Driving with guardian: Blending user inputs with safety ensuring barriers

Michael Arwashan, Tiancheng Ge, Zexiang Liu, Necmiye Ozay

Abstract— **This paper considers the practical aspects of using a control barrier function induced by a robust controlled invariant set for supervising user inputs. Our focus application is path following/lane keeping functionality in advanced driver assist systems. We define specific normalized barrier functions for ellipsoidal and polytopic invariant sets. Then, we use the magnitude of these barriers to determine an optimum safe control input as well to determine when to blend this input with the user's input. This method allows for a smooth overriding of the user input and is well defined in the case that the barrier is violated due, for instance, to model mismatch. The method is demonstrated in a realistic simulation environment. These simulations show the proposed method to be a computationally light way to safeguard user inputs in a smooth manner.**

I. Introduction

Improving the safety of passenger vehicles can have great societal impact as traffic accidents are one of the major causes of death around the world. Many accidents are a result of human error, hence automation can help eliminate some of these accidents. Two main approaches to tackle this issue are to take the human out of the driver's seat as in autonomous vehicles (AVs) or to enhance the human driving performance via advanced driver assist systems (ADAS). While many ADAS systems, such as adaptive cruise control and lane keeping, are being deployed in high-end vehicles, formal approaches to prove their safety are relatively recent [1], [2], [3]. One way to ensure safety in ADAS is to supervise the human's or driving algorithm's inputs via controlled invariant sets or barrier functions [2], [4], [5], [6], [7].

In this work, we develop new techniques for blending human driver's inputs with an input computed according to a safety barrier to ensure that the system stays in a safe set (e.g., within the lane boundaries, or away from the lead vehicle). Such shared autonomy or guardian architectures have been studied before [8], [9], [10], [11], [12], [13], which do not necessarily enjoy strong safety guarantees. There are also recent works that employ barriers and invariant sets for supervision, ensuring permissiveness and safety [14], [2] by projecting the user input to the set of safe inputs only when the input would cause the vehicle to the leave the invariant set. However, in practice, this projection method is likely to fail maintaining invariance if there is a model mismatch or a violation of the assumptions used in the design of the invariant set. Moreover, it leads to abrupt overrides and large control rates, which might be undesirable, counter-intuitive, and scary for the driver.

Motivated by the potential practical shortcomings of simple projection-based supervision using barriers, we develop several blending techniques that more smoothly integrate user input with safe input, while still providing some degree of permissiveness. We demonstrate these techniques on a path following/lane keeping example, where the safe set is defined as keeping the vehicle within a certain distance from the path/lane center. We consider two types of barriers, polytopic and ellipsoidal, and leverage recent results in incorporating input delays and preview in their computation [15]. Finally, these blending techniques are integrated in a realistic simulation model provided by Toyota Research Institute (TRI) and thoroughly evaluated in this simulation environment.

II. Invariant sets and barriers

Consider a linear discrete-time system:

$$x(k+1) = Ax(k) + Bu(k) + E_1 d_m(k) + E_2 d_{um}(k) \quad (1)$$

where x is the state, $u \in \mathcal{U}$ an input, $\mathcal{U}$ are the input bounds, d_m and d_{um} are measurable and unmeasurable disturbances with bounds $\mathcal{D}_m$ and $\mathcal{D}_{um}$, respectively. By measurable disturbance, we mean that the value $d_m(k)$ is known when deciding on the input value $u(k)$ so we have a preview on the value of measurable disturbances.

A robust controlled invariant set (or, invariant set, for short) [16] for a system of the form (1), is a set $\mathcal{C}_{inv}$ of states inside a safe set $\mathcal{X}_{safe}$ (i.e., $\mathcal{C}_{inv} \subseteq \mathcal{X}_{safe}$) that satisfies:

$$\begin{aligned} x \in \mathcal{C}_{inv} \Rightarrow \forall\ d_m \in \mathcal{D}_m\ \exists u \in \mathcal{U}\ \ \forall d_{um} \in \mathcal{D}_{um} \\ Ax + Bu + E_1 d_m + E_2 d_{um} \in \mathcal{C}_{inv} \end{aligned} \quad (2)$$

Given an invariant set $\mathcal{C}_{inv}$, a state x inside it, and a measured disturbance d_m, the safe (i.e., admissible) input set, $\mathcal{U}_{safe}(x, d_m)$, can be defined as:

$$\begin{aligned} \mathcal{U}_{safe}(x, d_m) = \{u \in \mathcal{U}\ | Ax + Bu + E_1 d_m + \\ E_2 d_{um} \in \mathcal{C}_{inv}, \forall d_{um} \in \mathcal{D}_{um}\}. \end{aligned} \quad (3)$$

This safe input set governs what input can be taken at a given state so that the system state is guaranteed to be kept within the invariant set.

In this work, we will consider two types of invariant sets: polytopic invariant sets and ellipsoidal invariant sets. When the safe set $\mathcal{X}_{safe}$, disturbance sets $\mathcal{D}_{um}$, $\mathcal{D}_m$, and the input set $\mathcal{U}$ are symmetric around the origin[1], we can restrict

The authors are with the Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI. {marwasha, gtcheng, zexiang, necmiye}@umich.edu.

[1] A set $\mathcal{X} \subseteq \mathbb{R}^n$ is symmetric around the origin if for all $x \in \mathcal{X}$, we have $-x \in \mathcal{X}$. In this case, we also write $\mathcal{X} = -\mathcal{X}$.

attention, without loss of generality, to invariant sets that are also symmetric around the origin.

Proposition 1: Let $\mathcal{C} \subseteq \mathbb{R}^n$ be an invariant set for a linear system, i.e., $\mathcal{C}$ satisfies (2), with symmetric safe, input, disturbance sets. Then, $\mathrm{Sym}(\mathcal{C}) \doteq \mathcal{C} \cup -\mathcal{C}$ also satisfies (2) with the same safe, input, disturbance sets.

