IEEE CSS

Received 8 December 2023; accepted 6 February 2024. Date of publication 22 February 2024;
date of current version 1 April 2024. Recommended by Senior Editor Prof. L. Pavel.

Digital Object Identifier 10.1109/OJCSYS.2024.3368850

A Computationally-Efficient Data-Driven Safe Optimal Algorithm Through Control Merging

MARJAN KHALEDI AND BAHARE KIUMARSI (Member, IEEE)

Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA

CORRESPONDING AUTHOR: BAHARE KIUMARSI (e-mail: kiumarsi@msu.edu)

This work was supported in part by the Office of Naval Research through the Department of Navy under Award N00014-22-1-2159, and in part by the National Science Foundation under Award ECCS-2227311.

ABSTRACT This article presents a proactive approach to resolving the conflict between safety and optimality for continuous-time (CT) safety-critical systems with unknown dynamics. The presented method guarantees safety and performance specifications by combining two controllers: a safe controller and an optimal controller. On the one hand, the safe controller is designed using only input and state data measurements and without requiring the state derivative data, which are typically required in data-driven control of CT systems. State derivative measurement is costly, and its approximation introduces noise to the system. On the other hand, the optimal controller is learned using a low-complexity one-shot optimization problem, which again does not rely on prior knowledge of the system dynamics and state derivative data. Compared to existing optimal control learning methods for CT systems, which are typically iterative, a one-shot optimization is considerably more sample-efficient and computationally efficient. The share of optimal and safe controllers in the overall control policy is obtained by solving a computationally efficient optimization problem involving a scalar variable in a data-driven manner. It is shown that the contribution of the safe controller dominates that of the optimal controller when the system's state is close to the safety boundaries, and this domination drops as the system trajectories move away from the safety boundaries. In this case, the optimal controller contributes more to the overall controller. The feasibility and stability of the proposed controller are shown. Finally, the simulation results show the efficacy of the proposed approach.

INDEX TERMS Control barrier functions (CBFs), data-driven initiative controller, optimality, safe control, unknown systems.

I. INTRODUCTION

Designing data-based or learning-enabled controllers that satisfy system specifications (e.g., safety, stability, and performance) has emerged as a critical concern recently in control engineering [1]. More specifically, as a prominent control design approach, data-driven control design involves learning controllers for an unknown system based solely on measurements obtained from the system and some prior information about its characteristics [2]. A common approach is to first identify a system model through input-output measurements of the system and then use the identified model to design a model-based controller [3]. This is also called indirect data-driven control. As an alternative, there has been a surge of interest recently in designing control systems directly from process data while bypassing an intermediary system identification stage [1], [2], [3], [4]. This is called direct data-driven control. In many situations, direct data-driven approaches can be advantageous to indirect data-driven approaches for many reasons, some of which are listed in [5].

In many control systems, safety is a critical concern, and it is of vital importance to design control systems that respect safety concerns. Safety concerns are typically imposed on the system as constraints that must be satisfied all the time. Safety, however, is the bare minimum requirement of safety-critical systems, and it is desired to design safe controllers with some performance guarantees. To provide performance guarantees, optimal control design aims at finding the best controller from a set of admissible controllers based on certain performance criteria, such as maximizing profits, minimizing costs, or maximizing efficiency. Both safety and optimality are important concepts in the control of autonomous systems, and finding the balance between them is essential to achieving acceptable outcomes.

To ensure safety, the concept of set forward invariance has been widely utilized, which requires the system's state to remain within a safe set once it starts from that set [6]. In most existing approaches, control barrier functions (CBFs) are employed to guarantee the forward invariance of the safe set. CBFs are based on the notion of defining a continuously differentiable function, known as a barrier function, which maps the system's state to a scalar value. These CBFs establish a linear inequality condition for the system's input to ensure the forward invariance of the safety set. In many CBF-based methods, safety and performance requirements are combined through a quadratic programming (QP) formulation. This is realized through minimizing the intervention with a nominal or optimal controller while imposing point-wise control Lyapunov functions (CLFs) and CBFs as soft and hard inequality constraints, respectively [7]. This approach is reactive in the sense that the safety of the system is certified myopically at every time instance. Other reactive safety certificates are also presented in [8], [9]. This myopic intervention can also lead to convergence to undesired stable equilibrium points on the boundary of the safe set, as shown in [10]. Besides being myopic, another challenge with these existing safety certificates is that they require a complete and accurate knowledge of the system model. Moreover, the nominal or optimal controller is also assumed to be given a priori. A fundamental challenge is to learn optimal controllers and safe controllers using only a single trajectory of input-state data and then combining them to manage conflicts proactively.

To diminish the complete reliance on the model, a data-driven approach to safe control design is presented in [11] using the concept of contraction sets [12]. Besides, a data-driven safe controller is designed in [4] by utilizing the control barrier certificate notions. However, one drawback of these approaches is that they require the state derivatives to be measured or approximated. This can be costly or introduce noise to the measurements, which can significantly deteriorate the performance and can also jeopardize the system's safety.

Besides efficient data-driven safe control design without the state derivative requirements, which is lacking, sample-efficient learning algorithms for learning optimal controllers for CT systems are also surprisingly lacking. Reinforcement learning (RL) algorithms such as policy iteration [13] and policy gradients [13] have been widely utilized to learn optimal control policies. However, the iterative nature of these algorithms makes them data-hungry. The need for online data to evaluate control policies or the cost gradients can also be costly and risky. It is highly desired to develop one-shot learning algorithms that can learn an optimal control policy using only a single data trajectory. Finally, it is highly desired to combine two control policies (i.e., safe control and optimal control) to avoid convergence to an undesired equilibrium. This will be in sharp contrast to existing myopic approaches that assume an optimal controller is given and certify the safety of its actions using CBFs.

This paper presents a sample-efficient data-based safe, and optimal controller for CT linear quadratic regulator (LQR) problems with safety constraints. The presented approach 1) learns both safe and optimal control policies using convex optimization formulations and using only input-state data in one shot; 2) combines the two policies using a computationally efficient interpolation technique. A convex optimization formulation of the LQR presented in [14] is leveraged to develop the one-shot optimization for optimality. Then, to design both safe and optimal controllers, the closed-loop system is represented based on data and is used in the optimization frameworks to turn them into data-based convex optimization frameworks. The safety of the resulting controller is guaranteed. Besides, convergence to the equilibrium is also guaranteed. Importantly, our approach neither relies on knowledge of the system dynamics nor requires measuring or approximating the derivative of the state as sampled data. It is also shown that the safe controller predominantly contributes to the overall controller near safety boundaries, ensuring safety takes precedence. However, as the system trajectories move away from the safety boundaries, the optimal controller takes over the contribution. The feasibility and stability of the controller are demonstrated, and simulation results illustrate the effectiveness of our method.

