Proceedings of xxx xxx xxx vol xxx:1–8, 2021

Supplementary Material for BarrierNet: A Safety-Guaranteed Layer for Neural Networks

Wei Xiao, Ramin Hasani, Xiao Li and Daniela Rus
{WEIXY, RHASANI, XIAOLI, RUS}@MIT.EDU

Abstract

This paper introduces differentiable higher-order control barrier functions (CBF) that are end-to-end trainable as part of a learning system. CBFs are usually overly conservative, while guaranteeing safety. Here, we address their conservativeness by softening their definitions using environmental dependencies without losing safety guarantees, and embed them into differentiable quadratic programs. These novel safety layers, termed a BarrierNet, can be used in conjunction with any neural network-based controller, and can be trained by gradient descent. BarrierNet allows the safety constraints of a neural controller be adaptable to changing environments. We evaluate them on a series of control problems such as traffic merging and robot navigation in 2D and 3D spaces, and demonstrate their effectiveness compared to state-of-the-art approaches.

Keywords: Neural Network, Safety Guarantees, Control Barrier Function

1. Summary of Notations

Table 1: Notation

Symbol	**Definition**
t	time
θ	BarrierNet parameters
$\boldsymbol{x} \in \mathbb{R}^n$	system state
$\boldsymbol{z} \in \mathbb{R}^d$	observation variable
$\boldsymbol{u} \in \mathbb{R}^q$	control
$\alpha : [0, a) \to [0, \infty), a > 0$	class $\mathcal{K}$ function
$\beta : \mathbb{R} \to \mathbb{R}$	extended class $\mathcal{K}$ function
$b : \mathbb{R}^n \to \mathbb{R}$	safety constraint
$\psi : \mathbb{R}^n \to \mathbb{R}$	CBF
$p : \mathbb{R}^d \to \mathbb{R}^{>0}$	penalty function
$C \subset \mathbb{R}^n$	safe set
$l : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}$	similarity measure
$f : \mathbb{R}^n \to \mathbb{R}^n$	affine dynamics drift term
$g : \mathbb{R}^n \to \mathbb{R}^{n\times q}$	affine dynamics control term

2. Traffic Merging Problem Setup

Experiment setup. The traffic merging problem arises when traffic must be joined from two different roads, usually associated with a main lane and a merging lane as shown in Fig.1. We consider the case where all traffic consisting of controlled autonomous vehicles (CAVs) arrive randomly at the origin (O and O') and join at the Merging Point (MP) M where a lateral collision may occur. The segment from the origin to the merging point M has length L for both lanes, and is called the Control Zone (CZ). All CAVs do not overtake each other in the CZ as each road consists of a single lane. A coordinator is associated with the MP whose function is to maintain a First-In-First-Out (FIFO) queue of CAVs based on their arrival time at the CZ. The coordinator also enables real-time communication among the CAVs that are in the CZ including the last one leaving the CZ. The FIFO assumption, imposed so that CAVs cross the MP in their order of arrival, is made for simplicity and often to ensure fairness.

Notation. x_k, v_k, u_k denote the along-lane position, speed and acceleration (control) of CAV k, respectively. t_k^0, t_k^m denote the arrival time of CAV k at the origin and the merging point, respectively. z_{k,k_p} denotes the along lane distance between CAV k and its preceding CAV k_p, as shown in Fig. 1.

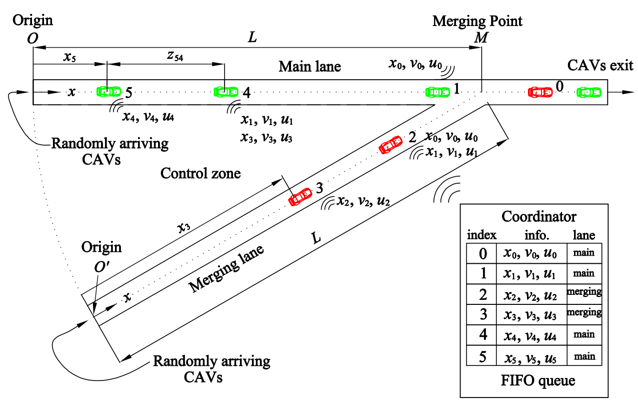


Figure 1: A traffic merging problem. A collision may happen at the merging point as well as everywhere within the control zone.

Our goal is to jointly minimize all vehicles' travel time and energy consumption in the control zone. Written as an objective function, we have

$$\min_{u_k(t)} \beta(t_k^m - t_k^0) + \int_{t_k^0}^{t_k^m} \frac{1}{2} u_k^2(t) dt, \tag{1}$$

where u_k is the vehicle's control (acceleration), and $\beta > 0$ is a weight controlling the relative magnitude of travel time and energy consumption. We assume double integrator dynamics for all vehicles.

