The Safety Force Field

David Nistér, Hon-Leung Lee, Julia Ng, Yizhou Wang

MAR19

Abstract

In this document, we present a general theory of safety at the obstacle avoidance level. The theory leads to a computational mechanism for mapping world perception into constraints on control that, if obeyed, prevents all collisions. Actors that do not obey the constraints can also be detected clearly. The intended corresponding practical implementation is to have a safety enforcement layer at the base of the autonomous vehicle control system that guarantees the safety requirements are met, provided the perception system performs as intended.

The theory defines a safety force field that is present between actors depending on their current state. The safety force field puts constraints on how one actor may act in the presence of another actor. The safety force field can be computed and is defined in such a way that if actors obey the constraints, they avoid unsafe states in a way that prevents collisions. An actor that ignores the safety force field can cause an unsafe condition, but this can be unambiguously detected.

The safety force field provides a computational mechanism at a basic level of obstacle avoidance. This component naturally combines with other software for planning and control, acting like a 'survival brain' that prevents unsafe behavior at the obstacle avoidance level. Additional components can be naturally stacked on top, including learning-based driving behaviors, map dependent constraints, wait conditions, yield and right-of-way requirements, and long tail traffic rules and regulations that may vary by country, location, time of day, or conditions.

We believe that a basic collision avoidance level should, as much as possible, function independently of the full complexity of software required to obey all traffic rules and drive courteously. This is similar to the way that an emergency braking system preferably does not depend on that full complexity. This allows a more practical safety decomposition and validation.

We outline a theoretical basis for a software component that is superior to previous proposals in several ways. It is constructive in the sense that it proposes how to perform computations to determine which actions are acceptable. The acceptable actions can be used as a safeguard at the motion planning layer of any other driving software. In this sense, it provides a general computational mechanism that serves as a platform. It allows doing better than a default safety procedure, when possible. This in turn allows it, in practice, to depend less on complex external structures such as, for example, a map, perceived road shape, or a complex world model beyond the representation of moving and static obstacles. This allows drawing upon the redundancy of obstacle perception, where the basis is strong, and depends less on the full complexity of a world model where the ability to achieve redundancy is less clear.

Introduction

The task of designing a system to drive a vehicle autonomously without human supervision at a level of safety required for practical acceptance is tremendously difficult. An attentive human driver draws upon a perception and action system that has an incredible ability to react to moving and static obstacles in a complex environment. In addition to the basic obstacle avoidance task, there are many rules of the road that are at a higher level than obvious obstacles in one's path, starting with traffic lights, directional dividing lines or arrows, one-ways, expected stop, yield and right-of-way behaviors, and going into a long tail of requirements such as school buses, ferry entrances, police, construction workers or emergency personnel directing traffic, commuter lanes that switch direction, rules to not block an area or intersection, or stop at an airport terminal, pay toll-booths, and signs with 'no U-turn on Thursdays 4-6 PM'.

In short, there are many ways to disobey formal traffic laws without causing an obvious obstacle avoidance hazard. We believe that while a system that can obey every traffic rule in every location, country, situation or condition is desirable, the foundation of safety should first be built in a way that does not depend on that long tail of rules. The basics of obstacle avoidance are the same in Redmond, Washington, Detroit, Munich, Tokyo, California, Calcutta, or Shanghai, although the rules and color coding on the road (if any) may not be.

We propose a software component that handles the basic fundamentals required to cooperate with other actors to avoid physical collisions. The control constraints implied on braking and steering guarantee no collisions under the assumptions of the obstacle avoidance world model they are derived from. It is important to recognize that any model is just that—a simplification of reality. Once the control constraints are part of a complete system that includes perception and actual physical action, the discrepancies between the world representation and the actual world, including but not limited to discrepancies in assumptions about control and actuation, will inevitably force careful validation and assessment of actual practical performance. This is unavoidable, and any claim to the contrary is irresponsible. We believe that the central challenge in designing safe autonomous vehicles is and will remain in accurate perception and modeling of a complex world and required driving behaviors. But the precision of the mathematical model facilitates the construction and verification of the system with a controlled understanding of what can go wrong and what needs to be validated at the basic level.

Redundancy, while it has limitations, is a way to combat the difficulty of modeling the world reliably. The constraints we calculate can be produced per sensor or sub-system, and the best control can be applied that satisfies, for example, at least two out of three sub-systems. This way, a false positive will require two sub-systems to produce false input constraints to block a control value and a false negative will require two sub-systems to miss constraints blocking a control value. This provides a very late form of fusion, where constraints are fused rather than world objects. Fusion can also be applied earlier before the constraints are produced. Our methodology provides computational tools and is flexible in how the world is modeled, and what planning and control is placed on top. In that sense, it is best viewed as computational tools for constructing a component of a modular system.

Our methodology preserves the strengths of previous proposals while removing important weaknesses. Previous proposals require a safety procedure to be used before an unsafe condition occurs, except in a number of cases that are specified by additional rules and moderated by input information such as the shape of the lane structure, or worse, the complete path structure in the environment drawn from a map, for which the redundancy story is weak, or where none of our mental models capture the real complexity of the world. Another potential problem is that the safety procedure is required even in cases where it does not help. For example, if another vehicle is tailgating us, this causes an unsafe condition, but if we apply our safety procedure and brake as hard as we possibly can, it would hurt rather than help. In previous proposals, this is handled by exemption rules. In our methodology, we instead use a direct calculation of whether our safety procedure will help and place our safety procedure as a baseline upon which we are allowed to do better. The key computational mechanism is to use the chain rule to calculate how much our safety procedure and other alternatives alleviate or contribute to the problem. This not only allows us to avoid applying difficult to design rules, but also means that our safety procedure does not have to be the perfect action for the situation, because we are enabled to improve upon it in a way tailored to the situation. The most important effect of this is that it lowers the dependency on input from complex rules, conventions, or maps. We achieve this while preserving the guarantee that safety is maintained if all actors follow the methodology.

The essence of the safety theory is simple. We first present the core concept in a nutshell, preserving its simplicity, making it easily accessible. We then make it more precise and generalize it to handle latency, time discretization, and visibility. We start with the nutshell level, which is neither complete nor mathematically precise, but gets the core idea across.