Proof: Take any $x' \in -\mathcal{C}$, then $x' = -x$ for some $x \in \mathcal{C}$. We can write (2) as $-x \in -\mathcal{C} \Rightarrow \forall -d_m \in -\mathcal{D}_m \;\; \exists -u \in -\mathcal{U} \;\; \forall -d_{um} \in -\mathcal{D}_{um} \quad -Ax - Bu - E_1 d_m - E_2 d_{um} \in -\mathcal{C}$. Define $u' \doteq -u$, $d'_m \doteq -d_m$, and $d'_{um} \doteq -d_{um}$, and use the symmetry of the sets $\mathcal{U}$, $\mathcal{D}_m$, and $\mathcal{D}_{um}$, to write $\forall d'_m \in \mathcal{D}_m \;\; \exists u' \in \mathcal{U} \;\; \forall d' \in \mathcal{D} \;\; Ax' + Bu' + E_1 d'_m + E_2 d'_{um} \in -\mathcal{C}$. Thus, there is an input for any point in $\mathcal{C}$ or $-\mathcal{C}$ (hence in their union) that ensures remaining in these sets so $\mathrm{Sym}(\mathcal{C})$ satisfies (2). Finally, since $\mathcal{X}_{safe}$ is symmetric, we have $\mathrm{Sym}(\mathcal{C}) \subseteq \mathcal{X}_{safe}$. ■

Given a compact invariant set $\mathcal{C}$ containing the origin, we associate with this set a novel barrier function $r : \mathbb{R}^n \to \mathbb{R}$ that satisfies:

$$r : \begin{cases} r(x) = 0 & \text{if}\, x = 0 \\ r(x) \in [0, 1) & \text{if}\, x \in int(\mathcal{C}) \\ r(x) = 1 & \text{if}\, x \in \partial\mathcal{C} \\ r(x) > 1 & \text{if}\, x \notin \mathcal{C} \end{cases} \tag{4}$$

where $int(\mathcal{C})$ and $\partial\mathcal{C}$ denote the interior and the boundary of the set $\mathcal{C}$, respectively. For a given state x, $r(x)$ is the *barrier magnitude* at x. If $r(x)$ is between 0 and 1, then the state x is within the invariant set and $r(x) > 1$ indicates that the state is outside the invariant set. The *barrier magnitude* measure gives a way to assess the safety of the current state relative to the origin. Technically the goal is to ensure that the state never goes out of the barrier; however this might happen due to model mismatch when pushing the vehicle to safety limits, or violation of the assumptions on the disturbance bounds. Therefore, we also want the barrier to be well-defined when outside the invariant set, giving a degree of robustness to the barrier-based design [17]. Then using the barrier, we define an optimal control input for safety as:

$$u^*(x, d_m) \in \arg\min_{u \in \mathcal{U}} \max_{d_{um} \in \mathcal{D}_{um}} r(Ax + Bu + E_1 d_m + E_2 d_{um}). \tag{5}$$

Note that when x is inside the invariant set and $d_m \in \mathcal{D}_m$, $u^*(x, d_m)$ is guaranteed to be in $\mathcal{U}_{safe}(x, d_m)$. A sufficient condition for the uniqueness of u^* is when B has full rank and $r(x)$ is strictly convex.

Remark 1: The barrier function definition in (4) is slightly different that what is commonly found in the literature where barrier value on the boundary of the set is 0 and it is positive in the inside and negative outside [17]. Equation (4) provides a normalization of the barrier magnitude for compact sets defined around the origin, which we find helpful while evaluating and comparing different blending methods using barriers induced by potentially different invariant sets for the same problem.

For the rest of the paper, we will assume the sets $\mathcal{X}_{safe}$, $\mathcal{U}$, $\mathcal{D}_m$, and $\mathcal{D}_{um}$ are polytopes, sets that can be represented by linear inequalities. For the invariant sets $\mathcal{C}_{inv}$, we will study two commonly used representations: polytopes and ellipsoids.

One way to represent an invariant set is using polytopes:

$$\mathcal{C}_{inv} = \{x \mid Hx \le h\}. \tag{6}$$

We assume $Hx \le h$ is a reduced representation of a polytope symmetric around the origin. By reduced representation, we mean that there are no redundant inequalities in the representation and each row of $Hx \le h$ corresponds to a unique facet of the polytope. There are methods to compute polytopic invariant sets for linear systems (please see the Appendix for details). Given a polytopic invariant set of the form (6) that is symmetric around the origin, a barrier can be defined as:

$$r_{poly}(x) = \max_i \left(\frac{H_i x}{h_i}\right), \tag{7}$$

where H_i and h_i are the ith row of H and h. It is easy to verify that this function satisfies the conditions in Eq. (4). The admissible input set $\mathcal{U}_{safe}$ is also a polytope in this case and can be computed by standard polytope operations [18]. Moreover, the computation of optimal input reduces to a robust linear program, which can further be reduced to a standard linear program.

Another invariant set representation we use is ellipsoids:

$$\mathcal{C}_{inv} = \{x \mid x^T M x \le 1\}, \tag{8}$$

where M is a positive definite matrix. There are techniques for computing ellipsoidal invariant sets inside polytopic safe sets [19], with some trade-off between computational efficiency and conservatism. We provide an alternative algorithm in the Appendix. The barrier induced by an ellipsoid of the form (8) can be defined as:

$$r_{ellip}(x) = x^T M x. \tag{9}$$

Again this function can be shown to satisfy the properties in Eq. (4). In this case, the admissible input set $\mathcal{U}_{safe}$ is an intersection of ellipsoids with the polytope $\mathcal{U}$, and the computation of u^* reduces to a convex quadratic program.

Moreover, in the case of a single input, i.e., $u \in \mathbb{R}$, we can compute u^* in Eq. (5) for both polytopic and ellipsoidal barriers from a bisection search algorithm.

