IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 May 2022, accepted 11 July 2022, date of publication 15 July 2022, date of current version 20 July 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3191343*

SURVEY

# Machine Learning in Event-Triggered Control: Recent Advances and Open Issues

**LEILA SEDGHI[1], ZOHAIB IJAZ[1], MD. NOOR-A-RAHIM[1], KRITCHAI WITHEEPHANICH[2], AND DIRK PESCH[1], (Senior Member, IEEE)**
[1]School of Computer Science and IT, University College Cork, Cork, T12 K8AF Ireland
[2]Department of Electrical and Electronic Engineering, Munster Technological University, Cork T12 P928, Ireland

Corresponding author: Leila Sedghi (leila.sedghi@cs.ucc.ie)

This publication has emanated from research conducted in part with the financial support of Science Foundation Ireland under Grant number 16/RC/3918 (CONFIRM Centre for Smart Manufacturing).

**ABSTRACT** Networked control systems have gained considerable attention over the last decade as a result of the trend towards decentralised control applications and the emergence of cyber-physical system applications. However, real-world wireless networked control systems suffer from limited communication bandwidths, reliability issues, and a lack of awareness of network dynamics due to the complex nature of wireless networks. Combining machine learning and event-triggered control has the potential to alleviate some of these issues. For example, machine learning can be used to overcome the problem of a lack of network models by learning system behavior or adapting to dynamically changing models by continuously learning model dynamics. Event-triggered control can help to conserve communication bandwidth by transmitting control information only when necessary or when resources are available. The purpose of this article is to conduct a review of the literature on the use of machine learning in combination with event-triggered control. Machine learning techniques such as statistical learning, neural networks, and reinforcement learning-based approaches such as deep reinforcement learning are being investigated in combination with event-triggered control. We discuss how these learning algorithms can be used for different applications depending on the purpose of the machine learning use. Following the review and discussion of the literature, we highlight open research questions and challenges associated with machine learning-based event-triggered control and suggest potential solutions.

**INDEX TERMS** Event-triggered control, networked control systems, machine learning, reinforcement learning, deep reinforcement learning, statistical learning.

## I. INTRODUCTION

Networked Control Systems (NCSs) have recently sparked a lot of interest as they provide solutions to a variety of technical problems in areas such as automotive systems, process control, smart manufacturing, smart grids, and autonomous driving. An NCS can be divided into two components: a cyber network and a physical component. This makes NCS a class of cyber-physical systems (CPSs) that are distributed across a network.

CPSs are the integration of physical processes, networking, and computation. Physical processes are controlled using a feedback loop, where the physical process has an influence on the computation of the system and vice versa [1]–[4]. As shown in Fig. 1, components of the NCS, such as sensors, controllers, and actuators, are connected via a communication network, such as an Ethernet-based Fieldbus, a wireless network, or even the Internet. An NCS is used to control a physical object or process and, through control feedback, adapts to changing environment conditions in real-time. Traditionally, wired network technology has been used in NCSs due to its reliability and stability. However, there has been a recent shift towards wireless NCSs due to ease of installation, flexibility, including mobility, and lower costs [5].

In general, communication between agents (, i.e. sensors, actuators, controllers) in an NCS has traditionally been

The associate editor coordinating the review of this manuscript and approving it for publication was Haipeng Yao.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

implemented using a time-periodic approach, with agents communicating at regular intervals. One of the difficulties with this approach is determining the sampling period (how frequently agents need to communicate). The sampling period is usually kept low to avoid data loss during system transients. In NCSs, the sampling period is typically set to approximately 20% of the total available network bandwidth [6]. However, allocating a constant sampling period is not always the best approach because it either negatively affects the system performance during transients by selecting too low a sampling period or chooses a high sampling period, which wastes limited network bandwidth by sending redundant control signals during steady-state. To address this issue, event-triggered control techniques have been proposed in which the communication is not time-periodic but the communication frequency increases during transients in the system and decreases during steady-state [7]. Event-Triggered Control (ETC) reduces the continuous utilization of network resources observed in time-periodic communication by acting only when the relevant information is available. These actions include the transmitting of a packet from a sensor to a controller or rescheduling the control tasks when several tasks are running concurrently on the same processor [8]. During execution intervals, ETC operates in an open-loop mode until the next update arrives.

In recent years, Machine Learning (ML) techniques have been combined with ETC to improve communication and control performance. In ETC, event-triggered condition is state dependent and full states information is required by the controller so that it can execute the next update [9]. It is not straightforward, however, to linearize the model uncertainties for the ETC [10] for systems that deal with dynamic changes such as in factory environments or where the system's dynamics change as the load changes, such as in smart grid systems. ML can be used to obtain a model estimate for the controller in order to implement ETC. In conjunction with ETC, ML can be implemented for example in smart grid systems, consensus-based Unmanned Aerial Vehicles (UAVs), and a wide variety of other applications. The only disadvantage of combining ML and ETC is that it will increase the controller's computational load, as ML requires a large amount of data and computations based on that data. In the literature, ML has been used in combination with ETC to accomplish three fundamental goals:

- **Model dynamics learning:** ETC requires access to the system model in order to work effectively, therefore, ML is widely used to learn plant models [11]. Unknown model parameters can be learned or improved by designing a state estimator using ML algorithms. Different ML approaches, such as Statistical Learning (SL), Reinforcement Learning (RL), and Neural Networks (NNs), have been developed to learn and update system models in order to make the model robust against disturbances and uncertainties. By using ML, we can even reduce the requirement for an actual model of the system, as the controller will learn the dynamics of the system using ML [12].
- **Solving an optimization problem:** It is often difficult to solve an optimal control problem for a nonlinear system because the equations are partial differential and usually do not have a closed form solution. For example, solving the Hamilton–Jacobi–Bellman (HJB) equations is a difficult problem in the field of optimal control. ML techniques such as RL or Adaptive Dynamic Programming (ADP) are used to find an approximate solution to optimization problems such as the HJB equations.
- **ML for joint learning and optimization:** While RL and ADP have been applied to solve nonlinear optimal control problems, the dynamics of some systems are excessively complex because of their highly nonlinear nature. This leads to a situation where the system model is partially or completely unknown to the controller, and learning the model should be taken into account prior to executing the control optimization step. A combination of RL and NN has been used to simultaneously learn the system dynamics and solve the optimization problem. The Actor-Critic-Identifier (ACI) approach, which is based on RL and is presented to approximate the Hamilton-Jacobi Bellman (HJB) equation, is widely used in this context. Typically, three NN structures are employed in ACI, with actor and critic NNs approximating the optimal control and optimal value functions, respectively, and NN identifier approximating the uncertain system dynamics [13]. Typically, in ML for joint learning and optimization problems, identifier NN is used to learn system dynamics and critic RL is used to solve event-triggered optimization problems.

While numerous techniques have been proposed in the literature to address the aforementioned goals, to the best of our knowledge, no survey has been conducted on how ML can be applied in ETC systems to address changing dynamics, resource management, uncertainties, and disturbances. To address this gap and to stimulate further research and innovation in this area, we present a comprehensive survey of the state-of-the-art of ML-based ETC. We provide a detailed discussion on ML for ETC, divided into three sections: ML for learning model dynamics, ML for optimal control and communication problems, and ML for joint learning and optimization. We have also categorized the state-of-the-art by determining whether ML can be used to learn control behavior (e.g., control inputs), communication behavior (decisions), or both. Both the control and communication policies are identified in the state-of-the-art overview tables. A number of promising research directions and trends are also discussed.

In summary, this study makes the following major contributions:

- Classification of the ML-based ETC literature: By analyzing ML-based ETC methods presented in [11], [12], [14]–[61], ML-based ETC can be classified into three

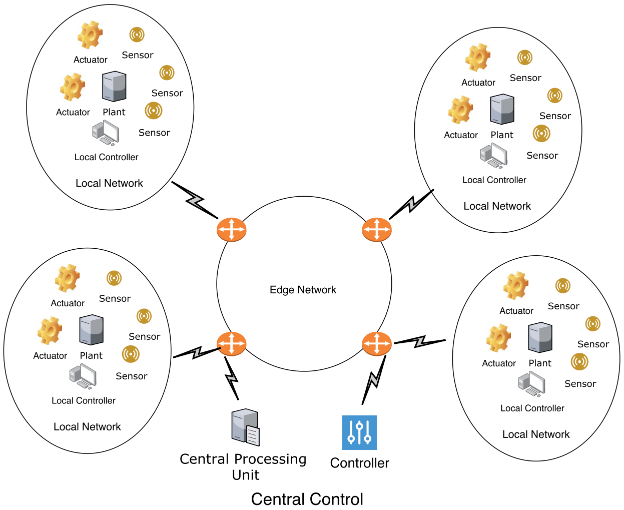

**FIGURE 1.** An example of a networked control system.

parts depending on the purpose of machine learning use; ML for model dynamics learning, ML for solving a control optimization problem, and ML for joint system dynamics learning and control optimization.

- Classification of ML-based ETC literature in accordance with learning techniques including SL, NN, RL, and Deep Reinforcement Learning (DRL). We also explain the advantages and disadvantages of these learning methods.
- Classification of ML-based ETC literature by control and communication properties, including control structure and controller.
- Reviewing articles that use ML to learn control laws or communication laws, or both, where a learning agent optimizes control actions and communication decision are made.
- Presentation of selected open issues and research trends in ML-based ETC.

The rest of the paper is organized as follows. We provide a brief overview of ETC and ML in Section II. In section III, we present ML techniques used for model dynamics learning in combination with ETC. Optimization-based ML is discussed in section IV. ML articles that discuss joint learning and optimization are reviewed in section V. Open issues and future research directions for ML-based ETC are discussed in section VI. Finally, we conclude our discussion in section VII.

## II. PRELIMINARIES

### A. EVENT-TRIGGERED AND SELF-TRIGGERED CONTROL

Traditional time-triggered networked control approaches use a fixed sampling period, leading to time-periodic communication between the agents in the system. This is often ineffective as it uses network resources even when no updates are required. ETC reduces the continuous utilization of network resources observed in time-periodic communication by acting only when the relevant information is available or needs to be communicated. This makes ETC reactive in nature to when an event is detected. Another, related event-driven control approach, Self-Triggered Control (STC), is proactive and predicts the occurrence of an event based on the system model and current measurements [62]. ETC requires extra hardware resources to continuously monitor the output of the system, which may increase the cost and complexity of the system [63]. To overcome this problem, STC was proposed in which the next sampling time is calculated at the current instant and the output of the system is only monitored at sampling instances [64]. Numerous studies have attempted to eliminate the need for continuous monitoring in ETC. For instance, continuous communication and self-state monitoring are avoided in [65]. In [66], mixed time- and event-triggered observers are presented to estimate the state of a system with discontinuous monitoring. The most significant benefit of merging time-triggered observers and event-triggered observers in an architectures is that it removes Zeno behavior by default.

Choosing suitable event-triggered threshold is crucial due to determining communication interval and controller updates. Conventionally, a state-dependent static threshold was employed, with communication occurring only when the difference between the current state and the previously transmitted state deviates from a predefined constant threshold [67]. Efforts have been made to make this threshold dynamic. In the dynamic event-triggered threshold case, the triggering not only depends on the output of the state but also on the internal variables of system dynamics to adjust the triggering mechanism dynamically with time [68]. Another triggering threshold mechanism is adaptive event-triggering, in which an optimization algorithm obtains a triggering threshold which is not only dynamically changing but also adaptive to the change in system dynamics [69]. To avoid Zeno behaviour and guarantee the minimum inter-sampling time, a hybrid event-triggering technique is proposed in [70]. While proposed event detection is not continuous, a minimum inter-sampling time is defined such as in periodic communication, which will ensure that the system will avoid Zeno behaviour. Event-detection is performed after this minimum sampling time has elapsed. Some recent advances in ETC are discussed in the following survey paper [9], [71], [72]. ETC has been used recently in various fields considering various uncertainties such as delay and packet loss. For example, ETC is applied to asynchronous control of cyber-attacks in [73]. Actuator non-linearity, and sensor saturation are also considered under a new design of ETC based on fuzzy Markov jump systems in [65].

### B. MACHINE LEARNING

ML is a set of algorithms that make decisions based on available data in order to predict and/or optimize the performance of a system. A good definition of what learning involves is as follows: "A computer program is said to learn from experience E with respect to some class of tasks T and a performance measure P if its performance at tasks in T, as measured by P, improves with experience E" [74]. In ML-based ETC, machine learning is used to achieve various goals, including learning model dynamics, solving optimal control problems, and joint learning and optimization. The methods used to achieve these goals are statistical learning (SL), neural networks (NN), reinforcement learning (RL), and Deep RL (DRL). These methods are briefly explained in the following.

