Training Neural Network Controllers Using Control Barrier Functions in the Presence of Disturbances

Shakiba Yaghoubi and Georgios Fainekos and Sriram Sankaranarayanan

***Abstract*— Control Barrier Functions (CBF) have been recently utilized in the design of provably safe feedback control laws for nonlinear systems. These feedback control methods typically compute the next control input by solving an online Quadratic Program (QP). Solving QPs in real-time can be a computationally expensive process for resource-constrained systems. In the presence of disturbances, finding CBF-based safe control inputs can get even more time consuming as finding the worst-case of the disturbance requires solving a nonlinear program in general. In this work, we propose to use imitation learning to learn Neural Network based feedback controllers which will satisfy the CBF constraints. In the process, we also develop a new class of High Order CBF for systems under external disturbances. We demonstrate the framework on a unicycle model subject to external disturbances, e.g., wind or currents.**

***Index Terms*— Barrier Function, Disturbance, Neural Network Controller, Imitation Learning**

I. Introduction

Control Barrier Functions (CBF) have enabled the design of provable safe feedback controllers for a number of different systems such as adaptive cruise control [3], bipedal robot walking and long term autonomy [2]. CBF - along with Control Lyapunov Functions (CLF) - are typically part of the constraints of a Quadratic Program (QP) whose solution computes sub-optimal control inputs that guarantee safe system operation (while stabilizing to a desired operating point or trajectory [17]). The CBF theory has been instrumental in developing safety critical controllers for nonlinear systems; however, it also has some limitations. First, the resulting controller may not be robust to external inputs and parameter or model inaccuracies, and to the best of our knowledge, robust high order control barrier functions have not been studied before. Second, finding the sub-optimal control inputs constrained to the CBF and CLF constraints requires the online/real-time solution of a QP, which typically cannot satisfy hard real time constraints.

In this work, first, we design robust control inputs based on high order control barrier functions, and second, we propose to use Neural Network (NN) based feedback controllers trained using the CBF-based QPs to accelerate computation of the desired control inputs in the run-time. Shallow NN-based controllers require limited memory and computational power and, therefore, they can address the second problem. In addition, NN-based controllers can be trained using both simulated and real data. By training the NNs to imitate the solution of the robust formulation of the CBF-based QPs, it can be used to quickly compute control inputs that tolerate model uncertainty and inaccuracies.

In particular, in this work, we make the following contributions. First, we extend results from CBF [2] and High Order CBF (HOCBF) [24] to nonlinear systems with affine controls and external disturbances. Second, we adapt an imitation learning algorithm ([20]) to train an NN-based controller from examples generated by the QP-based controller. Even though in these preliminary results, we did not use real data, we demonstrate that we can train the NN controller robustly over non-noisy system trajectories and apply the resulting controller to a system subject to external disturbances. Finally, even though in this paper, we do not address the provable safety of the resulting NN controller under all possible initial conditions, in the future, we plan to use tools like Sherlock [8], [9] to do so.

Related Work: Input-to-state safety of a set $C = \{x \mid h(x) \geq 0\}$ which ensures that trajectories of a nonlinear dynamical system in presence of "actuation errors" stay close to the set C, has been proved by enforcing the invariance of a larger set including C in [15]. The radius of the larger set depends on the L_∞ norm of the error. In this paper, firstly we consider a more general class of external inputs (that can represent measurement, or actuation errors as well as environment inputs like wind gusts and water currents) is applied to the system, and secondly, the control input is designed to reject the worst case disturbance to enforce invariance of the set C itself - when h has relative degree one w.r.t the control input - or its intersection with another set that depends on the derivatives of h.

The application of NN to control dynamical systems has a long history [14], [12], [21]. More recently, due to computational advances and available data [10], there has been a renewed interest in the utilization of NN in control systems. The works [26] and [7] utilize counterexample (adversarial sample) exploration to train NNs that seek to either satisfy a given property expressed in temporal logic or follow a reference trajectory. The work by [23] attempts to learn through simulations barrier certificates that can establish the safe operation of the closed loop system with an NN controller. On the other hand, [28] and [6] take a different

This work was partially funded by NSF CNS 1932068, NSF IIP 1361926 and the NSF I/UCRC Center for Embedded Systems

S. Yaghoubi is with the department of Computing Informatics and Decision System Engineering, Arizona State University, Tempe, AZ, USA syaghoub@asu.edu

G. Fainekos is with the department of Computing Informatics and Decision System Engineering, Arizona State University, Tempe, AZ, USA fainekos@asu.edu

S. Sankaranarayanan is with the department of Computer Science, University of Colorado Boulder, Boulder, CO, USA srirams@colorado.edu

approach: they approximate Model Predictive Controllers (MPC) using supervised reinforcement learning for an NN. Here, instead of approximating MPC, we approximate the solution of a QP constrained by HOCBF.

