same formulation of a constraint on the mover in the vehicle's scope and define $F^m_k(\mathbf{X}^m, \mathbf{U}^m)$ where $\mathbf{X}^m \equiv \{\mathbf{x}^m_k | 0 \leq k \leq T\}$ and $\mathbf{U}^m \equiv \{\mathbf{u}^m_k | 0 \leq k \leq T\}$.

Plan of Action A simple way of encoding a plan of action is putting a constraint function $C(\mathbf{X}^s, \mathbf{U}^s)$ on $\{\mathbf{X}^s, \mathbf{U}^s\}$. We can let the vehicle move toward a terminal goal position by setting a constraint on $\mathbf{x}^s_T$. Programming a complicated plan of action involving a guard condition requires reformulating the disjunction of constraints. Yet, IPOPT can only receive conjunctions of constraint functions as shown in (2). We will report a way of handling disjunctions in the future.

Static Safety Constraints Static safety constraints represent limits of the vehicle's operational capability validated by the coverage of the warranty or by design. We build a formula $S^s(\mathbf{X}^s, \mathbf{U}^s)$ in (4) using constraints on the velocity component v^s_k, on the input vectors $\mathbf{U}^s$ regarding acceleration a^s_k and steering τ^s_k, and on their differentials.

$$\begin{array}{c} S^s(\mathbf{X}^s, \mathbf{U}^s) := \bigwedge_{0\leq k\leq T} [\begin{array}{c} \underline{\mathbf{u}} \\ \underline{v} \end{array}] \leq [\begin{array}{c} \mathbf{u}^s_k \\ v^s_k \end{array}] \leq [\begin{array}{c} \bar{\mathbf{u}} \\ \bar{v} \end{array}] \wedge \\ \bigwedge_{0\leq k<T} \underline{\mathbf{du}} \leq \frac{1}{\Delta t}(\mathbf{u}^s_{k+1} - \mathbf{u}^s_k) \leq \overline{\mathbf{du}} \end{array} \qquad (4)$$

We define a formula of assumption on maneuverability of movers in the field $\mathbf{S}^m(\mathbf{X}^m, \mathbf{U}^m)$ that encodes by reusing the same form of the constraints.

Spatial Geometric Constraints Traffic rules instruct the vehicle to move on specified roadways. A spatial constraint appears in dynamic constraints on $\mathbf{X}^s$, and part of the assumptions on $\mathbf{X}^m$. Given a convex spatial region $W_i \sqsubseteq \mathbb{R}^2$ and a point $\mathbf{r} \in \mathbb{R}^2$, we can define a monotonic function $h(\mathbf{r}, W_i)$ that returns 0 if $\mathbf{r} \in W_i$, or that gives the Euclidean distance from the point and monotonically increases if $\mathbf{r} \notin W_j$. As a non-convex region W consists of concatenation of non-overlapping convex regions $\{W_i | 0 \leq i < |W|\}$, formula (5) encodes $\mathbf{r} \in W$.

$$W \equiv \underset{0\leq i<|W|}{\cup} W_i,\ h(\mathbf{r}, W) \equiv \underset{0\leq i<|W|}{min} h(\mathbf{r}, W_i) = 0 \qquad (5)$$

Dynamic Safety Constraints A basic dynamic safety constraint is a spatial constraint that originates from a roadway, a crosswalk, a sidewalk, and an intersection where the right of way is controlled by a traffic signal. It is formulated as a constraint function $D^s(\mathbf{X}^s) \equiv h(\mathbf{r}^s_k, W^s)$ where W^s represents a spatial region in which the vehicle can move. We will reuse the same form of the constraint function $D^s(\mathbf{X}^m) \equiv h(\mathbf{r}^m_k, W^m)$ regarding the mover and spatial region W^m. Another sort of dynamic safety constraints is concerned with the criterion (1). It is controlled by the position vector of the mover $\mathbf{r}^m_k$.

