A Data-enabled Dual Learning based Online Receding Horizon Safe-Critical Control for Nonlinear Systems under Uncertainty

Shawon Dey and Hao Xu

NAECON 2024 - IEEE National Aerospace and Electronics Conference | 979-8-3503-6762-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/NAECON61878.2024.10670666

***Abstract*— In this paper, a real-time reliable receding horizon control (RHC) with a guaranteed safe adaptation mechanism is developed for uncertain complex nonlinear systems. Ensuring receding horizon optimality and safety, particularly in the presence of uncertain nonlinear system dynamics, poses a significant challenge in both control and learning societies. To tackle this challenge, a novel safe-critical RHC framework has been developed to enhance classical RHC with the capability of prioritizing system safety by timely recognizing and adapting environmental uncertainties. Specifically, the developed framework utilizes a novel dual-learning approach with slow learning to recognize environmental uncertainties and further refine RHC along with a situation-aware physics-informed neural network (SA-PINN), and fast learning to ensure system safety by using a safe-critical control with fast learned adaptive control barrier (FA-CBF) function. Therefore, slow learning in the developed dual-learning approach can provide optimal RHC albeit with longer computation time, while the fast learning component provides safe control effectively adapting to the uncertain environment in real-time.**

I. Introduction

In real-time control scenarios for complex nonlinear systems, e.g. satellite control [1], autonomous transportation [2], unmanned aerial vehicle [3] etc., it is crucial to ensure system safety while pursuing performance optimality. Existing research has extensively explored approaches like Receding Horizon Control (RHC) [4] to strengthen practical control cost-effectiveness, and Control Barrier Function (CBF) [5] to guarantee safety. However, these methods face challenges in real-time applications due to their computational complexity and reliance on a fully known dynamics model. The existing approaches [6], [7] are impractical for rapid safe adaptation with performance optimality especially when the system dynamics are not fully known. To address this challenge as well as achieve an efficient balance between cost performance and system safety, a novel dual-learning approach is developed in this paper. Inspired by human-brain that can effectively balance optimality and safety, the developed dual learning algorithm includes a slow learning component mirroring the neocortical learning [8] and a fast learning component mimicking the hippocampal learning process [8] in the human brain. Neocortical learning, inspired by the human brain neocortex, allows the system to assimilate knowledge over time. Considering the system dynamic is unknown, slow learning can develop a neural network-based system identification approach that learns the unknown part of the agent dynamics first. Then, the slow learning component is able to learn the model over time through collected real-time operation data gradually. Please note that the computational complexity of model learning prohibits the instantaneous identification of unknown dynamics. In addition, a receding horizon control (RHC) method has been developed to improve system performance by guiding the agent to its predetermined destination. However, RHC requires solving optimization problems within a short time frame to provide time-efficient control. This becomes challenging in real-time implementation where rapid and safe adaptation is crucial. In this approach, optimal control is periodically updated and fed into the fast learning component to enhance overall system effectiveness. Here, the solution of the receding horizon optimal control problem relies on a partial differential equation (PDE) called the Hamilton-Jacobi-Bellman (HJB) equation [9], [10]. To solve the HJB equation with an uncertain nonlinear system and obtain the optimal control strategy, the physic-informed neural network (PINN) [11] is modified to a situation-aware physic-informed neural network (SA-PINN) based learning algorithm. Meanwhile, hippocampal learning allows the system to adapt to time-varying environments rapidly. The fast learning consists of two parts. First, a fast-learned adaptive control barrier function (FA-CBF) is formulated to ensure the forward invariance of a safe set. The proposed FA-CBF initially incorporates a bound threshold for uncertain dynamics to ensure the strict safety of the system. As the system model gradually improves through slow learning, fast learning updates the threshold bound, becoming less conservative over time. Furthermore, an adaptive control framework has been developed to integrate an RHC-based optimal controller from slow learning and a CBF-based safe controller into a unified framework that can simultaneously ensure system performance and safety. The key contributions of this method are as follows:

- A real-time control algorithm based on Receding Horizon Control (RHC) and safe control has been designed and integrated into a unified framework.
- To enable real-time intelligent safety-critical control, a dual learning strategy is developed, drawing inspiration from human behavior modeled from the hippocampus and neocortex. This approach mirrors fast and slow learning processes, with the slow learning component focusing on understanding uncertain dynamics and implementation of a receding horizon optimal control utilizing a learning algorithm based on situation-aware physics-informed neural networks (SA-PINN). Simultaneously, the fast learning component facilitates real-

The authors are with the Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV, 89557 USA e-mail: shawondey@nevada.unr.edu; haoxu@unr.edu.

time safe adaptation, striking a balance between real-time performance and safety considerations.