Core Concept in a Nutshell

Assume that the world is populated by dynamic actors and static obstacles. Assume that all dynamic actors have a safety procedure, typically to quickly slow down their lateral motion and slow down to a stop as fast as possible, subject to reasonable limits. The 'safety procedure' of static obstacles is simply to stay static. We think of the safety procedure as forming a family of trajectories that can be viewed as a point-set in space-time. The essence of the theory and what is required by an actor is this:

All actors are required to perform their safety procedure (or better) before and whenever the trajectory resulting from their safety procedure intersects with that of another actor.

Actors must begin their safety procedures before they intersect. If they do, their safety procedures play out and do not expand. Hence:

If all actors do what they are required, no collisions can occur.

Collisions are avoided if all actors do what is required. It can be detected unambiguously if they do not. By before, we intend that the actor is required to begin their safety procedure before an intersection occurs, and to continue the safety procedure whenever the intersection persists. We will define 'better' in the above statement more precisely later. This caveat is there to allow an actor to not engage in their safety procedure if it does not help, such as to not break when tailgated, or to do better than their safety procedure when possible, such as swerving to avoid a side swipe even when already going straight, provided it does not cause an unsafe condition to the other actors.

Core Concept Mathematics

We now proceed to specify the core concept at a more precise mathematical level.

State and Control

We want to work with parameterizations of the pose (such as position and orientation) of actors. To that end, we make:

***Definition 1**: The state of actor A is a vector $x_A(t) \in \mathbb{R}^m$ as a function of time that encodes the properties of actor A at time t. When viewed as a function of time, we refer to it as the state trajectory of actor A.*

Thus, the state of an actor is parameterized by an $\boldsymbol{m}$-dimensional state vector. For example, this could be a five-dimensional vector $\boldsymbol{x_A} = [\boldsymbol{y}^\top \ \boldsymbol{d}^\top \ \boldsymbol{v}]^\top$ holding the position $\boldsymbol{y}$ of the actor in two-dimensional space, a unit direction vector $\boldsymbol{d}$, and a scalar velocity $\boldsymbol{v}$. We will find it useful to consider sets of other actors.

***Definition 2**: The set Ω is the collection of the state spaces of all actors we consider, including static obstacles.*

In other words, for each actor $\boldsymbol{A}$, we can view its state as a function of time into the set $\boldsymbol{\Omega}$. We typically also need a control model for the state.

***Definition 3**: A control model $f(x_A, t, c)$ for actor A is a function f of the state x_A of the actor, time t, and control parameters c into $\mathbb{R}^m$.*

We use a control model function to specify that the derivative of the actor state with respect to time is governed by this function. That is, we set $\frac{dx_A}{dt} = f(x_A, t, c)$. In this way, control is formulated locally as an explicit differential equation with some parameters c that model user input, such as steering and braking. For example, building on the above, it could be $\frac{dx_A}{dt} = [vd^\top \; vbd_\perp^\top \; a]^\top$, where a is a scalar acceleration amount, b is a scalar steering parameter, and $d_\perp$ stands for the perpendicular to d generated by swapping its coordinates and negating the first coordinate. In this case, the control parameters are $c = [a \; b]^\top$. To do this properly, we have to close the loop by specifying the control parameters as a function $c(x_W, t)$ of the world state x_W (or in practice of perception of the world state) and time. When we do that, we get a control policy:

***Definition 4**: A control policy $\frac{dx_A}{dt} = f(x_W, t)$ for actor A is a function f of the joint state x_W of the world (all actors and static obstacles) and time t into $\mathbb{R}^m$ that is smooth and bounded, and that governs the derivative of the actor state with respect to time.*

Safety Procedure

Central to the safety theory is that each actor has a safety procedure.

***Definition 5**: The safety procedure S_A of actor A is a family of control policies that depend only on the actor starting state x_A and properties of the world that can be considered fixed, each of which brings the actor to a stop within a finite time. The safety procedure has a family of associated trajectories derived from any starting state x_A. We also require that the safety procedure results in a set of trajectories, each of which changes smoothly with its starting state x_A.*

While we allow the safety procedure to depend on fixed properties of the world such as, for example, the road shape or a map, it seems advantageous to avoid it if possible. A simple example building on the above: the safety procedure freezes the direction vector to straight ahead ($b = 0$) and begins slowing down by a range of acceleration values $[a_{min}, a']$ where a_{min} denotes the minimum acceleration amount (negative of maximum braking amount) to a complete stop and $a_{min} < a' < 0$, namely,

$$S_A = \left\{\frac{dx_A}{dt} = [vd^\top \; 0 \;\; a]^\top : a \in [a_{min}, a']\right\}.$$

We assume that actors start from rest at some point in time so that safety is guaranteed at some starting point.

Pose, Occupied Sets, Occupied Trajectories and Claimed Sets

We also want to consider the volume of space occupied by an actor given its pose. We assume that actors move around in and occupy points in n-dimensional real space $\mathbb{R}^n$. For practical simplicity, we will typically apply the theory with two-dimensional space modeling a top-down view of the real world, but the theory works equally well for three-dimensional space (or any other dimension). We also allow an optional safety margin.

Definition 6*: The occupied set $o_A(x_A) \subseteq \mathbb{R}^n$ of actor A is the set of points in space that the actor occupies as a function of its state x_A. This includes points physically occupied, as well as points needed to maintain a safety margin.*

If a point is in the occupied set of an actor, we say that the actor occupies the point. To again give a simple example, the occupied set could be a circle around the position of the actor. We extend this notion to a trajectory over time. T denotes the set of possible times for all actors we consider.

***Definition** 7: The occupied trajectory $O_A(X) = \{(y,t): y \in o_A(x_A(t)), t \ \in T\} \subseteq \mathbb{R}^n \times T$ of actor A is the set of points in space-time that the actor will occupy over time as a function of its trajectory X.*

We are particularly interested in the occupied trajectory that results from applying a control policy from the safety procedure. This is because when we casually refer to intersection of safety procedures, what we really mean is intersection of the occupied trajectories of actors when they apply their safety procedure. We will refer to the union of occupied trajectories of an actor when applying its safety procedure as the claimed set. The motivation for this name is that when points are in this set, the actor needs those points to maintain the integrity of its safety procedure.