#### 1) STATISTICAL LEARNING

SL approximate solutions to complicated control problems that are costly to solve exactly [75]. To analyze and learn from data sets, SL focuses on statistical properties. For example, Empirical Risk Minimization (ERM) is a concept in SL that defines algorithms that yield theoretical bounds on performance [75]. In general, a SL algorithm is fed a training set as an input, which is sampled as an unknown distribution and labeled with a target function, and the output is a predictor that finds the minimized error with respect to an unknown distribution and target function. Since the learner does not have any information about the unknown distribution and target function, the true error is not accessible directly to the learner. An error that can be calculated by the learner is the training error that a classifier incurs over the training sample, which is also known as an empirical error or empirical risk. ERM searches for the solution that minimizes the empirical error. SL has been used in existing ML-based ETC works to learn the dynamics of the model. For example, SL is used to learn a new model based on statistical properties of inter-communication time. It is also used in combination with controllers such as Linear Quadratic Control (LQR) or Model Predictive Control (MPC) and makes these controllers robust against uncertainties via increasing prediction and estimation. As SL is a combination of statistics and ML, building an SL model requires a good understanding of statistical properties of the data.

#### 2) NEURAL NETWORKS

A neural network (NN) is an ML technique in which data is processed similarly to the way the human brain works with sensory data. When an input is provided to the NN, it will generate the best possible results via adopting to changing input without redesigning the output criteria. For example, Radial Basis Function (RBF) NNs have been used as a tool for modeling nonlinear functions in control engineering due to their simple structure and good accuracy [76]. RBF networks can approximate an unknown function with a linear combination of a group of nonlinear functions, called base functions. In nonlinear systems, the system dynamics are unknown, which means the ETC framework cannot be directly applied. RBF NN is a powerful method applied in many areas of engineering due to its flexibility in adapting data distribution, fast training, and short run-time [77]. However, NN usually requires a lot of data in comparison with more traditional ML algorithms, which makes the approach not suitable for many problems where data is limited.

#### 3) REINFORCEMENT LEARNING

RL is an ML method in which agents take actions by trial and error and, in return, receive rewards based on those actions from the environment in which they operate. At each time step, the agent takes an action that may result in a transition to a new state of the environment. Then, the agent receives a reward based on the quality of this transition. RL agents estimate policy and value functions. The value function looks at the agent's current situation in the environment, while the policy function looks at how the agent makes decisions.

RL can be categorized into three types: actor-only, critic-only, and actor-critic methods, where the terms "actor" and "critics" are used instead of policy and value function, respectively [78]. Although the value function method has been successful with discrete lookup table parameterization, this method failed to generalize when applied to continuous function approximation. Q-learning and deep Q-learning are examples of this method. On the other hand, policy function methods have strong convergence guarantees in comparison with the value function method, which is quite inefficient even when applied to simple examples with few states [79]. While the policy function approach has been successful in continuous and stochastic environments and has a faster convergence, value functions are more sample efficient and steady. Therefore, actor-critic methods merge these two approaches to benefit from both and achieve a better result. In the actor-critic approach, the actor performs an action on the environment, and the critic evaluates the values of the action and sends feedback information to the actor [52]. Based on our review, RL algorithms using both critic and actor-critc methods have been developed to learn the model of the system, solve an optimization problem, and perform joint learning and optimization. However, the actor-only method is not used in any of the reviewed articles. Using RL means taking action based on rewards helps to learn the dynamics of a system accurately. However, RL is not preferable for solving simple problems or for solving problems that need a lot of data.

#### 4) DEEP REINFORCEMENT LEARNING

Traditional ML approaches exhibit problems when dealing with high-dimensional data, which has recently become more widely available. This has led to the development of the concept of deep-learning (DL) [80]. DL is a subset of ML based on NNs that uses multiple layers of non-linear information processing for both supervised and unsupervised feature extraction of data. [81]. DL can be combined with RL, which helps to overcome RL's limitation to domains with fully observed and low-dimensional state-spaces. This combination, deep DRL, can easily find compact low-dimensional features in high-dimensional data [82].

## III. LEARNING MODEL DYNAMICS

In this section, we review works that have used ML to learn the dynamics of the system model reported in Table. 1. As the accuracy of the available system model has a direct impact on control performance, it is possible to improve closed-loop performance by updating and improving the model during operation using data. The available model can be improved by learning an uncertainty compensation model and designing a state estimator supported by ML or by learning unknown model parameters. In ETC, the efficacy of the control approach dependents on the availability of an accurate dynamic model. ML can be used to learn such a model. Combining ML with ETC makes the control system more robust against disturbances and uncertainties, while computational load as well as communication bandwidth can be reduced significantly.

### *A. LEARNING MODEL DYNAMICS WITH STATISTICAL LEARNING*

Although learning methods have the potential to improve system performance, performing a learning task is expensive (e.g., including communication resources and computation costs). Therefore, several articles [11], [14], [15] consider event-triggering rules for model learning, which determine when a new model should be learned based on statistical properties of intercommunication time. These articles develop learning triggers through the derivation of model-induced probability distributions and the observation of intercommunication times. Additionally, statistical estimates can be obtained by using concentration inequalities such as Hoeffding's inequality and the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality.

A novel Event-Triggered Learning (ETL) approach is applied to linear Gaussian systems by combining State Estimation and SL in [11]. This combination will result in higher prediction accuracy and cheaper communication costs, even when compared to Event-Triggered State Estimation (ETSE), because the model will be improved through learning. ETSE's effectiveness in reducing communication is entirely dependent on the prediction's accuracy, or in other words, the model's quality. When the present model's prediction performance is low, learning experiments are triggered to improve the model using the available data. Model learning can be solved with a standard least-square estimator. Hoeffding's inequality is considered here as the concentration inequality to quantify the confidence level. The approach demonstrated a reduction in communication effort in both simulation and hardware implementation of a cart-pole system. The same authors' subsequent work [14] extended ETL by adding a Kalman filter and demonstrating the impacts with new illustrative use cases. Moreover, they used the DKW inequality in addition to Hoeffding's inequality to provide more detailed statistical information because it provides bounds on the empirical Cumulative Distribution Function (CDF). However, no disturbance is considered in [11] and [14] and the results are only developed for linear Gaussian systems. The work could be extended to non-linear dynamic systems to investigate how ML may support ETL with non-linear systems.

The framework used in [11] is further developed in [15] to include a control loop based on the concept of event-triggered pulse control mixed with SL to learn dynamic models, as illustrated in Fig. 2. This strategy is beneficial when the initial model is poor or when the dynamics have changed, causing learning to occur. A learning trigger decides whether or not the system model is accurate enough. If the accuracy is insufficient, the learning of a new model is triggered (see the green section in Fig. 2). The authors introduced two different triggers, (i) state trigger ($\gamma_{ctrl}$), which initiates communication of control commands when $\delta$ (a user-defined threshold) is exceeded; and (ii) a learning trigger ($\gamma_{learn}$), which initiates learning in the case of poor performance. When an event occurs, the system is reset to its equilibrium state via the application of a pulse whose duration is determined by the dynamics of the system. A plant is considered with sensors and actuators with noise and disturbance ($v$ and $\epsilon$ ). Hoeffding's inequality is used to quantify the confidence level in the estimation. According to the authors, two major characteristics of this method are its ability to adapt to changing dynamics and to discover an appropriate alternative for the integral control component used in periodic control. Numerical simulation demonstrates that learning system dynamics has the effect of reducing communication effort, coping with load disturbances, and changing dynamics. All numerical simulations, however, were conducted on first-order systems, not higher-order systems.

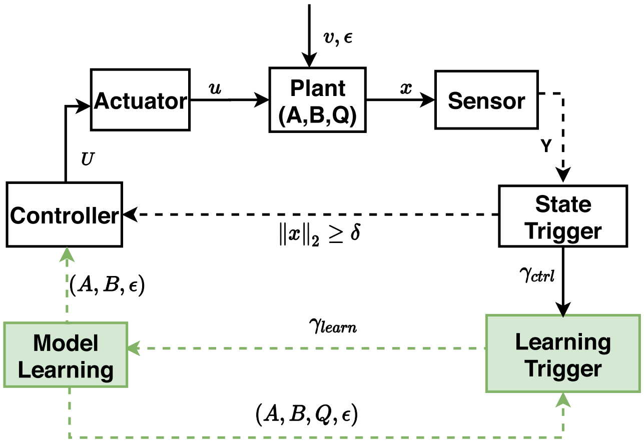

**FIGURE 2.** **Event triggered learning diagram [15].**

In contrast to [11], [14], and [15], which use the model of the system for prediction of the next trigger event, which is based on communication, the authors of [18] use the model for control purposes, and triggering is based on control performance. LQR is used in [18] to minimize the expected value of the cost function by combining it with SL theory. Model learning is activated when performance improvements are required. Hoeffding's inequality and the Chernoff bound are used as concentration inequalities to obtain an effective trigger with enough theoretical guarantees. In this approach, when the empirical cost of the Riccati equations over a finite horizon exceeds the Chernoff Bound, learning is triggered. Least square estimation is used to perform the learning. The proposed method is implemented to control the pole-balancing performance of a rotary pendulum. The authors validate whether the trigger is capable of detecting these changes and evaluate model accuracy by adjusting the ball joint and magnetic weights in the pendulum. SIMULINK is used to implement a switched controller with a sample rate of 500 Hz, and an LQR with an integrator is used to stabilize the upright position. Least squares estimation is used to perform the learning.

While the previously described works applied event triggering to the learning process to build an accurate model, the authors of [16] study event triggering applied to the control process. Here, event-triggered MPC is combined with ERM, which makes the control system adaptive and robust to uncertainties and state errors. MPC is a form of optimal control that can tackle multi-variable systems and handle hard constraints on input, state, and output variables by solving a finite-horizon open-loop optimization problem [83]. The main goal of [16] is to attenuate the unknown disturbance by designing an uncertainty compensator using ML, or, more precisely, by using ERM with kernel regression to predict the system state subject to uncertainties and learning an uncertainty compensation model to obtain the bound of uncertainty. In fact, by applying the ERM as an SL approach, restrictions that require a known upper bound of uncertainty or a known structure of uncertainty (constant or harmonic), which are standard assumptions when designing robust MPC or adaptive control, are eased. It is worth noting that [16] does not use online ML and the compensator is not updated during the control operation.

### *B. NEURAL NETWORKS (NNs)*

An online approximate ETC is designed for nonlinear multi-input-multi-output (MIMO) uncertain systems using NNs in [17]. The controller is approximated by utilizing a linearly parameterized NN. A system state vector-based event-triggered condition is described, and the condition is made adaptive (state dependent, monotonically increasing) in order to achieve a trade-off between ETC approximation and resource utilization. A novel NN weight update law, as shown in Fig. 3, ensures the reduction in network resource utilization and relaxes the required knowledge of the whole dynamic system. NN weights are updated through an event trigger mechanism in an aperiodic manner, as illustrated in Fig. 3. As a result, the proposed method requires less computation than traditional NNs that update periodically. Event-triggered communication in [17] is also a function of NN weight estimates and system states, whereas traditional ETC is a function of system states, resulting in overall less computation. To demonstrate Lyapunov stability, the event-triggered system is modeled as a nonlinear impulsive dynamical system. According to the authors' simulations, the strategy produced a 45 percent reduction in computation burden when compared to the periodic method.

In contrast to [17], which uses a linearly parameterized NN to approximate the controller, [19], [22], [24] use RBF NN to approximate unknown functions. Similar to [17], the NN weights are also updated at event-triggered instances in [19]. An adaptive ETC is proposed without making any assumptions about input-to-state stability, even while the model of the plant is uncertain, in order to formulate a practical dynamic system. The adaptive ETC approach is used in conjunction with a NN to develop a pure feedback controller for nonlinear systems with nonlinear model uncertainty. The proposed method uses adjustable dynamic sampling states to update the controller model and the adaptive control law. In contrast to a constant state-dependent threshold, a dynamic threshold is employed for ETC. The NN weights and controller are only updated when the desired control specification cannot be guaranteed. While utilizing Lyapunov stability to evaluate the model, the proposed ETC reduces computation and communication load. Since general state-feedback systems are not considered in [19], an event-triggered based controller for pure feedback systems is proposed using adaptive NN tracking in [24]. Here, the mean value theorem is used to transform the pure-feedback nonlinear system into a strict-feedback nonlinear system. A NN approximates the output tracking error in finite time to bound the error close to zero using finite-time prescribed performance, which will guarantee the same performance for both transient and steady-state. ETC is used to obtain a large (both fixed and variable) threshold. The adaptive NN ensures that all signals in a closed loop are bounded, which is verified by applying the Lyapunov stability theory.

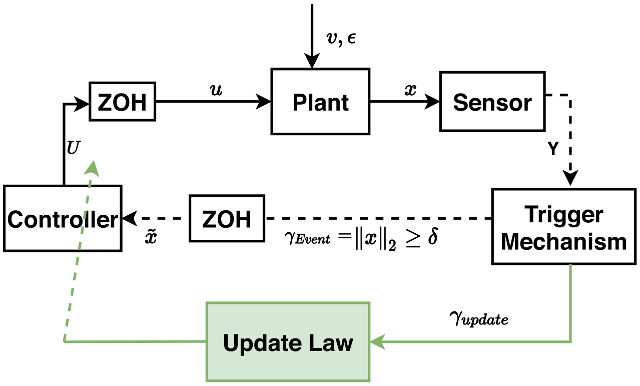

**FIGURE 3.** NN-based ETC [17].

[19], [24] an adaptive ETC problem is studied for a class of electromagnetic suspension systems with unknown parameters utilizing back-stepping technology in [22]. RBF NN is used to approximate the unknown functions. An ETC with a fixed threshold strategy and the relative threshold control approach are devised and compared in order to reduce communication resources. The authors proposed for future work the development of an intelligent controller that can switch between fixed threshold and relative threshold based on threshold size.