II. Problem Formulation

Consider the following nonlinear affine dynamic system

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

with $x(t) \in \mathbb{R}^n$ denote the system state vector, $u(t) \in \mathbb{R}^m$ represents the control input vector. Different than others [12], [13], the intelligent safety-critical control input $u(t) = u_k(t) + u_s(t)$ developed in this paper is the combination of the optimal control $u_k(t)$ which is directed to the fast learning at k-th time instant and the safe control $u_s(t)$. The functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ describe the intrinsic dynamics of the system. Please note the dynamics of the system are not known in advance. The primary goals of this research are:

1) Develop a feedback control strategy for an uncertain nonlinear dynamic agent to attain real-time optimal performance and strict safety while navigating towards a predefined destination. The strict safety requirements ensure that the agent trajectory remains within a safe set, satisfying the condition $b(x) \geq 0$ for $t > 0$, where $b(x)$ represents a continuously differentiable barrier function.
2) To achieve this real-time control, design a dual learning approach inspired by human behavior learning from the hippocampus and neocortex region, corresponding to fast and slow learning, respectively.

Next, the details of the developed dual-learning algorithm are given.

III. A Dual Learning Algorithm

The learning algorithm outlined in this section draws inspiration from the human learning process as described in [14]. It emulates how humans learn through the interaction of experiences, swiftly adapting to uncertain situations. For achieving optimal performance towards a predefined goal, the receding horizon control has been employed. However, implementing this control strategy in real-time is challenging due to computational complexity and the lack of built-in safety measures. To enable real-time decision-making, control, and ensure prompt safety measures, we introduce a fast or hippocampal learning component that starts with conservative bounds and progressively relaxes conditions over time. Specifically, the fast learning component implements the controller in real-time, receiving optimal updates from the slow learning to strike a balance between performance and safety. The applied controller feedback is then looped back to the slow learning component, allowing the algorithm to gradually accumulate experience related to both optimal performance and safety.

A. Slow Learning: A Neocortical Approach to Learn Model Uncertainty and Optimal Strategies

To implement receding horizon optimal control, it is necessary to know the system dynamics which is unrealistic. Therefore, this paper aims to relax the requirement of known system dynamics. Specifically, model learning is adopted where the agent begins with a nominal model and gradually learns the original system.

1) Neural Network-based System Approximator: The actual system dynamic of the agent can be represented as

$$\dot{x}(t) = Ax + f_{\text{true}}(x) + g_{\text{true}}(x)u(t) \tag{2}$$

with $u(t) = u_k(t) + u_s(t)$ and $A \in \mathbb{R}^{n\times n}$ represents a constant matrix. In accordance with the universal approximation property [15] of neural network (NN), the ideal nonlinear functions can be represented as

$$f_{\text{true}}(x) = \theta_f^T \phi_f(x) + \varepsilon_f(x)\ ;\ g_{\text{true}}(x) = \theta_g^T \phi_g(x) + \varepsilon_g(x) \tag{3}$$

with θ_f^T and θ_g^T are the respective weights of the NNs. Moreover, $\phi_f(x)$ and $\phi_g(x)$ are the activation functions and $\varepsilon_f(x)$ and $\varepsilon_g(x)$ denotes reconstruction errors. Then, the actual system dynamic can be rewritten as:

$$\begin{aligned}\dot{x}(t) &= Ax + \theta_f^T \phi_f(x) + \theta_g^T \phi_g(x)u(t) + \varepsilon_f(x) + \varepsilon_g(x)u(t)\\ &= Ax + \theta^T \phi_{\text{true}}(x)\bar{u} + \bar{\varepsilon}\end{aligned} \tag{4}$$

with $\theta = \begin{bmatrix}\theta_f & \theta_g\end{bmatrix}^T$, $\phi_{\text{true}}(x) = \text{diag}\{\phi_f(x), \phi_g(x)\}$, $\bar{u} = \begin{bmatrix}1 & u(t)\end{bmatrix}^T$ and $\bar{\varepsilon} = \varepsilon_f(x) + \varepsilon_g(x)u(t)$. Now, the initial nominal model of the agent system is as follows:

$$\dot{\hat{x}} = A\hat{x} + f_{\text{nomi}}(x) + g_{\text{nomi}}(x)u(t) \tag{5}$$