II. Preliminaries

Consider a nonlinear control system without disturbances and with affine control inputs:

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in X \subset \mathbb{R}^n$ is the system state, $u \in U \subset \mathbb{R}^l$ is the control input, and $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ are locally Lipschitz functions. Given an initial condition $x(0)$, we denote the solution of the system at time t with $x(t)$. A function $\alpha : \mathbb{R} \to \mathbb{R}$ is said to be an extended class $\mathcal{K}$ function iff α is strictly increasing and $\alpha(0) = 0$ [2].

Definition II.1 (Set Invariance [5])**.** *A set $C \subseteq \mathbb{R}^n$ is forward invariant w.r.t the system (1) iff for every $x(0) \in C$, its solution satisfies $x(t) \in C$ for all $t \geq 0$.*

Definition II.2 (Barrier Function)**.** *Let $h : X \to \mathbb{R}$ be a continuously differentiable function, $C = \{x \in X | h(x) \geq 0\}$, and α be a locally Lipschitz extended class $\mathcal{K}$ function. h is a* barrier function *iff for all $x \in C$*

$$\dot{h}(x) \geq -\alpha(h(x)) \tag{2}$$

Lemma II.1 ([11])**.** *If h is a barrier function for C, and α is as defined in Def. II.2 then C is a forward invariant set.*

Definition II.3 (Control Barrier Function [2])**.** *A continuous, differentiable function $h(x)$ is a Control Barrier Function (CBF) for the system (1), if there exist a class $\mathcal{K}$ function α such that for all $x \in C$:*

$$L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0 \tag{3}$$

where $L_f h(x) = \frac{\partial h}{\partial x}^\top f(x), L_g h(x) = \frac{\partial h}{\partial x}^\top g(x)$ are the first order Lie derivatives of the system.

Any Lipschitz continuous controller $u \in K_{cbf}(x) = \{u \in U \mid L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0\}$ results in a forward invariant set C for the system of Eq. (1).

Definition II.4 (Control Input Relative Degree of a Function)**.** *A continuously differentiable function h has a control input relative degree m w.r.t the system (1), if the first time that the control u appears in the derivatives of h along the system dynamics is in its m^{th} derivative.*

If the function h has a relative degree $m > 1$, $L_g h(x) = L_g^{m-1} h(x) = 0$. As a result Eq. (3) cannot be directly used for choosing safe controllers $u \in K_{cbf}(x)$. High Order Control Barrier Functions (HOCBF) were introduced in [18], and [24] to derive necessary conditions for guaranteeing set invariance. Assuming that the function h has a relative degree m w.r.t the system (1), define the series of functions $\psi_i : \mathbb{R}^n \to \mathbb{R}, i = 0, 1, \cdots, m$ and the corresponding sets $C_1, \cdots, C_m$ as follows:

$$\begin{aligned}
&\psi_0(x) = h(x) \\
&\psi_1(x) = \dot{\psi}_0(x) + \alpha_1(\psi_0(x)) && C_1 = \{x \mid \psi_0(x) \geq 0\} \\
&\psi_2(x) = \dot{\psi}_1(x) + \alpha_2(\psi_1(x)) && C_2 = \{x \mid \psi_1(x) \geq 0\} \\
&\quad\vdots && \quad\vdots \\
&\psi_m(x) = \dot{\psi}_{m-1}(x) + \alpha_m(\psi_{m-1}(x)) && C_m = \{x \mid \psi_{m-1}(x) \geq 0\}
\end{aligned} \tag{4}$$

where $\alpha_1, \alpha_2 \cdots, \alpha_m$ are class $\mathcal{K}$ functions of their arguments.

Definition II.5 (High Order Barrier Functions)**.** *A function $h : \mathbb{R}^n \to \mathbb{R}$ with a control input relative degree m is a High Order Barrier Function (HOBF) for system (1), if there exist differentiable class $\mathcal{K}$ functions $\alpha_1, \alpha_2 \cdots, \alpha_m$ such that for all $x \in C_1 \cap C_2 \cap \cdots \cap C_m$, we have: $\psi_m(x) \geq 0$. Under this condition, the set $C_1 \cap C_2 \cap \cdots \cap C_m$ is forward invariant.*

Definition II.6 (High Order Control Barrier Functions [24])**.** *A function $h : \mathbb{R}^n \to \mathbb{R}$ with a relative degree m is a High Order Control Barrier Function (HOCBF) for system (1), if there exist differentiable class $\mathcal{K}$ functions $\alpha_1, \alpha_2 \cdots, \alpha_m$ such that for all $x \in C_1 \cap C_2 \cap \cdots \cap C_m$:*

$$\begin{aligned}
\psi_m(x) = L_f^m h(x) + L_g L_f^{m-1} h(x)u + O(h(x)) \\
+ \alpha_m(\psi_{m-1}(x)) \geq 0
\end{aligned} \tag{5}$$

where $O(.)$ denotes the remaining Lie derivatives along f with degree less than or equal to $m - 1$.