Each vehicle k should satisfy the following rear-end safety constraint if its preceding vehicle k_p is in the same lane:

$$z_{k,k_p}(t) \geq \phi v_k(t) + \delta, \ \ \forall t \in [t_k^0, t_k^m], \tag{2}$$

where $z_{k,k_p} = x_{k_p} - x_k$ denotes the along-lane distance between k and k_p, ϕ is the reaction time (usually takes $1.8s$) and $\delta \geq 0$.

The traffic merging problem is to find an optimal control that minimizes (1), subject to (2). We assume vehicle k has access to only the information of its immediate neighbors from the coordinator (shown in Fig. 1), such as the preceding vehicle k_p. This merging problem can be solved analytically by optimal control methods (Xiao and Cassandras, 2021), but at the cost of extensive computation, and the solution becomes complicated when one or more constraints become active in an optimal trajectory, hence possibly prohibitive for real-time implementation.

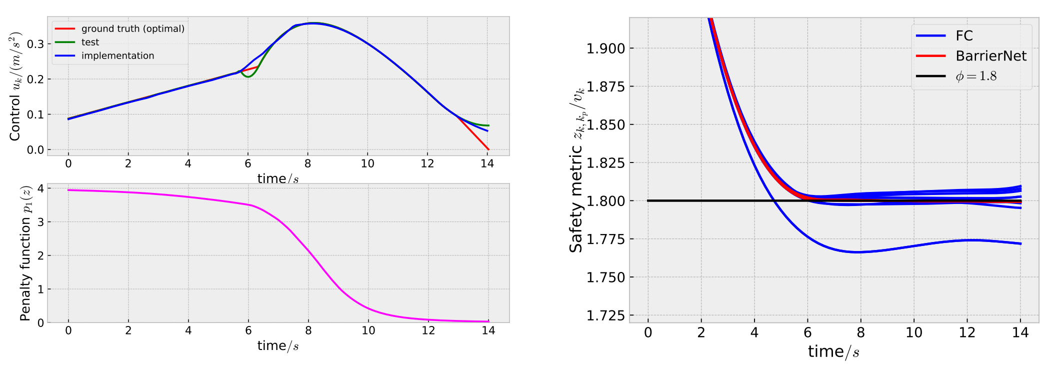
BarrierNet design. We enforce the safety constraint (2) by a CBF $b(z_{k,k_p}, v_k) = z_{k,k_p}(t) - \phi v_k(t) - \delta$, and any control input u_k should satisfy the CBF constraint which in this case (choose α_1 as a linear function in a CBF) is :

$$\varphi u_k(t) \leq v_k(t) - v_{k_p}(t) + p_1(\boldsymbol{z})(z_{k,k_p}(t) - \phi v_k(t) - \delta) \tag{3}$$

where v_k is the speed of vehicle k and $\boldsymbol{z} = (x_{k_p}, v_{k_p}, x_k, v_k)$ is the input of the neural network model (to be designed later). $p_1(\boldsymbol{z})$ is called a penalty in the CBF that addresses the conservativeness of the CBF method. The cost in the neuron of the BarrierNet is given by:

$$\min_{u_k} (u_k - f_1(\boldsymbol{z}))^2 \tag{4}$$

where $f_1(\boldsymbol{z})$ is a reference to be trained (the output of the FC network). Then, we create a neural network model whose structure is composed by a fully connected (FC) network (an input layer and two hidden layers) followed by a BarrierNet. The input of the FC network is $\boldsymbol{z}$, and its output is the penalty $p_1(\boldsymbol{z})$ and the reference $f_1(\boldsymbol{z})$. While the input of the BarrierNet is the penalty $p_1(\boldsymbol{z})$ and the reference $f_1(\boldsymbol{z})$, and its output is applied to control a vehicle k in the control zone.



(a) The control and penalty from the BarrierNet. (b) Safety metrics.

Figure 2: The safety comparison (under 10 trained models) between the BarrierNet and a FC network when training using the optimal controller ($\delta = 0$). If $z_{k,k_p}/v_k$ is above the line $\phi = 1.8$, then safety is guaranteed. We observe that only neural network agents equipped with BarrierNet satisfy this condition.