Then the NN-based estimated system dynamic is derived as:

$$\dot{\hat{x}} = A\hat{x} + f_{\text{nomi}}(x) + f_l(x) + [g_{\text{nomi}}(x) + g_l(x)]u(t) \tag{6}$$

where $f_l(x)$ and $g_l(x)$ are neural network estimation functions. Also, $\mathcal{L} \in \mathbb{R}^{n\times n}$ be a design matrix with $A_o = A - \mathcal{L}$. Then, the state estimation error is $\zeta = x(t) - \hat{x}(t)$. Now, using estimated weight and activation functions, the equation (6) is rewritten with feedback term:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + f_{\text{nomi}}(x) + f_l(x) + [g_{\text{nomi}}(x) + g_l(x)]u(t) + \mathcal{L}\zeta\\ &= A\hat{x} + \hat{\theta}^T \phi_{\text{true}}(\hat{x})\bar{u} + \mathcal{L}\zeta\end{aligned} \tag{7}$$

Here, $\hat{\theta}$ is the estimated weight. The system dynamic model approximation error is derived using equations (4) and (7):

$$\dot{\zeta} = A_o\zeta + \theta^T \tilde{\phi}_{\text{true}}\bar{u} + \tilde{\theta}^T \hat{\phi}_{\text{true}}\bar{u} + \bar{\varepsilon} \tag{8}$$

Now the objective function is defined as:

$$J_{\text{dyna}} = \frac{1}{2}\dot{\zeta}^T\dot{\zeta} = \frac{1}{2}\|\dot{x} - A\hat{x} - \hat{\theta}^T \phi_{\text{true}}(\hat{x})\bar{u} - \mathcal{L}\zeta\|^2 \tag{9}$$

Depending on this objective function, the gradient descent-based weight update law is defined as:

$$\begin{aligned}\dot{\hat{\theta}} &= -\alpha_d \frac{\partial J_{\text{dyna}}}{\partial \hat{\theta}}\\ &= \alpha_d \hat{\phi}_{\text{true}}\bar{u}[A_o\zeta + \theta^T \tilde{\phi}_{\text{true}}\bar{u} + \tilde{\theta}^T \hat{\phi}_{\text{true}}\bar{u} + \bar{\varepsilon}]^T\end{aligned} \tag{10}$$

Here α_d denotes the learning rate of the neural network. The weight approximation error dynamics is:

$$\dot{\tilde{\theta}} = -\alpha_d \hat{\phi}_{\text{true}}\bar{u}[A_o\zeta + \theta^T \tilde{\phi}_{\text{true}}\bar{u} + \tilde{\theta}^T \hat{\phi}_{\text{true}}\bar{u} + \bar{\varepsilon}]^T \tag{11}$$

2) Receding Horizon Control: Since the Receding Horizon Control (RHC) is computationally expensive and hard to implement in real-time, the slow learning framework has been used and integrated with RHC to learn the optimal control over time. Specifically, We have considered a finite-length prediction horizon. Here, $\{t_k\}_{k=0}^{\infty}$ represents a sequence over the timeline. The iteration period between sequence k and $k+1$ is denoted as $\varepsilon_{\text{ita}} \in \mathbb{R}^+$ that is $t_{k+1} = t_k + \varepsilon_{\text{ita}}$. Now, at each time instant t_k the RHC controller solves the optimal control problem over the time interval $[t_k, t_k + T]$, where T is the length of the prediction horizon. It's important to note that the prediction horizon duration T is greater than or equal to the iteration period ε_{ita}. Then the cost function can be formulated as follows:

$$J(x,u) = V_T(x(t_k+T)) + \int_{t_k}^{t_k+T} [\|x\|_Q^2 + \|u\|_R^2]dt \quad (12)$$
$$\text{s.t. } \dot{x}(t) = \hat{f}(x) + \hat{g}(x)u(t)$$

where V_T is the terminal cost, $\hat{f}$ and $\hat{g}$ are the estimated functions. Also, Q and R are the positive definite weight matrix for the state and control input. Now, the receding horizon optimal control is formulated for the estimated continuous system dynamic given in (1) and the cost function in (12). In addition, an optimal control policy is formulated to achieve the optimal cost function in (12). Now, the Hamiltonian [16] of this optimal control problem is defined as:

