Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2021) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2021)

Safe Learning for Control using Control Lyapunov Functions and Control Barrier Functions: A Review

Akhil Anand[a,*], Katrine Seel[a,*], Vilde Gjærum[a], Anne Håkansson[b], Haakon Robinson[a], Aya Saad[a]

[a] *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*
[b] *Department of Computer Science, Arctic University of Norway (UIT), Tromsø, Norway*

Abstract

Real-world autonomous systems are often controlled using conventional model-based control methods. But if accurate models of a system are not available, these methods may be unsuitable. For many safety-critical systems, such as robotic systems, a model of the system and a control strategy may be learned using data. When applying learning to safety-critical systems, guaranteeing safety during learning as well as testing/deployment is paramount. A variety of different approaches for ensuring safety exists, but the published works are cluttered and there are few reviews that compare the latest approaches. This paper reviews two promising approaches on guaranteeing safety for learning-based robust control of uncertain dynamical systems, which are based on control barrier functions and control Lyapunov functions. While control barrier functions provide an option to incorporate safety in terms of constraint satisfaction, control Lyapunov functions are used to define safety in terms of stability. This review categorises learning-based methods that use control barrier functions and control Lyapunov functions into three groups, namely reinforcement learning, online and offline supervised learning. Finally, the paper presents a discussion of the suitability of the different methods for different applications.

Keywords: safe learning; control barrier functions; control Lyapunov functions

1. Introduction

In the last decade, learning algorithms have been widely explored for designing control policies for complex and uncertain dynamical systems ranging from robotic manipulators to autonomous underwater vehicles. Both supervised learning (SL) and reinforcement learning (RL) algorithms have proved to be useful tools in learning for control. Especially RL algorithms have proved to be successful for a variety of complex and high-dimensional control tasks [1, 2].

[*] Corresponding author.
E-mail address: akhil.s.anand@ntnu.no, katrine.seel@ntnu.no

For systems with uncertain dynamics, learning-based methods represent an alternative to conventional robust controllers, by modelling the uncertainty from data. Providing theoretical safety guarantees for such learning algorithms is central to applying learning-based methods to control real-world safety-critical systems. These algorithms have motivated the research community to consider several methods for ensuring safety of physicals systems for which learning-based algorithms are used to find control policies [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13].

The notion of safety in learning-based control for real-world systems, such as robotic systems, has two aspects. The first aspect is ensuring safety during the learning process or the exploration phase, in cases when learning is done online. Online learning is here used to describe the process of learning by simultaneously interacting with and sampling from the system. The second aspect is guaranteeing safety of an already learned control policy. Specifically for most real-world robotic systems, learning the model or the control policy for the system, should be done partially or entirely online using the physical system, depending on the availability and accuracy of a prior model of the system. Even if a model of the system is available, learning a policy in simulation will often require fine tuning on the physical system, to compensate for inaccuracies of the prior model. This demands guaranteeing safety during the online learning process on the physical system in order to avoid any kind of damage to the robotic system or its environment. At the same time, the resulting learned control policy should have provable safety guarantees to facilitate its deployment on the real-world robotic system. For most learning-based algorithms, incorporating these guarantees may be done either by transforming the optimization problem or by changing the exploration process [14].

For safety-critical systems, the research community has mainly focused on safety in terms on constraint satisfaction, and mainly on state constraints. However, few of the learning algorithms used for designing learning-based controllers are capable of naturally incorporating state and input constraints. One of the predominant approaches for ensuring safe learning-based control, has been model predictive control (MPC) frameworks, that naturally incorporate state and input constraint satisfaction. MPC has been combined with SL methods in order to build or improve the prediction model as in [15, 16]. There are also many examples of MPC being combined with RL, for example as a value function approximator [17, 18], where stability and constraint satisfaction are ensured by design.

Another approach for ensuring safe learning-based control, that is more agnostic with respect to the control framework, is the use of *safety filters*. Safety filters function by verifying if a learned controller ensures safety in terms of the system states remaining inside a predefined safe set for all future times. If the learned controller does not pass the verification, it can be replaced by a known, safe controller. Alternatively, the learned control input can be minimally modified using an optimization problem, such that it satisfies the safety constraints. For both types of filters, a predefined safe set is necessary. One prominent method to calculate the safe set, is using reachability analysis [19]. However, this can be computationally demanding or result in potentially conservative approximations.

A different method for defining the safe sets, is employing control barrier functions (CBFs) [20]. CBFs gained popularity within the conventional control community during recent years, but have been utilized more as a safety filter for an existing nominal controller [21]. Many recent works in the learning-based control field use CBFs [22, 13]. By utilising probabilistic data-driven methods such as Gaussian processes (GPs), complete prior system knowledge is no longer needed, and probabilistic safety guarantees can be provided [6, 23, 24, 25].

In a different but slightly more conservative approach, safety can be guaranteed using stability guarantees of the closed-loop system. These approaches are typically based on Lyapunov stability verification, using control Lyapunov functions (CLFs)[26]. Compared to the safe sets, found either by reachability analysis or using CBFs, level sets of CLFs are both invariant and attractive. If the safe set can be designed as a subset of the region of attraction (ROA) defined by a Lyapunov function, then CLFs can be used to guarantee safety of the closed-loop system, and exploration in closed-loop is then typically limited to this region. A combination of CLF and CBF can be used to guarantee a safe and stabilizing controller [27], which is also utilized in learning to guarantee stability [28]. The combination of functions has been used in MPC frameworks [29], but are seldomly used in model-based RL algorithms [9].