$$D^m(\mathbf{X}^s, \mathbf{X}^m) := \neg \bigvee_{0\leq k\leq T} haz(\mathbf{r}^s_k, \mathbf{r}^m_k) \qquad (6)$$

Formula (6) suggests that the vehicle never attempts an unsafe move $\{\mathbf{r}^s_k | 0 \leq k \leq T\}$ when a nearby mover is within ϵ at any time and that the vehicle moves evasively at the time of a collision. We reiterate that (6) is irrelevant to the unobservable variables $\mathbf{U}^s$ and $\mathbf{U}^m$ that represent the both parties' intent of action. Here $\neg D^m(\mathbf{X}^s, \mathbf{X}^m)$ is the disjunction of a predicate $haz(\mathbf{r}^s_k, \mathbf{r}^m_k)$ that consists of two constraint functions. However, IPOPT can receive only a conjunction of constraint functions. Here, we can reformulate $s = \underset{0\leq k\leq T}{min} \|\mathbf{r}^s_k - \mathbf{r}^m_k\|^2$ to (7) by using auxiliary variables $\{q_k \in \mathbb{R} | 0 \leq k \leq T\}$ and a slack variable $s \in \mathbb{R}$.

$$\begin{array}{c} s = \overset{[}{k} = 0]T\sum q_k \left\|\mathbf{r}^s_k - \mathbf{r}^m_k\right\|^2 \wedge \overset{[}{k} = 0]T\sum q_k = 1 \wedge \\ \bigwedge_{0\leq k\leq T} \left[\left\|\mathbf{r}^s_k - \mathbf{r}^m_k\right\|^2 \geq s \wedge 0 \leq q_k \leq 1 \right] \end{array} \qquad (7)$$

The formula (7) is satisfiable, and the index k that satisfies $\left\|\mathbf{r}^s_k - \mathbf{r}^m_k\right\|^2 = s$ also satisfies $q_k = 1$ while all of other $\{q_k\}$ are zero. We can reformulate $H^m(\mathbf{X}^s, \mathbf{X}^m) \equiv \neg D^m(\mathbf{X}^s, \mathbf{X}^m)$ as (8) by using auxiliary variables $\{R_k \in \mathbb{R} | 0 \leq k \leq T\}$ and a slack variable $w \in \mathbb{R}$. Also, s and w correspond to each predicate in (1), $s \leq \epsilon^2$ and $w/\epsilon \geq \delta$.

$$\begin{array}{c} \left[\begin{array}{c} w \\ s \end{array} \right] = \overset{[}{k} = 0]T\sum q_k \left[\begin{array}{c} v^s_k(cos\theta^s_k, sin\theta^s_k) \\ \mathbf{r}^m_k - \mathbf{r}^s_k \end{array} \right] (\mathbf{r}^m_k - \mathbf{r}^s_k) \wedge \\ \bigwedge_{0\leq k\leq T} \left[\left\|\mathbf{r}^s_k - \mathbf{r}^m_k\right\|^2 - s = R_k \wedge 0 \leq q_k \leq 1 \wedge 0 \leq R_k \right] \wedge \\ \left[\begin{array}{c} w \\ -s \end{array} \right] \geq \left[\begin{array}{c} \epsilon\delta \\ -\epsilon^2 \end{array} \right] \wedge \overset{[}{k} = 0]T\sum q_k \left[\begin{array}{c} 1 \\ R_k \end{array} \right] = \left[\begin{array}{c} 1 \\ 0 \end{array} \right] \end{array} \qquad (8)$$

Assumptions on Maneuverability An assumption on maneuverability of the mover $A^m(\mathbf{X}^m, \mathbf{U}^m)$ is given as (9).

$$\bigwedge_{0\leq k\leq T} F^m_k(\mathbf{X}^m, \mathbf{U}^m) \wedge S^m(\mathbf{X}^m, \mathbf{U}^m) \qquad (9)$$

All reachable states $\mathbf{X}^m$ is enclosed by (9). We can use a moderate assumption and enclose a subspace of the reachable states by tightening the constraint $S^m(\mathbf{X}^m, \mathbf{U}^m)$. We will reuse the same form of a formula $\mathbf{A}^s(\mathbf{X}^s, \mathbf{U}^s)$ to express the assumption on maneuverability of the vehicle.