$$H(\hat{x},u) = \|\hat{x}\|_Q^2 + \|u\|_R^2 + \partial_x J(\hat{x},u)[\hat{f}(x) + \hat{g}(x)u(t)] \quad (13)$$

with $\hat{f}(x)$ and $\hat{g}(x)$ are the estimated nonlinear functions from the neural network. Now, the optimal control at the kth iteration from the slow learning part is defined as:

$$u(t) = u_k(t) = -1/2\ R^{-1}\hat{g}^T(x)\partial_x J(\hat{x},u) \quad (14)$$

This control input is inserted into (13) to achieve the HJB equation as follows:

$$\|x\|_Q^2 + \partial_x J(\hat{x},u)\hat{f}(x) - 1/4\ R^{-1}\hat{g}^T(x)\partial_x J(\hat{x},u)\hat{g}^T(x)$$
$$\partial_x J(\hat{x},u) = 0 \quad (15)$$

Addressing the receding horizon optimal control problem involves tackling the Hamilton-Jacobi-Bellman (HJB) equation to derive the optimal value function. Substituting this value function into the control equation (14) allows for the determination of the optimal control. Nevertheless, solving the HJB equation with an uncertain nonlinear system poses challenges due to its inherent nonlinearity. Here, the solution of the HJB equation involves the integration of a learning algorithm with SA-PINNs.

Situation Aware Physics-Informed Neural Network (SA-PINN): In this part, the agent aims to predict the optimal value function through the SA-PINNs learning algorithm with the unknown nonlinear system. By incorporating SA-PINN, the agent can gradually and accurately learn the optimal action without requiring an excessive amount of data and exact system dynamic information, achieved through solving the PDE known as the HJB equation. Now, the ideal value function is defined:

$$J(\hat{x},u) = W_J^T\phi_J(\hat{x},u) + \varepsilon_J(\hat{x}) \quad (16)$$

Here, W_J denotes the neural network weight, $\phi_J(\hat{x},u)$ is the activation function, and $\varepsilon_J(\hat{x})$ represents the reconstruction error. Next, substituting (16) into the HJB equation (15), the impact of the reconstruction error is taken into account:

$$\|\hat{x}\|_Q^2 + W_J^T\partial_x\phi_J(\hat{x},u)\hat{f}(x) - 1/4\ W_J^T\partial_x\phi_J(\hat{x},u)\mathcal{D}(\hat{x})$$
$$W_J^T\partial_x\phi_J(\hat{x},u) + \varepsilon_{\mathcal{R}}(\hat{x}) = 0 \quad (17)$$

with $\mathcal{D}(\hat{x}) = \hat{g}(x)R^{-1}\hat{g}^T(x)$ and $\varepsilon_{\mathcal{R}}(\hat{x}) = -1/2W_J^T\partial_x$ $\phi_J(\hat{x},u)\mathcal{D}(\hat{x})\partial_x\varepsilon_J(\hat{x}) - 1/4\ \partial_x\varepsilon_J(\hat{x})\mathcal{D}(\hat{x})\partial_x\varepsilon_J(\hat{x}) + \partial_x\varepsilon_J$ $(\hat{x})\hat{f}(x)$. Now, the value function approximation is:

$$\hat{J}(\hat{x},u) = \hat{W}_J^T\hat{\phi}_J(\hat{x},u) \quad (18)$$

Inserting the approximated value function in (15), the residual error can be calculated as:

$$\mathcal{R}_e = \|\hat{x}\|_Q^2 + \hat{W}_J^T\partial_x\hat{\phi}_J(\hat{x},u)\hat{f}(x) - 1/4\ \hat{W}_J^T\partial_x\hat{\phi}_J(\hat{x},u)$$
$$\mathcal{D}(\hat{x})\hat{W}_J^T\partial_x\hat{\phi}_J(\hat{x},u) \quad (19)$$

Then, substituting (19) into (17):

$$\mathcal{R}_e = \tilde{W}_J^T\partial_x\hat{\phi}_J(\hat{x},u)\hat{f}(x) + W_J^T\partial_x\tilde{\phi}_J(\hat{x},u)\hat{f}(x) + 1/4$$
$$W_J^T\partial_x\phi_J(\hat{x},u)\mathcal{D}(\hat{x})W_J^T\partial_x\phi_J(\hat{x},u) - 1/4\ \hat{W}_J^T\partial_x\hat{\phi}_J(\hat{x},u)$$
$$\mathcal{D}(\hat{x})\hat{W}_J^T\partial_x\hat{\phi}_J(\hat{x},u) - \varepsilon_{\mathcal{R}} \quad (20)$$