Any Lipschitz controller $u \in K_{hocbf}(x) = \{u \in U \mid L_f^m h(x) + L_g L_f^{m-1} h(x)u + O(h(x)) + \alpha_m(\psi_{m-1}(x)) \geq 0\}$ renders the system safe, and the set $C_1 \cap C_2 \cap \cdots \cap C_m$ forward invariant.

III. Control Barrier Functions in presence of Disturbance

In this paper, the nonlinear control system (1) is considered in presence of disturbances as in the following:

$$\Sigma : \quad \dot{x} = f(x) + g(x)u + Mw, \quad x(0) \in X_0 \tag{6}$$

where x, u, f, g are defined as for the system of Eq. (1), X_0 is the set of initial conditions, and $w \in W \subset \mathbb{R}^l$ is the disturbance input. Each dimension of W which we denote by W_i defines an interval $[\underline{w_i}, \overline{w_i}]$ that the i^{th} element of w belongs to, M is a $n \times l$ zero-one matrix with at most one non-zero element in each row.

When disturbance is present, in order to guarantee the forward invariance of the set C, which we call the safe set, the condition in inequality (2) needs to be satisfied for all $w \in W$, including its worst case where it minimizes the left hand side of the inequality.

Definition III.1. *The continuously differentiable function h with control input relative degree one is a CBF in presence of Disturbance (CBFD) for the system of Eq. (6), if there exist* *a class $\mathcal{K}$ function α such that for all $x \in C$ and $w \in W$, the following inequality is satisfied*

$$L_f h(x) + L_g h(x)u + L_M h(x)w + \alpha(h(x)) \geq 0 \quad (7)$$

where $L_M h(x) = \frac{\partial h}{\partial x}^\top M$. Equivalently for all $x \in C$ the following inequality needs to hold:

$$L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq F^*_{L_M}(x) \quad (8)$$

where $F^*_{L_M}(x) = \max_{w \in W}(-L_M h(x)w)$

Definition III.2 (Disturbance Relative Degree of a Function)**.** *A continuously differentiable function h has a disturbance relative degree q w.r.t the system (1), if the first time that the disturbance w appears in the derivatives of h along the system dynamics is in its q^{th} derivative.*

If the disturbance relative degree of h is greater than 1, $L_M h(x) = 0$ and a CBF is a CBFD too. Otherwise, since $L_M h(x)w$ is linear in w, and $w \in W$ imposes linear constraints on w, the program $\max_{w \in W}(-L_M h(x)w)$ is a linear program for each $x \in C$ whose solution can be found and replaced in inequality (8) to define the set of control values that satisfy the following inequality:

$$K_{cbfd}(x) = \{u \in U \mid L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq F^*_{L_M}(x)\} \quad (9)$$

Theorem 1. *Given a CBFD h from Def. (III.1), any Lipschitz continuous controller $u \in K_{cbfd}(x)$ renders the set C forward invariant.*

Proof. The proof can be directly derived from Lemma II.1. To be explicit, if for all $x \in C$ and $w \in W$, $\dot{h}(x) = L_f h(x) + L_g h(x)u + L_M h(x)w \geq -\alpha(h(x))$, then the solutions to system (6) with $x(0) \in C$, satisfy $h(x(t)) \geq 0$. So based on Def. II.1, C is forward invariant. □

If the function h has a control input relative degree higher than one, the multiplier of u in Eq. (8), $L_g h(x)$ is equal to zero, so the choice of u will not affect the satisfaction of inequality (8). In the following section we will study HOCBFs in presence of disturbance.

IV. High order Control Barrier Functions in Presence of Disturbance

Assume that the continously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ has control input relative degree m and consider the series of functions $\psi_i : \mathbb{R}^n \to \mathbb{R}, i = 0, ..., m$ and their corresponding sets $C_1, \ldots, C_m$ as defined in Eq. (4).