Definition 8*: The claimed set $C_A(x_A) \subseteq \mathbb{R}^n \times T$ of actor A from state x_A is the union of occupied trajectories that results if the actor applies its safety procedure S_A starting from state x_A.*

Building on our string of examples, the claimed set would be the union of trajectories generated for each $a \in [a_{min}, a']$ by moving a circle from the center of the actor in the direction of the velocity vector by the distance profile $vt + at^2/2$. This comes to a stop at distance $-v^2/(2a)$ at time $-v/a$ and corresponds to the velocity profile $v + at$.

A key requirement on the safety procedure is that when following any trajectory of the safety procedure, the claimed set at each subsequent time must be contained in the original claimed set. For example, defining a safety procedure with a fixed compact and convex subset of control parameters, all with negative acceleration, would satisfy all requirements.

Unsafe Set and Safety Potential

We now consider the combined state space $\boldsymbol{\Omega_A \times \Omega_B}$ of two actors $\boldsymbol{A}$ and $\boldsymbol{B}$. We first name the indicator set in the combined state space signaling when the safety procedures intersect.

***Definition 9**: The unsafe set $U_{AB} \subseteq \Omega_A \times \Omega_B$ of the actor pair A, B is the set of points $(x_A,\ x_B)$ in the combined state space for which the claimed sets intersect, namely, $C_A(x_A) \cap C_B(x_B) \neq \emptyset$. Its complement $(\Omega_A \times \Omega_B) \setminus U_{AB}$ is called the safe set.*

Then we define a potential function on the combined state space.

***Definition 10**: A safety potential $\rho_{AB}: \Omega_A \times \Omega_B \to \mathbb{R}$ of the actor pair A, B is a real-valued function on the combined state space that is strictly positive on the unsafe set and non-negative elsewhere, and non-increasing when both actors apply any trajectories in their safety procedures.*

The safety potentials we intend to work with are exactly zero, some small distance away from the unsafe set. They serve as a smooth indicator function for the safe set. As specified in the definition, the key property is that the joint safety procedures of the actors prevent the safety potential from climbing. We want to consider the contribution of each actor to the change in the safety potential. The key observation using the chain rule is:

***Lemma 1**: The change of a safety potential with respect to time decomposes into $\frac{d\rho_{AB}}{dt} = \frac{\partial \rho_{AB}}{\partial x_A}\frac{dx_A}{dt} + \frac{\partial \rho_{AB}}{\partial x_B}\frac{dx_B}{dt}$, provided that ρ_{AB} is differentiable at (x_A, x_B).*

It will be natural as we construct safety potentials to assume the stricter requirement that the contribution of each of the individual safety procedures to the change in the safety potential is smaller or equal to zero. For that reason, it is useful to note that:

***Lemma 2**: A real-valued differentiable function $\rho_{AB}: \Omega_A \times \Omega_B \rightarrow \mathbb{R}$ on the combined state space that is strictly positive on the unsafe set and non-negative elsewhere, and for which $\frac{\partial \rho_{AB}}{\partial x_A} s_A \leq 0$ and $\frac{\partial \rho_{AB}}{\partial x_B} s_B \leq 0$ (for any $s_A \in S_A, s_B \in S_B$) is a safety potential.*

Proof: This follows from the previous Lemma and the definition of a safety potential. ■

From now on, we assume that the safety potential must be differentiable on the pair of states we consider. In a coming section, we will show that an everywhere differentiable and meaningful safety potential can always be constructed.

The Safety Force Field and Safe Control Policies

As we saw above, the local change in the safety potential decomposes into a contribution from each of the two actors. Moreover, the local contribution from each actor is the dot product between the control policy and the gradient of the safety potential (with respect to the state of that actor). This suggests a control policy that is the negative gradient of the safety potential, subject to limits, i.e., for the actor to move down the gradient of the safety potential. This motivates the following definition:

***Definition 11**: The negative gradient $F_{AB} = -\frac{d\rho_{AB}}{dx_A}$ of the safety potential ρ_{AB} is called the safety force field on actor A from actor B.*

Moving along the gradient of the safety potential whenever it is non-zero would be appropriate if there were only two actors, us and one more. This would result in us repelling from the other actor whenever our safety procedures are about to overlap. We can think of this as other actors inducing a safety force field on us via our safety procedures. Note that the safety force field is zero when our safety procedures are not close to overlapping. However, when there are two or more other actors, there are two or more gradients and they are in general different. We do not want to combine the safety force fields from multiple other actors in a linear or other ad-hoc manner, because doing so would give us no guarantees and would not arbitrate in a principled way between conflicting constraints. For example, if another vehicle is approaching us from the side while we are going straight or along the road, it is appropriate to move over before our claimed sets start overlapping. However, if there is also a vehicle on the other side that is also traveling appropriately, we do not want to keep moving over when our claimed set is also about to start overlapping with the 'innocent' vehicle's. This is important in order to guarantee that problems do not propagate in an uncontrolled way. This is where the safety procedure comes in. We have set things up so that our safety procedure is good enough in the sense that if all actors apply their safety procedures before they start overlapping, no collisions can occur. We want to behave in a way that is at least as good as our safety procedure with respect to each other actor. This is accomplished, while also allowing to do even better when possible, by:

Definition 12*: A safe control policy* $\frac{dx_A}{dt}$ *for actor A with respect to a set* $\Theta \subseteq \Omega$ *of actors is one for which* $F_{AB}\frac{dx_A}{dt} \geq \min_{s_A \in S_A} F_{AB}s_A$ *for each other actor* $B \in \Theta$.