Moreover, the HJB equation is subject to the initial and terminal conditions and is defined as:

$$J(\hat{x},u,0) = J_{\text{ini}}(\hat{x},u)\ \ ;\ \ V_T(x(t_k+T)) = V_{\text{ter}}(\hat{x},u) \quad (21)$$

Then, the physics-informed learning model can be trained based on the following objective function:

$$J_{\text{PINN}}(\hat{W}_J) = 1/2\ w_{ic}J_{ic}(\hat{W}_J) + 1/2\ w_{tc}J_{tc}(\hat{W}_J)$$
$$+ 1/2\ w_{\mathcal{R}_e}\|\mathcal{R}_e(\hat{W}_J)\|^2 \quad (22)$$
$$\text{with, } J_{ic}(\hat{W}_J) = \|J_{\text{ini}}(\hat{x},u) - \hat{W}_J^T\hat{\phi}_J(\hat{x},u,0)\|^2$$
$$= \|W_J^T\phi_J(\hat{x},u,0) - \hat{W}_J^T\hat{\phi}_J(\hat{x},u,0) + \varepsilon_{\text{ini}}\|^2 \quad (23)$$
$$J_{tc}(\hat{W}_J) = \|V_{\text{ter}}(\hat{x},u) - \hat{W}_J^T\hat{\phi}_J(\hat{x},u,t_k+T)\|^2$$
$$= \|W_J^T\phi_J(\hat{x},u,t_k+T) - \hat{W}_J^T\hat{\phi}_J(\hat{x},u,t_k+T) + \varepsilon_{\text{ter}}\|^2 \quad (24)$$

The weight update rule through the given objective function is derived as follows:

$$\dot{\hat{W}}_J = -\alpha_J\frac{\partial J_{\text{PINN}}(\hat{W}_J)}{\partial\hat{W}_J}$$
$$= \alpha_j[w_{ic}\hat{\phi}_J(\hat{x},u,0)\{W_J^T\phi_J(\hat{x},u,0) - \hat{W}_J^T\hat{\phi}_J(\hat{x},u,0)$$
$$+ \varepsilon_{\text{ini}}\}^T + w_{tc}\hat{\phi}_J(\hat{x},u,t_k+T)\{W_J^T\phi_J(\hat{x},u,t_k+T) -$$
$$\hat{W}_J^T\hat{\phi}_J(\hat{x},u,t_k+T) + \varepsilon_{\text{ter}}\}^T + w_{\mathcal{R}_e}1/2\ \partial_x\hat{\phi}_J(\hat{x},u)$$
$$\mathcal{D}(\hat{x})\hat{W}_J^T\partial_x\hat{\phi}_J(\hat{x},u)\mathcal{R}_e^T] \quad (25)$$

Here, α_J is the learning rate. Next, the weight approximation error dynamic is defined as:

$$\dot{\tilde{W}}_J = -\alpha_j[w_{ic}\hat{\phi}_J(\hat{x},u,0)\{W_J^T\phi_J(\hat{x},u,0) - \hat{W}_J^T\hat{\phi}_J(\hat{x},u,0)$$

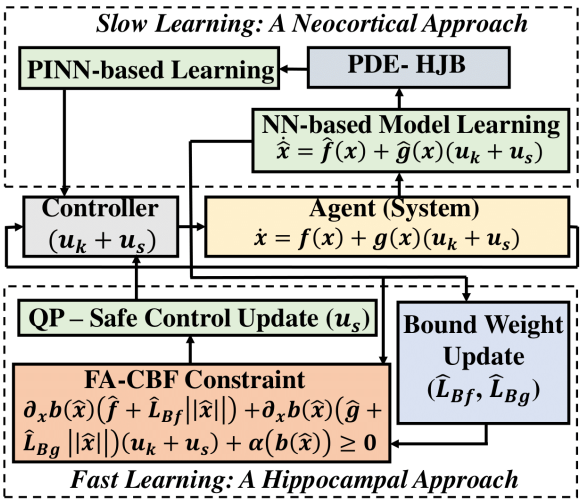


Fig. 1: The structure of the dual learning approach.