Detection of Infeasible Plan of Action

A simple way of losing functional integrity is to assert an infeasible plan of action. We can detect such an infeasibility by checking if formula (10) is unsatisfiable.

$$A^s(\mathbf{X}^s, \mathbf{U}^s) \wedge C(\mathbf{X}^s, \mathbf{U}^s) \wedge D^s(\mathbf{X}^s) \tag{10}$$

If it is satisfiable, we get a solution vector $\boldsymbol{U}^s = \{\mathbf{u}^s_k | 0 \le k \le T\}$ such that the plan of action is realized. Otherwise if unsatisfiable, then the plan of action needs to be withdrawn.

Preemptive Detection of Unsafe Motion

The proposed scheme should offer a generic interface to connect to a variety of motion planners and serve as a generic independent subsystem. We can translate the plan of action formulated as $C(\mathbf{X}^s, \mathbf{U}^s)$ into a satisfiable assignment $\mathbf{X}^s$ that contains at least a temporal series of positions $\{\mathbf{r}^s_k = \mathbf{p}^s_k \in \mathbb{R}^2 | 0 \le k \le T\}$. Then, we test if formula (11) has a satisfiable assignment $(\mathbf{X}^m, \mathbf{U}^m)$.

$$A^m(\mathbf{X}^m, \mathbf{U}^m) \wedge D^s(\mathbf{X}^m) \wedge H^m(\mathbf{X}^s, \mathbf{X}^m) \tag{11}$$

If (11) is satisfiable, we get a solution vector $\mathbf{X}^m$ that contains a temporal series of the mover's trajectory $\{\mathbf{r}^m_k | 0 \le k \le T\}$ that actually violate criteria (1) and we find a hazard risk that the vehicle is liable for. Otherwise, if (11) is unsatisfiable, the posed trajectory is free from liability so long as the assumption on maneuverability A^m holds. We learned that IPOPT inefficiently repeats iterations and extends the decision time, if an unsatisfiable NLP instance is posed. As we mostly try to check if (11) is unsatisfiable, the decision time in the unsatisfiable case should be short as well. Here, we can exploit the property that satisfiability of (8) solely depends on the feasible bounds on the pair $\{s, w\}$. Thus, we define $\hat{H}^m$ that excludes $s \le \epsilon^2$ and $w \geqq \epsilon\delta$ from $H^m(\mathbf{X}^s, \mathbf{X}^m)$ in (8) and build the NLP instance (12). Next, we check if $f(s, w) = 0$, in which case $s \le \epsilon^2 \wedge w \ge \epsilon\delta$ is satisfiable. Despite the local discontinuity of $\nabla f(s, w)$ at $s = \epsilon^2$ and $w = \delta$, the second derivatives are 0. Thus, the behavior of IPOPT remains stable.

$$\begin{aligned} &min\, f(s, w) = \begin{cases} s > \epsilon^2 & \to s - \epsilon^2 \\ s \le \epsilon^2 \wedge w < \epsilon\delta & \to -w + \epsilon\delta \\ s \le \epsilon^2 \wedge w \ge \epsilon\delta & \to 0 \end{cases} \\ &s.t.\ A^m(\mathbf{X}^m, \mathbf{U}^m) \wedge D^m(\mathbf{X}^m) \wedge \hat{H}^m(\mathbf{X}^s, \mathbf{X}^m) \end{aligned} \tag{12}$$