Results and analysis under optimal nominal controller. In an optimal controller, the original safety constraint is active after around $6s$, as shown in Fig. 2a. Therefore, the sampling trajectory is on the safety boundary, and inter-sampling effect becomes important in this case. Since we do not consider the inter-sampling effect in this paper, the safety metric of the BarrierNet might go below the lower bound $\phi = 1.8$, as the red curves shown in Fig. 2b. However, due to the Lyapunov property of the CBF, the safety metric will always stay close to the lower bound $\phi = 1.8$. The solutions for 10 trained models are also consistent. In a FC network, the safety metrics vary under different trained models, and the safety constraint might be violated, as the blue curves shown in Fig. 2b.

3. 2D Robot Navigation

Experiment setup. We consider a robot navigation problem with obstacle avoidance. In this case, we consider nonlinear dynamics with two control inputs and nonlinear safety constraints. The robot navigates according to the following unicycle model for a wheeled mobile robot:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5}$$

where $\boldsymbol{x} := (x, y, \theta, v), \boldsymbol{u} = (u_1, u_2)$, x, y denote the robot's 2D coordinates, θ denotes the heading angle of the robot, v denotes the linear speed, and u_1, u_2 denote the two control inputs for turning and acceleration.

BarrierNet design. The robot is required to avoid a circular obstacle in its path, i.e, the state of the robot should satisfy:

$$(x - x_o)^2 + (y - y_o)^2 \geq R^2, \tag{6}$$

where $(x_o, y_o) \in \mathbb{R}^2$ denotes the location of the obstacle, and $R > 0$ is the radius of the obstacle.

The goal is to minimize the control input effort, while subject to the safety constraint (6) as the robot approaches its destination, as shown in Fig. 3.

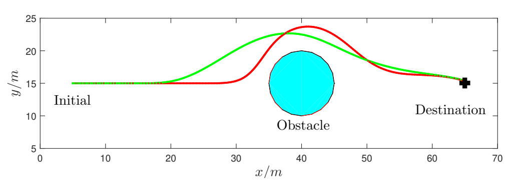


Figure 3: A 2D robot navigation problem. The robot is required to avoid the obstacle in its path. The trajectories (the red and green ones) vary under different definitions of HOCBFs that enforce the safety cosntraint (6).

The relative degree of the safety constraint (6) is 2 with respect to the dynamics (5), thus, we use a HOCBF $b(\boldsymbol{x}) = (x - x_o)^2 + (y - y_o)^2 - R^2$ to enforce it. Any control input $\boldsymbol{u}$ should satisfy the HOCBF constraint which in this case (choose α_1, α_2 in a HOCBF as linear functions) is:

$$-L_g L_f b(\boldsymbol{x})\boldsymbol{u} \leq L_f^2 b(\boldsymbol{x}) + (p_1(\boldsymbol{z}) + p_2(\boldsymbol{z}))L_f b(\boldsymbol{x}) + (\dot{p}_1(\boldsymbol{z}) + p_1(\boldsymbol{z})p_2(\boldsymbol{z}))b(\boldsymbol{x}) \tag{7}$$

where

$$\begin{aligned} L_g L_f b(\boldsymbol{x}) &= [-2(x-x_o)v\sin\theta + 2(y-y_o)v\cos\theta, \quad 2(x-x_o)\cos\theta + 2(y-y_o)\sin\theta] \\ L_f^2 b(\boldsymbol{x}) &= 2v^2 \\ L_f b(\boldsymbol{x}) &= 2(x-x_o)v\cos\theta + 2(y-y_o)v\sin\theta \end{aligned} \tag{8}$$

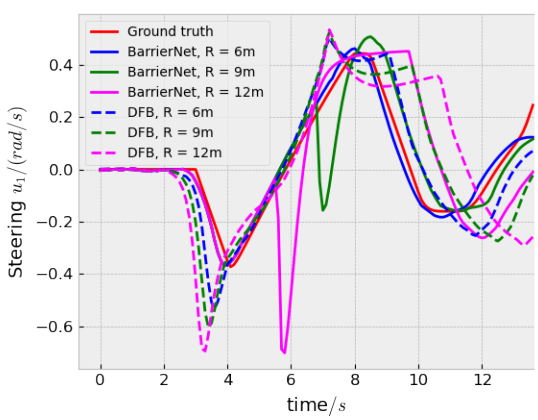
In the above equations, $\boldsymbol{z} = (\boldsymbol{x}, x_d)$ is the input to the model, where $x_d \in \mathbb{R}^2$ is the location of the destination, and $p_1(\boldsymbol{z}), p_2(\boldsymbol{z})$ are the trainable penalty functions. $\dot{p}_1(\boldsymbol{x})$ could be set as 0 due to the discretization solving method of the QP.