$$+ \varepsilon_{\text{ini}}\}^T + w_{tc}\hat{\phi}_J(\hat{x}, u, t_k + T)\{W_J^T\phi_J(\hat{x}, u, t_k + T) - \hat{W}_J^T\hat{\phi}_J(\hat{x}, u, t_k + T) + \varepsilon_{\text{ter}}\}^T + w_{\mathcal{R}_e}1/2\ \partial_x\hat{\phi}_J(\hat{x}, u)\mathcal{D}(\hat{x})\hat{W}_J^T\partial_x\hat{\phi}_J(\hat{x}, u)\mathcal{R}_e^T] \quad (26)$$

Substituting the ideal value function from (16) to the optimal control equation (14), the optimal control at the kth iteration from the slow learning part is rewritten as:

$$u_k(t) = -1/2\ R^{-1}\hat{g}^T(x)\partial_x[W_J^T\phi_J(\hat{x}, u) + \varepsilon_J(\hat{x})] = -\frac{1}{2}\ R^{-1}\hat{g}^T(x)W_J^T\partial_x\phi_J(\hat{x}, u) - 1/2\ R^{-1}\hat{g}^T(x)\partial_x\varepsilon_J(\hat{x}) \quad (27)$$

The estimated optimal control depends on the estimated value function from (18) as

$$\hat{u}_k(t) = -1/2\ R^{-1}\hat{g}^T(x)\hat{W}_J^T\partial_x\phi_J(\hat{x}, u) \quad (28)$$

The optimal control approximation error is derived as:

$$\tilde{u}_k(t) = -1/2\ R^{-1}\hat{g}^T(x)[\tilde{W}_J^T\partial_x\phi_J(\hat{x}, u) + \partial_x\varepsilon_J(\hat{x})] \quad (29)$$

Subsequently, the optimal control derived from the slow learning phase is transferred to the fast learning phase. In this fast learning phase, the agent implements real-time control, effectively achieving a balance between safety and performance considerations.

B. Fast Learning: A Hippocampal Approach for Real-Time Safe Control with Guaranteed Performance

In this section, the focus is on ensuring the safety of the system. The safety framework is defined by an invariant safe set denoted as $\mathcal{S}$. This set $\mathcal{S}$ is considered the super level set of a continuously differentiable smooth function $b : \mathbb{R}^n \to \mathbb{R}$. Given that the dynamics of the agent's system are initially unknown and gradually learned in the slow learning phase, an adaptive bound for the system dynamics is utilized to guarantee the strict safety of the agent. This bound undergoes a gradual relaxation over time, reflecting the evolving understanding of the system acquired during the slow learning phase. Now the agent system dynamic in the fast learning part is estimated as:

$$\dot{\hat{x}} = \hat{f}(x) + \hat{L}_{\text{Bf}}\|\hat{x}\| + [\hat{g}(x) + \hat{L}_{\text{Bg}}\|\hat{x}\|](u_k(t) + u_s(t)) \quad (30)$$

with $\hat{L}_{\text{Bf}}$ and $\hat{L}_{\text{Bg}}$ being the adaptive bound weight. Initially, a conservative bound is given to ensure strict safety and this bound is updated and relaxed over time. The system state errors including the bound is described as follows:

$$\dot{\tilde{x}} = [f(\hat{x}) - \hat{f}(\hat{x})] + [g(\hat{x}) - \hat{g}(\hat{x})](u_k(t) + u_s(t)) - \hat{L}_{\text{Bf}}\|\hat{x}\| - \hat{L}_{\text{Bg}}\|\hat{x}\|(u_k(t) + u_s(t)) \quad (31)$$

Next, the bound weights are updated as:

$$\dot{\hat{L}}_{\text{Bf}} = -\eta_{\text{Bf}}[(1/2)\ \partial\{\dot{\tilde{x}}^T\dot{\tilde{x}}\}]/\partial\hat{L}_{\text{Bf}} = \eta_{\text{Bf}}\|\hat{x}\|\dot{\tilde{x}}^T \quad (32)$$

$$\dot{\hat{L}}_{\text{Bg}} = -\eta_{\text{Bg}}[(1/2)\ \partial\{\dot{\tilde{x}}^T\dot{\tilde{x}}\}]/\partial\hat{L}_{\text{Bg}} = \eta_{\text{Bg}}\|\hat{x}\|(u_k + u_s)\dot{\tilde{x}}^T \quad (33)$$

where η_{Bf} and η_{Bg} are the tuning rate. Also, the approximated weight error is defined as:

$$\dot{\tilde{L}}_{\text{Bf}} = -\eta_{\text{Bf}}\|\hat{x}\|\dot{\tilde{x}}^T \ ; \ \dot{\tilde{L}}_{\text{Bg}} = -\eta_{\text{Bg}}\|\hat{x}\|(u_k + u_s)\dot{\tilde{x}}^T \quad (34)$$