An adaptive ETC problem is extended for non-affine nonlinear multiagent systems with uncertainties including dynamic disturbance, model-free dynamics, and dead-zone input in [25]. In [25], RBF NN is used to approximate the unknown function, similar to [19], [22], [24] and the ETC back stepping design procedure is combined. Both unknown dead-zone and model-free dynamics are considered simultaneously in [25] for multiagent systems. Adaptive ETC RBFNN back stepping control is studied more, including in [26] for MIMO switched nonlinear systems with output and state constraints and non-input-to-state practically stable (ISpS) model-free dynamics, in [20] for completely unknown nonlinear functions with dynamic gain, and in [27] for under-actuated marine surface vessels using an NN-based disturbance estimator. Adaptive ETC RBF NN is also presented in [21] for a class of single-input-single-output uncertain nonlinear continuous-time (CT) systems by integrating input-to-state linearization techniques, impulsive dynamical system and RBF NN with adaptive ETC threshold.

### C. REINFORCEMENT LEARNING

An STC based on Gaussian Process (GP) regression is developed in [23] for NCS with unknown system dynamics. In [23], a joint learning algorithm is proposed that uses RL to learn the dynamics of the plant and a self-triggered controller to reduce the number of communication time steps for NCS. An infinite horizon optimal control problem has been formulated that takes into account both the control and communication costs. The MPC problem is solved by using the GP dynamics of the plant to obtain control input for STC. The authors divide this framework into two parts, the execution phase and the learning phase. During the first phase, the STC is implemented in an epsilon-greedy [84] fashion, in which a random control input with one-step intercommunication time is sampled; otherwise, the computed optimal control and communication policy is executed. In the second (learning) phase, the learning agent uses the training data to update the GP model of the plant and compute the optimal control and communication policies.

Deep RL (DRL) is based on combining RL with a deep learning algorithm to generate an efficient learning algorithm. In [12], DRL has been used to simultaneously learn control and communication behaviour of a model-free system and then utilise this DRL for ETC to reduce sampling. A RL problem is formulated as a resource-aware control strategy, where the learning agent optimizes its control input and communication decisions to maximize the expected reward over the time horizon. The reward function comprises two terms, one is to capture control performance, and the other gives a reward for time steps without communication. Two learning ETC approaches are proposed. In the first learning approach, only communication is considered using feedback control, but in the second learning approach, both control and communication are simultaneously considered, which is called end-to-end learning. In terms of ETC, end-to-end learning emphasizes learning of both communication and control models simultaneously, rather than separating them. During the training of the agent, it receives negative rewards for bad performance (early termination of the episode) and for every communication. In an RL task, the agent's interaction with the environment is divided into episodes. An agent receives a constant positive reward to prevent an unwanted early termination of the episode. The authors use joint learning for control and communication, which reduces communication by using a parameterized action space Markov decision process.

### D. ACTOR-CRITIC RL

Adaptive tracking control based on dead-zone event-triggered RL is presented in [28] for a nonlinear CT system with external disturbances and unknown dynamics without the Persistently Exciting (PE) condition and initial stabilizing control. To approximate an unknown long-term performance index, controller, critic, and action NNs are used. To demonstrate the developed controller's performance, an autonomous underwater vehicle model was chosen for simulation. The ETC threshold increases monotonically, and the system is Uniformly Ultimately Bounded (UUB).

## IV. ML FOR OPTIMAL (CONTROL AND COMMUNICATION) PERFORMANCE

In this section, we discuss papers that employ ML to address optimization problems, as reported in Table. 2. RL and Adaptive Dynamic Programming (ADP) can satisfy both optimal control policy and optimal performance simultaneously [45]. In general, the RL method includes an actor to improve performance via interacting with the external environment and a critic to evaluate the control performance of the actor [33]. RL approaches have been applied to solve a variety of optimization problems, including optimal regulation problems [85], robust control problems [86], and differential games, including zero-sum games [87] and non zero-sum games [88]. Moreover, ADP and RL methods were developed to estimate the solution of the Hamilton–Jacobi–Bellman (HJB) [89] and the Hamilton– Jacobi–Isaacs (HJI) [90] equations. ADP, as a potential technique for obtaining satisfying solutions to HJB equations, can be classified into three primary categories: Heuristic Dynamic Programming (HDP), Dual Heuristic Dynamic Programming (DHDP), and Globalized Dual Heuristic Dynamic Programming (GDHP) [49].

Recently, ETC has been integrated with RL and ADP algorithms to increase computing efficiency and conserve communication resources. To solve non-convex optimization problems, a distributed stochastic gradient descent algorithm combined with an event-triggered communication mechanism has been proposed in [29]. In the following, we have classified papers into two categories based on their approach to solving optimization problems: critic-only method and actor-critic method. In some works, the critic-only method replaces the common actor-critic structure to simplify the iterative framework and implementation process.

### A. REINFORCEMENT LEARNING - CRITIC-ONLY

Actor Critic Learning (ACL) has been successfully applied to a variety of robust control problems as a technique that combines dynamic programming and NN to create a highly effective method for solving specific optimization problems [35]. However, the control system required for ACL implementation must be persistently excited. Concurrent learning (CL)

**TABLE 1.** Existing ETC systems proposed in the literature - ML for dynamic model learning.

| Reference | System Type | Learning Approach | Communication policy | Control policy | Multi-agent | Centralized / Decentralized / Distributed | Control Mechanism | Experimental Validation |
|---|---|---|---|---|---|---|---|---|
| [11] , [14] | Linear Gaussian System | SL | ✗ | ✓ | ✗ | Cent. | State Feedback Controller | ✓ |
| [15] | Linear time-invariant | SL | ✗ | ✓ | ✗ | Cent. | Simple Gain | ✗ |
| [16] | Nonlinear (DT) | SL | ✗ | ✓ | ✗ | Cent. | MPC | ✗ |
| [17] | Affine Nonlinear (CT) | NNs | ✗ | ✓ | ✗ | Cent. | State Feedback Controller | ✗ |
| [18] | Nonlinear | SL | ✗ | ✓ | ✗ | Cent. | NN | ✓ |
| [19] | Non-affine Nonlinear | RBFNN | ✗ | ✓ | ✗ | Cent. | Backstepping Control | ✓ |
| [22] | Nonlinear | RBFNN | ✗ | ✓ | ✗ | Cent. | Backstepping Control | ✗ |
| [25] | Non-affine Nonlinear | RBFNN | ✗ | ✓ | ✓ | Distributed | Backstepping Control | ✗ |
| [26] | Nonlinear |  | ✗ | ✓ | ✗ | Cent. | Backstepping Control | ✗ |
| [27] | Nonlinear CT | RBFNN | ✗ | ✓ | ✗ | Cent. | Feedback Linearization | ✗ |
| [21] | Nonlinear | RBFNN | ✗ | ✓ | ✗ | Cent. | Backstepping Control | ✗ |
| [20] | Nonlinear | RBFNN | ✗ | ✓ | ✗ | Cent. | Backstepping Control | ✗ |
| [24] | Non-affine Nonlinear | RBFNN | ✗ | ✓ | ✗ | Cent. | Backstepping Control | ✓ |
| [23] | Nonlinear | RL | ✓ | ✓ | ✗ | Dist. | Optimal control | ✗ |
| [12] | Nonlinear | DRL | ✓ | ✓ | ✗ | Cent. | State Feedback Control | ✓ |
| [28] | Nonlinear (CT) | RL | ✗ | ✓ | ✗ | Cent. | Adaptive Control | ✗ |

or Experience Replay (ER), which combine historical and current state data, may allow the Persistent Excitation (PE) condition to be relaxed. CL's main idea is to apply batch-like dynamics to parameter estimation dynamics by utilizing recorded input and output data [91]. For instance, [30] and [42] applied CL to guarantee parameter convergence without requiring PE.

Event-triggered Concurrent Learning (ETCL) is presented in [30] for solving the HJI equation of a $H_\infty$ control problem for a class of CT nonlinear systems with external perturbation. The authors defined the $H_\infty$ control problem as a two-player zero-sum game in which the control minimizes the cost function in the worst-case disturbance. An adaptive triggering condition is also obtained for the closed-loop system using an ETC policy and a time-triggered disturbance policy. In the ETCL algorithm, a single critic NN is used for implementation purposes. Additionally, a novel critic tuning law based on the CL technique is used, which allows the traditional PE condition to be relaxed. The results are compared with a concurrent RL algorithm used in [92] and a synchronous policy iteration algorithm (SPIA) in [93], and the ETCL method is found to be superior in terms of performance against disturbances. Notably, concurrent learning-based ETC requires robust estimation techniques because it requires knowledge of state derivatives, which are typically not directly sensed.

An event-triggered optimal control problem with Integral Reinforcement Learning (IRL) is proposed to solve the HJB equation of CT nonlinear systems with partially unknown dynamics in [31]. IRL is a class of RL methods, which are developed based on policy iteration and value iteration, using iterative methods to achieve the optimal solution asymptotically by minimizing the integral temporal difference error at each step [94]. In comparison to [52], which uses joint model learning and optimization, this method does not require any NN-identifier to identify the unknown internal dynamics. A single-critic NN is used in [31] to approximate the optimal value function and the optimal control policy for implementation. The UUB of critic weights are validated via the Lyapunov theory. However, no disturbance is considered, and the method is dependent on the initial admissible policy. The number of controller updates is significantly reduced during the simulation result learning process.

The method's applicability is limited by the fact that the approaches in [30] and [31] are dependent on an initial stabilizing control policy and consider an undiscounted cost function. Therefore, the goal of [32] and [33] is to benefit from ML to obtain the event-triggered nonlinear discounted optimal control law that is independent of the initial condition.

In [32], training NNs using a learning rule resulted in a near-optimal discounted event-based control law that is independent of the initial condition in an adaptive critic framework. Discounted optimal control considers stage costs, which are weighted by a time-varying decaying term [95].

The discount factor in the cost function can adjust the convergence speed of the regulation design and reduce the final value of the optimal cost function. In [32], the mentioned method is applied to industrial systems such as power systems, as an example of an affine nonlinear system. The stability of a closed-loop system is considered an impulsive model, and its stability is determined using the Lyapunov technique. The controller's performance demonstrates that when the discount factor is increased, the optimal cost decreases, validating the results of event-based near-optimal control performance with discounted cost functions. Additionally, controller updates are reduced by up to 66.76% during the learning process for power applications. However, the proposed method requires knowledge of the dynamic model and constrained control inputs are not taken into account.

Event-triggered $H_\infty$ tracking control is combined with RL for a CT nonlinear system with external disturbances in [33]. An event-triggered tracking HJI equation is developed based on an augmented system with the tracking error dynamics and a discounted cost function to solve the $H_\infty$ problem. The HJI nonlinear partial difference equation is solved using a novel RL with a critic network that approximates the optimal cost function independently of the initial admissible control policy. The Lyapunov theory is used to determine the stability of closed-loop systems. The proposed method has several characteristics, including UUB of weights in critic NNs and asymptotic convergence of tracking error to zero. The simulation results indicate that ETC requires 55 samples, whereas a time-triggered controller requires 200 samples, demonstrating the reduction in computing burden while achieving asymptotic tracking. Constrained control inputs, on the other hand, are not considered.

Input constraints such as actuator saturation are significant physical characteristics of actuators in industrial applications and must be considered. Therefore, to address the weakness of not considering constrained control inputs in [32], input constraints are taken into account in [35] and [34] through the use of a discounted cost function. Moreover, while the disturbance policy is updated using a time-driven strategy and the control policy is updated using an event-triggered mechanism in [30] and [33], both the control and the disturbance policies in [35] and [34] are updated using an event-driven mechanism, significantly reducing the computational load in comparison to other works in the literature that only update the control policy in the event-driven mechanism.

ETC is used in conjunction with adaptive critic design in [34] to study nonlinear systems with mismatched perturbations and input constraints. By defining an infinite-horizon cost function, the robust stabilization problem is transformed into a constrained $H_2$ optimal control problem. As a result of solving the event-triggered HJB equation, the system states are UUB. A single network adaptive critic design, which is used for solving HJB, is tuned via the gradient descent method. All signals in the closed-loop system are proven to be UUB via the Lyapunov method. The proposed method has a limit when applied to nonlinear, complicated systems due to the difficulty of computing the Moore–Penrose pseudoinverse of the control matrix function.