Selective Use of Assumptions

Multi-layered assumption set comprised of the moderate and the pessimistic one is helpful to provide a preemptive decision quickly after the moderate assumption gets violated. As far as the pessimistic one is satisfied, the vehicle can have a chance of recovering functional integrity using the pessimistic one. We argue that a practically viable setting is to monitor any violation of the moderate assumptions and on satisfiability of (11), until there remains a satisfiable alternative plan of action subject to the pessimistic ones. The assumption monitor using moderate one works as a threshold for preemptively detecting a potential risk of hazard. It safeguards the vehicle against the risk of hazard when the mover could actually attempt an adverse move aggressively beyond the moderate one. We do not definitely dismiss a conservative setting of using the pessimistic ones only. But the vehicle's pessimistic decision on the risk of hazard and resulting unusual driving experience can confuse neighboring movers. Indeed, sudden and frequent deceleration can unexpectedly inflict rear-end collision.

Experiments

We used a dual-core Intel Core-i3 CPU of 2.0GHz running on Linux kernel 4.0.8 and the following software libraries; IPOPT 3.12.3 with OpenBLAS 0.2.14 and Lapack 1.5.0. The parameters used in (1) are ϵ=1.0 m and ν=0.5 m/s. The parameters regarding the assumption on maneuverability of a vehicle was expressed using $\{\underline{v},\bar{v}\}$={-1.,20.}, $\underline{\mathbf{u}}$={-7.,-0.5}, $\overline{\mathbf{u}}$= {7.,0.5}, $\underline{\mathbf{du}}$={-3.,-0.25}, and $\overline{\mathbf{du}}$ ={3.,0.25}. The assumption on maneuverability of pedestrians was expressed using $\{\underline{v},\bar{v}\}$={-10.,10.}, $\underline{\mathbf{u}}$ ={-1.,-5.}, $\overline{\mathbf{u}}$={3.,5.}, $\underline{\mathbf{du}}$={-1.,-1.57} and $\overline{\mathbf{du}}$={1.,1.57}. We used options to configure IPOPT suitable for solving satisfiability problems; nlp_scaling_method=gradient-based, mu_strategy=adaptive, theta_max_fact=10, mu_max_fact=10, corrector_type=primal-dual.

Detection of Infeasible Plan of Action

A route planner can ignore the actual situation and force the vehicle to try any unreasonable plan of action. Fig. 2 shows a situation where the vehicle attempts to turn right at a four-way intersection. The temporal steps T is 50, $\Delta t = 0.1[s]$, and spatial geometric constraints were applied every 3 steps. A satisfiable plan of action was created by solving (10) using an initial state $\hat{\mathbf{x}}^s_0 = \{-2., -9., 3., 1.57\}$ and $\hat{\mathbf{u}}^s_0 = \{0., 0.\}$, and a reference plan of action $C(\hat{\mathbf{x}}^s_T)$ using (13) and a final state $\hat{\mathbf{x}}^s_T = \{18., 1.8, 6., 0.\}$;

$$C(\hat{\mathbf{x}}^s_T) = \|\hat{\mathbf{x}}^s_0 - \hat{\mathbf{x}}^s_T\| \le 0.1 \tag{13}$$

The NLP instance (2) consisted of n=306 variables and m=429 constraints. Initially, it took IPOPT 35 iterations (56 ms) to decide that the plan of action was feasible. Afterward, it took only one iteration (5 ms) by reusing the previous satisfiable solution vector $\{\mathbf{X}^s, \mathbf{U}^s\}$. If the actual initial state of the vehicle could be $\mathbf{x}^s_0$={-2.,-9.,18.,1.57} and $\mathbf{u}^s_0$={0.,0.}, such an aggressive attempt of turn would result in a departure from the plan. We replaced $\hat{\mathbf{x}}^s_0$ with $\mathbf{x}^s_0$ and solved (10) to check if the posed plan of action $C(\hat{\mathbf{x}}^s_T)$ was infeasible. It took IPOPT 77 ms to decide that the plan was infeasible after IPOPT was forced to terminate at 50 Newton iterations. It took IPOPT 18-69 ms afterward by the reuse. If the initial velocity v^s_0 was higher than 10 m/s, the resulting motion was infeasible because of the limit on the steering input τ^s_k; the vehicle entered the sidewalk, as shown in Fig. 2. The route planner had to withdraw the original plan, before the infeasible motion was actually attempted and loss of functional integrity occurred. A manufacturer of the vehicle would be liable for this hazard.