Definition IV.1. *The function h is a High Order Barrier Function in presence of disturbance (HOBFD) for system (6), if there exist differentiable class $\mathcal{K}$ functions $\alpha_1, \alpha_2, \ldots, \alpha_m$ that define the functions $\psi_1, \cdots, \psi_m$, such that for all $x \in C_1 \cap C_2 \cap \cdots \cap C_m$, we have: $\psi_m(x) \geq 0$*

Definition IV.2. *The function h is a High Order Control Barrier Function in presence of disturbance (HOCBFD) for system (6), if there exist differentiable class $\mathcal{K}$ functions $\alpha_1, ..., \alpha_m$ that define the functions $\psi_1, ..., \psi_m$, s.t for all $x \in C_1 \cap C_2 \cap ... \cap C_m$ and $w \in W$:*

$$\psi_m(x) = L_f^m h(x) + L_g L_f^{m-1} h(x)u + P(x, w) + O(h(x)) + \alpha_m(\psi_{m-1}(x)) \geq 0 \quad (10)$$

where $P(x, w)$ is a function of x and w that separates all the terms including w in $\psi_m(x)$ from the rest, $O(.)$ denotes the remaining Lie derivatives along f with degree less than or equal to $m - 1$. Since equation (10) needs to be satisfied for all $w \in W$, we can equivalently write it as:

$$L_f^m h(x) + L_g L_f^{m-1} h(x)u + O(h(x)) + \alpha_m(\psi_{m-1}(x)) \geq F^*_P(x) \quad (11)$$

where $F^*_P(x) = \max_{w \in W}(-P(x, w))$

If the disturbance relative degree of h is greater than m, $P(x, w) = 0$ and any HOCBF is a HOCBFD, else if the disturbance relative degree is m, $P(x, w) = L_M L_f^{m-1} h(x)w$ and $\max_{w \in W}(-P(x, w))$ is a linear program. Otherwise $P(x, w)$ is a nonlinear function of w in general, and the solution to the nonlinear program $F^*_P(x) = \max_{w \in W}(-P(x, w))$ can be used to find the set of control inputs that satisfy inequality (11):

$$K_{hocbfd}(x) = \{u \in U \mid L_f^m h(x) + L_g L_f^{m-1} h(x)u + O(h(x)) + \alpha_m(\psi_{m-1}(x)) \geq F^*_P(x)\}$$

Theorem 2. *Given a HOCBFD h from Def. (IV.2), any Lipschitz continuous controller $u \in K_{hocbfd}(x)$ renders the set $C_1 \cap C_2 \cap \cdots \cap C_m$ forward invariant.*

Proof. Any controller $u \in K_{hocbfd}(x)$ enforces $\psi_m(x) \geq 0$ or equivalently $\dot{\psi}_{m-1}(x) \geq -\alpha_m(\psi_{m-1}(x))$ irrespective of the value of $w \in W$. Assuming that $x(0) \in C_1 \cap C_2 \cap \cdots \cap C_m$, and hence $x(0) \in C_m$, we have $\psi_{m-1}(x(0)) \geq 0$ which based on lemma II.1, leads to $\psi_{m-1}(x) \geq 0$ $(x \in C_m)$ or equivalently $\dot{\psi}_{m-2}(x) \geq -\alpha_{m-1}(\psi_{m-2}(x))$, again since $x(0) \in C_{m-1}$ this results in $\psi_{m-2}(x) \geq 0$ $(x \in C_{m-1})$. Continuing this reasoning, we can prove that $C_1 \cap C_2 \cap \cdots \cap C_m$ is forward invariant. □

Remark 3. *The functions $F^*_{L_M}, F^*_P(x)$ are Lipschitz and hence, it is possible to find Lipschitz continuous controllers $u \in K_{cbfd}(x)$ or $u \in K_{hocbfd}(x)$. In the following we prove Lipschitz continuity of , $F^*_P(x)$. Lipschitz continuity of $F^*_{L_M}(x)$ will follow.*

$$\begin{aligned}
&||F^*_P(x_2) - F^*_P(x_1)|| = ||\max_w(-P(x_2, w)) - \max_w(-P(x_1, w))|| = \\
&||\max_w(-P(x_2, w) + P(x_1, w) - P(x_1, w)) - \max_w(-P(x_1, w))|| \leq \\
&||\max_w(P(x_1, w) - P(x_2, w)) + \max_w(-P(x_1, w)) - \max_w(-P(x_1, w))|| \\
&= ||\max_w(P(x_1, w) - P(x_2, w))|| \leq L_p||x_2 - x_1||
\end{aligned}$$

The first inequality is true since $\max(f + g)(w) \leq \max f(w) + \max g(w)$, and the second inequality is true since for all w including the one that maximizes $(P(x_1, w) - P(x_2, w))$, we have $||(P(x_2, w) - P(x_1, w))|| \leq L_p||x_2 - x_1||$ where L_p is the Lipschitz constant for P.