An event-driven HJI equation associated with a two-person zero-sum game is proposed in [35] for CT nonlinear systems with a disturbance. An $H_\infty$ control problem with asymmetric input constraints has been proposed. The $H_\infty$ control problem is converted into a zero-sum game that can be solved using ACL. ADP, ACL, and RL algorithms are often similar as they have the same characteristics. ACL uses historical data and instantaneous state data to update both control and disturbance in the event-driven mechanism. Zeno behavior is also excluded without the requirement of properly selecting disturbance attenuation. Then, using ACL, the event-driven HJI equation is solved and its weight parameters are tuned by applying the gradient descent method. UUB is guaranteed using the Lyapunov approach. The results indicate that when ACL is used with both event-driven control and event-driven disturbance, the computational load is reduced by up to 60%. This method is also applicable to systems that have an equilibrium point at the origin. This is a limitation of the method, as obtaining information about controlled systems and knowledge of their control matrix is difficult in real-world applications. Similar to [35], a zero-sum game problem is solved in [36], for nonlinear safety-critical systems with safety constraints and input saturation using a barrier function. A critic NN is developed to approximate the optimal safety value function of the HJI equation, and a novel event-triggered scheme is used to obtain the update instant of the control law and the disturbance law. The CL is also used to relax the PE condition.

While [35], [36] present zero-sum games, [37], [38] applies an event-triggered IRL algorithm to a non-zero-sum game problem. To address asymmetric input saturation, novel non-quadratic value functions with a discount factor are used in [37]. To alleviate the need for a comprehensive understanding of the game, an IRL-based coupled Hamilton–Jacobi equation is derived. To relax the PE condition, the weights of a single critic NN are tuned based on the ER method.

While [30]–[35], [37], [39], [42], [45] studied optimal control problems for nonlinear CT systems, [40] investigates ETC near-optimal problems for input-constrained nonlinear discrete time systems with the input-to-state stability (ISS) attribute subject to actuator saturation. First, the robust control problem of the uncertain system is converted to a near-optimal control problem via the designed cost function. Then adaptive ETC is designed to save computational resources. To improve the control performance, a goal representation adaptive critic design is presented, which consists of two NNs, namely the goal network and the critic network. A goal network is used to learn the external reinforcement reward and provide a more efficient internal reinforcement reward for the critic network with non-periodic weight updating.

In [41], an adaptive self-learning control approach is designed with matched uncertainties (for a plant whose model is uncertain) using an event-triggered critic cost control approach. ADP is used to solve the optimal control problem in a learning-based, forward-in-time approximation fashion. An event-triggered cost control approach using a self-learning technique for nonlinear systems is designed. The controller design is transferred to an optimal control problem with an event-based strategy to have a robust optimal control design. The event-triggered threshold dynamically changes in response to changes in the system's states. A NN is used to implement event-based optimal control with stability guarantees using Lyapunov stability. Learning and guaranteed cost control of the proposed method are limited to nonlinear systems with matched uncertainties and do not include unmatched uncertainties. The proposed method could be improved to track a trajectory based on learning.

ML-based ETC for the decentralized structure of nonlinear systems with uncertain interconnections is discussed in [42]. Decentralized ETC is developed with ACL and ER for a class of CT nonlinear systems with uncertain interconnections. A critic network is used to solve the event-triggered HJB equations related to optimal ETC laws of the subsystems. Gradient descent and ER are used to update the critic network's weights. ER helps to relax the PE condition. The estimated weight vectors used in the critic networks are proven to be UUB through a classic Lyapunov approach. Overall stability is also achieved based on the stability of decentralized ETC subsystems. Controller updates decreased by up to 60%, indicating a significant reduction in computational load. However, prior knowledge of the interconnected system is necessary in the proposed method, which limits the applicability of this method to a wide variety of engineering industries. ML-based decentralized ETC is also studied in [43] for nonlinear large-scale decentralized control problems with matched interconnections. A single critic network is used to solve the optimal control problem of HJB for nominal isolated subsystems, which decreases the computational cost and avoids the approximation error caused by the actor network. The critic network is updated via modified gradient descent with an additional stability term, and there is no requirement for the initial stabilizing control.

Additionally, ML-based ETC is applied to systems that are subject to denial of service attacks. For example, in [44] an iterative single critic learning framework is used in conjunction with ETC to consider the denial-of-service attack for autonomous driving systems, which effectively balances the frequency and changes in adjusting the vehicle's control during the running process. A single critic network is designed to approximate the optimal cost function and obtain an HJB solution.

### B. REINFORCEMENT LEARNING-ACTOR CRITIC

Online IRL is applied to nonlinear CT systems with external disturbances via an event-triggered mechanism based on robust constrained control problems in [45]. The event triggered $H_\infty$ tracking control problem is formulated in [45] as a two-player zero-sum game with a non-quadratic function for constrained inputs. The $H_\infty$ controller provides a robust optimal design for nonlinear systems. An $H_\infty$ optimal control problem could be formulated in the zero-sum game, based on Basar and Bernhard's theory [96]. Solving zero-sum games, which is a min-max optimization problem, is normally more preferable than directly solving the $H_\infty$ problem. The solution to the event-triggered condition is approximated through an actor-critic structure and a HJI equation. Event-triggered optimal constraint control is obtained through actor NN, and the optimal cost is evaluated based on ADP through a critic NN. Lyapunov stability is also used to validate the closed-loop system's stability.

In [46], an infinite-horizon optimal adaptive learning problem is formulated to design the control and triggering mechanism of a model-free system. Based on Q-learning, a model-free approach has been derived which will also guarantee the exclusion of Zeno behaviour. An actor-critic structure is selected to adaptively tune the optimal ETC and Q-function for a model-free system using RL to optimize the problem online in order to minimize cost. The system is validated using Lyapunov stability analysis.

Similar to [46], infinite horizon integral control is used in [47]. The authors begin by converting the event-triggered robust nonlinear control problem into an event-triggered nonlinear optimal control problem by constructing an infinite horizon integral cost for the nominal system, whose dynamics are unknown. Then the robust ETC of the original system can be derived via solving the event-triggered nonlinear optimal control problem. A recurrent NN is used to develop the unknown system dynamics and, using these dynamics, a critic network is proposed using adaptive critic design to find the solution to the ET HJB equation. The event-triggered threshold is considered constant and static. The system is validated using Lyapunov stability to show that the system is UUB to origin for all states.

Nonlinear multiagent systems are studied in [48] via distributed recursive RL ETC. The RBF NN critic and actor are applied to estimate the long-term strategic utility function and the uncertain dynamics in multiagent systems, respectively. The multi-gradient recursive strategy is tailored to learn the NN weights, which avoids the local optimal problem in gradient descent-based methods and decreases the dependence of the initial value. Semi-global UUB of all signals in a multiagent system is proven. Combining RL and ETC improved the energy conservation of multiagent systems by reducing the amplitude of the controller signal and the controller update frequency, respectively.

A DHDP strategy combined with self-learning optimal regulation for an event-driven adaptive control algorithm has been proposed in [49]. The DHDP strategy is used to formulate an event-based optimal regulation for discrete time nonlinear systems to reduce the cost. The input-to-state stability (ISS) analysis is proposed for a nonlinear plant. DHDP is a sub-domain of ADP and is used to solve the HJB equations. As shown in Fig. 4, the solid lines denote the flow of state information and the dashed lines denote the back propagation path for both actor and critic networks, where $u_k$ denotes the control input and $x_k$ denotes the state of the system. When compared to the traditional DHDP method, the proposed DHDP technique significantly reduces computation cost and resource utilization while maintaining performance. The ETC has a dynamic threshold. However, in [49], an additional assumption is required that the state norm is bounded by the supremum of the control input norm. For reducing these kinds of assumptions, ETC explainable GDHP is presented for nonlinear discrete-time systems to deal with asymmetric input constraints by integrating an integral function and the actor network in [50]. An explainable GDHP algorithm is presented to solve the HJB equation online, and the calculations for the derivative of the cost function are relaxed without matrix dimensionality transformations. However, the triggering condition is based on the state feedback scheme, and full-state feedback is required.

In [51], using parallel control, a novel event-triggered near-optimal control problem for unknown discrete-time nonlinear systems is studied. To achieve parallel control, the control input is introduced into the feedback system via an augmented nonlinear system with an augmented performance index. The control stability of an augmented nonlinear system is analyzed, and by selecting an appropriate augmented performance index, the optimal control of the augmented system can be viewed as close to optimal control of the original system with the original performance index. Then, a novel ETC based on parallel control and critic-actor network structure is applied without reconstructing unknown systems, thereby avoiding identification errors caused by other learning model approaches. In this method, the initial control input can be set arbitrarily, but control constraints are not considered.

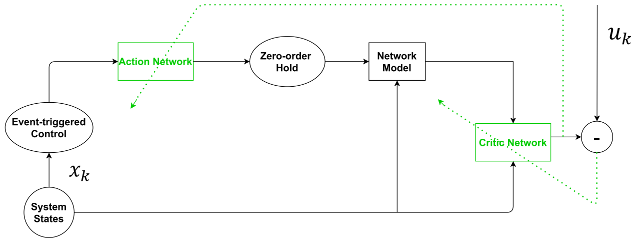

**FIGURE 4.** **Event-triggered Adaptive Critic Architecture. [49].**

## V. JOINT LEARNING AND OPTIMIZATION

In this section, we review articles, as shown in Table. 3, that aim to achieve two main goals: learning system dynamics, which are unknown, and solving optimization problems for event-triggered optimal control problems. To achieve these goals, an identifier-critic architecture is used by combining RL and NN. In the first step, the identifier NN is applied to learn the system dynamics, and in the second step, the critic NN is utilized to obtain the event-triggered optimal controller [52]. In some cases, the actor NN is also used.

### A. REINFORCEMENT LEARNING-CRITIC ONLY

The model-free RL approach is utilized to simultaneously learn an optimal ETC and the model of the system through an identifier-critic architecture in [52]. More precisely, the feed-forward NN identifier is used to learn the unknown system dynamics, and the critic NN is used to obtain the event-triggered optimal controller. Standard back-propagation algorithms and e-modification methods [97] are used together to update the identifier NN. A modified gradient descent method is also used to tune the critic NN. Closed-loop system stability is analysed based on the Lyapunov method, and a single-link robotic arm system is chosen as a nonlinear example for simulation. However, this method is inapplicable to nonlinear systems with non-affine inputs. Similar to [52], an event-driven DRL optimization algorithm is developed in [53] to reduce the energy consumption of data centers. The advantage of ETC over fixed periodic control is the ability to make decisions based on specific events (such as overheating). Event-driven optimization significantly reduces the number of regulatory decisions while assuring adequate system performance. Combining DRL with the high-dimensionality and high dynamics of data centers enables the nonlinear, dynamic aspects of the IT workload and thermal process to be captured. It is demonstrated that event-driven DRL can detect events more effectively, reduce regulatory decisions by 70% to 95%, and achieve comparable or even greater energy efficiency. The results of [53] are compared to [98] and [99].

$H_\infty$ event-driven control design based on ACL has been developed to deal with the data-based optimization for a class of unknown nonlinear systems in [54]. A two-player zero-sum differential game adaptive critic controller is designed by combining the event-driven design formulation with a data driven learning identifier used to formulate a nonlinear $H_\infty$ control problem. The unknown dynamics of the plant are learned using a NN-based data-driven design. A unique critic network is considered to solve the event-driven HJI equation. However, disturbance updating is in the time-driven mechanism, which will necessitate choosing the prescribed level of disturbance attenuation appropriately to keep the event-triggering threshold non-negative state dependent and monotonically increasing. The system is validated using Lyapunov stability analysis.

In [55], a neuro-dynamic programming-based ETC method for unknown non-affine nonlinear systems with input constraints is presented. Similar to [52], [54], a NN identifier is created to discover the unknown system dynamics given input constraints. The value function for solving the event-triggered HJB equation is then approximated using a critic NN. The ETC method can decrease computational load, communication expenses, and bandwidth. This method is applicable to both affine and non-affine systems employing NN identifiers with measurable input and output data.

In [56], a decentralized ETC problem is studied for a class of constrained nonlinear interconnected systems. By assigning a distinct cost function to each restricted auxiliary subsystem, the control problem is transformed into the selection of optimal control policies. An event-triggered HJB

**TABLE 2.** Existing ETC systems proposed in the literature - ML for optimal Control.

| Reference | System Type | Learning Approach | Communication policy | Control policy | Multi-agent | Centralized / Decentralized / Distributed | Control Mechanism | Experimental Validation |
|---|---|---|---|---|---|---|---|---|
| [33] | Nonlinear (CT) | RL | ✗ | ✓ | ✗ | Cent. | $H_\infty$ Control | ✗ |
| [30] | Nonlinear (CT) | CL | ✗ | ✓ | ✗ | Cent. | $H_\infty$ Control | ✗ |
| [39] | Affine Nonlinear (CT) | ACL | ✗ | ✓ | ✗ | Cent. | $H_\infty$ Control | ✗ |
| [34] | Nonlinear (CT) | ACL | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| [45] | Nonlinear (CT) | IRL | ✗ | ✓ | ✗ | Cent. | $H_\infty$ Control | ✗ |
| [31] | Nonlinear (CT) | IRL | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| [42] | Nonlinear (CT) | RL & ER | ✗ | ✓ | ✓ | Decent. | HJB | ✗ |
| [43] | Nonlinear (CT) | RL | ✗ | ✓ | ✓ | Decent. | HJB | ✗ |
| [32] | Affine Nonlinear (CT) | NNs | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| [29] | Nonlinear | DL | ✓ | ✗ | ✓ | Cent. | – | ✗ |
| [35] | Nonlinear (CT) | ACL | ✗ | ✓ | ✗ | Cent. | $H_\infty$ control | ✗ |
| [36] | Nonlinear (CT) | RL (Critic-only) | ✗ | ✓ | ✗ | Cent. | $H_\infty$ control | ✗ |
| [37] | Nonlinear (CT) | IRL | ✗ | ✓ | ✗ | Cent. | Optimal control | ✗ |
| [40] | Nonlinear-Discrete | RL | ✗ | ✓ | ✗ | Cent. | State Feedback Optimal control | ✗ |
| [46] | Linear | RL | ✓ | ✓ | ✗ | Cent. | LQR | ✗ |
| [47] | Nonlinear | ACD | ✗ | ✓ | ✗ | Cent. | Optimal Control | ✗ |
| [48] | Nonlinear | ACL | ✗ | ✓ | ✓ | Distributed | Optimal Control | ✗ |
| [49] | Nonlinear | DHDP | ✗ | ✓ | ✗ | Cent. | Optimal Control | ✓ |
| [50] | Nonlinear | Explainable GHDP | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| citelu2022event | Nonlinear | ACL | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| [38] | Affine Nonlinear | IRL | ✗ | ✓ | ✗ | Cent. | Adaptive Control | ✗ |
| [44] | Nonlinear | RL | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |

solution has rendered the system stable and UUB. Utilizing an identifier-critic network architecture relaxes the system's dynamic constraints. An identifier network and a critic network are utilized to identify unknown internal dynamics and approximate optimal cost functions, respectively. Optimizing the weights of the critic network using gradient descent. Combining ETC and RL results in less data transfer and enhanced system performance (less control cost and shorter convergence time).