For safety-critical systems, it can be paramount to ensure safety, either in terms of constraint satisfaction, or in terms of stability or both. Ensuring safety and stability by using CBFs and CLFs are particularly suited for learning, as both properties can be expressed using functions that can be learned. This paper aims to review approaches using CLFs and CBFs for ensuring safety and stability in learning-based methods considering their suitability in a learning framework and because they can be applied to a broad wide range of control frameworks. Even though this review is motivated by robotic applications, these approaches are generalizable to other types of control applications e.g. process control. The rest of the paper is organized as follows: Section 2 presents related work. The review of the learning algorithms is organised as follows: Section 3, presents a technical background of how CLFs and CBFs are used to provide safety guarantees. Section 4, treats RL algorithms, section 5, presents a review of online SL algorithms, followed by offline SL methods in Section 6. The mentioned categories and the corresponding relevant references treated for each of them, are listed in Table 1. To the best of the authors' knowledge there are currently no published research work on the combination of CLFs and CBFs for offline SL methods. Section 7 provides a short summary of the algorithms treated in the former section, and a discussion of the suitability of the methods for different applications. Finally, in Section 8, the concluding remarks are presented.

Table 1. Overview of the relevant literature.

	CBF	CLF	CLF + CBF
RL	Section 4.1 : [6, 13]	Section 4.2 : [30, 5, 31, 11]	Section 4.3: [9]
Online SL	Section 5.1 : [23, 24, 22]	Section 5.2 : [32, 33, 34, 35, 36]	Section 5.3 : [28, 25]
Offline SL	Section 6.1 : [37, 38, 39, 40, 41]	Section 6.2 : [42]	-

2. Related work

There are a few number of review papers in the area of learning for control, discussing different methods for ensuring safety. A survey on model learning of autonomous systems is presented in Nguyen-Tuong et al. [43], discussing safety challenges in existing methods. In [44], a comprehensive review on safety criteria and metrics, controller design along with mechanical design and actuation for domestic robotic systems is presented, where learning-based approaches were mentioned briefly. A survey on safe RL [14], discusses cases where it is important to respect safety constraints during learning and/or deployment. In this survey the authors categorise safe RL methods, based on whether the safety criterion is incorporated into the optimality criterion or by modifying the exploration process. In [45] and [46], the authors present a comprehensive survey of methods for safe human-robot interaction. A review work on learning-based MPC with emphasis on safe learning is presented in [47], categorising the methods based on learning the system dynamics, learning the control policy or using safety filters. In [48], a review of safe learning and optimization methods is presented as a continuation of [14] with an additional review of active learning and optimization methods. Kim et al. [48] review learning and optimization algorithms that ensure safety, where safety is guaranteed by adding constraints and/or ensuring that any unsafe states are avoided. But none of these surveys specifically address the CLF- and CBF-based approaches in learning-based control.

3. Safety Guarantees using Control Barrier Functions and Control Lyapunov Functions

Throughout this paper, nonlinear dynamical systems in its general form (1) and a control affine form (2) is considered, where f and g are locally Lipschitz, $x \in \mathcal{X} \subseteq \mathbb{R}^n$ denotes the state and, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ denotes the control input.

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

$$\dot{x} = f(x) + g(x)u\,. \tag{2}$$

For the rest of this section, we will use the following notation. A continuous function $\alpha : [0, a) \to [0, \infty)$ is called a class $\mathcal{K}$ function, if $\alpha(0) = 0$ and it is strictly increasing. A function is called a class $\mathcal{K}_\infty$ function, if it belongs to class $\mathcal{K}$, $a = \infty$ and $\lim_{r\to\infty} \alpha(r) = \infty$. Let $L_m Q(x)$ denote the Lie derivative of a function $Q(x)$ along another function $m(x)$ i.e. $L_m Q(x) := \frac{\partial Q(x)}{\partial x} m(x)$.

3.1. Notion of safety

For a dynamical system controlled by a control policy in order to perform a task, a general notion of safety is considered. For the system to be safe, it should be guaranteed that the system will never enter any unsafe region under the current policy. Safety can be enforced by ensuring the forward invariance of a safe set [21]. That is, all trajectories starting in the set of safe states will remain within the safe set for all $t \geq 0$.

Definition 1. *(Safe control) Consider a general form of dynamical systems (1), where the control policy $u = u(x)$ is a mapping from state to the optimal control action, $u : \mathcal{X} \to \mathcal{U}$. Consider a given set of unsafe states $\mathcal{X}_u \subseteq \mathcal{X}$, a set of initial condition $\mathcal{X}_0 \subseteq \mathcal{X}$ and a set of target/goal states $\mathcal{X}_g \subseteq \mathcal{X}$ where $\mathcal{X}_u \cap \mathcal{X}_0 = \emptyset$ and $\mathcal{X}_u \cap \mathcal{X}_g = \emptyset$. If for all the possible trajectories $x(t)$ evolving from the set of initial conditions to the set of goal states, such that $x(t) \notin \mathcal{X}_u$, for all time, $t \in T \subseteq \mathbb{R}^+$, the system is guaranteed to be safe under the control policy $u(x)$.*

3.2. Control Lyapunov Functions

The notion of control Lyapunov function [49] to design asymptotically stabilizing controllers to a nonlinear system, was introduced by Artstein and generalised by Sontag [50, 51]. Extending the Lyapunov function to a control Lyapunov function (CLF) helps to find a control law that ensures stability of a dynamical system in (2).