III. Vehicle model and safe set

We focus on the safety in lane keeping/path following scenarios. The lane keeping problem is one of keeping a vehicle within its lane bounds. This can be done for either Autonomous Vehicles (AVs) where one has full control over the vehicles actions or in Advanced Driver Assistance Systems (ADAS) where one monitors a human driver's input to keep safe driving [20], [21], [22]. Same problem also occurs when, instead of a physical lane, the aim is to follow a path, within a bounded error, generated by a path planner or by predicting human intent. For this purpose, we consider the following lateral dynamics model of the vehicle:

$$\begin{bmatrix} \dot{l} \\ \dot{\tilde{\theta}} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} V sin\tilde{\theta} \\ \alpha_5 V \delta + \alpha_6 V^2 \delta + \dfrac{V\kappa(s) cos\tilde{\theta}}{l\kappa(s) - 1} \\ \alpha_7(\delta^{cmd} - \delta) \end{bmatrix}. \tag{10}$$

The states are offset error l (m), heading error $\tilde{\theta}$ (rad), and steering angle δ (rad). Offset is the lateral deviations of the center of the vehicle from the center of the path/lane, heading error is the rotational deviations from the angle of the path, and steering angle is the angle of the wheels. α_5, α_6, and α_7 are learned parameters which Toyota Research Institute (TRI) provided for their test vehicle. The input to the system, δ^{cmd}, is the wheel angle commanded by the steering wheel, and the disturbance, $\kappa(s)$ (m^{-1}), is the curvature of the road which we have preview of. A visual representation can be found in Fig. 1. The other variables in Eq. (10) are constant parameters of the model.

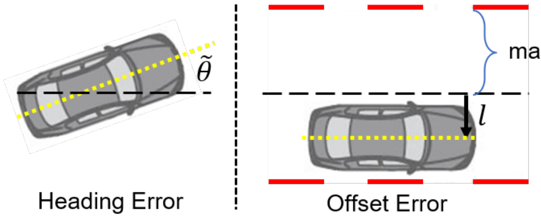


Fig. 1: Visual representation of the states of $\tilde{\theta}$ and l.

This model is then linearized at a constant velocity, v_0, and a constant road curvature κ_0:

$$A = \begin{bmatrix} 0 & v_0 & 0 \\ v_0\kappa_0^2 & 0 & (\alpha_5 v_0 + \alpha_6 v_0^2) \\ 0 & 0 & \alpha_7 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ \alpha_7 \end{bmatrix}, \quad E_1 = \begin{bmatrix} 0 \\ -v_0 \\ 0 \end{bmatrix}. \tag{11}$$

Then, for a given sampling time T, a time-discretized model is obtained with matrices:

$$A_d = e^{AT}, \quad [B_d \;\; E_{1,d}] = \int_0^T e^{A\tau} d\tau [B \;\; E_1]. \tag{12}$$

Finally, we also include an additional unmeasurable disturbance term with $E_{2,d} = [1, 1, 0]^\intercal$ and disturbance set $\mathcal{D}_{um}$ to account for the mismatch between the linearized model and the nonlinear model, where the last term in $E_{2,d}$ is 0 since that part of the dynamics is already linear.

Another thing to account for, is a rather significant pure delay on the input of p time steps (overall delay is pT seconds) which TRI provided for their test vehicle. Combining the above, we obtain:

$$x(k+1) = A_d x(k) + B_d u(k-p) + E_{1,d} d_m(k) + E_{2,d} d_{um}(k) \tag{13}$$

where $x \doteq [l, \tilde{\theta}, \delta]^\intercal$, $u \doteq \delta^{cmd}$, and $d_m = \kappa$.

The safe set is defined as

$$\mathcal{X}_{safe} = \{x \mid -0.5 \leq l \leq 0.5\,(m), -\pi/4 \leq \delta \leq \pi/4\,(rad)\}.$$

The actuator limits impose constraints on the input via $\mathcal{U} = [-\pi/4, \pi/4]$ (rad). And, the road curvatures that will be navigated are assumed to be bounded by the disturbance set $\mathcal{D}_m = [1/100, -1/100]$ (m^{-1}).

One of the main results in [15] shows that for systems of the form (13) with delay and preview, there is a slightly modified auxiliary system without delay whose invariant sets can be used for constructing invariant sets for the delayed system. We use this reduction to compute an ellipsoidal and a polytopic invariant set for the discrete auxiliary linear dynamics. These invariant sets are shown in Fig. 2.

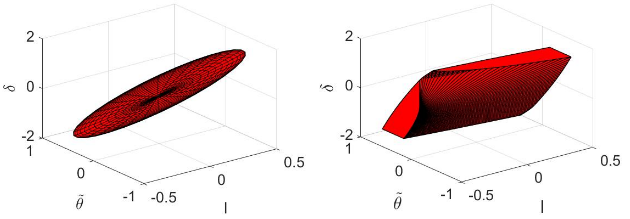


Fig. 2: A visual representation of the polytopic $\mathcal{C}_{inv}$ (right) and the ellipsoid $\mathcal{C}_{inv}$ (left) for the lane-keeping problem. The ellipsoid $\mathcal{C}_{inv}$ has a volume of 0.0440 from a ~2 second computation time while the polytope $\mathcal{C}_{inv}$ has a volume of 0.9743 from a ~3 hour computation time. For this polytope H contains 246 rows,

In order to account for the delay in supervision and blending (as discussed in the next section), we predict the state p time steps ahead using the current state, past control inputs, and disturbance preview information:

$$\begin{aligned} \widehat{x}(k+p) = & A_d^p x(k) + \sum_{i=0}^{p-1} A_d^i B_d u(k-1-i) + \\ & \sum_{i=0}^{p-1} A_d^i E_{1,d} d_m(k+p-1-i) \end{aligned} \tag{14}$$

This state, $\widehat{x}(k+p)$, in combination with the disturbance preview, $d_m(k+p)$, are the values used for the calculation of u^* and $\mathcal{U}_{safe}$ for supervising this system. d_{um} is ignored as the prediction error $\|x(k+p) - \widehat{x}(k+p)\|$ is bounded and taken into account by the controlled invariant set constructed in [15] so that the safety constraints are satisfied.