A. NOTATIONS

In this paper, $\mathbb{N} > 0$ $(\mathbb{N}^+)$ shows the set of positive natural numbers, excluding zero. Also, $\mathbb{R} > 0$ $(\mathbb{R}^+)$ demonstrates the positive real numbers, excluding zero. $\mathbb{X}_u$ represents the unsafe set. $\mathbb{U}$ shows the set of all admissible control inputs. The safe control input is shown by u^s, and u^L presents the learning-based optimal control input. $\mathbb{I}_n$ is represented as an identity matrix in $\mathbb{R}^{n\times n}$. A prime symbol has been used to show the transpose of the matrix. Therefore, the transpose of matrix A has been represented as A'. A symmetric matrix $P \in \mathbb{R}^{n\times n}$ is said to be positive definite, denoted by $P \succ 0$, if all of its eigenvalues are positive. Note that $(X_{0,T})^{\dagger}$ represents the Pseudo inverse of $X_{0,T}$.

II. PROBLEM FORMULATION AND PRELIMINARIES

A. PROBLEM FORMULATION: SAFE OPTIMAL CONTROL

This subsection formalizes the safe optimal control problem and provides a background on its challenges.

Consider a continuous-time linear system described by

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state of the system, and $u(t) \in \mathbb{U}$ indicates the control input. For the notation simplicity, $x(t)$ and $u(t)$ are replaced by x and u, respectively, in the rest of the paper.

Our main objective is to design a control input u for (1), certifying safety and optimality while no prior knowledge of the system dynamics is required.

To take into account optimality, the objective function for system (1) is considered as

$$J(x(t), u(t)) = \int_{t}^{+\infty} (x'(\tau)Qx(\tau) + u'(\tau)Ru(\tau))d\tau, \quad (2)$$

in which R and Q are positive-definite symmetric matrices (i.e., $R = R' > 0, Q = Q' > 0$).

Definition 1 (Control Barrier Functions): Suppose $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$ represents the super level set of a continuously differentiable function $h(x) : \mathcal{D} \to \mathbb{R}$. Then, $h(x)$ is called a Control Barrier Function (CBF) for the control system (1) if there exists an extended class $\mathcal{K}$ function γ such that [7]:

$$\sup_{u \in \mathbb{U}} [\dot{h}(x) + \gamma(h(x))] \geq 0.$$

Based on Definition 1, upon existence of a CBF $h(x)$, the set of all control inputs that render $\mathcal{C}$ safe is [7]:

$$K_{cbf}(x) = \{u \in \mathbb{U} |\ \dot{h}(x) + \gamma(h(x)) \geq 0\}. \quad (3)$$

We now formalize the safe optimal control problem based on the performance (2) and the safety requirement (3) by the following optimization problem.

Problem 1 (Safe Optimal Control): Find a control policy u that solves

$$\min_{u \in \mathbb{U}} J(x, u) \quad (4a)$$

$$\text{s.t.} \quad \dot{h}(x) + \gamma(h(x)) \geq 0, \quad (4b)$$

$$\dot{x} = Ax + Bu, \quad (4c)$$

In this optimization, the cost $J(x, u)$ defined in (2) is optimized while ensuring that the safety constraint is certified (i.e., the control input belongs to the set K_{cbf} defined in (3)) in the presence of uncertainty in the system dynamics. Solving the constrained optimal control problem (4) is very difficult in general, even if the system dynamic is known. This is because it is computationally intractable to find a feedback controller u that minimizes the objective function and satisfies the safety constraint simultaneously. Due to the difficulty of directly solving the optimization (4), we separate safety concern and optimality concern by designing a safe controller and an optimal controller separately and combining them with each other while optimizing the contribution of the optimal controller.

While reinforcement learning (RL) algorithms have been developed to solve the optimal LQR problem without requiring the knowledge of the system dynamics, the following challenges remain: 1) RL algorithms such as policy iteration [13] iteratively solve the LQR problem and they might need to perform many iterations to learn an optimal control policy. Besides, they can be data-intensive, especially if they need to generate new data at every iteration to evaluate a new policy. 2) Safe control design approaches are typically model-based methods, which can ruin the model-free nature of the RL algorithms, as the whole safe optimal control design still requires the system model. To observe the model dependence of the CBF-based safe control design, since the safety constraints depend on a general class $\mathcal{K}$ function, in the following, we assume that $\gamma(h(x)) = \gamma h(x)$, for which the safety constraint in the optimization (4) turns into

$$\frac{\partial h(x)}{\partial x}(Ax + Bu) + \gamma h(x) \geq 0 \quad (5)$$

As can be seen, the system dynamic shows up in safety constraint (5). Even though data-based safe control design is considered in [4], state derivative measurements are required, and no performance requirements are considered. To resolve these disadvantages of safe optimal control learning methods, this paper aims to learn safe and optimal controllers using a single trajectory of state-input data and in a one-shot or non-iterative fashion.

B. BACKGROUND ON STATE-INPUT REPRESENTATION OF THE CLOSED-LOOP SYSTEM

The input and state measurements collected from the system are organized as follows

$$U_{0,T} = [u(t_0) \quad u(t_0 + \tau) \quad \cdots \quad u(t_0 + (T-1)\tau)], \quad (6a)$$

$$X_{0,T} = [x(t_0) \quad x(t_0 + \tau) \quad \cdots \quad x(t_0 + (T-1)\tau)], \quad (6b)$$

where $T \in \mathbb{N}^+$ shows the number of collected samples and $\tau \in \mathbb{R}^+$ demonstrates the sampling time. To compare with the existing results, we also introduce the collected state derivative data as follows, even though our learning approach will not need it.

$$X_{1,T} = [\dot{x}(t_0) \quad \dot{x}(t_0 + \tau) \quad \cdots \quad \dot{x}(t_0 + (T-1)\tau)].$$

Assumption 1: The matrix $X_{0,T} \in R^{n\times T}$ in (6) is full row rank.