Intuitively, the quantity $\min_{\mathrm{s_A \in S_A}} \mathrm{F_{AB}s_A}$ represents the worst local performance of any trajectory in the safety procedure. The inequality says that a safe control policy is one that performs at least as well as the worst trajectory in the safety procedure. From now on we make some reasonable assumptions on the safety procedure set so that the minimum in Definition 12 is always attained, say, the set is compact. Notice that the inequality in Definition 12 is equivalent to $\max_{s_A \in S_A} F_{AB}\left(\frac{dx_A}{dt} - s_A\right) \geq 0$. In other words, we want our control policy (when mapped into the full state derivative) to have a dot product against the safety force field from each other actor that is at least as large as some member of our safety procedure. We can also see our control policy as additive relative to our safety procedure. Seen this way, we want the addition to have a dot product that is at least zero against the safety force field from each other actor. This means that each other actor either produces a zero safety force field and no restriction, or puts a restriction that is exactly that our additive control policy is in the half-volume defined by a plane through the origin with the safety force field as normal vector, in the direction of the safety force field. Thus, the safe control policies form a convex polyhedron containing the safety procedure (and limited by limits such as jerk limits, which the safety procedure likely sits up against). To comment on the practical implication, this means that we can use the constraints as a safety layer applied to any desired control policy. We take our desired control at any moment and check it against the polyhedron. If it is in the polyhedron, we can just apply it as is. If it is not, we have to find some control parameters that map to a point in the polyhedron. For example, the closest realizable point in the polyhedron to the desired control. The polyhedron can never be empty because it contains the safety procedure. Note that the polyhedron in the full state derivative space must be mapped back to constraints on the control parameters.

Lemma 3*: A safe control policy exists.*

Proof: All we have to do is note that any member in the safety procedure is a safe control policy. ■

***Theorem 1**: Two actors with safe control policies with respect to each other do not collide with each other.*

Proof: As the two actors move along their combined trajectory with respect to time, the local change in the safety potential can be split into the two contributions from the two actors. With safe control policies, those contributions are no larger than by some members of the safety procedures of both actors applied at that point. This in turn we have constructed to be no larger than zero. Expressing the same in algebra, assume $s_A^* \in S_A$ and $s_B^* \in S_B$ are the trajectories that attain the minima: $F_{AB} s_A^* = \min_{s_A \in S_A} F_{AB}\, s_A$ and $F_{BA} s_B^* = \min_{s_B \in S_B} F_{BA}\, s_B$. Then one has

$$\frac{d\rho_{\mathrm{AB}}}{dt} = \frac{\partial \rho_{AB}}{\partial x_A}\frac{dx_A}{dt} + \frac{\partial \rho_{AB}}{\partial x_B}\frac{dx_B}{dt} \leq \frac{\partial \rho_{AB}}{\partial x_A} s_A^* + \frac{\partial \rho_{AB}}{\partial x_B} s_B^* \leq 0.$$

Here, the first equality relies on the chain rule, then the first inequality relies on the definition of a safe control policy, and the final inequality relies on the definition of a safety potential (that the safety potential is non-increasing when both actors apply any trajectories in their safety procedures). Since the safety potential is non-negative and the change in the safety potential is smaller or equal to zero everywhere along the combined trajectory, it can never leave zero, which means that the combined trajectory can never enter the unsafe set, which means the actors do not collide. ■

Bump Functions and Mollifiers

We now touch upon some theory enabling us to construct useful safety potential functions. In particular, we want to be able to cover sets with a smooth function and to smooth a function that is not infinitely differentiable. This section draws upon well-known material from the theory of bump functions and mollifiers [1,2,3], through which one can generate such covering functions with bump functions that are infinitely differentiable with compact support and smooth a function by convolving it with such a bump function, which is then called a mollifier.

***Lemma 4**: For any $\epsilon > 0$ and any ball in $\mathbb{R}^N$ for any N there is an infinitely differentiable scalar function that is strictly positive inside the ball, zero on the boundary of the ball, and zero outside the ball.*

Proof:

$$h(x) = \begin{cases} e^{-\frac{1}{1-|(x-p)/\epsilon\,|^2}}, & |(x-p)/\epsilon| < 1 \\ 0, & otherwise \end{cases}$$

is such a function for a ball centered at $\boldsymbol{p}$. It is just a shifting and scaling of the function

$$h(x) = \begin{cases} e^{-\frac{1}{1-|x|^2}}, & |x| < 1 \\ 0, & otherwise \end{cases}$$

which is a well-known bump function. We may shift and scale the function to center on any point in space and have any radius. ∎

We refer to such a function as a bump function. This allows us to guarantee the existence of and construct the type of covering function we want.

***Lemma 5**: For any $\epsilon > 0$ and any set in $\mathbb{R}^N$ for any N there is an infinitely differentiable scalar function that covers the set and evaluates to zero for any point further than ϵ from all points in the set.*

Proof: For a bounded set, it is easy to construct such a function by using bump functions of a diameter smaller than epsilon. First pick a bounding sphere for the set, then a finite cover of that bounding sphere by bump functions of a fixed diameter smaller than epsilon. Then make the function the addition of each bump function that includes at least one point in the set. This function is clearly covering the set, and it is infinitely differentiable since it is a finite sum of infinitely differentiable functions, and it is clearly zero since otherwise a ball function includes both a point in the set and a point epsilon or further from the set, which would be a contradiction. For an unbounded set, we can extend the definition of the function by an expanding sequence of spheres where new bump functions are added to cover the additional set between one bounding sphere and the next. This carries with it a sequence of bump functions, and we define the function as the (now infinite sequence) sum of bump functions. The function is still well defined and infinitely differentiable at any point since for an epsilon-neighborhood of a point there is only a finite set of bump functions from the sequence that affect it. The function also clearly covers any point in the set and evaluates to zero epsilon away from the set since the analysis for any point falls back to the finite case. ∎

Construction of Safety Potentials

Our construction of safety potentials relies on variants of non-negative measures of the intersection between the safety procedures of the two actors. Note that as both the safety procedures are applied, the trajectories of the two actors just 'play out', which means that an intersection measure works on the same claimed sets, shrunk versions of them, or whatever is left of them as the actors progress through them. As what is left decreases, typical intersection measures do not increase. This is at the heart of our constructions as this allows us to satisfy the non-climbing property of a safety potential. Hence, if we can get smooth measures of the intersection, we get a safety potential.