Definition 2. *A positive definite function $V : \mathbb{R}^n \to \mathbb{R}$, is a CLF to the system (2), if there exist a class $\mathcal{K}$ function, γ, and a control law,*

$$u_{clf} = u \in U, \quad s.t \quad L_f V(x) + L_g V(x) u + \gamma(V(x)) \leq 0\,, \tag{3}$$

which render the system asymptotically stable at the point $x^* = 0$ where $V(x^*) = 0$. For a system (2), the existence of a CLF implies that for all x in the level set $\mathcal{V}(c) = \{x \in \mathcal{X} | V(x) \leq c\}$ for $c > 0$, the control law u_{clf} asymptotically stabilizes the system. The largest level set is referred to as the region of attraction (ROA), and is forward invariant and attractive.

3.3. Control Barrier Functions

A set C, defined as a superlevel set of a continuously differentiable function $h : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, is safe if,

$$C = \{h(x) \geq 0\}\,, \qquad \partial C = \{h(x) = 0\}\,, \qquad \mathrm{Int}(C) = \{h(x) > 0\}\,, \tag{4}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ and ∂C represents the boundary of C. In order to address safety while controlling dynamical systems, the control barrier function (CBF) is introduced [20]. The general definition considered here is from [21].

Definition 3. *h in (4) is a CBF for a dynamical system in (2), if there exists a class $\mathcal{K}_\infty$ function, α defined over the entire real line, $\mathbb{R}$, and a control law $u_{cbf} = u \in U$, s.t*

$$L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0\,, \tag{5}$$

for all $x \in \mathcal{X}$. Given the control barrier condition (CBC) in (5), the safe set C is forward invariant for the system (2).

4. Control Barrier Functions and Control Lyapunov Functions in Reinforcement learning

Complex dynamical systems, such as robotic systems, are often difficult to model accurately due to their highly nonlinear and uncertain dynamics. Model-based RL can be used to estimate the unknown system dynamics online, while simultaneously learning an optimal policy to perform a particular task using samples from the learned model. Whereas model-free RL algorithms can be used to learn the policy directly. Both types of algorithms come at the cost of demanding online safety certification during learning, in addition to safety certification of the learned policy. Therefore safety certification of RL algorithms for systems with unknown dynamics is two-fold, (1) safety certification during learning and (2) offline safety certification of the learned policy i.e. after convergence. Depending on the selected RL algorithm, both regards may be ensured simultaneously or separately.

4.1. Control Barrier Functions

There are a few different approaches utilizing CBFs to guarantee safety in a RL setting. In [6], a framework for combining model-free RL algorithms with model-based CBFs is proposed. The approach ensures safety and improves the exploration efficiency of the RL algorithm. The model of the dynamical system to be controlled is assumed unknown. As the RL algorithm explores the system's states, measurements are used to update a GP model used to learn the unknown system dynamics. The GP model is in turn used to derive the CBC defined in Section 3.3. Here, safety is determined by adding a compensating CBF-based controller to the model-free RL policy. This is formulated using a quadratic program (QP) with CBC constraints as a safety filter, which aims to modify the RL-policy as little as possible, while ensuring that the state remains within the safe set. This approach resembles the general safety filter formulated in [4]. As policy iteration is done considering the altered RL-policy, i.e. with the CBF-addition, the learned policy is encouraged to operate in the safe part of state space. To avoid solving a QP every time-step during deployment the method is extended by approximating the compensating CBF-based controller using a neural network (NN) during the learning process.

Another approach is presented in [13], where an off-policy actor-critic method is used to learn a policy without requiring knowledge of the system dynamics, i.e a model-free RL algorithm. A safe, possibly conservative policy is used to explore while the algorithm learns, and by adding a CBF to the value function, the learned policy will stay inside the safe set determined by the CBF condition. A coefficient is used to define a trade-off between optimality, defined in terms of the original utility function, and safety, determined using the CBF. Using an off-policy algorithm, where the resulting policy is approximated using a NN, the safety guarantees will only carry over given that the NN converges to the optimal solution.

4.2. Control Lyapunov Functions

For RL algorithms, a distinction is typically made between policy-gradient methods as opposed to value-based methods. For value-based methods, the Lyapunov-based approach has generated interest, because the value function can be used as a Lyapunov function. This idea is exploited in [11], where safety constraints are defined using the ROA, i.e all states inside the ROA are safe. The ROA can be approximated for a fixed policy, by formulating a Lyapunov function and taking the largest level set as the ROA, as defined in Section 3.2. In [11], the uncertain dynamics are learned using GPs with measurements collected online by sampling from the system. Confidence intervals, defined based on the learned system dynamics model, are used to check the Lyapunov decrease condition for the system. An optimization problem constrained by the Lyapunov decrease condition is then solved to find a policy that results in the largest possible level set (largest possible ROA). The exploration strategy is based on information maximization. By choosing state-action pairs where the dynamics are most uncertain, the confidence interval will shrink so the ROA can be expanded incrementally. This is the same exploration strategy used in [36], which is treated in Section 5.2.