For Assumption 1 to be held, $T \geq n$ must be satisfied, where $n \in \mathbb{N}$ shows the dimension of x. This condition can be verified since the matrix $X_{0,T}$ is constructed based on the sampled data.

The following results from the literature provide a data-based representation of the closed-loop system while leveraging the state-input-state derivative data.

Lemma 1 ([4]): Consider the system (1) under the nonlinear control input $u(x) = F(x)x$ where $F(x)$ is a polynomial function. Let Assumption 1 hold and $W(x)$ is defined as a $T \times n$ matrix such that

$$\mathbb{I}_n = X_{0,T} W(x).$$

If $u = F(x)x = U_{0,T}W(x)x$, then the closed-loop dynamics (1) is represented as

$$\dot{x} = X_{1,T}W(x)x, \quad (7)$$

or equivalently, the closed-loop system can be written as

$$A + BF(x) = X_{1,T}W(x), \quad (8)$$

in which (7) and (8) are data-based representation of the closed-loop system.

Remark 1: (8) provides a data-based representation of the closed-loop system dynamics and can replace the constraint

(5) to find the safe controller in a data-driven manner. However, the drawback of data-driven approaches presented in [3], [4] is that the derivatives of the state of the system are needed, which are typically unavailable as direct measurements and have the following representation [1]

$$X_{1,T} = AX_{0,T} + BU_{0,T} = [B \quad A]\begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix}. \tag{9}$$

Moreover, the performance is not considered in the safe control design approaches presented in [4].

To address the challenges outlined above in existing approaches, this paper presents a new learning-enabled safe optimal control design approach to solve the optimization problem (4) in a data-driven manner. Addressing both safety and optimality simultaneously presents its own set of challenges, particularly when formulating the problem solely based on data, which adds further complexity. However, the proposed approach in this paper is able to tackle these issues and can entail the construction of data-driven safe control input, as well as the development of a one-shot optimization to find optimal control input. This methodology is achieved by bypassing the necessity for explicit knowledge of system dynamics and eliminating the requirement for knowledge of state derivatives from sampled data. Instead, our method relies solely on measured input/state data to trade-off between safety and optimality.

III. DATA-DRIVEN SYSTEM REPRESENTATION

To design a safe controller while the system dynamic is unknown and the derivative of the state of the system is not available, it is desired to write constraint (5) in terms of only input and state measurements. Besides, to design an optimal controller in a one-shot and data-efficient manner, a data-based representation of the closed-loop system is required. This section provides a data-based representation of open-loop systems for polynomial controllers and a data-based closed-loop representation for linear controllers.

To ensure that the safe control input covers the entire safe region, the following general format is considered for the safe controller:

$$u^s = F(x)x. \tag{10}$$

By applying (10) in (1) instead of u, the closed-loop system is described as

$$\dot{x} = (A + BF(x))\ x. \tag{11}$$

To obviate the requirement of the derivative of the state of the system in the proposed approach, the following theorem is presented.

Theorem 1: Consider the system (1). Let x be a continuous-time signal ($x_{[0,nT]} \subseteq \mathbb{R}$) for some $n \in \mathbb{N}^+$ and $T \in \mathbb{R}^+$. Then, the input-state representation of A and B is

$$[B \quad A] = (\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))D'(DD')^{-1}, \tag{12}$$

in which

$$\mathcal{H}_T\,(x(t)) = \begin{bmatrix} x(t) & x\,(t+T) & \cdots & x\,(t+(n-1)T\,) \end{bmatrix}, \tag{13}$$

and

$$D = \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix}(X'_{0,T}(X_{0,T}.X'_{0,T})^{-1})(\textstyle\int_0^T \mathcal{H}_T(x)d\tau). \tag{14}$$

Proof: (13) represents a time-varying matrix defined on the interval $t \in [0, T]$. The proof relies on the expression of $\mathcal{H}_T(\dot{x}(t))$ using data first and then removing its dependence on the state derivative in two steps. To this end, using (11) one has

$$\mathcal{H}_T(\dot{x}(t)) = \mathcal{H}_T((A + BF(x))x(t)). \tag{15}$$

Based on Lemma 1, in $A + BF(x) = X_{1,T}W(x)$, the state derivative measurements $X_{1,T}$ are required, which is not desired. However, $X_{1,T}$ can be expressed in terms of state and input data as (9). Therefore, using (9) while substituting (8) in (15) and taking the integral of (15) on the interval $[0, T]$ result in

$$\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)) = [B \quad A]\begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix}$$
$$\times(X'_{0,T}(X_{0,T}.X'_{0,T})^{-1})(\textstyle\int_0^T \mathcal{H}_T(x)d\tau). \tag{16}$$

where $X'_{0,T}\,(X_{0,T}.X'_{0,T})^{-1}$ is the pseudo-inverse of $X_{0,T}$ represented as $(X_{0,T}\,)^{\dagger}$. As can be seen, the derivative of the state of the system does not show up in (16), which makes the presented formulation more practical. By considering D from (14) in (16), we get (12) and this completes the proof. ■

By considering (12) in constraint (5), we can find the safe control policy using only input and state measurements. Optimal LQR control, however, requires a linear controller, which can further reduce the data requirements. The following corollary shows that for linear controllers, as a special case of polynomial controllers, the entire closed-loop system can be represented by data. Learning the closed-loop system directly to satisfy safety or optimality requires less amount of data than learning the open-loop dynamics first and then designing a safe controller.

Corollary 1: Consider the system (1) under the linear controller $u = Kx$. Let the assumptions and conditions of Theorem 1 be satisfied. Then, the state-input data-based representation of the closed-loop system is

$$\mathcal{H}_T(\dot{x}) = (A + BK)\mathcal{H}_T(x), \tag{17}$$

where

$$(A + BK) = (\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))(\textstyle\int_0^T \mathcal{H}_T(x)d\tau)^{-1}. \tag{18}$$

Moreover, this closed-loop representation can be leveraged to design safe or optimal controllers if the rank of $X_{0,T}$ is n and at least $n+1$ data samples are collected.