One variant is to use the sum of the time intervals between the first time that there is an intersection between just slightly dilated occupied sets (in time-slices of the claimed set) and the time that each actor is fully stopped, zero if there is no intersection. This function is bounded, non-negative, and strictly positive on the unsafe set. It also stays constantly zero if there is no intersection, stays constant until the intersection if there is one, and decreases after the intersection if there is one. It also does not increase if the claimed sets do not expand. Hence, its smoothness is the only thing left to ensure, which is typically already the case when there is an intersection and when there is not. In the transition just when the intersection appears, we can handle this by using a monotonic function of the time interval that flattens space near zero like a bump function does at its boundaries. One advantage of this variant is that it is efficient to compute in practice.

Another variant is what we can think of as the dot product between two smooth functions covering the claimed sets of the actors. If the initial covering functions are near constant on the claimed sets, this results in a smoothed version of volume of overlap between the covering functions. Construct smooth covering schemes $h_A(C_A(x_A)), h_B(C_B(x_B))$ for each of the claimed sets, that are non-increasing when the claimed sets shrink. Our output is then

$$\rho_{AB}(x_A, x_B) = \int_{\Omega_A \times \Omega_B \times T} h_A(C_A(x_A)) h_B(C_B(x_B))\, dx_A dx_B dt$$

where we assume that the integration takes place from the current time to the first time that both actors are fully stopped. The resulting function is smooth, bounded, non-negative, strictly positive on the unsafe set, and non-increasing when the claimed sets shrink.

Visibility

Visibility deserves special mention. Thus far we have considered the world as known. In practice this takes place through perception (or maps, vehicle-to-vehicle, or vehicle-to-infrastructure). The key is that the limitations to perception are understood. If perception was limited but in completely unpredictable ways, then the only safe way to act is not to move. Thus, we have to make sure that limitations, such as visibility, range, and uncertainty are understood. A way to handle uncertainty is to assume that all actors (actor states) that are possible under perception uncertainty are possible. For example, a vehicle an uncertain distance away would be modeled by the set of its possible positions. Note that the modeling cannot be fundamentally probabilistic and at the same time provide absolute guarantees. To handle visibility and range, we assume that perception can confirm that some actors are present, and deny some actor states (i.e., confirm that no actor is present in a particular state) and leave some actor states as unknown, or 'invisible'. We encode this via:

***Definition 13**: The visible set $V \subseteq \Omega$ is the collection of actors confirmed by perception. An invisible actor is an actor who could be present, but can neither be confirmed nor denied by world perception. The invisible set $\Lambda \subseteq \Omega$ is the collection of states of all invisible actors.*

In other words, $V \cup \Lambda \subseteq \Omega$ is the collection of actors who could be present. Then, we can handle visibility limitations by assuming the worst. Before we do, we want a mechanism for excluding extreme states of actors that may be physically possible but would force us to behave too conservatively for practical use if we have to take them into account. Such actors will be declared as not satisfying the requirements solely based on their state. An example would be an actor who is traveling at 150 mph and is about to cross our route, but not yet visible to us. We could handle such actors by saying that they could not be present, but we prefer to acknowledge that they could be present but call them unreasonable, so that we can include them in our analysis.

Definition 14*: We divide the set Ω of actor states into reasonable and unreasonable, where unreasonable states are states we consider so extreme we do not want to take them into account unless they are visible. The set $\Psi \subseteq \Omega$ is the collection of all reasonable actors (and its complement $\overline{\Psi} = \Omega \setminus \Psi$ is the collection of all unreasonable actors).*

While we in principle allow the set of unreasonable states to be some complex set of states, such as depending on a map, we intend the use of a simple definition, such as all states above a certain speed. We typically also want to assume that we never ourselves enter an unreasonable state. One way to achieve that is to always monitor our safety procedure and make sure that it never comes close to an unreasonable state (if it does, we have to apply it). Of course, we probably want to do this with a much smoother braking profile to allow smooth transitions between speed zones, for example. The simplest case is if we can use a single upper limit for reasonable speed (likely significantly above speed limit) for a particular application. The same way that we assume that actors start in safe states at some point in time, we assume that the actors start in reasonable states at some point in time.

With this definition, we can think of the set $V \cup (\Lambda \cap \Psi)$ as the collection of all actors who can reasonably be present. Now we are ready to define a visibility-aware control policy.

Definition 15*: A visibility-aware control policy for an actor is a control policy that is safe with respect to all other visible and reasonable invisible actors. In other words, a visibility-aware control policy $\frac{dx_A}{dt}$ for actor A is one for which $F_{AB}\frac{dx_A}{dt} \geq \min_{s_A \in S_A} F_{AB} s_A$ for all $B \in V \cup (\Lambda \cap \Psi)$.*

Note that the safety procedure is a visibility-aware control policy, so it is clear that a visibility-aware control policy exists. We also immediately get:

Lemma 6*: A visibility-aware control policy is safe with respect to all visible and reasonable invisible actors (despite the visibility limitations).*

Proof: This is a direct consequence of the worst-case assumption and the definition of a safe control policy. Since we assume that all constraints (from visible and invisible actors) that could be present are present, we must be obeying all relevant constraints. ■

Extensions to Handle Latency and Discretization

Latency, discretization, and reaction time are other important practical limitations of real systems and actors that we want to model. We want to ensure that we will act safely even with those limitations. We take a similar approach to this as we did for visibility. Again, we are dealing with a limitation in perception, or more precisely perception and action, in the sense that when an actor takes action, it is inevitably based on perception that is not completely current. Whether the latency is in perception or action is immaterial—the end effect is that when the actor takes action, it is based on perception of the world at some earlier (while hopefully very recent) point in time. Assume that the amount of latency is Δt. To handle that, we need a form of worst-case forward prediction:

Definition 16*: The forwarded set $\Phi_A(x_A, \Delta t)$ of actor A by a time interval Δt is the set of all states that actor A could possibly get to at the time interval Δt after being in state x_A. The forwarded set $\Phi(\Theta, \Delta t) = \bigcup_{A \in \Theta} \Phi_A(x_A, \Delta t)$ of a collection Θ of actors by a time interval Δt is the union of the forwarded sets of all actors in Θ.*