Lypunov functions have also been used with RL algorithms to achieve a different understanding of safety. In [5], an agent's behaviour policy is defined to be safe, if the cumulative cost constraint of the constrained Markov decision problem is satisfied. The Lyapunov function is designed to be a uniform upper bound on the constraint cost, such that the corresponding algorithm guarantees feasibility and optimality under certain conditions. This approach, is also extended to policy-gradient methods in [30], for which the policy function, that is a mapping from state to action, is learned directly.

In a different approach, Lyapunov functions have been used to construct safe RL agents that switch among a safe base of controllers. This was first done in [31], where Lyapunov functions are used to provide stability guarantees for each controller.

4.3. Control Lyapunov Functions and Control Barrier Functions

In [9], system uncertainty is estimated in the CLF and the CBFs using deep NNs. An actor-critic RL algorithm is used to minimize the effect of model uncertainty in the learned CLF and CBF using a reward function that penalizes the estimation errors in the CLF and CBF. The learned CBF and CLF constraints, compensating for model uncertainty, are exploited in a QP to find a safe and stable controller for the uncertain system, using input-output linearization. This work assumes that CLFs and CBFs designed for the nominal model, will also serve as a CLFs and CBFs for the true system. This assumption holds for systems where the nominal model and the true system have the same relative degree [22].

5. Control Barrier Functions and Control Lyapunov Functions in Online Supervised Learning

Online SL methods are here understood as either continuous or episodic learning of system components in closed-loop. In this setting, SL methods are used for learning a model with data collected by sampling from the controlled system. Unlike RL-algorithms, online SL methods are usually not used to optimize a control policy directly. However, for a uncertain system, CBFs or CLFs can be learned in order to determine a safe controller. Often a safe controller is found by solving an optimization problem, with constraints formulated using the learned models.

5.1. Control Barrier Functions

In [22], the authors present an approach to improve safety of a dynamical system by estimating the model uncertainty using CBFs. Instead of estimating the uncertainty by learning the system dynamics from measurements, the uncertainty is modelled directly in the CBF. This approach is less restrictive on the types of system uncertainties, as it estimates both uncertainties due to parametric errors and unmodelled dynamics. The uncertainty is learned episodically using NNs, and included as a constraint in an optimization problem modifying a nominal controller to be safe. It is assumed that if there exists a valid CBF for the nominal model of the system, then it is also a valid CBF for the uncertain model. For learning the uncertainties, an episodic learning approach is used, which alternates between collecting data using the current controller and synthesizing a new controller by solving a QP. At every iteration of the episodic learning, a heuristically weighted blend of the newly synthesised controller and the nominal controller is used to explore new data.

A different approach for learning the safe region of an unknown dynamical system is presented in [24]. This paper considers a dynamical system on the form (2) with an additional unknown affine disturbance term, which is modelled using a GP. A high probability confidence interval is defined over this GP model. It is assumed that an initial safe region and the corresponding barrier certificates are given. With online learning, the safe region is expanded until no more improvement is obtained with further exploration. A QP is formulated to maximize the volume of the barrier certified safe region with CBC as the constraint. An adaptive sampling method of the discretized state space, namely an information-maximization-based exploration method, inspired by [36], is proposed. The system is driven to any selected state by employing a nominal controller augmented with a safety filter (the QP with CBC as constraint). The approach is successfully demonstrated on a quadrotor to learn maximally aggressive movements in the vertical direction with an uncertain model and limited thrust.

A more general approach in the direction of safe online learning of system dynamics is presented in [23], for a nonlinear control affine dynamical system (2). The unknown system dynamics are modelled as a GP and used to optimize the system behavior and to guarantee safety with high probability. A chance constraint, i.e. a constraint that needs to hold for the entire state or input trajectory with a given probability, is specified using a predefined CBF defined by the estimated dynamics. The chance constrained version of an optimization problem for the control input is solved providing safety guarantees for a zero-order hold (ZOH) controller over a control time step. The safe control input provided by solving the constrained QP is then used to explore in order to train the GP. Similar probabilistic safety guarantees were extended to systems with arbitrary relative degree using exponential CBFs [21].

5.2. Control Lyapunov Functions

Online SL methods can be used for finding safe controllers using Lyapunov analysis. In [32], GP regression is used to model the uncertainties in the system. The resulting stochastic model is used to formulate a stochastic CLF, included as a chance constraint in a second-order cone program (SOCP). A stabilizing controller for the system with probabilistic guarantees is derived by solving the SOCP.

An accurate estimate of the ROA is useful for ensuring safety as addressed in Section 4.2. Learning can be used to estimate the unknown parts of the dynamic model of an uncertain system and thereby expand the ROA as exploited in [36]. In [36], a GP is used to learn the uncertain parts of the dynamics and the ROA is taken as the largest level-set of the resulting CLF. The next state to be explored is chosen as the point with largest variance within the current estimate of the ROA. A fixed locally safe controller is used to drive the system to the chosen next state. The episodically updated GP model incrementally decreases its variance and thereby increases the accuracy of the ROA estimation.