Proof: Using $u = Kx$ in (1), the closed-loop system becomes

$$\dot{x} = (A + BK)x. \tag{19}$$

Applying (13) in (19) results in (17). Taking the integral from this equation yields

$$\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)) = (A + BK)\int_0^T \mathcal{H}_T(x)d\tau.$$

As a result, $(A + BK)$ can be written as in (18). Besides, when the rank of $X_{0,T}$ is n, a solution to (18) exists. Moreover, when at least $n+1$ data samples are collected, the inverse is not unique, and thus $(\mathcal{H}_T(x)d\tau)^{-1}$ can be considered as a decision variable to directly learn a closed-loop system with safety satisfaction properties. ■

Since only the closed-loop system with the size $n \times n$ must be learned, $rank(X_{0,T}) = n$ needs to be satisfied. However, learning open-loop dynamics requires a state-input data matrix of rank $n + m$. This advantage will also be leveraged in the design of an optimal controller in the subsequent sections. When it comes to safety, however, even though a linear controller $u^s = K^s x$ makes the design simpler and more data-efficient, this safe control input might not cover the entire safe region for some systems and some safe sets.

IV. DATA-DRIVEN OPTIMAL CONTROL DESIGN

Finding the optimal control input that minimizes (2) for the linear system (1) leads to solving the algebraic Riccati equation [15], which requires complete knowledge of the system dynamics. To obviate this requirement, iterative RL-based approaches [13], [16], [17] have been presented to learn the optimal control policy. Despite their advantages, iterative RL algorithms have disadvantages compared to the one-shot optimization methods. A significant drawback of iterative RL approaches pertains to their demanding computational requirements. In contrast, the one-shot optimization strategies center around directly solving optimization problems to discover the best solution at once, displaying the potential for greater efficiency in computational processing and data utilization. Even though one-shot learning of LQR for discrete-time systems is considered [1], it is lacking for continuous-time systems.

To this end, in this paper, we present a one-shot learning approach for solving the LQR problem for continuous-time systems (i.e., learning the optimal controller u^L that minimizes (2)) using only state-input measured data. To this end, we leverage the following semi-definite program (SDP) representation of the LQR problem and turn it into a data-based optimization.

Lemma 2 ([14]): Consider the linear system (1) with the quadratic cost function (2). Then, the control gain that optimizes the cost is obtained by $K = YP^{-1}$ where P and Y are the solutions to

$$\min_{P,Y}(Trace(QP) + Trace(R^{\frac{1}{2}}YP^{-1}Y'R^{\frac{1}{2}})) \tag{20a}$$

$$\text{s.t. } AP + PA' - BY - Y'B' + I \prec 0, \tag{20b}$$

$$P = P' \succ 0, \tag{20c}$$

According to [14], [18], the objective function in (20) comprises the sum of two components, and the second term $\phi(P, Y) = Trace(R^{\frac{1}{2}}YP^{-1}Y'R^{\frac{1}{2}})$ can be represented as:

$$\phi(P, Y) = \min(Trace(X))$$

$$\text{s.t. } \begin{bmatrix} X & R^{\frac{1}{2}}Y \\ Y'R^{\frac{1}{2}} & P \end{bmatrix} \succ 0.$$

Therefore, (20) can be written as [14]:

$$\min \beta \tag{21a}$$

$$\text{s.t. } C(\beta, P, Y, X) \succ 0, \tag{21b}$$

where $C(\beta, P, Y, X)$ is given by [14]:

$$C(\beta, P, Y, X) = diag(C_1, C_2, C_3), \tag{22a}$$

$$C_1(\beta, P, Y, X) = -Trace(QP) - Trace(X) + \beta, \tag{22b}$$

$$C_2(\beta, P, Y, X) = -AP - PA' + BY + Y'B' - I, \tag{22c}$$

$$C_3(\beta, P, Y, X) = \begin{bmatrix} X & R^{\frac{1}{2}}Y \\ Y'R^{\frac{1}{2}} & P \end{bmatrix}. \tag{22d}$$

As discussed in [14], several issues related to optimal control problems can be expressed as convex programs containing a finite set of variables. The conditions (21) and (22) corresponding to the LQR problem are well-posed if R is positive definite ($R > 0$), (A, B) is controllable, and (A, Q) is observable. The following theorem presents a data-based version of the SDP optimization (21).

Theorem 2: Consider the system (1) and the optimization problem in (21). Let $u^L = Kx$ and consider the collected data in (6). The one-shot data-based representation of the SDP optimization problem in (21) with the constraints (22) becomes

$$\min \beta \tag{23a}$$

$$\text{s.t. } C(\beta, P, Y, X) = diag(C_1, C_2, C_3) \succ 0, \tag{23b}$$

where

$$C_1 = -Trace(QP) - Trace(X) + \beta, \tag{24a}$$

$$C_2 = \left((\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))D'(DD')^{-1}\right)\begin{bmatrix} Y \\ -P \end{bmatrix} + \begin{bmatrix} Y \\ -P \end{bmatrix}'\left((\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))D'(DD')^{-1}\right)' - I, \tag{24b}$$

$$C_3 = \begin{bmatrix} X & R^{\frac{1}{2}}Y \\ Y'R^{\frac{1}{2}} & P \end{bmatrix}. \tag{24c}$$

Proof: For system (1), if $u = Kx$, then, the closed-loop system is

$$\dot{x} = (A + BK)x.$$

According to [19] and (13), $\mathcal{H}_T$ has the capability to gather the past data. Therefore,

$$\mathcal{H}_T(\dot{x}) = (A + BK)\mathcal{H}_T(x). \tag{25}$$

Since directly measuring the state derivative $\dot{x}$ is not a typical scenario, as previously stated, (25) can be integrated to derive the following:

$$\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)) = (A + BK)\int_0^T \mathcal{H}_T(x)d\tau.$$

An aspect that worth taking into consideration is that C_2 in (22) could be written as

$$\begin{bmatrix} B & A \end{bmatrix}\begin{bmatrix} Y \\ -P \end{bmatrix} + \begin{bmatrix} Y \\ -P \end{bmatrix}'\begin{bmatrix} B & A \end{bmatrix}' - I \succ 0. \tag{26}$$

Note that the system dynamic shows up in the constraint (26). Based on (12), (14), and (16), the optimization problem in (21) with the mentioned constraints in (22) are turned into (23) and the constraints in (24), which completes the proof. ■

In (24), C_2 is replaced with data while the system dynamics is not required. Equation (23) with the mentioned constraints in (24) presents the data-driven optimization problem of (21) while the state derivative measurements are not required. Inspired by the proposed method in the previous section, this part proposes a data-driven version of (21).