Note that an actor typically has a better ability to predict its own state than that of other actors. In particular, in the control system of an autonomous vehicle, the actual command sequence that was previously sent is known, providing an ability to predict where the actor itself will be when the actuation command that is deliberated now is actually issued. For practical purposes this can allow the forwarded set to include only one point, effectively being 'deterministic forwarding', resulting in a single actor state, while in general the forwarding mechanism is 'non-deterministic forwarding', resulting in a set of states. While in principle, we could use non-deterministic forwarding of the actor itself and require that the control policy is safe for all the possible states the actor could be in, we can keep things simple by assuming deterministic forwarding of the actor itself. We refer to this simply as a control policy for the forwarded actor, assuming implicitly that the state parameterization we work with is updated with a prediction based on all the actuation commands in queue up to the one deliberated now. Note that with these assumptions, the control command will apply to the actor state considered, and the only thing delayed is the information regarding the other actors.

Definition 17*: A forwarded control policy with respect to a perceived collection Θ of actors by a time interval Δt is one that is safe with respect to the forwarded set $\Phi(\Theta, \Delta t)$ of Θ.*

It is straightforward to see that:

***Lemma** 7: A forwarded control policy is safe at the current time with respect to wherever the perceived collection of actors moved, despite the latency limitations between perception and action.*

Proof: This is again a direct consequence of the worst-case assumption and the definition of a safe control policy. Since we assume that all constraints (from wherever other actors can get to when our control applies) that could be present are present, we must be obeying all relevant constraints. ■

We now combine latency awareness with visibility awareness and add that we should not enter unreasonable states. We call the combination a sound control policy. We carefully elect to consider the set $\Phi(\mathrm{V}, \Delta\mathrm{t}) \cup (\Phi(\Lambda, \Delta\mathrm{t}) \cap \Psi)$. First, visibility is taken into account to provide a 'complete' collection representing all the actors (visible and invisible) in the world that we want to consider at one point in time. Then latency is taken into account on that complete world representation by forwarding both sets. Finally, we exclude unreasonable actors from the forwarded set of invisible actors. We prefer not to exclude unreasonable visible actors since it would be odd to ignore actors that are actually perceived. We could have excluded unreasonable actors before forwarding, but that is less preferable because of unreasonable actors who make it into reasonable states during forwarding.

***Definition 18**: A sound control policy* $\frac{dx_A}{dt}$ *for an actor A is one for which* $F_{AB}\frac{dx_A}{dt} \geq \min_{s_A \in S_A} F_{AB} s_A$ *for all* $B \in \Phi(V, \Delta t) \cup (\Phi(\Lambda, \Delta t) \cap \Psi)$ *and that never enters an unreasonable state.*

The important result is:

***Lemma** 8: Actors with sound control policies do not collide with each other (despite the limitations of perception and action).*

Proof: Actors with sound control policies do not enter unreasonable states. Their control policies are also safe with respect to all other actors in reasonable states. Hence, they have control policies that are safe with respect to each other. Hence, they do not collide with each other. ■

For typical computer controlled autonomous actors, there is also a discretization at some level of the control system, so that action sequences are determined in discrete time intervals and then actuated by some lower level control mechanism. The world and the actuation itself are of course continuous—the world does not stop turning while the actor is thinking—but this means that strictly speaking, we have to guarantee that every point in the entire control sequence that will play out in the next actual interval are part of a sound control policy. This entails checking that entire sequence against forwarded actor sets, either forwarded to each time in the actuation interval, or more conservatively, forwarded to the end of the actuation interval.

Out of Policy Detection

Based on the safety force field, we are now able to quantify precisely at any moment whether and how much actors contribute to raising or lowering the safety potential with respect to each other actor. Based on this strength, it is straightforward to clearly detect when an actor is not satisfying the requirements. We first need a few basic definitions.

Definition 19*: An actor A for which* $F_{AB}\frac{dx_A}{dt} < \min_{s_A \in S_A} F_{AB} s_A$ *is said to behave out of policy with respect to the actor* B*. In words, an actor is behaving out of policy when they contribute to raising the safety potential with respect to another actor.*

Note that for two actors to collide, the safety potential between them must have been raised from zero. This is clear because their occupied sets intersect at the collision time, implying that they are in the unsafe set, implying that the safety potential is strictly positive.

Definition 20*: The uninterrupted time interval before a collision between actors A and B when the safety potential* ρ_{AB} *is strictly positive is called the out of policy interval.*

This allows us to clearly detect out of policy behavior as it relates to a particular collision:

Definition 21*: An actor A is out of policy with respect to a collision if it was in an unreasonable state* $(x_A \in \overline{\Psi})$ *at any time in the out of policy interval. An actor A is also out of policy with respect to a collision if they behaved out of policy with respect to actor B at any time in the out of policy interval when B was in a reasonable state.*

An important result is:

***Lemma 9**: At least one actor is out of policy with respect to a collision.*

Proof: At least one actor must have contributed to raising the safety potential during the out of policy interval. That actor either is out of policy with respect to the collision, or the other actor was in an unreasonable state in the out of policy interval and therefore is out of policy with respect to the collision. ■

***Lemma 10**: An actor with a sound control policy is never out of policy with respect to a collision.*

Proof: An actor with a sound control policy is not in an unreasonable state. It also does not contribute to raising the safety potential with respect to any other actors other than those who are in an unreasonable state. ■

Note that the contra-positive of this statement (which is therefore also true) is that an actor who is out of policy with respect to a collision does not have a sound control policy.

Right of Way and Wait Conditions

There is a long tail of cases, such as traffic lights corresponding to different paths, yield-patterns at multi-way stops, stop or yield lines in roundabouts, construction workers or police directing traffic, ferry entrances, school buses, and these rules vary by country. The common theme of these cases is that there are rules that require us to stop or yield above and beyond what is obvious from an obstacle avoidance perspective. We collectively refer to these cases as wait conditions. Because of the variety of such cases, a complete formalization that fully describes how an autonomous vehicle must behave is out of scope of this document. We just note that there is a higher level function that gives some actors right-of-way over others.