Some data-driven model research [57], [58] has been conducted based on constructing models with recurrent neural networks (RNNs) for completely unknown nonlinear systems in order to eliminate identification error and respond quickly to dynamic system changes in system identification. In [57], a data-driven model based on RNNs is developed to construct the system uncertainties, including the drift dynamics and the input gain matrix. In the data-driven model, the modeling error caused by NN approximation is eliminated by including a compensation term. A critic NN can approximate the solution of the HJB equation, which significantly simplifies the ACL implementation architecture. In their problem, the authors of [58] incorporated input constraints and external disturbances to extend the work in [57]. By developing an integral Bellman equation in IRL, the authors of [59] eliminate the system identification procedure. The proposed IRL makes the algorithm suitable for systems whose drift dynamics are unknown. The ETC ADP technique for tracking control of partially unknown systems with constraints and uncertainties is developed. After constructing an augmented function, the optimal tracking control problem with uncertainty is transformed into the optimal regulation of the nominal augmented system with a discounted value function; consequently, the requirement for partial system knowledge is relaxed through the use of IRL. The critic and actor NNs are used, the learning of NN weights is event-triggered, and the initial admissible control requirement is relaxed. However, this method cannot be applied to systems with unmatched uncertainty

### B. REINFORCEMENT LEARNING - ACTOR CRITIC

An event-triggered HDP $\lambda$ optimal control strategy for nonlinear discrete time systems with unknown dynamics has been developed in [60]. Iteratively, HDP $\lambda$ takes into account a parameter for long-term prediction, the $\lambda$. Although long-term prediction increases accuracy and accelerates the rate of learning, it poses a formidable challenge to control systems with limited bandwidth and computational units. Therefore, ETC ensures system stability and reduces the need for computation and communication. ACI structure or model-actor–critic NN structure is utilized, in which the model NN or identifier NN evaluates the system state in order to obtain $\lambda$-return of the current time target value. Then, actor and critic NN are employed to approximate the event-triggered optimal control signal and the one-step return value, respectively. The Lyapunov approach is used to ensure the UUB stability of the system and the absence of NN weight errors [60].

In [61], an event triggered distributed $H_\infty$ constrained control problem for physically interconnected large-scale partially unknown systems with constrained-input and external disturbance is studied. Using an event-triggered feed-forward control policy, the control of physically interconnected large-scale systems is transformed into equivalent event-triggered control of decoupled multiagent systems. This method has the advantage of learning the solution to the HJI equation by combining the NNs of the critic, identifier, actor, and disturber into one. By omitting three NNs for each agent in a multiagent system, computational complexity and resources are significantly decreased.

## VI. DISCUSSION AND OPEN ISSUES

Based on our review of the literature, we can identify a number of open issues and challenges for ML-based ETC/STC systems. We outline some key issues in the following and suggest approaches to address them.

### A. COMMUNICATION ERRORS

Reliable real-time data transmission is critical for wireless automation, as it requires real-time system state information from remote observers to determine appropriate control actions. Network-induced packet errors and loss, as well as long and variable communication delays, often occur in wireless communication networks. This is caused by erroneous wireless channels, contention in multi-access wireless communication [100] and packet re-transmissions to reduce packet error rates. In particular, quantization errors, communication delays, and packet loss can cause instability in closed-loop control systems. The existing works on ML-based event-driven control in our comprehensive review in sections III, IV, and V have not considered the impact of network-induced imperfections in their learning algorithms. In other words, they assume a perfect communication scenario in the sensor-controller communication link and the controller-actuator communication link.

Ignoring network-induced imperfections makes the current results of ML-based event-driven control superficial for real-world applications. Therefore, developing a framework that considers relevant network-induced imperfections is necessary by extending and integrating current results. To cope with communication delays and packet dropouts, several measures need to be taken into account, including: (i) building new data sets; (ii) adapting learning techniques based on the imperfections; and (iii) developing strategies to tackle packet loss and delay. In the following, we present the impact of communication imperfections on ML-based event-driven control in more detail.

#### 1) PACKET LOSS

The majority of prior ML-ETC research has assumed that information transmitted by a sender is always successfully received by the receiver. In practice, however, this is not the case. If a packet is lost during transmission from the sensor to the controller or from the controller to the actuator, the ML-ETC will be unaware of the current state. Numerous strategies can be used to mitigate the effect of packet loss in ML-ETC systems. One possibility is to predict the lost information (state) and then use the predicted states to obtain the control input. Another approach to mitigating packet loss is through appropriate error correction design, which can be accomplished via forward error correction (FEC) or backward error correction (BEC) (a.k.a. Automatic Repeat reQuest, ARQ). However, BEC, which is based on re-transmissions, may lead to undesirable communication delays. For example, authors in [101] proposed Deep Reed-Solomon (DeepRS) coding, as a novel FEC algorithm which predicts packet loss using deep NNs to determine the amount of redundant packets. While there is research in the literature attempting to combine event-based control and ML in order to deal with packet loss, the majority of these works make assumptions that limit the applicability of the proposed methods. For instance, [102] extend event based state-feedback control to cope with communication delays and packet losses. The maximum tolerable communication delay bound is found, which guarantees the event-based state-feedback control is stable. The results are shown for a communication link with additional packet losses. However, the paper assumes that the dynamics of the plant are considered to be accurately known, the states are measurable, and the communication delay is bounded, which limits the applicability of this method. From a control perspective, robust controllers [103], [104] and MPC [105] are well-known to be robust against packet loss and delay in NCSs. However, the performance of these methods usually depends on the dynamic model accuracy and availability, which may strongly vary for real systems. Even if a dynamic model is available, the model of the system might change because of a dynamically changing environment, which can deteriorate the performance of a closed-loop system. This issue can be addressed with the help of ML.

**TABLE 3.** Existing ETC systems proposed in the literature - ML for Joint Learning and Optimization.

| Reference | System Type | Learning Approach | Communication policy | Control policy | Multi-agent | Centralized / Decentralized / Distributed | Control Mechanism | Experimental Validation |
|---|---|---|---|---|---|---|---|---|
| [52] | Affine Nonlinear | RL | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| [54] | Nonlinear (CT) | RL | ✗ | ✓ | ✗ | Cent. | $H_\infty$ Control | ✗ |
| [55] | Non-affine nonlinear | RL | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| [60] | Nonlinear | HDP | ✗ | ✓ | ✗ | Cent. | Optimal Control | ✗ |
| [61] | Nonlinear | RL | ✗ | ✓ | ✓ | Distributed | $H_\infty$ Control | ✗ |
| [53] | Nonlinear | DRL | ✗ | ✓ | ✗ | Cent. | Optimal Control | ✗ |
| [56] | Nonlinear Intercon-nected | RL | ✗ | ✓ | ✗ | Decent. | HJB | ✗ |
| [57] | Nonlinear (CT) | RL & RNN | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| [58] | Nonlinear (CT) | RL & RNN | ✗ | ✓ | ✗ | Cent. | HJB | ✗ |
| [59] | Affine Nonlinear (CT) | IRL & NN | ✗ | ✓ | ✗ | Cent. | Optimal Control | ✗ |

#### 2) NETWORK DELAYS

As previously stated, existing ML-based ETC methods have not considered communication delays in their problem formulation. There are three types of delays in networked control systems, sensor–controller delays, controller–actuator delays, and controller processing delays. In control theory, these delays cause phase shifts that limit the control bandwidth and affect closed-loop stability [106]. In order to overcome the pernicious effects of delay on closed-loop systems, ML can be used in various ways. For example, the average end-to-end delay in communication networks can be modeled accurately using NNs, resulting in improved control with sufficient knowledge of delay uncertainties [107]. ML can be used for learning models to cope with various uncertainties, such as delays and packet loss. For example, in [108], an MPC is designed for Unmanned Aerial Vehicles (UAVs) and a GP is applied to learn an unknown nonlinear model, whereas [109] also applied a GP-based approach to compensate for random communication delays, which is independent of the UAV's dynamic model. In fact, the pattern of network-induced effects is learned. While the literature presents ML algorithms for learning a system's model and making it robust against delay, delay is not taken into account in ML-ETC methods.

In [110], the authors considered delays only in the filtering phase and not in the control phase. An event-triggered $H_\infty$ filter is presented for the description of a Markovian jump system, which considers network-induced delay with the disturbance and an unknown nonlinear perturbation. A NN based on back propagation is used to dynamically adjust the communication threshold to reduce the burden of the network communication. A novel $H_\infty$ filtering error system model is used to cope with delays. The NN-based event-triggered scheme is compared with the traditional event-triggered scheme, and the advantage of adjusting the communication threshold dynamically to save more limited communication bandwidth is proven in simulation results. Future research could model delay as stochastic coupled leakage time-varying delays and develop a relaxed Lyapunov–Krasovskii functional for studying the delayed system [111]. Similar efforts have been made in [112] to develop a novel Lyapunov-Krasovskii functional and reveal all intrinsic relationships between time delay and sampling interval in the system. For example, memory event-triggerd $H_\infty$ output feedback control for neural networks with mixed delays including discrete and distributed delay problem is considered in [113]. The communication delay among neurons is modeled as a distributed delay term with a kernel representing the probability density and the integral term resulting from the proposed memory event-triggered system can be considered as a second distributed delay term. For designing an event-triggered $H_\infty$ controller, the Lyapunov–Krasovskii functional with the distributed delay kernel and a generalized integral inequality helped to form linear matrix inequalities.

### B. QUANTIZATION ERROR

Quantization errors occur in many digital systems during the process of converting signals from analog to digital as a result of the transmission of a plant's state information from a sensor to the controller/learning agent. All the ML-based ETC methods reviewed in this survey consider perfect quantization (i.e., errorless quantization), which limits their application to sensitive control systems. Moreover, quantization plays a significant role in event-driven control systems. As mentioned in ETC, an event is triggered by comparing the norm of state or the norm of the state error, which is a function of the plant's real state information. Both of these comparisons are considered based on non-quantized measurements that are assumed to be known with certainty in the papers we reviewed. This assumption might lead to system instability in practical scenarios [114]. Therefore, triggering conditions should be devised based on the available quantized state values. For example, a quantization level based ETC algorithm is presented under measurement uncertainties in [115]. Note that various types of quantizers are available in the literature, such as static, logarithmic, or dynamic quantizers [116]. The impact of these methods should be taken into account in future research studying ML-based event driven control.

### C. MOBILITY AWARE COMMUNICATION AND CONTROL

Mobility has a significant impact on real-time and sensitive control applications such as autonomous cars, robots, unmanned aerial vehicles, and vehicle platoons, where the objects are usually highly dynamic. For these applications, ETC can be a potential tool for ensuring real-time and reliable control actions while conserving wireless communication resources.

Current research on ETC either considers static agents or uses a predefined mobility model in which agents can only move in specific directions, complicating the system model's implementation. Because ML can be used to learn a system's behavior through experience, it can be combined with ETC while taking into account the impact of mobility on the system model. Consideration of the mobility model's availability limits the scope of application. To address this issue, a learning technique can be used, and a preliminary attempt has been made in [117], where a data-driven mobility model has been developed. Event detection analysis is conducted based on GPS location readings. This mobility model may be expanded further by allowing both the local and mobile hosts to learn about one another's position using ML algorithms and develop the mobility model based on their experience.

### D. SCALABILITY

Many of the learning algorithms in the works we reviewed here require significant computational resources, even when using event/self-triggered approaches. This will affect scalability in real-world applications, which is impacted by computational efficiency and reduced communication. The large amounts of data generated by many NCS implementations can be beneficial for learning in order to improve the quality of control via ML and large-scale optimization advances. This can be beneficial, in particular in the very networks that link control systems together. In fact, learn-and-adapt network management schemes result in decreased service delays, increased system resilience, and adaptability. However, in general, learning using large amounts of parameters and data can suffer from the curse of dimensionality, negatively affecting scalability. For further information on these issues, readers are referred to [118].