Remark 2: Using approaches presented in Sections III and IV, the data-driven safe control design approach and data-driven one-shot approach to find optimal control input can be obtained. However, it is desired to consider both safety and optimality simultaneously in the design of control inputs for systems. To do this, in the next section, a new approach is presented to combine safe and optimal control inputs proactively.

V. DATA-DRIVEN INITIATIVE CONTROLLER

To consider both safety and optimality in the control design, an initiative control input u is considered as

$$u = \alpha(t)u^L + (1 - \alpha(t))u^s, \tag{27}$$

where $\alpha(t) \in [0, 1]$ determines the share of safe and optimal controllers in overall control input u.

In control policy (27), two special scenarios could happen: 1) the system trajectories are far from the safety boundary. Therefore, $\alpha(t) = 1$. In this case, $u = u^L$ and the performance specifications of the system are fully certified. 2) the system trajectories are very close to the safety boundary. So, the safe controller must dominate the optimal controller. Then, $\alpha(t) = 0$ and consequently, $u = u^s$. In this case, the system should only care about safety as it is a hard constraint all the time and should not be violated. In other situations, the goal is to find the largest value of $\alpha(t)$ so that the safety constraints are always satisfied while we can get as much performance as possible.

The goal is formulated as the following maximization problem

$$\max \alpha(t) \tag{28a}$$

$$\text{s.t.} \quad \frac{\partial h}{\partial x}(Ax + Bu) + \gamma h \geq 0. \tag{28b}$$

Solving the above optimization problem provides a proactive control approach that takes into account both safety and optimality aspects for systems described in (1). This approach differs significantly from existing reactive approaches [8], [9], where the system reacts with sharp control actions when safety constraints are about to be violated. Such reactive actions can be disruptive and potentially detrimental to the system.

Theorem 3: Consider the system (1) and the maximization problem in (28). Consider the collected data in (6). The data-based representation of the maximization problem in (28) with its constraint could be written as

$$\max \alpha(t) \tag{29a}$$

$$\begin{aligned} &\frac{\partial h}{\partial x}((\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))D'(DD')^{-1} \\ &\quad \times \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix}(X_{0,T})^{\dagger}x \\ &-\alpha(t)(RK^LP^{-1})'(u^L - U_{0,T}(X_{0,T})^{\dagger}x)) + \gamma h \geq 0. \end{aligned} \tag{29b}$$

Proof: For system (1), by substituting the proposed initiative control input (27) in the constraint of (28), one has

$$\frac{\partial h}{\partial x}(Ax + B(\alpha(t)u^L + (1 - \alpha(t))u^s)) + \gamma h \geq 0. \tag{30}$$

(30) can be simplified as

$$\frac{\partial h}{\partial x}((Ax + Bu^s) + B\alpha(t)(u^L - u^s)) + \gamma h \geq 0. \tag{31}$$

Substituting $u^s = F(x)x$ into (31) yields

$$\frac{\partial h}{\partial x}((Ax + BF(x)x) + B\alpha(t)(u^L - F(x)x)) + \gamma h \geq 0. \tag{32}$$

By using (8) and (9) in (32), one has

$$\begin{aligned} &\frac{\partial h}{\partial x}(\begin{bmatrix} B & A \end{bmatrix}\begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix}(X_{0,T})^{\dagger}x + B\alpha(t)(u^L - F(x)x)) \\ &\quad + \gamma h \geq 0. \end{aligned} \tag{33}$$

Substituting (12) into (33) yields

$$\begin{aligned} &\frac{\partial h}{\partial x}((\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))D'(DD')^{-1} \\ &\quad \times \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix}(X_{0,T})^{\dagger}x + B\alpha(t)(u^L - F(x)x)) + \gamma h \geq 0. \end{aligned}$$

Since $F(x) = U_{0,T}W(x)$, the maximization problem in (28) will be changed into:

$$\max \alpha(t) \tag{34a}$$

$$\frac{\partial h}{\partial x}((\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))D'(DD')^{-1}$$

$$\times \begin{bmatrix} U_{0,T} \\ X_{0,T} \end{bmatrix} (X_{0,T})^{\dagger} x$$
$$+B\alpha(t)(u^L - U_{0,T}(X_{0,T})^{\dagger} x)) + \gamma h \geq 0. \tag{34b}$$

As can be seen, only the B matrix shows up in (34). To obviate the requirement of B matrix, inspired by the fact that B can be computed by using predetermined u^L as

$$u^L = -R^{-1}B'Px,$$

here u^L, R and P are all known. Since $u^L = K^L x$, this computation can be expressed as follows:

$$K^L = -R^{-1}B'P \Rightarrow RK^L = -B'P, \tag{35a}$$
$$RK^L P^{-1} = -B' \Rightarrow B = -(RK^L P^{-1})'. \tag{35b}$$

In this part, the Lyapunov function candidate V has the following format:

$$V = x'Px,$$

where as mentioned earlier $P \succ 0$.

Substituting matrix B from (35) in (34) leads to (29) and this completes the proof. ■

Corollary 2: Consider the system (1). Assume the linear controller $u^s = K^s x$. The maximization problem in (28) can be presented in a data-driven manner as follows.

$$\max \alpha(t) \tag{36a}$$
$$\frac{\partial h}{\partial x}(\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))\big(\textstyle\int_0^T \mathcal{H}_T(x)d\tau\big)'$$
$$\Big((\textstyle\int_0^T \mathcal{H}_T(x)d\tau)(\int_0^T \mathcal{H}_T(x)d\tau)'\Big)^{-1} x$$
$$-\alpha(RK^L P^{-1})'(u^L - U_{0,T}(X_{0,T})^{\dagger} x)) + \gamma h \geq 0. \tag{36b}$$

Proof: If we consider $u^s = K^s x$ in (28), for this special case the constraint (31) will be turned into:

$$\frac{\partial h}{\partial x}(((A + Bk)x) + B\alpha(t)(u^L - K^s x)) + \gamma h \geq 0. \tag{37}$$

After substituting (18) in (37)

$$\frac{\partial h}{\partial x}((\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))\big(\textstyle\int_0^T \mathcal{H}_T(x)d\tau\big)'$$
$$\times \Big((\textstyle\int_0^T \mathcal{H}_T(x)d\tau)(\int_0^T \mathcal{H}_T(x)d\tau)'\Big)^{-1} x$$
$$+ B\alpha(u^L - K^s x)) + \gamma h \geq 0. \tag{38}$$