A similar approach is used in [35], where the goal is to find an accurate estimate of the ROA. Here a GP is used for learning the Lyapunov function rather than dynamics as in [36]. Using the converse Lyapunov theorem, which states that for a stable system there exists a Lyapunov function, a GP is trained with closed-loop data in order to infer the Lyapunov function, and in turn estimate the ROA like described above. Safe samples are collected using an algorithm that aims to balance the trade-off between exploration, in order to expand the resulting estimate of the ROA, and exploitation, i.e. reducing the uncertainty of the GP-learned Lyapunov function.

Learning a Lyapunov function can be useful for stability analysis of MPC algorithms. In [33], a NN is used for learning a Lyapunov function for the closed-loop system, used as a terminal cost in the MPC scheme. For uncertain nonlinear systems, this often needs to be conservatively approximated, with implications on closed-loop performance. By learning the terminal cost from data, the learned Lyapunov function is used to guarantee stability of the closed-loop system, in addition to ensuring robustness with respect to model errors in the prediction model.

Stability properties formulated using Lyapunov functions may be used to add knowledge about an existing closed-loop system, for which the system dynamics are unknown. This is investigated in [34], where the stochastic dynamics of a closed-loop system are learned based on training data and a constrained likelihood maximization problem, exploiting the fact that the closed-loop system is exponentially stable. The stability property is expressed in terms of a Lyapunov function, included as a constraint to the maximization problem.

5.3. Control Lyapunov Functions and Control Barrier Functions

There are two notable approaches combining CBFs and CLFs in an online SL framework to ensure safety and stability. In [25], the authors augment the approach presented in [23] by using both CLFs and CBFs. The system dynamics are learned online while satisfying safety constraints. The computationally efficient matrix variate Gaussian process regression method is used to learn the drift and input gain terms of control affine dynamical system. In addition to a CBF-based chance constraint in [23], a CLF-based chance constraint is included for specifying stability constraints. This method is extended to systems with arbitrary relative degree to synthesize a safe control policy by solving a deterministic SOCP.

The second approach presented in [28], uses SL to learn a safe and optimal goal reaching policy using a barrier function and a Lyapunov-like function respectively, for dynamical systems of the form (1) The condition for asymptotic stability is translated to goal reaching utilizing a Lyapunov-like function, considering the equilibrium point as the goal. The proposed Lyapunov-like function is less restrictive as it allows for specifying a set of goal states rather than just a fixed point. In addition, the Lie derivative is not required to always be negative definite. The policy, barrier function and the Lyapunov-like function are parameterized using deep NNs. The loss function for learning the barrier function is designed to penalize the violation of any constraints. Similarly, another loss function is defined to penalise the violation of Lyapunov-like function constraint. The two loss functions are then combined into a single optimization objective called the *total certificate risk*, which is positive semi-definite. Joint training of the barrier certificate, Lyapunov-like function and the policy networks is achieved by minimizing this objective. Additionally, a verification procedure is introduced to confirm the validity of the barrier and Lyapunov-like networks. One drawback of this approach is that it does not guarantee safety during the learning process. Guarantees are only provided for the final policy obtained from the learning process upon verification of the networks.

6. Control Barrier Functions and Control Lyapunov Functions in Offline Supervised Learning

Similarly to online SL and RL, offline SL can be utilized to learn the unknown system dynamics, CBFs and/or CLFs from collected data. Unlike in RL or online SL, offline SL is not an active learning method, for which an algorithm chooses the data points from the sampling space of the system using an exploration strategy. However, provided with an adequate data set, offline supervised learning can achieve the same final goals.

6.1. Control Barrier Functions

There are a few interesting approaches to learn CBFs for nonlinear systems with unknown dynamics in an offline supervised learning set-up. Saveriano et al. [40] focus on incremental learning of a set of linear parametric zeroing control barrier functions (ZCBFs)[21]. ZCBF is a type of CBF, which approaches infinity in the boundary of its safe set. ZCBFs are combined with a dynamical system-based motion planner such as dynamic movement primitives to ensure the constraint satisfaction for planned trajectories. The state constraint for the motion trajectory can be learned from human demonstrations and formulated as ZCBFs. A QP can then be used to find a stabilizing controller, where the states of the motion planner are constrained by the ZCBF. This enables the motion planner to generate a feasible motion trajectory satisfying the safety constraints. Another approach for estimating ZCBFs (both in offline and online settings) of control affine robotic system from sensor data is presented in [37]. A support vector machine (SVM) approach, namely the kernel-SVM method, is used to classify the set of safe and unsafe states in the data set. An online approach is defined for the scenario where the full set of unsafe samples from the environment is not available for offline learning. Robey et al. [41] present an approach to learn CBFs for nonlinear control affine dynamical systems of the form (2) using expert demonstrations of safe trajectories. This approach is agnostic to the parameterization used to represent CBFs. An optimization method is defined to synthesize valid local CBFs from the collected expert demonstrations.

An approach to synthesize NN-based controllers for nonlinear dynamical systems where safety guarantees are provided by NN-based barrier functions is proposed in [38]. The controller and the barrier functions are simultaneously trained using the same data set using a modified Stochastic Gradient Descent (SGD) optimization technique. A formal verification to guarantee safety of the synthesized controller is provided.