IV. Blending with Human Input

Given the invariant sets, in this section, we describe how these sets and induced barriers can be used to monitor and supervise a human driver within a guardian architecture. We start by explaining a baseline approach from the literature.

A. *Safe Input Projection*

One way to blend the user input with the barrier is to project the user input u_D onto the safe input set [2], [14][2]:

$$u = \underset{u \in \mathcal{U}_{safe}}{\arg\min} ||u - u_D||_2, \tag{15}$$

[2] For simplicity of notation, we use $\mathcal{U}_{safe}$ to denote $\mathcal{U}_{safe}(\widehat{x}(k+p), d_m(k+p))$ in the rest of this section.

Note that when the user input $u_D \in \mathcal{U}_{safe}$, it is unaltered, that is, we have $u = u_D$, providing permissiveness. This projection guarantees that the vehicle will stay within the invariant set under the modelled dynamics, however, has no statements on the control rate. Since it tends to react when the system operates close to the boundary of the invariant set C_{inv}, it may lead to overriding actions with very high control rates, which physically results in jerking the system away from the user, which is undesirable. Another problem with reacting close to the boundary of $\mathcal{C}_{inv}$ is that this leaves no room for error, namely that $\mathcal{C}_{inv}$ can be violated in application if the modelled dynamics are unable to fully encapsulate the true dynamics. If $\mathcal{C}_{inv}$ is exited, $\mathcal{U}_{safe} = \emptyset$, resulting in no corrective action. We will use this method as a baseline.

B. *Barrier Magnitude*

These limitations with the safe input projection lead to the idea of using barrier magnitude to blend. The barrier magnitude evaluates how safe the current state is, and if the user's input needs to be modified. The input is modified by blending the user input with the optimal control input available via:

$$u = cu^* + (1-c)u_D, \tag{16}$$

where u^* is a shorthand for the optimal input $u^*(\widehat{x}(k+p), d_m(k+p))$ defined in (5), u_D is the driver's input (with both $u_D, u \in \mathcal{U}$), and c is the barrier value dependent blending ratio.

To account for the effect of the user input, we define

$$\begin{aligned} r^*(\widehat{x}(k+p), u_D(k)) = \max_{d_{um}} \; & r(A_d\widehat{x}(k+p) + B_d u_D(k) + \\ & + E_{1,d} d_m(k+p) + A_d^p E_{2,d} d_{um}). \end{aligned} \tag{17}$$

This gives us a worst case prediction of the barrier magnitude due to the user's action. Note that if $u_D \in \mathcal{U}_{safe}$, Eq. (17) gives $r^* \leq 1$. For simplicity of notation, we use r or $r(k)$ to denote $r^*(\widehat{x}(k+p), u_D(k))$ in the rest of this section.

We take c to be a piece-wise affine function of r:

$$c = f_1(r) = \begin{cases} 0 & \text{if } r \leq r_3 \\ (r - r_3)/(r_4 - r_3) & \text{if } r \leq r_4 \\ 1 & \text{if } r \geq r_4 \end{cases} \tag{18}$$

where r_3 is the barrier magnitude prediction in which blending first occurs and r_4 is the barrier magnitude prediction where the user is fully overridden by the optimal input. This is similar to applying a proportional feedback term on the barrier magnitude once r_3 is surpassed. With this analogy, we notice that if r is increasing rapidly towards r_3, the supervisor will respond with a respectfully fast blending action resulting in an undesired high control rate. To respond we introduce damping on the barrier magnitude rate r', which we compute in discrete-time as $r'(k) = (r(k) - r(k-1))/T$, where T is sampling time, by first redefining c:

$$c = c_o + c_b \quad 0 \leq c \leq 1 \tag{19}$$

where c_o is a blending term based on the barrier magnitude, r, and c_b is the a blending term based also on the barrier magnitude rate, r'. We then compute these terms using:

$$c_o = f_1(r), \quad c_b = f(r, r') = f_{b(r)}(r'), \tag{20}$$

where $f_{b(r)}$ and b are defined as:

$$f_{b(r)}(r') = \begin{cases} b(r)r' & \text{if } r' > 0 \\ 0 & \text{if } r' \leq 0 \end{cases} \tag{21}$$

$$b(r) = \begin{cases} 0 & \text{if } r \leq r_1 \\ b_{max}(r - r_1)/(r_2 - r_1) & \text{if } r \leq r_2 \\ b_{max} & \text{if } r \geq r_2 \end{cases} \tag{22}$$

Here b acts as a damping coefficient. The parameters r_1, r_2, r_3, r_4, and b_{max} are tuned while obeying that $0 < r_1 < r_2 \leq r_3 < r_4$. r_1 is selected so that the user has a region in the $\mathcal{C}_{inv}$ where they have full control, i.e., no blending occurs. Once this region is exited, $r > r_1$, damping is introduced with a ramp to prevent discontinuous jump of c. r_2 is selected so that the damping saturates at a known b_{max}, to give a greater level of control for tuning. r_3 is when proportional action on r begins more strongly correcting the system. Between r_3 and r_4, c can be seen as a proportional/derivative controller that aims to correct the potential violations of the barrier. Finally, r_4 is the threshold when full overriding occurs, which gives an upper bound on r for the system. These functions are plotted in Fig. 3.

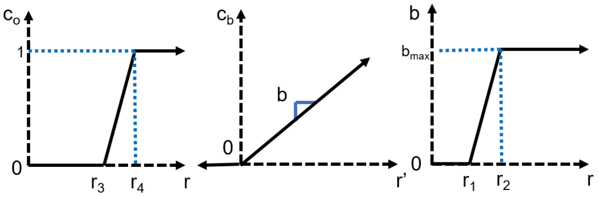


Fig. 3: Functions described in Eqs. (18), (21), and (22).