Substituting $K^s = U_{0,T}(X_{0,T})^{\dagger}$ in (38) results in

$$\frac{\partial h}{\partial x}((\mathcal{H}_T(x(T)) - \mathcal{H}_T(x(0)))\big(\textstyle\int_0^T \mathcal{H}_T(x)d\tau\big)'$$
$$\times \Big((\textstyle\int_0^T \mathcal{H}_T(x)d\tau)(\int_0^T \mathcal{H}_T(x)d\tau)'\Big)^{-1} x$$
$$+ B\alpha(u^L - U_{0,T}(X_{0,T})^{\dagger} x)) + \gamma h \geq 0. \tag{39}$$

By considering (39) and (35) instead of the constraint in (28), we get (36), and this completes the proof. ■

Note that by considering a linear safe control structure $u^s = K^s x$, the solution of the maximization problem in (36) is more conservative than the one in (34).

VI. STABILITY AND FEASIBILITY ANALYSIS

In this part, the proofs of recursive feasibility and asymptotic stability of the system, which play a prominent role, are provided. Since these proofs are inspired by the interpolating control method, first, the interpolating control is reviewed.

Interpolating control relies on a vertex representation approach, which involves smoothly interpolating between a controller located at the vertex of a feasible set and a high-gain feedback controller that prioritizes safety and robustness. The interpolation process is designed to ensure that the resulting control input stays within the feasible set while still achieving the desired performance. Here, based on the interpolating control theory, an interpolation coefficient is presented, which is $\alpha(t)$. Based on interpolating theory, $x(t)$ can be decomposed as follows [20]:

$$x(t) = \alpha(t)x^L(t) + (1 - \alpha(t))\,x^s(t), \tag{40}$$

where $\alpha(t) \in [0, 1]$. Given the interpolation coefficient $\alpha(t)$, one can obtain the control law (27). The proposed control law in (27) provides a smooth transition between two controllers. The primary benefit of using the vertex control scheme is that it provides a large domain of attraction for the system.

In the concept of interpolating control, the following constraint should be satisfied for the controlled invariant set $C_{\bar{t}}$:

$$C_{\bar{t}} = \big\{x \in R^n : F_{\bar{t}} x \leq g_{\bar{t}}\big\}$$

where $F_{\bar{t}}$ is a constant matrix, and $g_{\bar{t}}$ is a constant vector. The possible values for the feasible domain of vertex control, denoted as $C_{\bar{t}}$, can be as extensive as those for any other limited control method. One approach to address this limitation is to switch to a different, more aggressive, nearby controller, such as u^L, once the state reaches $\Omega_{\max}$, where $\Omega_{\max} = \{x \in R^n : F_0 x \leq g_0\}$ is the maximal invariant set. Interpolating control greatly decreases the amount of computational work needed [21].

Theorem 4: Recursive feasibility: The control method described by (27) and (40), which involves interpolating at regular intervals, is capable of achieving a feasible solution for system (1), for all initial states $x_0 \in C_{\bar{t}}$.

Proof: In terms of recursive feasibility, it has to be proven that $u(t) \in \mathbb{U}$, which means $F_u u(t) \leq g_u$ and $\dot{x}(t) \in C_{\bar{t}}$. To show $u(t) \in \mathbb{U}$, by using (27) one has

$$F_u u(t) = F_u(\alpha(t)u^L(t) + (1 - \alpha(t))u^s(t))$$
$$= \alpha(t)F_u u^L(t) + (1 - \alpha(t))F_u u^s(t)$$
$$\leq (\alpha(t)g_u + (1 - \alpha(t))g_u) = g_u,$$

and also by the use of (40) beside (27), one has

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$= A(\alpha(t)x^L(t) + (1 - \alpha(t))x^s(t)\,) + B(\alpha(t)u^L(t)$$

$$+ (1 - \alpha(t))u^s(t))$$
$$= \alpha(t)(Ax^L(t) + Bu^L(t)) + (1 - \alpha(t))(Ax^s(t) + Bu^s(t)).$$

Since $(Ax^L(t) + Bu^L(t)) \in \Omega_{\max} \subseteq C_{\bar{t}}$ and $Ax^s(t) + Bu^s(t) \in C_{\bar{t}}$, it follows that $\dot{x}(t) \in C_{\bar{t}}$. ■

Theorem 5: Asymptotic stability: Considering system (1), the control law (27) guarantees asymptotic stability for all initial states x_0. That is, the system will eventually reach a stable state, regardless of the initial starting point x_0.

Proof: It has to be proven that all solutions starting in $C_{\bar{t}} \backslash \Omega_{\max}$ will reach $\Omega_{\max}$ in finite time. One obtains

$$x(t) = \alpha^*(t)x^{*L}(t) + \left(1 - \alpha^*(t)\right) x^{*s}(t),$$
$$u(t) = \alpha^*(t)u^L(t) + \left(1 - \alpha^*(t)\right) u^s(t).$$

Also, the following is valid:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$= \alpha^*(t)\dot{x}^L(t) + \left(1 - \alpha^*(t)\right) \dot{x}^s(t),$$

where

$$\dot{x}^L(t) = Ax^{*L}(t) + Bu^L(t) \in \Omega_{\max},$$
$$\dot{x}^s(t) = Ax^{*s}(t) + Bu^s(t) \in C_{\bar{t}}.$$

We have considered $V = 1 - \alpha^*(t)$ as a Lyapunov candidate, which is a non-negative function. Consider the following linear programming problem:

$$\alpha^*(t) = \alpha^*(x(t)) = \max_{\alpha, r^L} \alpha \tag{41a}$$
$$\text{s.t.} \quad F_0 r^L \leq \alpha g_0, \tag{41b}$$
$$F_N(x(t) - r^L) \leq (1 - \alpha)g_N, \tag{41c}$$
$$0 \leq \alpha \leq 1, \tag{41d}$$

where $r^L = \alpha x^L$. The optimal solution of (41) is $\alpha^*(t)$. $\max(\alpha) = \alpha^*$ byfinding the best x^{*s} and x^{*L}. Also, we have:

$$\dot{\alpha}^*(t) = \alpha^*(\dot{x}(t)) = \max_{\alpha, r^L} \alpha \tag{42a}$$
$$\text{s.t.} \quad F_0 r^L \leq \alpha g_0, \tag{42b}$$
$$F_N(\dot{x}(t) - r^L) \leq (1 - \alpha)g_N, \tag{42c}$$
$$0 \leq \alpha \leq 1. \tag{42d}$$