The cost in the neuron of the BarrierNet is given by:

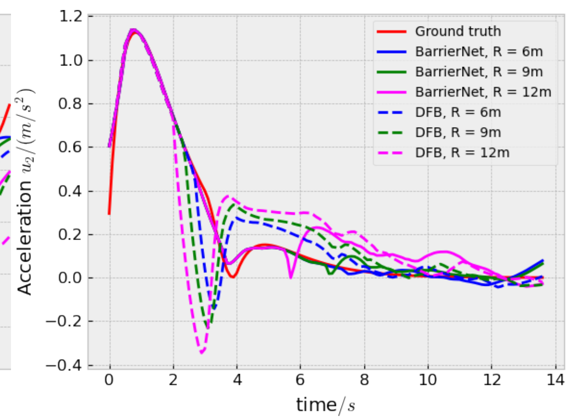
$$\min_{\boldsymbol{u}} (u_1 - f_1(\boldsymbol{z}))^2 + (u_2 - f_2(\boldsymbol{z}))^2 \tag{9}$$

where $f_1(\boldsymbol{z}), f_2(\boldsymbol{z})$ are references controls provided by the upstream network (the outputs of the FC network).

Results and dicussion. When we increase the obstacle size during implementation, the controls u_1, u_2 from the BarrierNet and DFB deviate from the ground true, as shown in Figs. 4a and 4b. This is due to the fact that the BarrierNet and DFB will always ensure safety first.



(a) Control u_1.



(b) Control u_2.

Figure 4: The controls from the BarrierNet and DFB under different obstacle sizes. The BarrierNet and DFB are trained under the obstacle size $R = 6m$. The results refer to the case that the trained BarrierNet/DFB controller is used to drive a robot to its destination. When we increase the obstacle size during implementation, the outputs (controls of the robot) of the BarrierNet and the DFB will adjust accordingly in order to guarantee safety, as shown by the blue and cyan curves. However, the BarrierNet tends to be less conservative for unseen situations.

The profiles of the penalty functions $p_1(\boldsymbol{z}), p_2(\boldsymbol{z})$ in the BarrierNet are shown in the paper (Xiao et al., 2021). The values of the penalty functions vary when the robot approaches the obstacle and gets to its destination, and it shows the adaptivity of the BarrierNet in the sense that with the varying penalty functions, a BarrierNet can produce desired control signals given by labels (ground truth). This is due to the fact the varying penalty functions soften the HOCBF constraint without loosing safety guarantees.

4. 3D Robot Navigation

Experiment setup. We consider a robot navigation problem with obstacle avoidance in 3D space. In this case, we consider complicated superquadratic safety constraints. The robot navigates according to the double integrator dynamics. The state of the robot is $\boldsymbol{x} = (p_x, v_x, p_y, v_y, p_z, v_z) \in \mathbb{R}^6$, in which the components denote the position and speed along x, y, z axes, respectively. The three control inputs u_1, u_2, u_3 are the acceleration along x, y, z axes, respectively.
BarrierNet design. The robot is required to avoid a superquadratic obstacle in its path, i.e, the state of the robot should satisfy:

$$(p_x - x_o)^4 + (p_y - y_o)^4 + (p_z - z_o)^4 \geq R^4, \tag{10}$$

where $(x_o, y_o, z_o) \in \mathbb{R}^3$ denotes the location of the obstacle, and $R > 0$ is the half-length of the superquadratic obstacle.

The goal is to minimize the control input effort, while subject to the safety constraint (10) as the robot approaches its destination, as shown in Fig. 5.

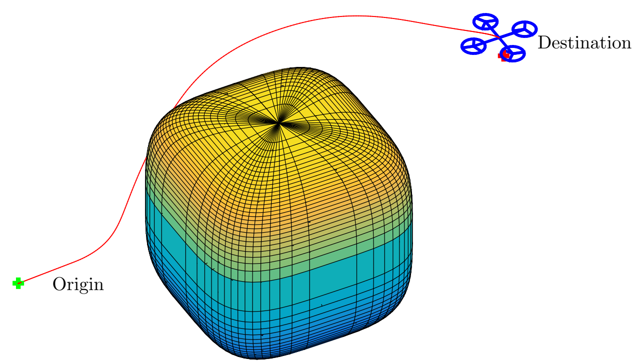


Figure 5: A 3D robot navigation problem. The robot is required to avoid the obstacle in its path.