One question is whether right-of-way, however it is derived, should allow the actor who has it to behave more aggressively than the safety force field normally allows, or require the actor who should give it to behave less aggressively. Regarding the former, it is at least the case that if another actor is visible and in such a state that a safety force is applied to us, then we should obey that force as usual. To not do it would be almost to observe that mutual help from both parties is required to not collide, and then not provide help. This is not strictly true since the actors can do better than the safety procedure, but serves to make the point that ignoring the safety force field based on right-of-way is ill-advised. Perhaps the only case where this might be warranted is in very low visibility situations, such as a completely blind corner controlled by a traffic light. In this extreme case, the traffic light could be viewed as a substitute for perceiving the fact that no other vehicles should reasonably be coming at high speeds across our route. It is as if we perceive that guarantee via the traffic light, like perceiving via a mirror assisting to view around a blind corner. Such exceptions should be used with care, however. If the traffic light was green to both actors, then blame could only be placed on the traffic lights. So, in summary, the safety force field should be obeyed regardless of right-of-way. On the other hand, if we are expected to yield, additional requirements are on us beyond the safety force field. The recurring theme is that we should strive to behave in such a way that we do not cause a safety force field on the actor to whom we should yield. If we do, the other actor should still yield, but we have failed to meet the expectation. It is interesting to note that an actor can 'bully' other actors with their safety force field, and we believe this is actually a good model of what happens in highly congested traffic, with actors strong-arming their way into a lane or similar. So the essence of giving right-of-way is to avoid constraining the other actor by making sufficient concessions. This is perhaps harder to make mathematically precise because once you have merged into the path of another actor who had right of way, at some point you may constrain them due to traffic in front or a congested intersection.

Attentiveness

The safety force field is designed based on the assumption that both actors in a pair perceive each other whenever necessary and understand the safety procedure of the other actor, or take the necessary precautions based on their own perception limitations. We believe this is necessary to model actual driving behavior in, for example, highway driving. However, it is useful to be able to handle actors that fail to meet these requirements. We may have noticed that a pedestrian is looking at their phone instead of at us as they cross a street, or we may have reason to believe that a merging vehicle has not even seen us yet. In those cases, when can we assume that the other actor will see us, if at all? Different answers to this question lead to alternative constraints. One quite conservative answer is to assume that other actors see us only once we are literally in the path that they will end up taking, and that they after that time will need sufficient time to slow down. This is one example of several, all of which can be designed and combined with the safety force field.

Model Summary

We have presented a computational mechanism that provides a safety layer for collision avoidance purposes. The essence of the model is to take perception to a set of constraints on control. At this point it is useful to summarize the content of the model. The model consists of: a collection of actor state spaces and a definition of a reasonable set of actors, a perception mechanism defining visible and invisible actor sets, a control model and family of specific control policies called the safety procedure for each actor, an occupied set function for each actor, and a safety potential between actors.

Implementation Example

Let us now get back to our running example and work out the details of a safety potential. For simplicity of this illustration, we will take a safety procedure that contains just a single control policy, hence a single trajectory.

We use

$$x_A = \begin{bmatrix} y \\ d \\ v \end{bmatrix} \qquad \frac{\delta x_A}{\delta t} = \begin{bmatrix} vd \\ vbd_\perp \\ a \end{bmatrix} \qquad S_A = \begin{bmatrix} vd \\ 0 \\ a_{min} \end{bmatrix} \qquad s(t) = vt + \frac{a_{min}t^2}{2}$$

$$t_{stop} = -\frac{v}{a_{min}} \qquad y(t) = y_0 + s(t)d \qquad R(t) = [d\ d_\perp]$$

where x_A is a five-dimensional vector holding the position y of the actor in two-dimensional space, a unit direction vector d, and a scalar velocity v. For the control, a is a scalar acceleration amount, a_{min} its minimum, b is a scalar steering parameter, and $d_\perp$ stands for the perpendicular to d generated by flipping its coordinates and negating the first coordinate. In this case, the control parameters are $c = [a\ b]^\mathsf{T}$. We have also included the trajectory resulting from the safety procedure represented as a pose transformation from vehicle coordinates to world coordinates. The pose transformation is stated in terms of a reference point $y(t)$ of the vehicle in the world and a rotation $R(t)$ from vehicle to world, assisted by a distance traveled $s(t)$.

An alternative if we want to get closer to a physically plausible model of a vehicle is to include b in the state, and introduce a steering rate β that is part of the control instead. In this case, we have

$$x_A = \begin{bmatrix} y \\ d \\ v \\ b \end{bmatrix} \qquad \frac{\delta x_A}{\delta t} = \begin{bmatrix} vd \\ vbd_\perp \\ a \\ \beta \end{bmatrix} \qquad S_A = \begin{bmatrix} vd \\ vbd_\perp \\ a_{min} \\ 0 \end{bmatrix} \qquad s(t) = vt + \frac{a_{min}t^2}{2}$$

$$t_{stop} = -\frac{v}{a_{min}}$$

$$y(t) = y_0 + \frac{1}{b}(\sin(s(t)b)d + (1 - \cos(s(t)b))d_\perp)$$

$R(t) = [d\ d_\perp]\begin{bmatrix} \cos(s(t)b) & -\sin(s(t)b) \\ \sin(s(t)b) & \cos(s(t)b) \end{bmatrix}$ where x_A is instead six-dimensional.

The occupied set can, for example, be a circle, or a bounding box (more generally a polygon). If we want to use a safety potential where the earliest intersection is involved, we have to find that time and its derivatives with respect to x_A. As a practical calculation, this is best done by first performing collision detection (in space-time $\mathbb{R}^\mathrm{n} \times T$). For example, by searching through time for the earliest intersection point $p \in \mathbb{R}^n \times T$ between circles or bounding boxes, and then performing differential analysis on the intersection point p once we have found it. Our safety potential is

$$\rho_{AB} = \left\| \left(t_{A_{stop}} - p_t, t_{B_{stop}} - p_t\right) \right\|_k$$

where p_t is the time coordinate of p. Notice that the k-norm in this definition of safety potential can be replaced by any norm where $k \geq 1$, including max-norm. We now want to compute the derivative $\frac{\partial p}{\partial x_A}$ since that is the hard part of the expression to differentiate.