The profile of f_1 ensures that as we reach the barrier boundary, the optimum solution is more heavily weighted. However, this is leading to very high control rates when the barrier is approached rapidly. To accommodate for that, we add a damping term c_b determined in f_b to reduce severe control rate takeovers. To avoid similar discontinuous jumps in control rates we blend the damping coefficient, $b(r)$.

Although these blending techniques are heuristics with tunable parameters, as long as $r_4 \leq 1$, the supervisor will fully override the user with u^*, a control input steering the system towards the origin of the $\mathcal{C}_{inv}$ before the boundary is met. This means that as long as assumptions on the model and disturbance bounds hold, these techniques inherit the invariance and safety guarantees provided by the barrier. Even if assumptions fail, since r is well-defined outside of the $\mathcal{C}_{inv}$, these techniques are also well-defined outside of the $\mathcal{C}_{inv}$, providing a certain degree of robustness.

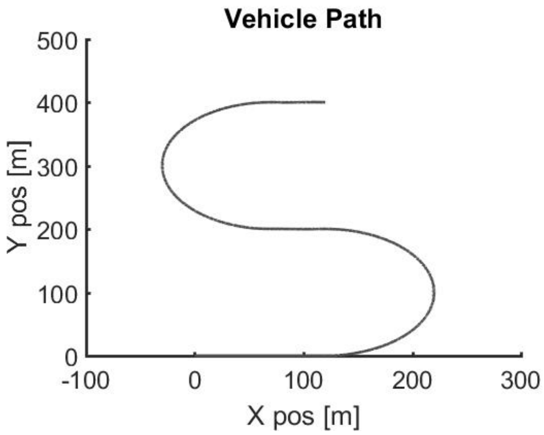


Fig. 4: The target trajectory for all simulations. This course has two turns with a curvature of 1/100m and is performed at a constant velocity of 10m/s. This trajectory was chosen since it was deemed most difficult to operate at the disturbance bounds.

V. Results and evaluation

In this section, we evaluate the proposed blending techniques and compare them with the projection-based baseline. A simulator, provided by TRI, with nonlinear vehicle dynamics was used for the tests. The road profile/path used for the test is shown in Fig. 4, that is navigated by keeping a constant velocity at 10m/s. The road has two turns with curvature $\kappa = 1/100$.

A user input was used that purposefully violated the barrier during the turns and the blending methods of safe input set, blending without damping, and blending with damping were tested for both the polytopic and ellipsoidal barriers. The parameters used for these tests are given in Table I.

TABLE I: The parameters used for both the polytope and ellipsoid blending simulations

Parameter	Without Damping	With Damping
r_1	0.00	0.40
r_2	0.00	0.75
r_3	0.85	0.85
r_4	0.95	0.95
b_{max}	0.00	0.20

One thing to note is that the polytopic and ellipsoidal barrier magnitudes vary greatly in scale as shown in Fig. 5 due to the difference in the sizes of the corresponding invariant sets. To account for this, the user inputs were scaled to violate each barrier with the same profile to make a fairer comparison. In particular, we generated an aggressive human input to test the polytope barrier and a mild human input to test the ellipsoid barrier. Their lateral offsets are show in Fig. 6.

We depict the simulation results in Figs. 7 and 8 for polytopic and ellipsoidal barriers, respectively. These simulations had very similar patterns with the difference of barrier magnitude measurements being taken into account for scaling human input. The human input for each respective simulation shown in Figs. 7a and 8a violate each respective barrier magnitude by the same respective amount for each simulation. In Figs. 7b and 8b, the human input is supervised with the input projection method. This method of supervision results in undesirable behavior as it corrects the human input with high control rates, which physically would result in jerking of the steering wheel from the driver which we deem an unpleasant experience. In Figs. 7c and 8c, the human input is supervised with barrier magnitude blending with no damping. This results in keeping the system farther away from the boundary as the controller engaged at a barrier magnitude value of $r = 0.85$. This, however, also results in severe blending control rates, and some oscillatory behavior as the vehicle bounces off the lane boundary on both sides for overly corrective blended inputs. In Figs. 7d and 8d, the human input is supervised with barrier magnitude blending with damping. With this damping term, the blended control rates are *two orders of magnitude smaller* than those of input projection/blending without a damping term. This is a significantly better user experience as the steering wheel is gently pushed to the correct position to guide the user, with the proportional term in case the system still approaches the barrier boundary. These more proactive, smoother blending resulted in smaller overall barrier magnitudes when compared to the other blending methods. Not seen in simulation but important to note is that another advantage of this method is that since the barrier magnitude is well defined outside of the barrier, the safety controller is still able to operate if the barrier is violated.

We also investigate some quantitative metrics as summarized in Table II. In particular, we consider maximal control rate u'_{max}; time blended T_b, defined as sampling time T times the cardinality of the set $\mathcal{K} = \{k \in [0, k_{total}] \mid u(k) \neq u_D(k)\}$; number of engagements, which is the number of consecutive segments in the set $\mathcal{K}$; total deviation Δ from

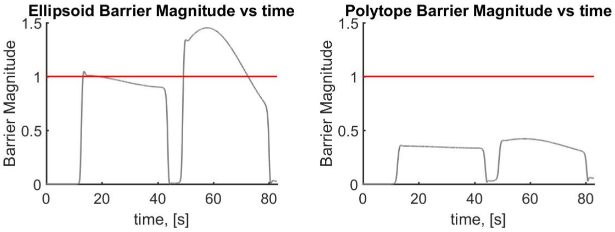


Fig. 5: For the same trajectory, the polytope barrier magnitude was much smaller than the ellipsoid barrier magnitude. This is as the polytopic invariant set is larger than the ellipsoidal one, cf. Fig. 2, thus leading to points which are on the ellipsoids boundary being within the polytope.