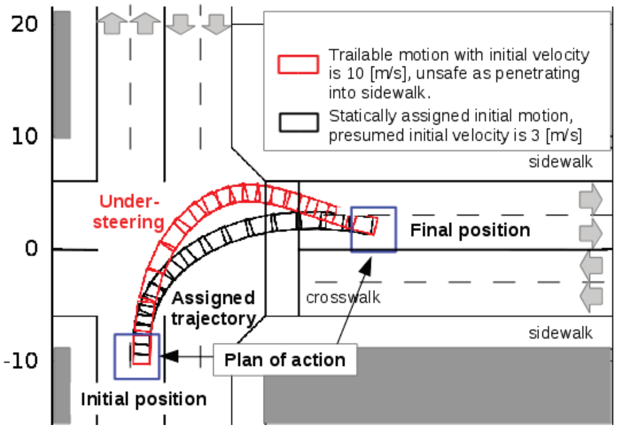
20
10
0
-10
Trailable motion with initial velocity is 10 [m/s], unsafe as penetrating into sidewalk.
Statically assigned initial motion, presumed initial velocity is 3 [m/s]
sidewalk
Under-steering
Final position
Assigned trajectory
crosswalk
sidewalk
Plan of action
Initial position

Figure 2: Infeasible plan of action.

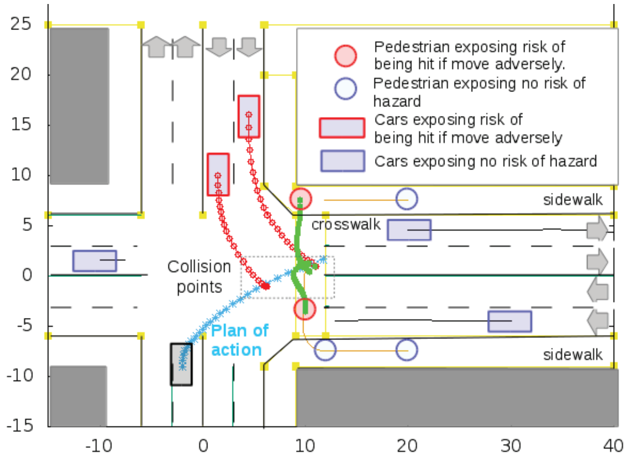

Figure 3: Detection of unsafe motion.

Preemptive Detection of Unsafe Motion

What primarily impacts on the decision time for deciding satisfiability using the proposed NLP encoding technique is geometric complexity of the environment and the degree of congestion measured by the number of movers who attempt conflicting move that satisfies (1). Thus, we studied the same situation at a four-way intersection at which the vehicle attempted to turn right. As shown in Fig. 3, we set five vehicles moving forward on opposite lanes and five pedestrians on sidewalks and a crosswalk. Each mover's right-of-way was controlled by traffic lights. The vehicles in opposite lanes could move in their lanes or cross the intersection. We set a moderate assumption that pedestrians can move on sidewalks or the crosswalk. The union of these geometric regions corresponds to W in (5) and is non-convex. Here, the vehicle attempted to follow a trajectory $\{\mathbf{r}^s_k = \mathbf{p}^s_k | 0 \leq k \leq T\}$ computed from the same plan of action $C(\mathbf{x}^s_T)$ in (13) where $\mathbf{x}^s_T = \{12., 1.8, 6., 0.\}$. We checked if (12) was unsatisfiable and dismissed a risk that the mover can reach the trajectory in a way that the vehicle is liable for. Here, $\Delta t = 0.25\,s$, $T = 20$, and spatial geometric constraints were imposed every four steps. As this planning problem constrained only the initial point, we had no clue as to whether the posed one was satisfiable or not, and no clue as to the location of the final solution vector. We studied a good initial solution vector and selected $\{\mathbf{x}^m_k = \mathbf{x}^m_0 | 1 \leq k \leq T\}$ and $\{\mathbf{u}^m_k = 0 | 1 \leq k \leq T\}$.