Remark 4. *In order to use HOCBFDs to prove that all the trajectories of the system 6 starting from X_0 will never exit C_1, the sets $C_1, C_2, \cdots, C_m$ should have a nonempty interior, and the set of initial conditions of the system, X_0, should be a subset of $C_1 \cap C_2 \cap \cdots \cap C_m$. Note that if $X_0 \subset C_1$ ($h(x(0)) \geq 0$) except for special cases (see [24]) which we do not consider here, we can always choose $\alpha_1, \alpha_2 \cdots, \alpha_m$ such that $x_0 \in C_2 \cap \cdots \cap C_m$.*

Note that the problem $\max_{w \in W} (-P(x, w))$, is in general a nonlinear program and finding its optimal - or even suboptimal - solution can be time consuming. A special case of the problem is if we consider the linear class $\mathcal{K}$ functions $\alpha_1, \cdots, \alpha_{m-1}$ which will form Exponential Control Barrier Functions [2]. This makes $P(x, w)$ a polynomial function of degree at most m in w. In case of polynomial functions $\alpha_1, \cdots, \alpha_m$, $P(x, w)$ will be a polynomial function of w - potentially of higher degree than m.

A special case is when $m = 2$, and $\alpha_1, \cdots, \alpha_m$ are linear functions. In this case $P(x, w)$ is a quadratic function of w, and $\max_{w \in W} (-P(x, w))$ is a QP for each $x \in X$ that can be solved efficiently.

Example 1. *Consider the system $\dot{x}_1 = x_2 + w$, $\dot{x}_2 = u$ with $w \in [\underline{w}, \overline{w}]$. The control input should be designed such that the function $h(x) = x_1^2 - 1$ is a HOCBFD. We consider $\alpha_i(y) = y, i = 1, 2$, so we have $\alpha_i'(y) = \frac{\partial \alpha_i}{\partial y} = 1$, and as a result:*

$$\begin{aligned}\psi_2(x) &= \ddot{h}(x) + \alpha_1'(h(x))\dot{h}(x) + \alpha_2(\dot{h}(x) + \alpha_1(h(x))) \\ &= 2x_1 u + \underbrace{(4x_2 + 4x_1)w + 2w^2}_{P(x,w)} + 2x_2^2 + 4x_1x_2 + x_1^2 - 1\end{aligned}$$

observing that $w_{opt} = \arg\max_{\underline{w} < w < \overline{w}} (-2w^2 - (4x_2 + 4x_1)w)$ is a quadratic program that can be solved at each x, any Lipschitz controller in the set $K_{hocbfd}(x) = \{2x_1u + 2x_2^2 + 4x_1x_2 + x_1^2 - 1 \geq -2w_{opt}^2 - (4x_2 + 4x_1)w_{opt}\}$ will make the set $C_1 \cap C_2 = \{x \mid h(x) \geq 0\} \cap \{x \mid \dot{h}(x) + h(x) \geq 0\}$ forward invariant, hence any trajectory starting from this set will never exit the set even in presence of the worst case disturbance. When $\underline{w} = -0.1, \overline{w} = 0.1$, this set is shown in Fig. 1.

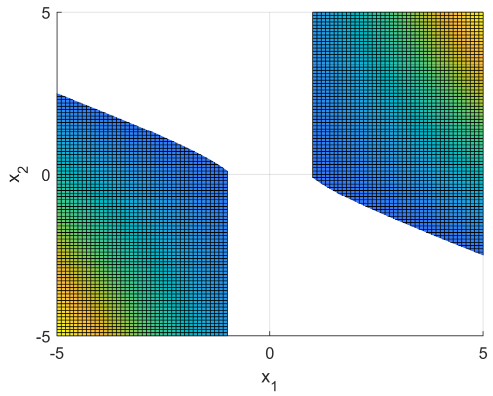


Fig. 1: The invariant sets of Example 1 for the worst case disturbance $w \in [-0.1, 0.1]$.

V. Control Optimization Problem with CBF Constraints

In order to find safe sub-optimal controllers, many recent works [16], [24], [3], [27], formulate optimization problems with quadratic costs in the control input u subject to CLF and CBF constraints (each CBF constraints corresponds to an unsafe set) which are linear in u. These QPs are solved every time new information about the states x is received, and the resulting control value u is used in the time period before new information is received. In presence of disturbances, in order to formulate the QPs with constraints of type (8) or (11), w_{opt} should be computed as a prerequisite. To compute w_{opt} one need to solve $\max_{w \in W} (-L_M h(x)w)$ or $\max_{w \in W} (-P(x, w))$ - depending on the relative degree m - for each barrier function or unsafe set. After computing w_{opt} it can be used in the following QP to find the semi-optimal CBFD-based control input:

$$\begin{aligned}&\min_{u \in U} u^T Q u \\ &\text{s.t. Eq (8) if } m = 1 \text{ or Eq. (11) if } m > 1\end{aligned}$$

As a result, formulating the quadratic program and solving it for evaluating the control input u may not be possible at run-time. In the following section, we present a paradigm for training NN controllers that predict the value of the control input resulting from the quadratic programs.