Next, Nagumo's theorem [17] provides a necessary and sufficient condition for set invariance in dynamical systems. Applying this theorem to the system described in equation (30), the condition for set invariance can be expressed as: $\mathcal{S} = \{x \in \mathbb{R}^n : b(\hat{x}) \geq 0\}$, $\partial\mathcal{S} = \{x \in \mathbb{R}^n : b(\hat{x}) = 0\}$ and $\text{Int}(\mathcal{S}) = \{x \in \mathbb{R}^n : b(\hat{x}) > 0\}$. Here, $\partial\mathcal{S}$ and $\text{Int}(\mathcal{S})$ represent the boundary and interior of the safe set $\mathcal{S}$ respectively. Now, b can be identified as a fast adaptive control barrier function (FA-CBF) if there is a function α belonging to the extended class $\mathcal{K}_\alpha$, and the provided dynamical system meets the following condition:

$$\sup_{u \in U}[\partial_x b(\hat{x})(\hat{f}(x) + \hat{L}_{\text{Bf}}\|\hat{x}\|) + \partial_x b(\hat{x})(\hat{g}(x) + \hat{L}_{\text{Bg}}\|\hat{x}\|)(u_k(t) + u_s(t))] \geq -\alpha(b(\hat{x})) \quad (35)$$

Here, the optimal control $u_k(t)$ is injected from the slow learning part. Please note that $u_s(t)$ is the safe control applied in the fast learning part which satisfies the above condition. Now, the extended $\mathcal{K}_\alpha$ function can be defined as follows:

Definition 1: A function $\alpha : \mathbb{R} \to \mathbb{R}$ is known as an extended class $\mathcal{K}_\alpha$ function if the function is strictly increasing and $\alpha(0) = 0$. Please see ([5]) for the definition.

The control inputs that meet the conditions specified in equation (35) and ensure the safety of the set $\mathcal{S}$ can be described as:

$$K_{\text{cbf}}(\hat{x}) = \{u \in U : \partial_x b(\hat{x})(\hat{f}(x) + \hat{L}_{\text{Bf}}\|\hat{x}\|) + \partial_x b(\hat{x})(\hat{g}(x) + \hat{L}_{\text{Bg}}\|\hat{x}\|)(u_k(t) + u_s(t)) + \alpha(b(\hat{x})) \geq 0\} \quad (36)$$

It's important to highlight that the optimal control, denoted as $u_k(t)$, obtained from slow learning, faces challenges in real-time implementation due to computational complexity and is inherently unsafe. In the fast learning phase, the

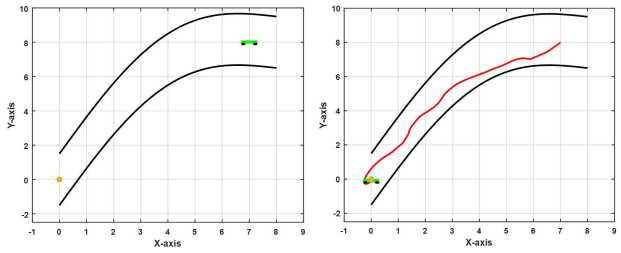
combination of a safe control strategy and optimal control ensures real-time control that is both safe and high-performing. The design of the controller responsible for maintaining the system state within a safe set and stable requires the incorporation of a Lyapunov function denoted as $L_e(\hat{x})$. The integration of this Lyapunov function is crucial to address the constraint on the derivative of $L_e(\hat{x})$ and to unify it with the Fast Adaptive-CBF constraint. This integration aims to ensure safety, stability, and system performance. To design this safe controller that filtered the optimal action from the slow learning part to ensure safety and real-time execution, a quadratic programming (QP) based approach has been chosen.

$$
\begin{aligned}
u_s(\hat{x}) = \underset{(u,\delta)}{\arg\min} \quad & \frac{1}{2}\|u_s\|_2 + p\delta^2 \\
\text{s.t.} \quad & \partial_x b(\hat{x})(\hat{f}(x) + \hat{L}_{\text{Bf}}\|\hat{x}\|) + \partial_x b(\hat{x})(\hat{g}(x) + \\
& \hat{L}_{\text{Bg}}\|\hat{x}\|)(u_k(t) + u_s(t)) + \alpha(b(\hat{x})) \geq 0 \\
& \dot{L}_e(\hat{x}) \leq \delta
\end{aligned} \tag{37}
$$

Note that δ represents a relaxation variable introduced to ensure the solvability of the quadratic program, while p denotes the coefficient for the relaxation factor.