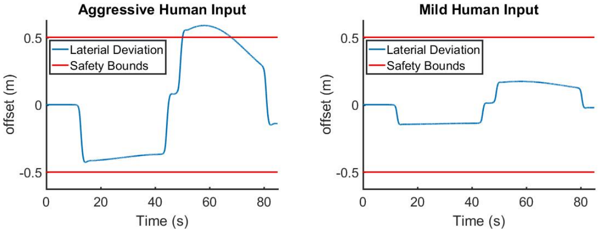


Fig. 6: The lateral offset of the Aggressive and Mild human inputs. It is shown that the aggressive human input violates $\mathcal{X}_{safe}$, while the mild user does not.

the user input, defined as $\Delta = \sum_{k=0}^{k_{total}} |u(k) - u_D(k)|$; and average deviation, that is, $\Delta/(T_b T)$. From Table II, it can be seen that u'_{max} and average deviation improved for both $\mathcal{C}_{inv}$ when the blending method went from input projection to barrier magnitude blending with damping. The polytopic $\mathcal{C}_{inv}$ is also tested with the mild human input, which results in zero blending action from all three methods, highlighting the permissiveness of the polytopic $\mathcal{C}_{inv}$.

These methods are also tested with different trajectory profiles, leading to similar results. We find that the difficulty to supervise increases with larger road curvature. Thus the trajectory in Fig 4 is chosen as it includes two turns at the minimum and maximum bounds of $\mathcal{D}_m$, acting as a difficult test case.

VI. Conclusions

In this paper, we propose a new blending method using barriers to safeguard human inputs. This blending method uses barrier magnitude to determine an optimal control input and how much blending action should be taken. A key insight is to incorporate the derivative of the barrier magnitude in blending decision, making the approach more proactive and less abrupt. The approach is demonstrated via lane keeping computer simulations with barriers safe guarding a human input that intentionally violates the barrier. Our results show that our proposed method has a much smoother override action on a user when compared to projecting the input onto a safe input set and provides a good trade-off between permissiveness and driving comfort (measured by input rates), while not compromising the theoretical safety guarantees of barrier-based approaches. Our future work is to bring these blending methods onto a real vehicle, and to adapt them to be used in a varying speed environment. We would also like to run some user tests (either on a real vehicle or on a simulator) to understand the trade-offs and user preferences better.

Acknowledgments

This work was supported in part by Toyota Research Institute (TRI). TRI provided funds to assist the authors with their research but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. The authors would like to thank Kevin Zaseck from TRI for valuable discussions and for providing the simulator used in this work, and Profs. Dimitra Panagou and Sze Zheng Yong for helpful suggestion throughout this project.

Appendix

A. Iterative invariant set computation

Robust controlled invariant sets can be computed using an iterative algorithm that starts with the safe set and shrinks it by eliminating the parts that cannot remain in the safe set at each iteration [23], [16]. A pseudo-code for such an algorithm is given in Alg.1 for a generic discrete-time system Σ.

Algorithm 1 Outside-in invariant set calculation within a safe set $\mathcal{X}_{safe}$

```
1: function GETCINVS($\Sigma$, $\mathcal{X}_{safe}$)
2:     $\mathcal{C}$ ←underapproximate($\mathcal{X}_{safe}$)
3:     while True
4:         $\mathcal{C}_{\text{pre}}$ ←underapproximate(pre($\Sigma, \mathcal{C}$) $\cap\, \mathcal{C}$)
5:         if $\mathcal{C} \subseteq \mathcal{C}_{\text{pre}}$
6:             break
7:         else $\mathcal{C} \leftarrow \mathcal{C}_{\text{pre}}$
8:     return $\mathcal{C}$
9: end function
```

In addition to standard set operations, the algorithm includes, the subrountines "underapproximate" and "pre". "underapproximate" refers to any techniques that returns a subset of the set in its argument, which might be needed when exact representation of the sets is impossible or costly. For a system Σ and a set $\mathcal{X}$ of states, "pre($\Sigma, \mathcal{X}$)" computes the controllable predecessors of $\mathcal{X}$, that is, the set of all states which can be enforced to be in $\mathcal{X}$ in one step by the choice of proper control input.

For the case with polytopic invariant sets and linear systems with polytopic safe sets, we can remove the "underapproximate" in lines 2 and 4 as the "pre" can be exactly computed, and if Alg. 1 terminates, the exact maximal controlled invariant set is computed. The toolbox PCIS [24] implements this algorithm and is used in this paper. On the other hand, for ellipsoidal sets under approximation of the exact sets can provide extra computational benefits. We briefly summarize how to implement the algorithm with ellipsoidal sets in a way that is computationally more efficient (yet more conservative) than polytopes in the next section.

B. Ellipsoid invariant set computation

Given $M \in \mathbb{R}^{n\times n}$ positive definite and $q \in \mathbb{R}^n$ we can define an ellipsoidal set centered at q as

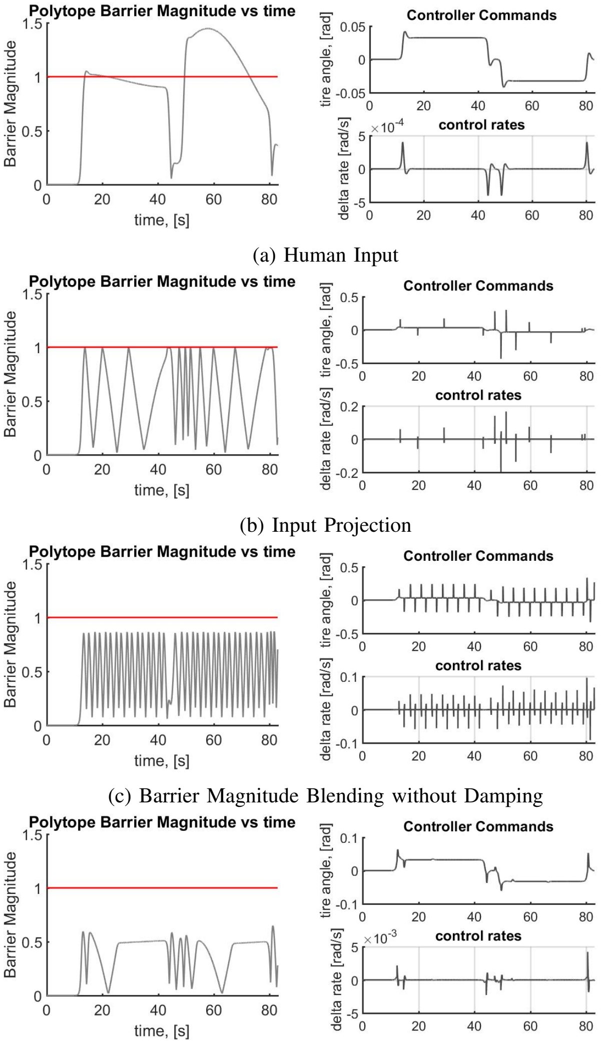
$$\mathcal{E}_{(M,q)} = \{x \in \mathbb{R}^n | (x-q)^T M (x-q) \leq 1\} \qquad (23)$$