A few recent articles have attempted to address scalability issues in specific areas such as semi-definite programming by incorporating methods from ML, control, and robotics [119]. Saving communication bandwidth is also critical in large-scale projects, and as a result, more scalable ETC techniques should be developed. For example, [120] developed the distributed event-triggered consensus problem for linear multiagent networks. The proposed adaptive event-based protocol is fully distributed and scalable, as it is not reliant on any global information about the network graph or its scale [120]. Event-triggered consensus of linear multiagent systems on undirected graphs is developed with no need to know the precise Laplacian of the communication graph, which keeps the protocol scalable and distributed [121]. Scalability should be taken into account in ML-based ETC literature, and additional research is necessary.

### E. CLOUD/EDGE COMPUTING

In ML-based ETC systems, agents will often need to execute sophisticated ML and control algorithms. In particular, ML algorithms can be computationally extensive due to their complex nature. Moreover, for an ML-based ETC system, these computations need to be performed in real-time. Due to hardware constraints, it may not always be feasible to perform the ML tasks on an individual agent's hardware platform. One feasible solution is to offload parts of or all of the computational tasks of an ML-based ETC system to cloud or edge nodes [122], [123]. Those nodes can perform the computation and transmit the necessary information back to the agents to assist in the decision-making process. This also relates to the issue of scalability pointed out above. Research needs to be carried out on the aspects of using cloud/edge computing to assist computation in ML-based ETC systems. For example, issues such as the assignment of different tasks to different computing components and locations or communication policy between the agents and cloud/edge nodes need to be thoroughly investigated.

### F. JOINT LEARNING OF SYSTEM AND NETWORK MODELS

Managing the wireless network can play a pivotal role when the control actions are performed over wireless channels. In a real-world scenario, both the system model and the network model may change rapidly. The significance of learning the system model has been understood, and several attempts to use different ML techniques to learn the dynamics of the system model (as discussed earlier) have been presented. Still, the literature assumes that the network model is perfect and always available to the controller, which may not be the case in a practical scenario. The network model also needs to be learned in real time to achieve the best performance for ETC. A preliminary attempt has been made in [124], in which DRL is used to learn the communication network dynamics rather than the plant model, as shown in Fig. 5, where $x_k$ represents the state of the communication channel, $u_k$ represents the input from the learning agent to the channel, and $\delta_k$ represents the event-triggered threshold.

In large scale NCSs, the number of subsystems may be distributed over a wide area [125]. The effect of controller awareness on large-scale NCS scheduling decisions is discussed separately in [124]. This is the first publication on transmission scheduling for control signals over shared communication channels. The authors use DRL-based iterative resource allocation (DIRA), in which the DIRA uses system state information and performance feedback (control

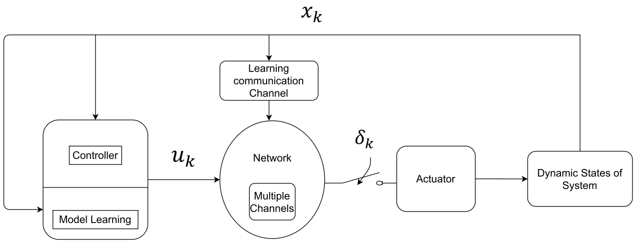

**FIGURE 5.** Joint Learning of System and Network Model [124].

cost evaluations) to achieve optimal control and optimized resource allocation. Further, DIRA can adapt to a given control policy that allows for such performance feedback. The proposed framework does not require a network model, and it implicitly learns the network parameters using DRL. This work can be further extended by using DIRA for state estimation and scheduling of the sensor-controller link along with a time-varying controller-actuator link.

### G. ENERGY EFFICIENT ML-BASED ETC

ETC considers a threshold to trigger control actions, resulting in an aperiodic system that is capable of saving computation and communication resources [71]. ETC can occasionally achieve higher performance with a lower sampling frequency than time-driven control [60]. By combining ETC and ML, it becomes more robust to disturbances and uncertainties and can potentially be more energy efficient.

In [16], event-triggered MPC is combined with SL to make it more adaptive to uncertainties and robust to state estimation errors. While standard MPC and event-triggered MPC do not account for uncertainties, which can result in tracking errors, learning-based event-triggered MPC can achieve accurate tracking results comparable to standard MPC. The simulation results show that triggering instances can be reduced when using learning-based event-triggered MPC versus event-triggered MPC, highlighting the critical role that ML can play for energy savings. In [15], model learning is used in an ETL framework when the existing model is not accurate. The benefits of learning system dynamics are demonstrated through a numerical study. After learning, better tracking performance and control signals are observed in comparison to the state prior to learning. Additionally, learning leads to an increase in intercommunication time, resulting in decreased communication. Therefore, ML-based ETC can be more energy efficient while maintaining accuracy. While the literature suggests that ETC can reduce communication and lead to energy savings, we believe that applying ML to ETC can result in even greater control accuracy and communication reductions.

### H. SELF-TRIGGERED CONTROL

As ETC is reactive in nature, it continuously monitors the system's states and triggers when the system deviates too far from a predefined threshold [64]. ETC requires extra hardware resources for continuous state monitoring, which increases the cost. This is a significant cost in large-scale settings. Another issue with ETC is that it requires full state information at all times, necessitating that the system is more robust throughout the execution of time control actions [64]. To overcome this, STC can be a better aperiodic triggering mechanism because it is proactive in nature. It calculates the next triggering time at the current time instance, and in-between these instances it remains idle [8], [64], [126]. It does not require full state information all the time, but rather at triggering instances. This property of STC makes it more suitable for combining with ML algorithms in order to avoid excessive learning of the system model. As discussed earlier in this article, ML is used to learn the dynamic model of the system or to optimize performance, and its combination with ETC also utilizes scarce resources, as ETC requires state information all the time. Therefore, replacing ETC with STC in combination with ML can be extremely beneficial, as it reduces resource waste.

A preliminary attempt has been made in [23], where the system dynamics are too difficult to obtain due to the system's complexity, necessitating optimal learning via STC-based learning. STC does not require the learning agent to constantly learn the system's model. This STC learning technique can be extended to learn the dynamics required for optimal system control. STC can also be used to combine various ML techniques mentioned earlier to achieve optimal utilization of resources.

### I. SECURITY ISSUES

In a multiagent system, or NCS, where different nodes communicate with each other, security can be a critical issue. Specifically, security and privacy will be a fundamental challenge to the adoption of large-scale NCSs. Recently, researchers have combined ML and event-triggered communication/control to study fault detection and fault-tolerant control. ML techniques are used to improve the recognition, detection, diagnosis, and prediction accuracy of fault features [127]. The most effective methods for feature classification are deep NNs, recurrent NNs, and conventional NNs. Additionally, an event-driven approach is used to trigger fault detection and localization in order to improve transmission efficiency [128]. From a fault-tolerant control perspective, the authors of [129] used ETC laws to effectively reduce the network transmission load from the controller to the actuators, and they used neural adaptive laws to compensate for unknown actuator faults online. Similar research is being conducted in [130] to address the attitude control problem for spacecraft against actuator faults. ML can also be used to design better control and communication mechanisms that can prevent data injection attacks. A major concern for the traditional training process is privacy, which the nodes may not want to compromise on by sharing training samples. Federated learning, which emerged in recent years to address the privacy and communication overhead issues associated with the training of ML models, has attracted extensive research interest for enhanced wireless networks [131], [132]. Federated learning may play a vital role in the design of future NCS systems. It is worth noting that using ETC for security issues requires smart ETC algorithms. For example, in [133], a novel event-triggered scheme has been developed, which is smarter and more flexible with features of avoiding chaotic triggering, increasing triggering exponentially, linear compensation, and linear triggering.

## VII. CONCLUSION

This article provides a survey of current ML techniques combined with ETC. We begin our discussion by highlighting the challenge of scarce bandwidth resources available to NCSs and how event-triggered communication can address this challenge. Furthermore, we reviewed various articles that discuss the limitations of implementing ETC for practical NCSs and potential solutions. The majority of the literature indicates that the availability of the model available to the controller is one of the most significant challenges in implementing ETC for NCSs. By learning the entire model or portions of a model, ML is a key technique to address the problems of changing dynamics in practical NCSs. Based on the application of ML in the ETC literature, we classify articles into three groups: dynamic model learning, ML for optimal performance, and joint model learning and optimization. While the literature discusses a variety of ML techniques, ML-based ETC appears to rely primarily on SL, NN, and (deep) RL approaches. Although ML-based ETC approaches have demonstrated promising results in addressing the various challenges outlined here, there is still scope to enhance existing ML approaches further or develop new solutions to address existing challenges. Among them, we highlighted how ML can be used to address issues such as learning the network as well as the system model or how the movement of agents affects the model. We concluded by proposing possible solutions to several of these open issues.

## REFERENCES

[1] C. Sun, G. Cembrano, V. Puig, and J. Meseguer, "Cyber-physical systems for real-time management in the urban water cycle," in *Proc. Int. Workshop Cyber-Phys. Syst. Smart Water Netw. (CySWater)*, Apr. 2018, pp. 5–8.

[2] D. G. Broo, U. Boman, and M. Törngren, "Cyber-physical systems research and education in 2030: Scenarios and strategies," *J. Ind. Inf. Integr.*, vol. 21, Mar. 2021, Art. no. 100192.

[3] H. Khujamatov, E. Reypnazarov, D. Khasanov, and N. Akhmedov, "IoT, IIoT, and cyber-physical systems integration," in *Emergence of Cyber Physical System and IoT in Smart Automation and Robotics*. Cham, Switzerland: Springer, 2021, pp. 31–50.

[4] D. G. S. Pivoto, L. F. F. de Almeida, R. da Rosa Righi, J. J. P. C. Rodrigues, A. B. Lugli, and A. M. Alberti, "Cyber-physical systems architectures for industrial Internet of Things applications in industry 4.0: A literature review," *J. Manuf. Syst.*, vol. 58, pp. 176–192, Jan. 2021.

[5] R. A. Gupta and M.-Y. Chow, "Networked control system: Overview and research trends," *IEEE Trans. Ind. Electron.*, vol. 57, no. 7, pp. 2527–2535, Jul. 2010.

[6] M. Mazo and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 435–440.

[7] M. Guinaldo, J. Sánchez, R. Dormido, and S. Dormido, "Distributed control for large-scale systems with adaptive event-triggering," *J. Franklin Inst.*, vol. 353, no. 3, pp. 735–756, Feb. 2016.

[8] U. Tiberi and K. H. Johansson, "A simple self-triggered sampler for perturbed nonlinear systems," *Nonlinear Anal., Hybrid Syst.*, vol. 10, pp. 126–140, Nov. 2013.

[9] C. Peng and F. Li, "A survey on recent advances in event-triggered communication and control," *Inf. Sci.*, vols. 457–458, pp. 113–125, Aug. 2018.

[10] S. Liu, W. Luo, and L. Wu, "Co-design of distributed model-based control and event-triggering scheme for load frequency regulation in smart grids," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 9, pp. 3311–3319, Sep. 2020.

[11] F. Solowjow, D. Baumann, J. Garcke, and S. Trimpe, "Event-triggered learning for resource-efficient networked control," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 6506–6512.

[12] D. Baumann, J.-J. Zhu, G. Martius, and S. Trimpe, "Deep reinforcement learning for event-triggered control," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 943–950.

[13] S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. Lewis, and W. Dixon, "An actor–critic–identifier architecture for adaptive approximate optimal control," in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Hoboken, NJ, USA: Wiley, 2012, pp. 258–280.

[14] F. Solowjow and S. Trimpe, "Event-triggered learning," *Automatica*, vol. 117, Jul. 2020, Art. no. 109009.

[15] F. Solowjow, K. H. Johansson, and S. Trimpe, "Event-triggered pulse control with model learning (if necessary)," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019 pp. 792–797.

[16] J. Yoo and K. H. Johansson, "Event-triggered model predictive control with a statistical learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 4, pp. 2571–2581, Apr. 2021.

[17] A. Sahoo, H. Xu, and S. Jagannathan, "Neural network-based event-triggered state feedback control of nonlinear continuous-time systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 497–509, Mar. 2015.

[18] H. Schlüter, F. Solowjow, and S. Trimpe, "Event-triggered learning for linear quadratic control," 2019, *arXiv:1910.07732*.

[19] Y. Li and G.-H. Yang, "Adaptive neural control of pure-feedback nonlinear systems with event-triggered communications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6242–6251, Dec. 2018.

[20] J. Li, H. Liu, Z. Zhang, X. Li, and X. Yang, "Event-triggered adaptive NN tracking control with dynamic gain for a class of unknown nonlinear systems," *Neurocomputing*, vol. 467, pp. 292–299, Jan. 2022.

[21] Z. Chen, B. Niu, X. Zhao, L. Zhang, and N. Xu, "Model-based adaptive event-triggered control of nonlinear continuous-time systems," *Appl. Math. Comput.*, vol. 408, Nov. 2021, Art. no. 126330.