NLP instance (2) for each mover consisted of n=170 variables and m=157 constraints. From the initial iteration when IPOPT started from the posed solution vector and began to explore a satisfiable one, it took 71-89 ms to check if (12) regarding four among 10 movers was satisfiable. As shown in Fig. 3, two vehicles in opposite lanes and two pedestrians who attempted to go on the crosswalk were detected. If they moved accordingly and (1) was actually satisfied, the vehicle would have been at least partially liable for hazard because of negligence. Other movers were irrelevant to the risk of liability for a hazard, as they could not attempt to make an adversarial motion that satisfied (12). This initial overhead was negligible, as it occurred only when a new mover appeared within the scope of the vehicle. In practice, we can exploit the temporal locality of the solution vector to shorten the decision time, as the positions of movers and the trajectory of the vehicles have changed only slightly Δt after the initial iteration. Once IPOPT finds a satisfiable solution vector, the decision time gets shorter by reusing the previous one. It took 28 ms - 59 ms to decide if (12) was satisfiable. The decision time depended on the geometric relation between the spatial component of the solution vector $\{\mathbf{r}^m_k | 0 \leq k \leq T\}$ and the non-convex spatial region in which the mover could move. The time for checking the satisfiability of (12) regarding the pedestrians varied, as their movements were within the concatenated region of the crosswalk and the sidewalks, which had a non-convex shape. In contrast, the time for checking the satisfiability of (12) regarding the vehicles was steady and short, as they moved in a simpler geometric region. Note that once the run-time monitor detected a violation of assumption A^m and the hazard became inevitable, it was time for insurers to get involved.

Discussion and Related Work

Potential undecidability of (1)(10)(11) could result from that assumption violation is unpredictable. It could also result from partial observability is discussed in (Nishi 2016a). While we inadvertently assume that the confirmed states in them are available, corruption of the state can hardly be self-detected by the sensors. We need to check integrity of redundant data and statistically correct predicates. Particularly, a hazard caused by false negative detection must be addressed. In that case, no relevant data would be recorded on the run-time log and a root-cause analysis would fail.

An idea of trajectory verification at the pre-market stage separately from a planning algorithm (Wongpiromsarn 2009) is prone to loss of functional integrity and violation of assumption. As the planner determines the actual motion at run-time, the violation must be detected at run-time in order to adapt to it (Bensalem 2014). If the validity of safety claim is impaired that way, we may hardly suppress the rise in the frequency of hazard. The proposed way of using the NLP solver replaces the method of analyzing a forward reachable set that handles non-linearity inadequately and that suffers from high computational costs (Althoff 2008). We judged that control logic synthesis is costly (Tabuada 2013) and limits the high-level decisions for handling fault and transient assumption violations by re-programming a complex plan and constraints at run-time.

Computing a satisfiable solution of (11) subject to an assumption on $\mathbf{U}^m$ serves as a verifiable run-time motion planner. Liability-free motion is a satisfiable solution to a quantified formula $\forall \mathbf{U}^m,\ A^s(\mathbf{X}^s, \mathbf{U}^s) \wedge D^s(\mathbf{X}^s) \wedge s > \epsilon^2 \vee \big(s \leq \epsilon^2 \wedge w \leq \delta\epsilon\big)$ using (s, w) in (12). This is close to a reach-avoid problem (Fisac 2015). We note that a crowd of pedestrians would not overburden the run-time scheme, as the constraints regarding their initial positions is replaced with a constraint function restricting their initial positions to be within the spatial region. Uncertainty due to imperfect sensors is handled in the same way. Lastly, we seek for a sufficient condition for resolution completeness using only abstract data in a compositional way (Yershov 2010).