In a different approach using GPs, presented in [39], the authors learn the unknown control affine nonlinear dynamics as GPs. A parametric nonlinear CBF is generated based on a counterexample guided inductive synthesis (CEGIS) method. A control policy is synthesized with safety guarantees in three steps. In the first step a GP is learned using a data set and a confidence interval is defined using the uncertainty of the learned model. The second step involves computing a parametric CBF using CEGIS. In the third step a controller can be synthesized by solving an optimization problem with a CBC-constraint.

6.2. Control Lyapunov Functions

For uncertain systems, offline supervised learning may be used to improve the estimate of a Lyapunov function. This approach is explored in [42], where the derivative of the Lyapunov function is learned offline. Using an episodic learning approach, the time derivative of the Lyapynov function is iteratively improved. This, in turn, is used to ensure that the nominal controller, augmented with a QP-based optimal controller, satisfies the necessary conditions of the time derivative of the Lyapunov function, and renders the closed-loop system stable. The formulation of the QP is similar to the formulation in [22], except that a CLF rather than a CBF, is used to formulate the constraints.

7. Discussion

This paper presented a review of three different learning methods that use CLFs and CBFs for ensuring safe learning-based control, namely RL, online and offline SL. RL and typically online SL are both active learning methods, but differ in the way the control policy is derived. RL offers a flexible framework to learn any complex control policies based on data, while SL methods are often used in combination with optimization to find a control policy. Offline SL differs from these two approaches in its data collection strategy as it uses pre-collected data sets. Active learning methods may be more data-efficient for learning control policies for systems with complex dynamics, compared to offline SL methods [52]. Especially for high dimensional robotic systems, the data requirements scale exponentially with the state space dimension, demanding very large pre-collected data sets. Active learning methods can explore state space efficiently, using for example an information maximization exploration strategy. Low quality data sets could adversely effect the accuracy of estimated model and thereby the resulting control policy. On the other hand, offline SL methods are suitable for learning from expert demonstrations.

Online SL and RL methods offer an option to incrementally update/optimize the control policy and use the same policy to sample. An example is model-based policy search algorithms, which are proved to be very sample efficient for policy optimization [52]. For robotic systems, a nominal safe controller may be used for collecting data or initial exploration of state space, but this may only be valid in small local areas. For offline SL methods, this can therefore result in small and local data sets, limiting the possible control policies that can be learned. Gradually obtaining a less conservative control policy using incrementally updated estimates of the system model, as in online SL and RL methods, can therefore be very convenient.

Using NNs to model the optimal policy in RL-algorithms enable approximation of arbitrary complex policies. On the downside, providing safety guarantees for the resulting policy is hard as it is only an approximation of the safe policy. SL methods will often be used in combination with optimization to derive the controller, for which it can be easier to provide stricter guarantees. Combining RL with an optimization-based safety filter could provide stricter safety guarantees, in addition to providing rich expressibility of the final policy. However, this comes at the cost of solving an additional optimization problem in real-time as discussed in Section 4.1. For optimization-based safety filters that alter the learned policy in order to satisfy safety constraints, a relevant question is how this modification affects the learning process. Depending on the applied RL-algorithm, this can disrupt the learning process such that the learned policy becomes suboptimal. This issue is addressed for several RL-algorithms in [53].

Offline SL methods can generate a model of the system dynamics, which can be used to formulate an optimal control problem for the system. Offline SL methods are predominantly used for learning CBFs rather than learning the dynamics model or CLFs, as observed in Section 6. For some dynamical systems, controlling the system may not be needed in order to derive the safety constraints. One example is in the case of collision avoidance, where a camera system can be used to detect possible collision objects and learn the corresponding safety constraints. In this case offline SL may be an ideal tool for providing safe control policies. Learning CBFs using NNs could be suitable as it can represent a complex safe set accurately and incorporate constraints in real-time which are otherwise hard to model. Robust and well established machine learning approaches, such as clustering and classification can be utilized in learning the safe sets and thereby CBFs or CLFs. Consequently, CBFs and CLFs learned through offline SL could be used with RL or online SL methods to derive less restrictive controllers than using conservative CBFs or CLFs defined for the uncertain system.

Considering robotics applications, both RL and online SL methods are suited as it may be hard to collect data offline. For applications demanding strict safety guarantees either online SL or RL with a safety filter can be used. Wang et al. [24] provide a very good example of the practical use of a safety filter where a quadcopter's dynamics are learned safely using CBFs in an online SL setting. Offline SL methods can aid RL and online SL methods in learning the system dynamics and controller less restrictively. For robotic systems the dynamics are often control affine, an approximate prior model is usually available and a major part of the uncertain dynamics are linked to the environment rather than the robot itself. These properties make CBFs particularly suited for guaranteeing safety for a wide variety of robotic systems. CBF-based approaches can be generalized to most real world robotic systems using exponential CBFs [21]. This includes systems such as robotic manipulators, bipedal robots, unmanned ground and aerial vehicles, autonomous underwater vehicles etc.