[22] L. Liu, X. Li, Y.-J. Liu, and S. Tong, "Neural network based adaptive event trigger control for a class of electromagnetic suspension systems," *Control Eng. Pract.*, vol. 106, Jan. 2021, Art. no. 104675.

[23] K. Hashimoto, Y. Yoshimura, and T. Ushio, "Learning self-triggered controllers with Gaussian processes," 2019, *arXiv:1909.00178*.

[24] C. Gao, C. Zhang, X. Liu, H. Wang, and L. Wang, "Event-triggering based adaptive neural tracking control for a class of pure-feedback systems with finite-time prescribed performance," *Neurocomputing*, vol. 382, pp. 221–232, Mar. 2020.

[25] H. Liang, G. Liu, H. Zhang, and T. Huang, "Neural-network-based event-triggered adaptive control of nonaffine nonlinear multiagent systems with dynamic uncertainties," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2239–2250, May 2020.

[26] F. Wang and L. Long, "Event-triggered adaptive NN control for MIMO switched nonlinear systems with non-ISpS unmodeled dynamics," *J. Franklin Inst.*, vol. 359, no. 4, pp. 1457–1485, Mar. 2022.

[27] S. Yu, J. Lu, G. Zhu, and S. Yang, "Event-triggered finite-time tracking control of underactuated MSVs based on neural network disturbance observer," *Ocean Eng.*, vol. 253, Jun. 2022, Art. no. 111169.

[28] X. Guo, W. Yan, and R. Cui, "Event-triggered reinforcement learning-based adaptive tracking control for completely unknown continuous-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3231–3242, Jul. 2020.

[29] J. George and P. Gurram, "Distributed deep learning with event-triggered communication," 2019, *arXiv:1909.05020*.

[30] Q. Zhang, D. Zhao, and Y. Zhu, "Event-triggered $H_\infty$ control for continuous-time nonlinear system via concurrent learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1071–1081, Jul. 2017.

[31] Q. Zhang and D. Zhao, "Event-triggered integral reinforcement learning for nonlinear continuous-time systems," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–6.

[32] D. Wang, H. He, X. Zhong, and D. Liu, "Event-driven nonlinear discounted optimal regulation involving a power system application," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8177–8186, Oct. 2017.

[33] L. Cui, W. Qu, L. Wang, Y. Luo, and Z. Wang, "Event-triggered $H_\infty$ tracking control of nonlinear systems via reinforcement learning method," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[34] X. Yang and H. He, "Event-triggered robust stabilization of nonlinear input-constrained systems using single network adaptive critic designs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 9, pp. 3145–3157, Sep. 2020.

[35] X. Yang and H. He, "Event-driven $H_\infty$-constrained control using adaptive critic learning," *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4860–4872, Oct. 2021.

[36] C. Qin, H. Zhu, J. Wang, Q. Xiao, and D. Zhang, "Event-triggered safe control for the zero-sum game of nonlinear safety-critical systems with input saturation," *IEEE Access*, vol. 10, pp. 40324–40337, 2022.

[37] S. Xue, B. Luo, D. Liu, and Y. Gao, "Event-triggered integral reinforcement learning for nonzero-sum games with asymmetric input saturation," *Neural Netw.*, vol. 152, pp. 212–223, Aug. 2022.

[38] H. Su, H. Zhang, S. Sun, and Y. Cai, "Integral reinforcement learning-based online adaptive event-triggered control for non-zero-sum games of partially unknown nonlinear systems," *Neurocomputing*, vol. 377, pp. 243–255, Feb. 2020.

[39] D. Wang, H. He, and D. Liu, "Improving the critic learning for event-based nonlinear $H_\infty$ control design," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3417–3428, Oct. 2017.

[40] S. Zhao, J. Wang, H. Wang, and H. Xu, "Goal representation adaptive critic design for discrete-time uncertain systems subjected to input constraints: The event-triggered case," *Neurocomputing*, vol. 492, pp. 676–688, Jul. 2022.

[41] D. Wang and D. Liu, "Learning and guaranteed cost control with event-based adaptive critic implementation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6004–6014, Dec. 2018.

[42] X. Yang and H. He, "Adaptive critic learning and experience replay for decentralized event-triggered control of nonlinear interconnected systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4043–4055, Nov. 2020.

[43] Y. Huo, D. Wang, and J. Qiao, "Adaptive critic optimization to decentralized event-triggered control of continuous-time nonlinear interconnected systems," *Optim. Control Appl. Methods*, vol. 43, no. 1, pp. 198–212, Jan. 2022.

[44] K. Zhang, R. Su, H. Zhang, and Y. Tian, "Adaptive resilient event-triggered control design of autonomous vehicles with an iterative single critic learning framework," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5502–5511, Dec. 2021.

[45] D. Yang, T. Li, H. Zhang, and X. Xie, "Event-trigger-based robust control for nonlinear constrained-input systems using reinforcement learning method," *Neurocomputing*, vol. 340, no. 12, pp. 158–170, May 2019.

[46] K. G. Vamvoudakis and H. Ferraz, "Model-free event-triggered control algorithm for continuous-time linear systems with optimal performance," *Automatica*, vol. 87, pp. 412–420, Jan. 2018.

[47] X. Yang and H. He, "Adaptive critic designs for event-triggered robust control of nonlinear systems with unknown dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2255–2267, Jun. 2019.

[48] W. Bai, T. Li, Y. Long, and C. L. P. Chen, "Event-triggered multigradient recursive reinforcement learning tracking control for multiagent systems," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 16, 2021, doi: 10.1109/TNNLS.2021.3094901.

[49] D. Wang, M. Ha, and J. Qiao, "Self-learning optimal regulation for discrete-time nonlinear systems under event-driven formulation," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 1272–1279, Mar. 2020.

[50] B. Sun and E.-J. van Kampen, "Event-triggered constrained control using explainable global dual heuristic programming for nonlinear discrete-time systems," *Neurocomputing*, vol. 468, pp. 452–463, Jan. 2022.

[51] J. Lu, Q. Wei, T. Zhou, Z. Wang, and F.-Y. Wang, "Event-triggered near-optimal control for unknown discrete-time nonlinear systems using parallel control," *IEEE Trans. Cybern.*, early access, May 6, 2022, doi: 10.1109/TCYB.2022.3164977.

[52] X. Yang, H. He, and D. Liu, "Event-triggered optimal neuro-controller design with reinforcement learning for unknown nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 9, pp. 1866–1878, Sep. 2019.

[53] Y. Ran, X. Zhou, H. Hu, and Y. Wen, "Optimizing data centre energy efficiency via event-driven deep reinforcement learning," *IEEE Trans. Services Comput.*, early access, Mar. 7, 2022, doi: 10.1109/TSC.2022.3157145.

[54] D. Wang, C. Mu, D. Liu, and H. Ma, "On mixed data and event driven design for adaptive-critic-based nonlinear $H_\infty$ control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 993–1005, Apr. 2018.

[55] S. Zhang, B. Zhao, and Y. Zhang, "Event-triggered control for input constrained non-affine nonlinear systems based on neuro-dynamic programming," *Neurocomputing*, vol. 440, pp. 175–184, Jun. 2021.

[56] X. Huo, H. R. Karimi, X. Zhao, B. Wang, and G. Zong, "Adaptive-critic design for decentralized event-triggered control of constrained nonlinear interconnected systems within an identifier-critic framework," *IEEE Trans. Cybern.*, early access, Jan. 5, 2021, doi: 10.1109/TCYB.2020.3037321.

[57] N. Xu, B. Niu, H. Wang, X. Huo, and X. Zhao, "Single-network ADP for solving optimal event-triggered tracking control problem of completely unknown nonlinear systems," *Int. J. Intell. Syst.*, vol. 36, no. 9, pp. 4795–4815, Sep. 2021.

[58] S. Liu, B. Niu, G. Zong, X. Zhao, and N. Xu, "Data-driven-based event-triggered optimal control of unknown nonlinear systems with input constraints," *Nonlinear Dyn.*, vol. 2022, pp. 1–19, May 2022.

[59] S. Xue, B. Luo, D. Liu, and Y. Gao, "Event-triggered ADP for tracking control of partially unknown constrained uncertain systems," *IEEE Trans. Cybern.*, early access, Mar. 4, 2021, doi: 10.1109/TCYB.2021.3054626.

[60] T. Li, D. Yang, X. Xie, and H. Zhang, "Event-triggered control of nonlinear discrete-time system with unknown dynamics based on HDP($\lambda$)," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6046–6058, Jul. 2022.

[61] L. N. Tan, "Event-triggered distributed $H_\infty$ constrained control of physically interconnected large-scale partially unknown strict-feedback systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 4, pp. 2444–2456, Apr. 2021.

[62] Z. Ijaz, M. Tahir, and S. Arshad, "Self-triggered control plane for cognitive radio networks," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5.

[63] J. Wang, H. Zhang, K. Ma, Z. Liu, and C. L. P. Chen, "Neural adaptive self-triggered control for uncertain nonlinear systems with input hysteresis," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 10, 2021, doi: 10.1109/TNNLS.2021.3072784.

[64] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 3270–3285.

[65] G. Ran, C. Li, H.-K. Lam, D. Li, and C. Han, "Event-based dissipative control of interval type-2 fuzzy Markov jump systems under sensor saturation and actuator nonlinearity," *IEEE Trans. Fuzzy Syst.*, 2020.

[66] S. Wang, Z. Shu, and T. Chen, "Cooperative output regulation with mixed time- and event-triggered observers," 2021, *arXiv:2105.02200*.

[67] E. Tian, Z. Wang, L. Zou, and D. Yue, "Probabilistic-constrained filtering for a class of nonlinear systems with improved static event-triggered communication," *Int. J. Robust Nonlinear Control*, vol. 29, no. 5, pp. 1484–1498, 2019.

[68] X. Ge, Q.-L. Han, L. Ding, Y.-L. Wang, and X.-M. Zhang, "Dynamic event-triggered distributed coordination control and its applications: A survey of trends and techniques," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 9, pp. 3112–3125, Sep. 2020.

[69] Z. Gu, E. Tian, and J. Liu, "Adaptive event-triggered control of a class of nonlinear networked systems," *J. Franklin Inst.*, vol. 354, no. 9, pp. 3854–3871, 2017.

[70] G. Zhao, C. Hua, and X. Guan, "A hybrid event-triggered approach to consensus of multiagent systems with disturbances," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 3, pp. 1259–1271, Sep. 2020.

[71] X. Ge, Q.-L. Han, X.-M. Zhang, and D. Ding, "Dynamic event-triggered control and estimation: A survey," *Int. J. Autom. Comput.*, vol. 18, no. 6, pp. 857–886, Dec. 2021.

[72] Z. Chen, Q.-L. Han, Y. Yan, and Z.-G. Wu, "How often should one update control and estimation: Review of networked triggering techniques," *Sci. China Inf. Sci.*, vol. 63, no. 5, pp. 1–18, May 2020.

[73] G. Ran, C. Li, R. Sakthivel, C. Han, B. Wang, and J. Liu, "Adaptive event-triggered asynchronous control for interval type-2 fuzzy Markov jump systems with cyberattacks," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 1, pp. 88–99, Mar. 2022.

[74] T. M. Mitchell, *Machine Learning*, vol. 1, no. 9. New York, NY, USA: McGraw-Hill, 1997.

[75] V. Koltchinskii *et al.*, "Statistical learning control of uncertain systems: It is better than it seems," Dept. ECE, Univ. New Mexico, Albuquerque, NM, USA, Tech. Rep., 2000.

[76] M. Papadimitrakis and A. Alexandridis, "Active vehicle suspension control using road preview model predictive control and radial basis function networks," *Appl. Soft Comput.*, vol. 120, May 2022, Art. no. 108646.

[77] S. Golnaraghi, O. Moselhi, S. Alkass, and Z. Zangenehmadar, "Predicting construction labor productivity using lower upper decomposition radial base function neural network," *Eng. Rep.*, vol. 2, no. 2, Feb. 2020, Art. no. e12107.

[78] B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: A comprehensive survey," *Artif. Intell. Rev.*, vol. 2021, pp. 1–46, Apr. 2021.

[79] J. Peters, S. Vijayakumar, and S. Schaal, "Natural actor-critic," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, 2005, pp. 280–291.

[80] V. Nasir and F. Sassani, "A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges," *Int. J. Adv. Manuf. Technol.*, vol. 115, no. 9, pp. 2683–2709, May 2021.

[81] G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, L. L. García, I. Heredia, P. Malík, and L. Hluchý, "Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey," *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 77–124, Jun. 2019.

[82] N. M. Ashraf, R. R. Mostafa, R. H. Sakr, and M. Z. Rashad, "A state-of-the-art review of deep reinforcement learning techniques for real-time strategy games," in *Applications of Artificial Intelligence in Business, Education and Healthcare*. Springer, 2021, pp. 285–307.

[83] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: An engineering perspective," *Int. J. Adv. Manuf. Technol.*, vol. 117, nos. 5–6, pp. 1327–1349, Nov. 2021.