It should be noted that $\alpha^*(t)$ is a feasible solution of (42) while $\dot{\alpha}^*(t)$ is the optimal solution of (42). Therefore, $\dot{\alpha}^*(t) \geq \alpha^*(t) > 0$. To guarantee asymptotic stability, we need to show that $\dot{V} = -\dot{\alpha}^* < 0$. Since $\dot{\alpha}^*(t) \geq \alpha^*(t) > 0$, the proof is completed. ■

The vertex control law ensures that the system is capable of achieving feasible solutions in a recursive manner, meaning that feasible solutions can be found repeatedly over time. Furthermore, it ensures that the system will be asymptotically stable. This means that the system will eventually converge to a stable state, and this stability will persist even if there are minor variations in the initial conditions of the system.

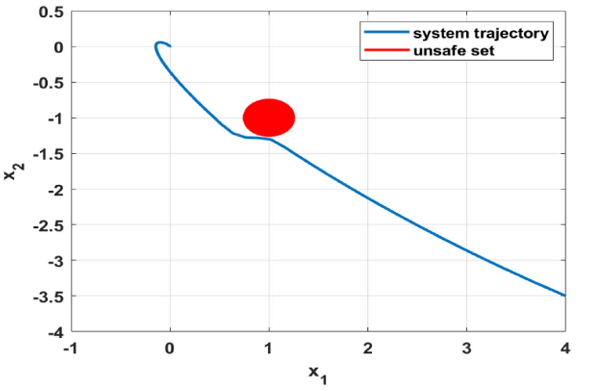

FIGURE 1. The trajectory of the system in the presence of the unsafe set.

VII. SIMULATION RESULTS

In this section, two numerical examples have been brought to verify the effectiveness of the proposed approach.

A. EXAMPLE 1:

Consider the double integrator system, which is a canonical example of a second-order control system, the state space model has the following format:

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = u.$$

The matrices regarding the cost function in (2) are being considered as follows:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$R = 1.$$

The initial condition has been set as $x_0 = [4, \ -3.5]'$, and $\gamma = 10$. The unsafe set $\mathbb{X}_u$ is defined as $\mathbb{X}_u = \{(x_1, x_2) : (x_1 - 1)^2 + (x_2 + 1)^2 - (1/4)^2 < 0\}$.

As can be seen in Fig. 1, the unsafe set is shown as a red ellipse and the system trajectory has been shown by the blue line. This figure shows the system trajectory converges to the equilibrium point. Fig. 2 demonstrates the control input of a data-driven initiative controller, which indicates both safe control and optimal control input. In Fig. 3, the trajectory of $\alpha(t)$ has been brought, which is, as mentioned earlier always between zero and one.

B. EXAMPLE 2:

In order to demonstrate our approach better and consider a realistic example, the lane-keeping problem has been considered for an autonomous vehicle as another numerical example.

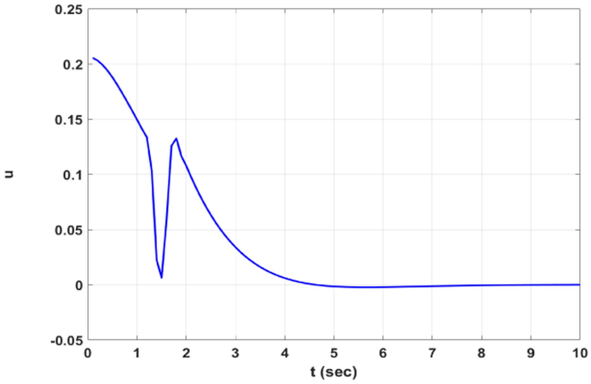

FIGURE 2. Initiative controller input.

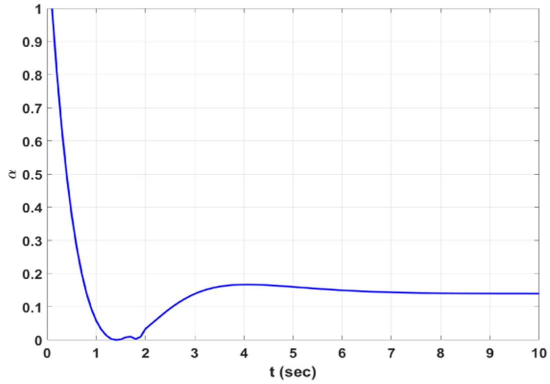

FIGURE 3. Evolution of $\alpha(t)$ VS time.

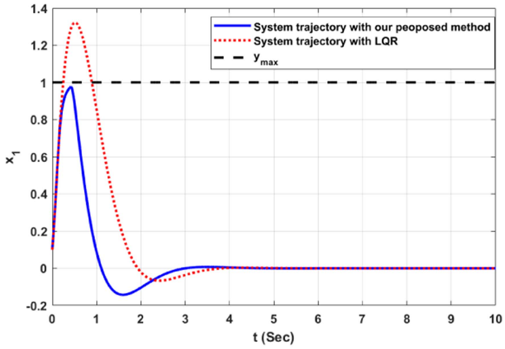

FIGURE 4. The trajectory of the lane-keeping system in the presence of the unsafe set.

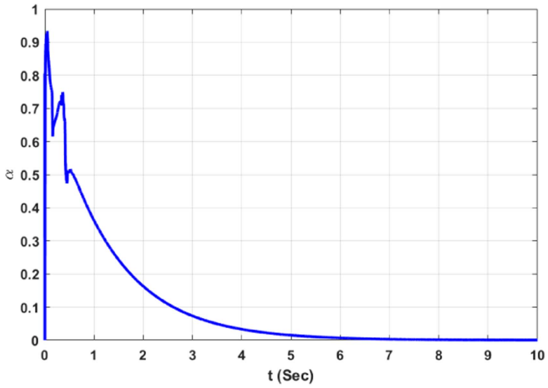

FIGURE 5. Evolution of $\alpha(t)$ VS time in the lane-keeping system.