VI. Learning NN Controllers from Control Barrier Functions Using the DAGGER Algorithm

Imitation learning methods, which use expert demonstrations of good behavior to learn controllers, have proven to be very useful in practice [13], [1], [4], [19], [22]. While a typical method to imitation learning is to train a classifier/regressor to predict an expert's behavior given data from the encountered observations and expert's actions in them, it's been shown in [20] that using this framework, small errors made by the learner can lead to large errors over time. The reason is that in this scenario, the learner can encounter completely different observations than those it has been trained with, leading to error accumulation. Motivated by this, [20] presents an algorithm called DAGGER (Dataset Aggregation) that iteratively updates the training dataset with new observations encountered by the learner and their corresponding expert's actions and retrains the learner.

As described in Section V, formulating and solving the required quadratic programs may not be feasible at run-time. As a result, we use an algorithm inspired by the DAGGER algorithm to train NN controllers that predict the outcome of the quadratic program. In this regard, the QP acts as an expert that a NN imitates. An NN controller that has been trained offline can be used in a feedback loop to produce the desired control values online. The NN training algorithm is described in Alg. 1 in which it is assumed that $\pi^*(x, \Sigma, U, W)$ is an

Algorithm 1 Data set Aggregation for training NN using Quadratic Programs

Data: The dynamical system (6), the set of admissible control inputs U, the set of external inputs W, the set of initial conditions X_0, the constant $0 < p < 1$, maximum number of iterations N

Randomly choose the set X_0^s by sampling from X_0

Sample trajectories of the system (6) with initial conditions in X_0^s and input $\pi_0 = \pi^*(x, \Sigma, U, W)$

Initialize D with the pairs of visited states and corresponding control inputs: $D = (x, \pi^*(x, \Sigma, U, W))$

Train NN controller $\hat{\pi}_1$ on D

for $i = 1, ..., N$ **do**

  $\beta = p^i$

  Sample trajectories of the system (6) with $x(0) \in X_0^s$ and input $\pi_i = \beta\pi^*(x, \Sigma, U, W) + (1-\beta)\hat{\pi}_i(x)$

  Get dataset $D_i = (x, \pi^*(x, \Sigma, U, W))$ of visited states and corresponding control inputs

  Aggregate datasets: $D \leftarrow D \cup D_i$

  Train NN controller $\hat{\pi}_i$ on D

end

return the best $\hat{\pi}_i$ on validation

expert that given the system Σ, W, and U performs the QP routine at x to output the desired control value.

VII. Reach Avoid Problem of a Water Vehicle Model

Consider the model of a surface water vehicle subject to wind gusts and water currents as:

$$\dot{x} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} w, \qquad x(0) \in X_0 \tag{12}$$

where the state $x \in \mathbb{R}^3$ consists of vehicle location (x_1, x_2) and the heading angle θ. The control input $u \in \mathbb{R}$ is the vehicle's steering angle. The velocity v is assumed to be constant ($v = 1$) as it has a different relative degree from the steering angle u[1]. The external disturbance is $w \in [-0.1, 0.1]$. System trajectories starting from the set $X_0 = [8, 9] \times [5, 11] \times [-\pi, \pi]$ should avoid the unsafe sets $\mathcal{U}_i, i = 1, ..., 5$ and reach the goal set $\mathcal{G}$:

$$\begin{aligned} \mathcal{U}_i =& \{x : (x_1 - p_i(1))^2 + (x_2 - p_i(2))^2 < r_i\}, \\ \mathcal{G} =& \{x : (x_1 - x_{g,1})^2 + (x_2 - x_{g,2}))^2 < 0.3\} \end{aligned}$$

where $p_1 = (4, 2.5)$, $r_1 = 0.7$, $p_2 = (5, 6.5)$, $r_2 = 0.5$, $p_3 = (7, 4.75)$, $r_3 = 0.4$, $p_4 = (2.5, 5)$, $r_4 = 0.3$, $p_5 = (7.5, 2.5)$, $r_5 = 0.5$, and $x_{g,1} = x_{g,2} = 1$.