[84] W. Dabney, G. Ostrovski, and A. Barreto, "Temporally-extended $\varepsilon$-greedy exploration," 2020, *arXiv:2006.01782*.

[85] T. Li, W. Bai, Q. Liu, Y. Long, and C. L. P. Chen, "Distributed fault-tolerant containment control protocols for the discrete-time multiagent systems via reinforcement learning method," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 1, 2021, doi: 10.1109/TNNLS.2021.3121403.

[86] C. Mu and Y. Zhang, "Learning-based robust tracking control of quadrotor with time-varying and coupling uncertainties," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 1, pp. 259–273, Jan. 2020.

[87] Z. Chen, W. Xue, N. Li, B. Lian, and F. L. Lewis, "A novel $Z$-function-based completely model-free reinforcement learning method to finite-horizon zero-sum game of nonlinear system," *Nonlinear Dyn.*, vol. 107, pp. 2563–2582, Jan. 2022.

[88] H. Su, H. Zhang, Y. Luo, and Q. Sun, "Event-based integral reinforcement learning algorithm for non-zero-sum games of partially unknown nonlinear systems," in *Proc. IEEE 10th Data Driven Control Learn. Syst. Conf. (DDCLS)*, May 2021, pp. 287–292.

[89] G. Wen, L. Xu, and B. Li, "Optimized backstepping tracking control using reinforcement learning for a class of stochastic nonlinear strict-feedback systems," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 26, 2021, doi: 10.1109/TNNLS.2021.3105176.

[90] S. Zhang, B. Zhao, D. Liu, and Y. Zhang, "Observer-based event-triggered control for zero-sum games of input constrained multi-player nonlinear systems," *Neural Netw.*, vol. 144, pp. 101–112, Dec. 2021.

[91] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 12, pp. 1775–1787, Dec. 2019.

[92] S. Yasini, A. Karimpour, M.-B. N. Sistani, and H. Modares, "Online concurrent reinforcement learning algorithm to solve two-player zero-sum games for partially unknown nonlinear continuous-time systems," *Int. J. Adapt. Control Signal Process.*, vol. 29, no. 4, pp. 473–493, Apr. 2015.

[93] K. G. Vamvoudakis and F. L. Lewis, "Online solution of nonlinear two-player zero-sum games using synchronous policy iteration," *Int. J. Robust Nonlinear Control*, vol. 22, no. 13, pp. 1460–1483, 2012.

[94] C. Peng and J. Ma, "Online integral reinforcement learning control for an uncertain highly flexible aircraft using state and output feedback," *Aerosp. Sci. Technol.*, vol. 109, Feb. 2021, Art. no. 106442.

[95] M. Granzotto, R. Postoyan, L. Busoniu, D. Nesic, and J. Daafouz, "Finite-horizon discounted optimal control: Stability and performance," *IEEE Trans. Autom. Control*, vol. 66, no. 2, pp. 550–565, Feb. 2021.

[96] T. Başar and P. Bernhard, *$H^{\infty}$ Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Boston, MA, USA: Springer, 2008.

[97] J. Na, S. Wang, Y.-J. Liu, Y. Huang, and X. Ren, "Finite-time convergence adaptive neural network control for nonlinear servo systems," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2568–2579, Jun. 2020.

[98] Y. Ran, H. Hu, X. Zhou, and Y. Wen, "DeepEE: Joint optimization of job scheduling and cooling control for data center energy efficiency using deep reinforcement learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 645–655.

[99] B. Sun, P. B. Luh, Q. S. Jia, and B. Yan, "Event-based optimization within the Lagrangian relaxation framework for energy savings in HVAC systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 4, pp. 1396–1406, Oct. 2015.

[100] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[101] S. Cheng, H. Hu, X. Zhang, and Z. Guo, "DeepRS: Deep-learning based network-adaptive FEC for real-time video communications," 2020, *arXiv:2001.07852*.

[102] D. Lehmann and J. Lunze, "Event-based control with communication delays and packet losses," *Int. J. Control*, vol. 85, no. 5, pp. 563–577, Feb. 2012.

[103] B. Xue, H. Yu, and M. Wang, "Robust $H_{\infty}$ output feedback control of networked control systems with discrete distributed delays subject to packet dropout and quantization," *IEEE Access*, vol. 7, pp. 30313–30320, 2019.

[104] Y. Shi, J. Wang, X. Fang, Y. Huang, and S. Gu, "Robust mixed $H_2/H_{\infty}$ control for an uncertain wireless sensor network systems with time delay and packet loss," *Int. J. Control, Automat. Syst.*, vol. 19, pp. 88–100, Jan. 2021.

[105] A. Ibrahim, D. Goswami, H. Li, and T. Basten, "Delay-aware multi-layer multi-rate model predictive control for vehicle platooning under message-rate congestion control," *IEEE Access*, vol. 10, pp. 44583–44607, 2022.

[106] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 978–1013, 2nd Quart., 2018.

[107] A. Mestres, E. Alarcón, Y. Ji, and A. Cabellos-Aparicio, "Understanding the modeling of computer network delays using neural networks," in *Proc. Workshop Big Data Anal. Mach. Learn. Data Commun. Netw.*, Aug. 2018, pp. 46–52.

[108] D. Jang, J. Yoo, C. Y. Son, H. J. Kim, and K. H. Johansson, "Networked operation of a UAV using Gaussian process-based delay compensation and model predictive control," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9216–9222.

[109] J. Yoo and K. H. Johansson, "Learning communication delay patterns for remotely controlled UAV networks," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13216–13221, 2017.

[110] Y. Wang, T. Zhang, S. Chen, and J. Ren, "Network-based $H_{\infty}$ filtering for descriptor Markovian jump systems with a novel neural network event-triggered scheme," *Neural Process. Lett.*, vol. 53, no. 1, pp. 757–775, 2021.

[111] X. Cai, J. Wang, S. Zhong, K. Shi, and Y. Tang, "Fuzzy quantized sampled-data control for extended dissipative analysis of T–S fuzzy system and its application to WPGSs," *J. Franklin Inst.*, vol. 358, no. 2, pp. 1350–1375, Jan. 2021.

[112] X. Cai, K. B. Shi, S. M. Zhong, J. Wang, and Y. Q. Tang, "Dissipative analysis for high speed train systems via looped-functional and relaxed condition methods," *Appl. Math. Model.*, vol. 96, pp. 570–583, Aug. 2021.

[113] S. Yan, Z. Gu, and S. K. Nguang, "Memory-event-triggered $H_{\infty}$ output control of neural networks with mixed delays," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 4, 2021, doi: 10.1109/TNNLS.2021.3083898.

[114] E. Garcia and P. J. Antsaklis, "Model-based event-triggered control for systems with quantization and time-varying network delays," *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 422–434, Feb. 2013.

[115] T. Zhou, G. Yue, and B. Niu, "Quantization level based event-triggered control with measurement uncertainties," *Inf. Sci.*, vol. 588, pp. 442–456, Apr. 2022.

[116] E. Mastani and M. Rahmani, "Dynamic output feedback control for networked systems subject to communication delays, packet dropouts, and quantization," *J. Franklin Inst.*, vol. 358, no. 8, pp. 4303–4325, May 2021.

[117] P. Kolios, G. Ellinas, and C. Panayiotou, "Tracking trip changes with event triggering," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 529–534.

[118] T. Chen, S. Barbarossa, X. Wang, G. Giannakis, and Z.-L. Zhang, "Learning and management for Internet of Things: Accounting for adaptivity and scalability," *Proc. IEEE*, vol. 107, no. 4, pp. 778–796, Apr. 2019.

[119] A. Majumdar, G. Hall, and A. A. Ahmadi, "Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 3, no. 1, pp. 331–360, May 2020.

[120] B. Cheng and Z. Li, "Fully distributed event-triggered protocols for linear multiagent networks," *IEEE Trans. Autom. Control*, vol. 64, no. 4, pp. 1655–1662, Apr. 2019.

[121] X. Li, Y. Tang, and H. R. Karimi, "Consensus of multi-agent systems via fully distributed event-triggered control," *Automatica*, vol. 116, Jun. 2020, Art. no. 108898.

[122] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, May 2016.

[123] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[124] A. Redder, A. Ramaswamy, and D. E. Quevedo, "Deep reinforcement learning for scheduling in large-scale networked control systems," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 333–338, 2019.

[125] D. Zhang, S. K. Nguang, and L. Yu, "Distributed control of large-scale networked control systems with communication constraints and topology switching," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1746–1757, Jul. 2017.

[126] U. Tiberi and K. H. Johansson, "A simple self-triggered sampler for nonlinear systems," *IFAC Proc. Volumes*, vol. 45, no. 9, pp. 76–81, 2012.

[127] G. Cirrincione, R. R. Kumar, A. Mohammadi, S. H. Kia, P. Barbiero, and J. Ferretti, "Shallow versus deep neural networks in gear fault diagnosis," *IEEE Trans. Energy Convers.*, vol. 35, no. 3, pp. 1338–1347, Sep. 2020.

[128] H. Habbouche, Y. Amirat, T. Benkedjouh, and M. Benbouzid, "Bearing fault event-triggered diagnosis using a variational mode decomposition-based machine learning approach," *IEEE Trans. Energy Convers.*, vol. 37, no. 1, pp. 466–474, Mar. 2022.

[129] G. Zhang, S. Chu, X. Jin, and W. Zhang, "Composite neural learning fault-tolerant control for underactuated vehicles with event-triggered input," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2327–2338, May 2021.

[130] C. Zhang, J. Wang, D. Zhang, and X. Shao, "Learning observer based and event-triggered control to spacecraft against actuator faults," *Aerosp. Sci. Technol.*, vol. 78, pp. 522–530, Jul. 2018.

[131] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.

[132] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[133] C. Zhao, X. Liu, S. Zhong, K. Shi, D. Liao, and Q. Zhong, "Secure consensus of multi-agent systems with redundant signal and communication interference via distributed dynamic event-triggered control," *ISA Trans.*, vol. 112, pp. 89–98, Jun. 2021.

**LEILA SEDGHI** was born in Iran, in 1990. She received the B.S. and M.S. degrees in control engineering from Qazvin Islamic Azad University, Iran, in 2012 and 2016, respectively. She is currently pursuing the Ph.D. degree in computer science with University College Cork, Ireland.

She has been lecturing at the Computer Science Department, Munster Technological University, since 2019. Her research interests include control engineering, event-triggered control, machine learning, and deep learning. She has applied various control algorithms to different applications, such as robotics, power systems, and microgrids. Her research is funded through the Confirm Centre for Smart Manufacturing for Ph.D. degree.

**ZOHAIB IJAZ** received the bachelor's degree in electronic engineering from the Islamia University of Bahawalpur, Pakistan, in 2014, and the master's degree in electrical engineering with a specialization in control systems from the University of Engineering and Technology, Lahore, Pakistan, in 2017. He is currently pursuing the Ph.D. degree with the School of Computer Science, University College Cork, Ireland, with SFI funding from the Confirm Centre for Smart Manufacturing. His research interests include event-based communication for cyber-physical systems and consensus control of multi-agent systems, such as synchronous motors and power systems.

**MD. NOOR-A-RAHIM** received the Ph.D. degree from the Institute for Telecommunications Research, University of South Australia, Australia, in 2015. He was a Postdoctoral Research Fellow with the Centre for Infocomm Technology (INFINITUS), Nanyang Technological University (NTU), Singapore. He is currently a Research Fellow with the School of Computer Science and IT, University College Cork, Ireland. His research interests include control over wireless networks, intelligent transportation systems, machine learning, signal processing, and DNA-based data storage. He was a recipient of the Michael Miller Medal from the Institute for Telecommunications Research (ITR), University of South Australia, for the most outstanding Ph.D. thesis, in 2015.

**KRITCHAI WITHEEPHANICH** received the Ph.D. degree in electronic and computer engineering from the University of Limerick, Ireland, in 2012. He is currently an Assistant Lecturer with the Department of Electrical and Electronic Engineering, Munster Technological University, and an Associate Investigator with the SFI Confirm Centre for Smart Manufacturing. His research interest includes control of cyber-physical systems.

**DIRK PESCH** (Senior Member, IEEE) received the Dipl.-Ing. degree from RWTH Aachen University, Germany, and the Ph.D. degree from the University of Strathclyde, Glasgow, U.K. Previously, he was the Head of the Nimbus Research Centre, Cork Institute of Technology (now Munster Technological University). He is currently a Professor with the School of Computer Science and Information Technology, University College Cork. He is also the Principal Investigator of the National Science Foundation Ireland funded Collaborative Centres CONNECT (Future Networks) and CONFIRM (Smart Manufacturing) and the Director of the SFI Centre for Research Training in Advanced Networks for Sustainable Societies. He has over 25 years research and development experience in both industry and academia and has coauthored over 200 scientific articles and book chapters. He has been involved in a number of EU funded research projects on smart and energy efficient buildings and urban neighbourhoods, including as a Coordinator. His research interests include architecture, design, algorithms, performance evaluation of low power, dense and vehicular wireless/mobile networks and services for the Internet of Things and cyber-physical systems applications in building management, smart connected communities, health and wellbeing, and smart manufacturing. He contributes to international conference organization and is a member of the editorial board of a number of journals.