The state space model of the lane-keeping problem is given as:

$$\begin{bmatrix}\dot y\\ \dot v\\ \dot\phi\\ \dot\psi\end{bmatrix} = \begin{bmatrix}0 & 1 & V_0 & 0\\ 0 & -\frac{C_f+C_r}{MV_0} & 0 & -\frac{bC_r-aC_f}{MV_0}-V_0\\ 0 & 0 & 0 & 1\\ 0 & -\frac{bC_r-aC_f}{I_zV_0} & 0 & 0\end{bmatrix}\begin{bmatrix}y\\ v\\ \phi\\ \psi\end{bmatrix} + \begin{bmatrix}0\\ \frac{C_f}{M}\\ 0\\ \frac{aC_f}{I_z}\end{bmatrix}u, \tag{43}$$

where $V_0 = 27.7\,(\text{m/s})$, $C_f = 133000\,(\text{N/rad})$, $C_r = 98800\,(\text{N/rad})$, $M = 1650\,(\text{kg})$, $I_z = 2315.3\,(\text{m}^2.\text{kg})$, $a = 1.11\,(\text{m})$, and $b = 1.59\,(\text{m})$.

The problem aims to seek to maintain the vehicle's position in the center of the driving lane. The details of this system model can be found in [22]. The states of the system could be denoted as:

$$\mathbf{x} = \begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix} = \begin{bmatrix}y\\ v\\ \phi\\ \psi\end{bmatrix}^T$$

The values of the parameters in (43) are brought from [23]. The initial condition has been set as $x_0 = [0.1, -0.1, 0.2, 0]'$, and $\gamma = 10$. The unsafe set $\mathbb{X}_u$ here is defined as $\mathbb{X}_u = \{y = x_1 : y = x_1 > 1\}$. The R and Q matrices regarding the cost function in (2) has been considered as:

$$Q = 10 \times I_{4\times 4},$$

$$R = 1.$$

As can be seen in Fig. 4, the unsafe set is shown as a black dashed line ($y_{\max}$) and the system trajectory with our proposed method has been shown by the blue line. In Fig. 5, the path of $\alpha(t)$ is presented, which ranges from zero to one.

VIII. CONCLUSION

This article has developed a novel control law that merges safety and optimality at the same time. The key advantage of our proactive approach is its independence from knowledge of the system dynamics. In our proposed method, a data-driven safe controller is formulated, and optimal control input is computed based on data using a one-shot optimization problem. The initiative maximization problem has been formulated based on input/state data, eliminating the need for state derivatives in the formulation. The recursive feasibility and asymptotic stability of the control input have been proven. It is shown that the coefficient allocated to safety and optimal control input is proportional to its distance from the safety boundary, indicating a trade-off between safety and optimality. Simulation results have been presented to demonstrate the effectiveness of the proposed method.

REFERENCES

[1] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 909–924, Mar. 2020.

[2] C. De Persis, M. Rotulo, and P. Tesi, "Learning controllers from data via approximate nonlinearity cancellation," *IEEE Trans. Autom. Control*, vol. 68, no. 10, pp. 6082–6097, Oct. 2023.

[3] M. Guo, C. De Persis, and P. Tesi, "Learning control for polynomial systems using sum of squares relaxations," in *Proc. IEEE 59th Conf. Decis. Control*, 2020, pp. 2436–2441.

[4] A. Nejati, B. Zhong, M. Caccamo, and M. Zamani, "Data-driven controller synthesis of unknown nonlinear polynomial systems via control barrier certificates," in *Proc. Learn. Dyn. Control Conf.*, 2022, pp. 763–776.

[5] C. De Persis and P. Tesi, "Low-complexity learning of linear quadratic regulators from noisy data," *Automatica*, vol. 128, 2021, Art. no. 109548.

[6] X. Tan and D. V. Dimarogonas, "Distributed implementation of control barrier functions for multi-agent systems," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 1879–1884, 2022.

[7] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. IEEE 18th Eur. Control Conf.*, 2019, pp. 3420–3431.

[8] H.-N. Nguyen, P.-O. Gutman, S. Olaru, and M. Hovd, "Implicit improved vertex control for uncertain, time-varying linear discrete-time systems with state and control constraints," *Automatica*, vol. 49, no. 9, pp. 2754–2759, 2013.

[9] W. Ren and D. V. Dimarogonas, "Event-triggered tracking control of networked multiagent systems," *IEEE Trans. Autom. Control*, vol. 67, no. 10, pp. 5332–5347, Oct. 2022.

[10] M. F. Reis, A. P. Aguiar, and P. Tabuada, "Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria," *IEEE Contr. Syst. Lett.*, vol. 5, no. 2, pp. 731–736, Apr. 2021.

[11] A. Luppi, A. Bisoffi, C. De Persis, and P. Tesi, "Data-driven design of safe control for polynomial systems," *Eur. J. Control*, vol. 75, 2024, Art. no. 100914.

[12] F. Blanchini et al., *Set-Theoretic Methods in Control*, vol. 78. Berlin, Germany: Springer, 2008.

[13] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[14] E. Feron, V. Balakrishnan, S. Boyd, and L. El Ghaoui, "Numerical methods for H2 related problems," in *Proc. Amer. Control Conf.*, 1992, pp. 2921–2922.

[15] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, NJ, USA: Wiley, 2012.

[16] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Third Quarter 2009.

[17] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, Jun. 2018.

[18] V. Balakrishnan and L. Vandenberghe, "Semidefinite programming duality and linear time-invariant systems," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 30–41, Jan. 2003.

[19] V. G. Lopez and M. A. Müller, "An efficient off-policy reinforcement learning algorithm for the continuous-time LQR problem," in *Proc. IEEE 62nd Conf. Decis. Control*, 2023, pp. 13–19.

[20] S. Scialanga, S. Olaru, and K. Ampountolas, "Reduced-complexity interpolating control with periodic invariant sets," *Int. J. Control*, vol. 96, pp. 757–769, 2023.

[21] H.-N. Nguyen, *Constrained Control of Uncertain, Time-Varying, Discrete-Time Systems* (Lecture Notes in Control and Information Sciences), vol. 451. Berlin, Germany: Springer, 2014.

[22] E. J. Rossetter and J. Christian Gerdes, "Lyapunov based performance guarantees for the potential field lane-keeping assistance system," *J. Dyn. Syst. Meas. Control*, vol. 128, no. 3, pp. 510–522, 2006.

[23] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.