When learning a controller for an uncertain system, desired safety and stability properties will dictate which approach is suited for ensuring safe control. The combination of CLFs and CBFs can be used to obtain control policies that ensure stability and safety for a wide range of safety-critical systems. If only interested in constraint satisfaction, then CLF-based methods will limit the set of possible control policies. This is because a policy derived from this approach will render the system asymptotically stable in addition to guaranteeing constraint satisfaction. Therefore CBFs are particularly suited for scenarios where safety is the primary goal, as it offers a less restrictive way to ensure constraint satisfaction compared to using CLFs. In cases where stability is of major importance, CLFs can provide constraint satisfaction in addition to stability guarantees at the expense of more restrictive conditions. In case of value function-based RL algorithms, CLFs can be incorporated naturally by using the value function itself as a Lyapunov function [11].

8. Conclusion

This paper presents a literature review of learning methods that incorporate CBFs and CLFs and their combination. The review summarizes the existing learning-based methods for safe control of dynamical systems with uncertainty, utilizing CBFs and CLFs. The relevant references are divided into three main categories, decided by the learning method that CBFs and CLFs are combined with, namely RL, online and offline SL as shown in Table 1. The similarities and differences between the methods used in the review references are highlighted and their suitability on different scenarios are discussed. It is observed that, despite steady progress, there still exists a large gap between theory and practical application of the methods. Because using CLFs and CBFs with learning is a rather new approach, a major challenge ahead is demonstrating their capabilities on real-world safety-critical systems. This widens the scope for future research in the area.

Acknowledgements

This work has been supported by the IDUN-project (NFR 295920) , TAPI (NFR 294544) and ROMO (NFR 270941).

References

[1] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.

[2] Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y Ng. An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19:1, 2007.

[3] Anayo K Akametalu, Jaime F Fisac, Jeremy H Gillula, Shahab Kaynama, Melanie N Zeilinger, and Claire J Tomlin. Reachability-based safe learning with gaussian processes. In *Conference on Decision and Control (CDC)*, pages 1424–1431. IEEE, 2014.

[4] Kim P Wabersich and Melanie N Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *arXiv preprint arXiv:1812.05506*, 2018.

[5] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *arXiv preprint arXiv:1805.07708*, 2018.

[6] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3387–3395, 2019.

[7] Tommaso Mannucci, Erik-Jan van Kampen, Cornelis de Visser, and Qiping Chu. Safe exploration algorithms for reinforcement learning controllers. *IEEE transactions on neural networks and learning systems*, 29(4):1069–1081, 2017.

[8] David D Fan, Jennifer Nguyen, Rohan Thakker, Nikhilesh Alatur, Ali-akbar Agha-mohammadi, and Evangelos A Theodorou. Bayesian learning-based adaptive control for safety critical systems. In *International Conference on Robotics and Automation (ICRA)*, pages 4093–4099. IEEE, 2020.

[9] Jason Choi, Fernando Castaneda, Claire J Tomlin, and Koushil Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584*, 2020.

[10] Felix Berkenkamp. *Safe exploration in reinforcement learning: Theory and applications in robotics*. PhD thesis, ETH Zurich, 2019.

[11] Felix Berkenkamp, Matteo Turchetta, Angela P Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *arXiv preprint arXiv:1705.08551*, 2017.

[12] Pavel Osinenko, Lukas Beckenbach, Thomas Göhrt, and Stefan Streif. A reinforcement learning method with closed-loop stability guarantee. *arXiv preprint arXiv:2006.14034*, 2020.

[13] Zahra Marvi and Bahare Kiumarsi. Safe reinforcement learning: A control barrier function optimization approach. *International Journal of Robust and Nonlinear Control*, 31(6):1923–1940, 2021.

[14] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16 (1):1437–1480, 2015.

[15] José María Manzano, Daniel Limon, David Muñoz de la Peña, and Jan-Peter Calliess. Robust learning-based mpc for nonlinear constrained systems. *Automatica*, 117:108948, 2020.

[16] Katrine Seel, Esten I Grøtli, Signe Moe, Jan T Gravdahl, and Kristin Y Pettersen. Neural nework-based model predictive control with input-to-state stability. In *American Control Conference (ACC)*. IEEE, 2021.

[17] Mario Zanon and Sébastien Gros. Safe reinforcement learning using robust mpc. *IEEE Transactions on Automatic Control*, 2020.

[18] Sebastien Gros and Mario Zanon. Towards safe reinforcement learning using nmpc and policy gradients: Part ii-deterministic case. *arXiv preprint arXiv:1906.04034*, 2019.

[19] Jeremy H Gillula and Claire J Tomlin. Guaranteed safe online learning via reachability: tracking a ground target using a quadrotor. In *International Conference on Robotics and Automation (ICRA)*, pages 2723–2730. IEEE, 2012.

[20] Peter Wieland and Frank Allgöwer. Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12):462–467, 2007.

[21] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *European Control Conference (ECC)*, pages 3420–3431. IEEE, 2019.

[22] Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for Dynamics and Control*, pages 708–717. PMLR, 2020.

[23] Mohammad Javad Khojasteh, Vikas Dhiman, Massimo Franceschetti, and Nikolay Atanasov. Probabilistic safety constraints for learned high relative degree system dynamics. In *Learning for Dynamics and Control*, pages 781–792. PMLR, 2020.