In order to reach the goal set, instead of using CLF based constraints, we formulate the stabilizing condition in the objective function. The desired heading angle is $\theta_{ref}(x) = \arctan(\frac{x_{g,2}-x_2}{x_{g,1}-x_1})$, and the desired input u to force θ to follow θ_{ref} is $u_{ref}(x) = K(\theta_{ref}(x) - \theta)$ where K is a positive constant, here we choose $K = 1$. The barrier function corresponding to the unsafe set $\mathcal{U}_j$ is $h_j(x) = (x_1 - p_j(1))^2 + (x_2 - p_j(2))^2 - r_j$ which has relative degree 2 w.r.t to the steering angle u. We consider $\alpha_1(y) = \alpha_2(y) = 2y$. The function $\dot{\psi}_{2,j}$ corresponding to each h_j can be computed based on Eq. (4) using Matlab's Symbolic toolbox, for example:

$$\begin{aligned} \dot{\psi}_{2,1} = & \underbrace{-(2\sin(\theta)(x_1 - 4) - 2\cos(\theta)(x_2 - 2.5))u}_{L_g L_f h_1(x) u} \\ & + \underbrace{4w^2 + 4(\cos(\theta) + \sin(\theta) + 2((x_1 - 4) + (x_2 - 2.5)))w}_{P_1(x,w)} \\ & + 4(x_1 - 4)^2 + 4(x_2 - 2.5)^2 + 4(\cos(\theta))(2x_1 - 8) \\ & + 4(\sin(\theta))(2x_2 - 5) - 0.8 \end{aligned}$$

The functions $P_j(x, w)$ corresponding to each unsafe set are quadratic in w. Let's call the portion of $\dot{\psi}_{2,j}$ that only depends on x, Ψ_j. Note that $\Psi_j(x) = L_f^2 h_j(x) + O(h_j(x)) + \alpha_2(\psi_{1,j}(x))$. As a result, in order to reach the goal set while avoiding the unsafe sets, first $F^*_{P,j}(x) = \max_{-0.1<w<0.1}(-P_j(x, w))$ needs to be computed and then the following quadratic program needs to be solved:

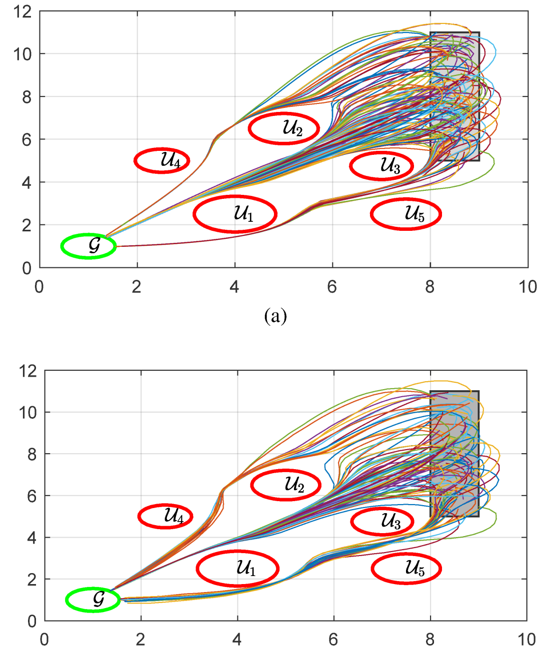
$$\begin{aligned} & \min_u \quad (u - u_{ref}(x))^2 \\ s.t \quad & L_g L_f h_j(x) u + \Psi_j(x) \geq F^*_{P,j}(x) \quad \forall j = 1, ..., 5 \end{aligned} \tag{13}$$

This QP is solved at each state visited by the vehicle under the controller π_i until reaching the goal set $\mathcal{G}$ as described in Alg. 1, to train NN controllers that can predict the expert's action online. Figure 2.(a) shows the trajectories of the system (12) guided by the solutions to QPs in Eq. (13) when $w = 0$. The NN controller successfully imitates the QPs at the 11^{th} iteration of the for loop in Alg. 1. Figure 2.(b) shows the system trajectories guided by the trained NN controller when randomized disturbance is applied to the system. As it is clear from the figures the controller is robust to disturbances as it has been trained with controllers that are able to compensate for the disturbance in the worst-case. It is worth mentioning that the inputs to the NN are the location states (x_1, x_2) in addition to $(\sin(\theta), \cos(\theta))$ - instead of the state θ itself. This data processing helps remove the discontinuities than happen when mapping θ to $[-\pi, \pi]$ and helps NN understand that $-\pi$ and π are indeed equivalent. Also, even-though input constraints are not enforced in this example, they can be added to problem (13) as linear constraints and considered in the NN architecture by adding a saturation function in the output.

VIII. Conclusions

In this work, we studied Control Barrier Functions (CBF) in presence of disturbances. These functions define constraints on the control input that can be used in an optimization problem to find safe sub-optimal control inputs. As

[1]Considering v as an input will make CBF constraints nonlinear in v, and the resulting problem will not be a quadratic program anymore. While this nonlinear program can be solved offline in this framework, in this paper we assume v is constant for simplicity.