There are a few cases. The earliest intersection can either be an intersection between two smooth surfaces, or the intersection between a vertex of A and a smooth surface of B, or between a smooth surface of A and a vertex of B.

Movement of Safety Procedure Surface with Respect to Actor State

The safety procedure is viewed as a trajectory of the pose represented by $y(t), R(t)$ applying to points on the occupied set definition of the actor. Let us say that we have a point z in the actor coordinate system. It then moves to point

$$w(t) = \begin{bmatrix} y(t) + R(t)z \\ t \end{bmatrix}$$

as a function of time as the safety procedure plays out. If we perform differential analysis of those world points with respect to change in x_A, that gives us an understanding of how the object surface moves in space-time due to change in x_A. To first order, the shape of the surface in space-time does not change (since that depends on second derivatives). Thus, if we know the local shape of the space-time surface of the safety procedure for an actor (either a surface normal or a vertex curve tangent), plus the derivative $\frac{\partial w}{\partial x_A}$, that gives us all we need to perform differential analysis of how the surface behaves. If we have that for both actors, we can combine the result to determine the change in the intersection point. We have

$$\frac{\partial w}{\partial x_A} = \frac{\partial y}{\partial x_A} + \frac{\partial R}{\partial x_A} z$$

For our first example, this can be concretized to

$$\frac{\partial w}{\partial x_A} = \begin{bmatrix} I & s(t)I + [z \quad z_\perp] & td \\ 0 & 0 & 0 \end{bmatrix}$$

Smooth vs. Smooth

In the case where the earliest intersection point happens between two smooth surfaces, such as those swept out by two circles moving over time, the local surfaces at that intersection point can both be approximated to first order by a plane (in space time). The plane can be found by computing $\frac{dw}{dt}$ at the intersection point and taking the cross product with a direction vector for a tangent vector to the smooth or polygonal shape at the same point (which for a polygon can be found by transforming its end points by $y(t), R(t)$ and subtracting them). Assume that we have such normal vectors to the local plane of both surfaces, related to both actors. Assume also that those vectors have been normalized to unit magnitude—call the result $A_\perp, B_\perp$. Then we observe that local motion of the A-surface is measured by $\frac{\partial w}{\partial x_A}$. The correction along the A-surface to 'get back to' the B-surface is then some multiple q of $(I - A_\perp A_\perp^\top)B_\perp$. The sum of those two is the vector

$$\frac{\partial p}{\partial x_A} = (I - A_\perp A_\perp^\top)B_\perp q + \frac{\partial w}{\partial x_A}$$

we are looking for. We also have the constraint that $B_\perp^\top \frac{\partial p}{\partial x_A} = 0$, which allows us to solve for

$$q = -\frac{B_\perp^\top \frac{\partial w}{\partial x_A}}{1 - (A_\perp^\top B_\perp)^2}$$

which yields

$$\frac{\partial p}{\partial x_A} = \frac{\partial w}{\partial x_A} + \frac{((A_\perp^\top B_\perp)A_\perp - B_\perp)}{1 - (A_\perp^\top B_\perp)^2}\left(B_\perp^\top \frac{\delta w}{\delta x_A}\right)$$

which is an efficient way to perform the calculation.

Vertex vs. Smooth

In the case where the intersection is between the curve swept out by a vertex of the A-surface and a smooth part of the B-surface, we instead assume that we have a tangent vector $A_\top = \frac{dw_A}{dt} / \left|\frac{dw_A}{dt}\right|$ to the curve and a normal vector to the space-time B-surface. Again, also assume that those vectors are normalized to unit magnitude. We get a very similar calculation as for the smooth-to-smooth case. The local motion of the A-curve is measured by $\frac{\partial w}{\partial x_A}$. The correction along the A-curve to 'get back to' the B-surface is then some multiple q of $A_\top$. The sum of those two is the vector

$$\frac{\partial p}{\partial x_A} = A_\top q + \frac{\partial w}{\partial x_A}$$

we are looking for. We also have the constraint that $B_\perp^\top \frac{\partial p}{\partial x_A} = 0$, which allows us to solve for

$$q = -\frac{B_\perp^\top \frac{\partial w}{\partial x_A}}{A_\top^\top B_\perp}$$

which yields

$$\frac{\partial p}{\partial x_A} = \frac{\partial w}{\partial x_A} - \frac{A_\top}{A_\top^\top B_\perp}\left(B_\perp^\top \frac{\partial w}{\partial x_A}\right).$$

Smooth vs. Vertex

In the case where the intersection is between a smooth part of the A-surface and the curve swept out by a vertex of the B-surface intersecting, we instead assume that we have a normal vector to the space-time A-surface and a tangent vector $B_{\top} = \frac{dw_B}{dt} / \left|\frac{dw_B}{dt}\right|$ to the curve. Again, also assume that those vectors are normalized to unit magnitude. The local motion of the A-surface is measured by $\frac{\partial w}{\partial x_A}$. The movement of the intersection point along the B-curve is $\frac{\partial p}{\partial x_A} = B_{\top} q$, which is some multiple q of $B_{\top}$. The subtraction of those two has to be in the plane of the A-surface and hence perpendicular to $A_{\perp}$, which yields

$$A_{\perp}^{\top}\left(B_{\top} q - \frac{\partial w}{\partial x_A}\right) = 0$$

and

$$q = \frac{A_{\perp}^{\top}\frac{\partial w}{\partial x_A}}{A_{\perp}^{\top} B_{\top}}$$

which yields

$$\frac{\partial p}{\partial x_A} = \frac{B_{\top}}{A_{\perp}^{\top} B_{\top}}\left(A_{\perp}^{\top}\frac{\partial w}{\partial x_A}\right).$$

References

[1] https://en.wikipedia.org/wiki/Bump_function

[2] https://en.wikipedia.org/wiki/Mollifier

[3] https://en.wikipedia.org/wiki/Non-analytic_smooth_function