Conclusions

We formalized a verification problem for supporting safety claims of fully autonomous vehicles. Our insight is that malfunctions of autonomous vehicles result from loss of functional integrity due to design defects or assumption violations. To fairly divide up the liability for hazard for each root-cause among the stakeholders, a safety claim should consist of only confirmed observable states and a commonly endorsed assumption set. We proposed a viable way of deciding liability for hazard and a safety claim in which the functional integrity with respect to the criteria is preserved. We proposed a run-time scheme that preemptively detects assumption violations and loss of functional integrity when infeasible plans of action and unsafe motion are attempted. A run-time scheme embodying the above ideas with encoding techniques using the NLP solver IPOPT was shown to be viable even in a real-world setting.

Acknowledgments

The date of initial submission was on October 23rd 2015. We appreciate anonymous reviewers of HSCC2016, NFM2016 and FM2016 for providing a mix of insightful, valuable, indifferent and critical comments.

References

Althoff, M. 2008. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. *47th IEEE Conference on Decision and Control.*

Anderson, J. M. 2014. Autonomous vehicle technology a guide for policymakers. RR-443-1-RC. RAND Corporation.

Barringer, H. 2004. Rule-based runtime verification. *International Workshop on Verification, Model Checking, and Abstract Interpretation.*

Bensalem, S. 2014. Verification and validation meet planning and scheduling. *International Journal on Software Tools for Technology Transfer* 16.1.

Fey, G. 2008. Automatic fault localization for property checking. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* 27.6.

Fisac, J. 2015. Reach-avoid problems with time-varying dynamics, targets and constraints. *Proc. of the 18th International Conference on Hybrid Systems: Computation and Control.*

Griesmayer, A. 2007. Automated fault localization for c programs. *Electronic Notes in Theoretical Computer Science* 174.4.

Jacklin, S. A. 2008. Closing the certification gaps in adaptive flight control software. *Proc. of AIAA Guidance, Navigation and Control Conference.*

Kalra, N. 2016. How many miles of driving would it take to demonstrate autonomous vehicle reliability? RR-1478-RC.

Manu, J. 2011. Cause clue clauses: error localization using maximum satisfiability. *ACM SIGPLAN Notices* 46.6.

Montemerlo, M. 2008. Junior: The stanford entry in the urban challenge. *Journal of field Robotics* 25.9.

NHTSA. 2013. Preliminary statement of policy on automated vehicle development.

NHTSA. 2016. Federal automated vehicles policy.

Nishi, M. 2014. Run-time conflict resolution mechanism for functional integrity of autonomous system. *53rd IEEE Conference on Decision and Control.*

Nishi, M. 2016a. Reduction of state observation problem to an identifiability problem. AAAI Fall Symposia 2016 Technical Report.

Nishi, M. 2016b. Towards bounded model checking using nonlinear programming solver. *Proc. of 31st IEEE/ACM International Conference on Automated Software Engineering.*

Rushby, J. 2008. Runtime certification. *Proc. of Runtime Verification.*

Tabuada, P. 2013. Linear time logic control of discrete-time linear systems. *IEEE Trans. on Automatic Control* 51(12).

Urmson, C. 2008. Autonomous driving in urban environments boss and the urban challenge. *Journal of Field Robotics* 25.8.

Villasenor, J. 2014. Products liability and driverless cars. *Issues and Guiding Principles for Legislation.* Brookings Institution.

Wachter, A. 2006. Line search filter methods for nonlinear programming motivation and global convergence. *SIAM Journal on Optimization* 16(1).

Wongpiromsarn, T. 2009. Receding horizon temporal logic planning for dynamical systems. *Proc. of 48th IEEE Conf. on Decision and Control.*

Yershov, D. S. 2010. Sufficient conditions for the existence of resolution complete planning algorithms. *Algorithmic Foundations of Robotics* IX.

Ziebart, B. D. 2009. Planning-based prediction for pedestrians. *Proc. of Int. Conference on Intelligent Robots and Systems.*