To implement Alg. 1 with ellipsoidal approximations, we need the following operations. On line 2, we need to find an ellipsoid inside the polytope $\mathcal{X}_{safe}$. On line 4, we need to compute an ellipsoidal under approximation of (i) "pre", for which we use Alg. 2, (ii) an intersection of two ellipsoids, which can be done in closed-form for concentric ellipsoids as the ones that we use. In Alg. 2, Minkowski Sum, $\oplus$, and Minkowski Difference, $\ominus$ operators are used, which are defined as:

$$\begin{aligned} \mathcal{S}_1 \oplus \mathcal{S}_2 &= \{x \mid \exists p_1 \in S_1, \exists p_2 \in S_2, x = p_1 + p_2\} \\ \mathcal{S}_1 \ominus \mathcal{S}_2 &= \{x \mid \forall p_2 \in \mathcal{S}_2, \exists p_1 \in \mathcal{S}_1, x = p_1 - p_2\} \end{aligned} \qquad (24)$$

TABLE II: A summary of the key metrics measured when evaluating the different blending methods with different barriers and user inputs.

$\mathcal{C}_{inv}$	Method	Human Input	u'_{max} (rad/s)	Time Blended (s)	# of Engagements	Total Deviation (rad)	Average Deviation (rad)
Polytope	Projection	Aggressive	0.1662	2.9920	13	20.3952	0.0530
Polytope	Blending w/o Damping	Aggressive	0.0954	7.5520	38	135.2832	0.1429
Polytope	Blending w/ Damping	Aggressive	0.0042	6.6640	10	9.0007	0.0021
Ellipsoid	Projection	Mild	0.0463	1.6160	9	3.4637	0.0157
Ellipsoid	Blending w/o Damping	Mild	0.0424	1.3360	9	3.4202	0.0192
Ellipsoid	Blending w/ Damping	Mild	0.0050	3.2480	8	2.9401	0.0007
Polytope	Projection	Mild	0.0005	0	0	0	0
Polytope	Blending w/o Damping	Mild	0.0005	0	0	0	0
Polytope	Blending w/ Damping	Mild	0.0005	0	0	0	0



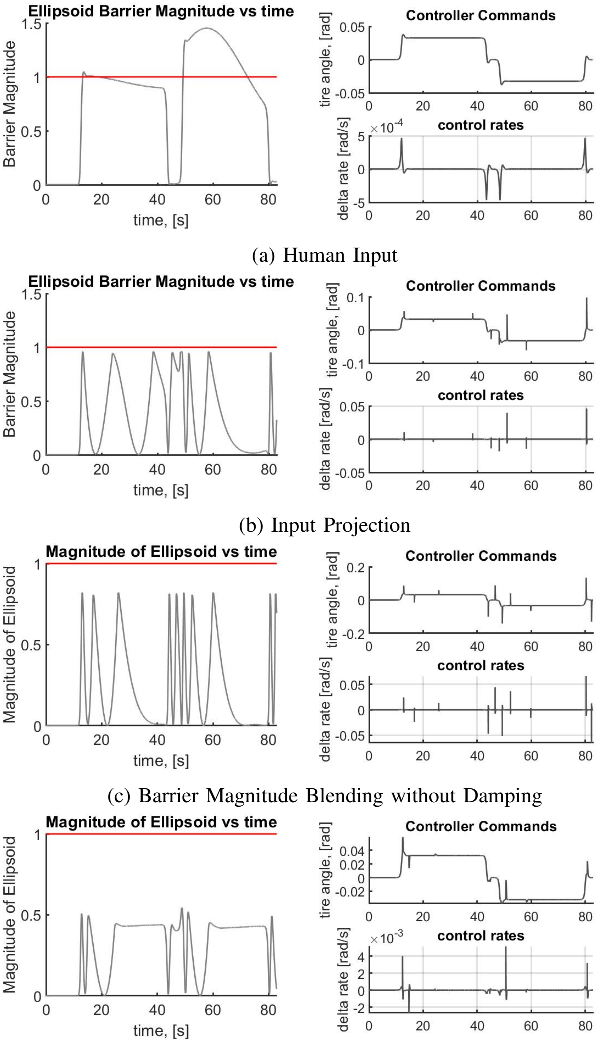
(a) Human Input

(b) Input Projection

(c) Barrier Magnitude Blending without Damping

(d) Barrier Magnitude Blending

Fig. 7: Polytope barrier magnitudes (red line representing barrier boundary) and control inputs for the following scenarios of unsupervised, supervised with admissible input set, supervised with barrier magnitude blending without damping, and barrier magnitude blending with damping. 7b, 7c, and 7d all supervised the input provided in 7a. 7a (aggressive human input) was selected to initially violate the polytope barrier during the turns.



(a) Human Input

(b) Input Projection

(c) Barrier Magnitude Blending without Damping

(d) Barrier Magnitude Blending

Fig. 8: Ellipsoid barrier magnitudes (red line representing barrier boundary) and control inputs for the following scenarios of unsupervised, supervised with admissible input set, supervised with barrier magnitude blending without damping, and barrier magnitude blending with damping. 8b, 8c, and 8d all supervised the input provided in 8a. 8a (mild human input) was selected to initially violate the ellipsoid barrier during the turns.