(a)

(b)

Fig. 2: Trajectories initiated from X_0^s as guided by (a) the QPs as expert when $w = 0$ and (b) the trained NN controller when random disturbance is applied to system

solving these optimization problems might not be possible in real-time, we presented a framework to train NN controllers that can be used online to predict the outcome of the optimization problems. Future work will use methods like [9] to establish safety of the learned controller and counter-example generation methods as in [25] to speed up training.

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[2] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. *European Control Conference (ECC)*, 2019.

[3] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pages 6271–6278. IEEE, 2014.

[4] JA Bagnell, Joel Chestnutt, David M Bradley, and Nathan D Ratliff. Boosting structured prediction for imitation learning. In *Advances in Neural Information Processing Systems*, pages 1153–1160, 2007.

[5] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[6] Steven Chen, Kelsey Saulnier, Nikolay Atanasov, Daniel D Lee, Vijay Kumar, George J Pappas, and Manfred Morari. Approximating explicit model predictive control using constrained neural networks. In *2018 Annual American Control Conference (ACC)*, pages 1520–1527. IEEE, 2018.

[7] Arthur Claviere, Souradeep Dutta, and Sriram Sankaranarayanan. Trajectory tracking control for robotic vehicles using counterexample guided training of neural networks. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 680–688, 2019.

[8] Souradeep Dutta, Xin Chen, Susmit Jha, Sriram Sankaranarayanan, and Ashish Tiwari. Sherlock - a tool for verification of neural network feedback systems: Demo abstract. In *22nd ACM International Conference on Hybrid Systems: Computation and Control*, pages 262–263, 2019.

[9] Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, and Ashish Tiwari. Learning and verification of feedback control systems using feedforward neural networks. In *Analysis and Design of Hybrid Systems*, 2018.

[10] Mohammad Farhadi Bajestani, Mehdi Ghasemi, Sarma Vrudhula, and Yezhou Yang. Enabling incremental knowledge transfer for object detection at the edge. *arXiv*, pages arXiv–2004, 2020.

[11] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE control systems letters*, 1(2):310–315, 2017.

[12] Martin T. Hagan, Howard B. Demuth, and Orlando De Jesus. An introduction to the use of neural networks in control systems. *International Journal of Robust and Nonlinear Control*, 12(11):959–985, 2002.

[13] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[14] Kenneth J. Hunt, Daniel G. Sbarbaro, Rafat Zbikowski, and Peter Gawthrop. Neural networks for control systems - a survey. *Automatica*, 28:1083–1112, 1992.

[15] Shishir Kolathaya and Aaron D Ames. Input-to-state safety with control barrier functions. *IEEE control systems letters*, 3(1):108–113, 2018.

[16] Lars Lindemann and Dimos V Dimarogonas. Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks. *IEEE control systems letters*, 2019.

[17] Keyvan Majd, Mohammad Razeghi-Jahromi, and Abdollah Homaifar. A stable analytical solution method for car-like robot trajectory tracking and optimization. *IEEE/CAA Journal of Automatica Sinica*, 7(1):39–47, 2019.

[18] Quan Nguyen and Koushil Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *2016 American Control Conference (ACC)*, pages 322–328. IEEE, 2016.

[19] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: Imitation learning via regularized behavioral cloning. *arXiv preprint arXiv:1905.11108*, 2019.

[20] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[21] Johann Schumann and Yan Liu. *Applications of Neural Networks in High Assurance Systems*, volume 268 of *SCI*. Springer, 2010.

[22] Jiaming Song, Hongyu Ren, Dorsa Sadigh, and Stefano Ermon. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 7461–7472, 2018.

[23] C. E. Tuncali, H. Ito, J. Kapinski, and J. V. Deshmukh. Reasoning about safety of learning-enabled components in autonomous cyber-physical systems. In *55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, 2018.

[24] Wei Xiao and Calin Belta. Control barrier functions for systems with high relative degree. *arXiv preprint arXiv:1903.04706*, 2019.

[25] Shakiba Yaghoubi and Georgios Fainekos. Gray-box adversarial testing for control systems with machine learning components. In *ACM International Conference on Hybrid Systems: Computation and Control (HSCC)*, 2019.

[26] Shakiba Yaghoubi and Georgios Fainekos. Worst-case satisfaction of stl specifications using feedforward neural network controllers: a lagrange multipliers approach. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s):107, 2019.

[27] Guang Yang, Bee Vang, Zachary Serlin, Calin Belta, and Roberto Tron. Sampling-based motion planning via control barrier functions. In *Proceedings of the 2019 3rd International Conference on Automation, Control and Robots*, pages 22–29. ACM, 2019.

[28] Xiaojing Zhang, Monimoy Bujarbaruah, and Francesco Borrelli. Near-optimal rapid mpc using neural networks: A primal-dual policy learning framework. *arXiv preprint arXiv:1912.04744*, 2019.