The relative degree of the safety constraint (10) is 2 with respect to the dynamics, thus, we use a HOCBF $b(\boldsymbol{x}) = (p_x - x_o)^4 + (p_y - y_o)^4 + (p_z - z_o)^4 - R^4$ to enforce it. Any control input $\boldsymbol{u}$ should satisfy the HOCBF constraint which in this case (choose α_1, α_2 in a HOCBF as linear functions) is:

$$-L_g L_f b(\boldsymbol{x})\boldsymbol{u} \leq L_f^2 b(\boldsymbol{x}) + (p_1(\boldsymbol{z}) + p_2(\boldsymbol{z}))L_f b(\boldsymbol{x}) + (\dot{p}_1(\boldsymbol{z}) + p_1(\boldsymbol{z})p_2(\boldsymbol{z}))b(\boldsymbol{x}) \tag{11}$$

where

$$\begin{aligned} L_g L_f b(\boldsymbol{x}) &= [4(p_x - x_o)^3, \quad 4(p_y - y_o)^3, \quad 4(p_z - z_o)^3] \\ L_f^2 b(\boldsymbol{x}) &= 12(p_x - x_o)^2 v_x^2 + 12(p_y - y_o)^2 v_y^2 + 12(p_z - x_o)^2 v_z^2 \\ L_f b(\boldsymbol{x}) &= 4(p_x - x_o)^3 v_x + 4(p_y - y_o)^3 v_y + 4(p_z - z_o)^3 v_z \end{aligned} \tag{12}$$

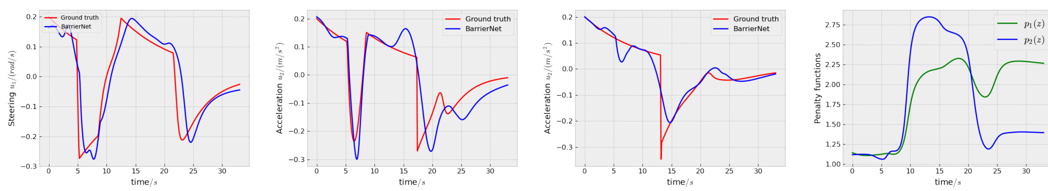
In the above equations, $\boldsymbol{z} = \boldsymbol{x}$ is the input to the model, $p_1(\boldsymbol{z}), p_2(\boldsymbol{z})$ are the trainable penalty functions. $\dot{p}_1(\boldsymbol{x})$ is also set as 0 as in the 2D navigation case.

The cost in the neuron of the BarrierNet is given by:

$$\min_{\boldsymbol{u}} (u_1 - f_1(\boldsymbol{z}))^2 + (u_2 - f_2(\boldsymbol{z}))^2 + (u_3 - f_3(\boldsymbol{z}))^2 \tag{13}$$

where $f_1(\boldsymbol{z}), f_2(\boldsymbol{z}), f_3(\boldsymbol{z})$ are references controls provided by the upstream network (the outputs of the FC network).

Results and dicussion. The training and testing results are shown in Figs. 6. The controls from the BarrierNet have some errors with repsect to the ground truth, and this is due to the complicated safety constraint (10). We can improve the tracking accuracy with deeper BarrierNet models (not the focus of this paper). Nevertheless, the implementation trajectory under the BarrierNet controller is close to the ground truth, as shown in (Xiao et al., 2021).



(a) Control u_1. (b) Control u_2. (c) Control u_3. (d) Penalty functions.

Figure 6: The controls and penalty functions from the and BarrierNet. The results refer to the case that the trained BarrierNet controller is used to drive a robot to its destination. The varying penalty functions allow the generation of desired control signals and trajectories (given by training labels), and demonstrate the adaptivity of the BarrierNet with safety guarantees.

The profiles of the penalty functions $p_1(\boldsymbol{z}), p_2(\boldsymbol{z})$ in the BarrierNet are shown in Fig. 6d. The values of the penalty function variations demonstrate the adaptivity of the BarrierNet in the sense that with the varying penalty functions, a BarrierNet can produce desired control signals given by labels (ground truth).

Acknowledgments

This research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was further supported by The Boeing Company,and the Office of Naval Research (ONR) Grant N00014-18-1-2830. We are grateful to the members of the Capgemini research team for discussing the importance of safety and stability of machine learning.

References

W. Xiao and C. G. Cassandras. Decentralized optimal merging control for connected and automated vehicles with safety constraint guarantees. *Automatica*, 123:109333, 2021.

Wei Xiao, Ramin Hasani, Xiao Li, and Daniela Rus. Barriernet: A safety-guaranteed layer for neural networks. *preprint arXiv:2111.11277*, 2021.