[24] Li Wang, Evangelos A Theodorou, and Magnus Egerstedt. Safe learning of quadrotor dynamics using barrier certificates. In *International Conference on Robotics and Automation (ICRA)*, pages 2460–2465. IEEE, 2018.

[25] Vikas Dhiman, Mohammad Javad Khojasteh, Massimo Franceschetti, and Nikolay Atanasov. Control barriers in bayesian learning of system dynamics. *arXiv preprint arXiv:2012.14964*, 2020.

[26] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.

[27] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Stabilization with guaranteed safety using control lyapunov–barrier function. *Automatica*, 66:39–47, 2016.

[28] Wanxin Jin, Zhaoran Wang, Zhuoran Yang, and Shaoshuai Mou. Neural certificates for safe control policies. *arXiv preprint arXiv:2006.08465*, 2020.

[29] Zhe Wu and Panagiotis D Christofides. Control lyapunov-barrier function-based predictive control of nonlinear processes using machine learning modeling. *Computers & Chemical Engineering*, 134:106706, 2020.

[30] Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

[31] Theodore J Perkins and Andrew G Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3(Dec): 803–832, 2002.

[32] Fernando Castañeda, Jason J Choi, Bike Zhang, Claire J Tomlin, and Koushil Sreenath. Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects. *arXiv preprint arXiv:2011.07183*, 2020.

[33] Mayank Mittal, Marco Gallieri, Alessio Quaglino, Seyed Sina Mirrazavi Salehian, and Jan Koutník. Neural lyapunov model predictive control. *arXiv preprint arXiv:2002.10451*, 2020.

[34] Jonas Umlauft, Armin Lederer, and Sandra Hirche. Learning stable gaussian process state space models. In *American Control Conference (ACC)*, pages 1499–1504. IEEE, 2017.

[35] Chao Zhai and Hung D Nguyen. Region of attraction for power systems using gaussian process and converse lyapunov function–part i: Theoretical framework and off-line study. *arXiv preprint arXiv:1906.03590*, 2019.

[36] Felix Berkenkamp, Riccardo Moriconi, Angela P Schoellig, and Andreas Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *Conference on Decision and Control (CDC)*, pages 4661–4666. IEEE, 2016.

[37] Mohit Srinivasan, Amogh Dabholkar, Samuel Coogan, and Patricio Vela. Synthesis of control barrier functions using a supervised machine learning approach. *arXiv preprint arXiv:2003.04950*, 2020.

[38] Hengjun Zhao, Xia Zeng, Taolue Chen, Zhiming Liu, and Jim Woodcock. Learning safe neural network controllers with barrier certificates. In *International Symposium on Dependable Software Engineering: Theories, Tools, and Applications*, pages 177–185. Springer, 2020.

[39] Pushpak Jagtap, George J Pappas, and Majid Zamani. Control barrier functions for unknown nonlinear systems using gaussian processes. In *Conference on Decision and Control (CDC)*, pages 3699–3704. IEEE, 2020.

[40] Matteo Saveriano and Dongheui Lee. Learning barrier functions for constrained motion planning with dynamical systems. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 112–119. IEEE, 2019.

[41] Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *Conference on Decision and Control (CDC)*, pages 3717–3724. IEEE, 2020.

[42] Andrew J Taylor, Victor D Dorobantu, Hoang M Le, Yisong Yue, and Aaron D Ames. Episodic learning with control lyapunov functions for uncertain robotic systems. *arXiv preprint arXiv:1903.01577*, 2019.

[43] Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340, 2011.

[44] Tadele Shiferaw Tadele, Theo de Vries, and Stefano Stramigioli. The safety of domestic robotics: A survey of various safety-related publications. *IEEE robotics & automation magazine*, 21(3):134–142, 2014.

[45] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. *A survey of methods for safe human-robot interaction*. Now Publishers, 2017.

[46] Angeliki Zacharaki, Ioannis Kostavelis, Antonios Gasteratos, and Ioannis Dokas. Safety bounds in human robot interaction: A survey. *Safety science*, 127:104667, 2020.

[47] Lukas Hewing, Kim P Wabersich, Marcel Menner, and Melanie N Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296, 2020.

[48] Youngmin Kim, Richard Allmendinger, and Manuel López-Ibáñez. Safe learning and optimization techniques: Towards a survey of the state of the art. *arXiv preprint arXiv:2101.09505*, 2021.

[49] Randy A Freeman and James A Primbs. Control lyapunov functions: New ideas from an old source. In *Conference on Decision and Control (CDC)*, volume 4, pages 3926–3931. IEEE, 1996.

[50] Zvi Artstein. Stabilization with relaxed controls. *Nonlinear Analysis: Theory, Methods & Applications*, 7(11):1163–1173, 1983.

[51] Eduardo D Sontag. A 'universal'construction of artstein's theorem on nonlinear stabilization. *Systems & control letters*, 13(2):117–123, 1989.

[52] Konstantinos Chatzilygeroudis, Vassilis Vassiliades, Freek Stulp, Sylvain Calinon, and Jean-Baptiste Mouret. A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Transactions on Robotics*, 36(2):328–347, 2019.

[53] Sebastien Gros, Mario Zanon, and Alberto Bemporad. Safe reinforcement learning via projection on a safe set: How to achieve optimality? *arXiv preprint arXiv:2004.00915*, 2020.