Algorithm 2 Calculate control pre set of a ellipsoidal set $\mathcal{E} = \mathcal{E}_{(M,q)}$, for a system of the form (1)

```
function PRE($\Sigma, \mathcal{E}$)
    $\mathcal{E}' \leftarrow$ underapproximate($\mathcal{E} \ominus E_2\mathcal{D}_{um}$)
    if $\mathcal{E}'$ is $\emptyset$
        return $\emptyset$
    $\mathcal{E}' \leftarrow$ underapproximate($\mathcal{E}' \oplus B\mathcal{U}$)
    $\mathcal{E}' \leftarrow$ underapproximate($\mathcal{E}' \ominus E_1\mathcal{D}_m$)
    if $\mathcal{E}'$ is $\emptyset$
        return $\emptyset$
    $\mathcal{E}'.M \leftarrow A^\intercal(\mathcal{E}'.M)A$
    $\mathcal{E}'.q \leftarrow A^{-1}(\mathcal{E}'.q)$
    return $\mathcal{E}'$
end function
```

Then, we use Alg. 1 to continually update the set $\mathcal{C}$ until it converges to an invariant set or returns an empty set. The key advantage of this method is that in practice is converges much more rapidly than PCIS. One relaxation is taking an intersection with $\mathcal{X}_{Safe}$ instead of $\mathcal{C}$ on line 4, for which one loses convergence guarantees that come with monotonic shrinkage but we observe termination is achieved much faster in practice. The implementation of Algs. 1 and 2 for ellipsoid invariant set computation are available in [25].

References

[1] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, 2010.

[2] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng, and P. Tabuada, "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1294–1307, 2015.

[3] P. Nilsson and N. Ozay, *Provably-Correct Compositional Synthesis of Vehicle Safety Systems*. Cham: Springer International Publishing, 2019, pp. 97–122. [Online]. Available: https://doi.org/10.1007/978-3-319-97301-2_6

[4] S. W. Smith, P. Nilsson, and N. Ozay, "Interdependence quantification for compositional control synthesis with an application in vehicle safety systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 5700–5707.

[5] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2017.

[6] K. Berntorp, R. Bai, K. F. Erliksson, C. Danielson, A. Weiss, and S. Di Cairano, "Positive invariant sets for safe integrated vehicle motion planning and control," *IEEE Transactions on Intelligent Vehicles*, 2019.

[7] C. Liu, C.-Y. Lin, and M. Tomizuka, "The convex feasible set algorithm for real time optimization in motion planning," *SIAM Journal on Control and optimization*, vol. 56, no. 4, pp. 2712–2733, 2018.

[8] M. Desai and H. A. Yanco, "Blending human and robot inputs for sliding scale autonomy," in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.* IEEE, 2005, pp. 537–542.

[9] J. G. Storms and D. M. Tilbury, "Blending of human and obstacle avoidance control for a high speed mobile robot," in *2014 American Control Conference*. IEEE, 2014, pp. 3488–3493.

[10] A. Goil, M. Derry, and B. D. Argall, "Using machine learning to blend human and robot controls for assisted wheelchair navigation," in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2013, pp. 1–6.

[11] C. Ezeh, P. Trautman, L. Devigne, V. Bureau, M. Babel, and T. Carlson, "Probabilistic vs linear blending approaches to shared control for wheelchair driving," in *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2017, pp. 835–840.

[12] A. Bhardwaj, A. H. Ghasemi, Y. Zheng, H. Febbo, P. Jayakumar, T. Ersal, J. L. Stein, and R. B. Gillespie, "Who's the boss? Arbitrating control authority between a human driver and automation system," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 68, pp. 144–160, 2020.

[13] S. Anderson, S. Peters, T. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, pp. 190–216, 10 2010.

[14] Y. E. Sahin, Z. Liu, K. Rutledge, D. Panagou, S. Z. Yong, and N. Ozay, "Intention-aware supervisory control with driving safety applications," in *2019 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2019, pp. 1–8.

[15] Z. Liu, L. Yang, and N. Ozay, "Scalable computation of controlled invariant sets for discrete-time linear systems with input delays," in *2020 American Control Conference (ACC)*. IEEE, 2020, (accepted). [Online]. Available: https://www.dropbox.com/s/ph37d02yy7pn5zy/root.pdf?dl=0

[16] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[17] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[18] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/~mpt.

[19] B. Legat, P. Tabuada, and R. M. Jungers, "Computing controlled invariant sets for hybrid systems with applications to model-predictive control," *IFAC-PapersOnLine*, vol. 51, no. 16, pp. 193–198, 2018.

[20] E. J. Rossetter and J. C. Gerdes, "Lyapunov Based Performance Guarantees for the Potential Field Lane-keeping Assistance System," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 3, pp. 510–522, 08 2005. [Online]. Available: https://doi.org/10.1115/1.2192835

[21] M. Bujarbaruah, X. Zhang, H. E. Tseng, and F. Borrelli, "Adaptive MPC for autonomous lane keeping," *CoRR*, vol. abs/1806.04335, 2018. [Online]. Available: http://arxiv.org/abs/1806.04335

[22] A. Benine-Neto, S. Scalzi, S. Mammar, and M. Netto, "Dynamic controller for lane keeping and obstacle avoidance assistance system," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2010, pp. 1363–1368.

[23] D. P. Bertsekas, "Infinite Time Reachability of State-Space Regions by Using Feedback Control," *IEEE Trans. Autom. Control*, vol. 17, no. 5, pp. 604–613, 1972.

[24] P. Nilsson, "PCIS," https://github.com/pettni/pcis.

[25] T. Ge, "Ellipsoidal Controlled Invariant Set," https://github.com/getc1995/ellipsoidal_cinv.