IV. Simulation Results

This simulation study implemented the developed algorithm in an autonomous vehicle (AV) to demonstrate real-time, reliable control that combines fast and slow learning for guaranteed performance and safety. The objective is to



(a) t = 0s (b) t = 20s

Fig. 2: The movement of the AV is depicted here. The red curve shows the path taken by the vehicle. Two black curves indicate the boundaries. The yellow circle marks the destination point.

enhance vehicle performance by integrating a slow learning approach based on the neocortex, while also swiftly making safe decisions through a fast learning framework inspired by the hippocampus. The initial state of the AV is selected as $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} 7 & 8 \end{bmatrix}^T$. Next, the intrinsic dynamic of the AV is defined as

$$
f(x) = \begin{bmatrix} -x_1 - x_2 \\ -0.5x_1 - 0.5x_2(1 - (cos(2x_1) + 2)^2) \end{bmatrix} \tag{38}
$$

$$
g(x) = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix} \tag{39}
$$

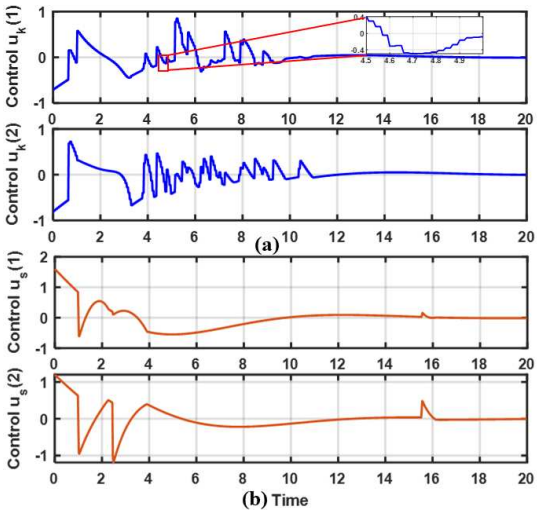


Fig. 3: Convergence of control input (a) control input u_k from slow learning (b) safe control input u_s from fast learning.

The algorithm comprises two components: a slow-learning module and a fast-learning module. The slow learning module focuses on understanding uncertain dynamics and facilitating receding horizon optimal control. The NN-based system approximation consists of a three-layer nonlinear feedforward network comprising one input, one hidden, and one output layer. The activation functions of NN-system approximation are selected as a hyperbolic tangent function, i.e. $tanh(*)$. This activation function is selected for both $f(x)$ and $g(x)$ approximation. The learning rate and error threshold of this NN is defined as $\alpha_d = 1 \times 10^{-3}$ and δ_ζ respectively. Also, the prediction horizon is set to $T = 4$ seconds, and the iteration period is defined as $\varepsilon_{\text{ita}} = 0.1$ seconds. The activation function for the SA-PINNs learning was selected from a polynomial expansion represented by the formula $\sum_{\beta=1}^{P}(\sum_{j=1}^{n} z_j)^\beta$, where n denotes the input dimension and P is a predetermined constant. Also, the learning rate and error threshold are selected as $\alpha_J = 1 \times 10^{-3}$ and $\delta_J = 1 \times 10^{-5}$. The tuning rates are chosen as $\eta_{\text{Bf}} = \eta_{\text{Bg}} = 0.1$. Additionally, the threshold is specified as $\delta_{\tilde{x}} = 1 \times 10^{-2}$. Now, the barrier function is defined as $b(\hat{x}) = (\|\hat{x} - x_{\max}\|_2 - \varepsilon_1)(\|\hat{x} - x_{\min}\|_2 - \varepsilon_2)$ with $\varepsilon_1 = 0.3$ and $\varepsilon_2 = 0.3$. Here, the ε_1 and ε_2 represent the minimum safe distance of the AV from the boundary. Figure 2 illustrates how the autonomous vehicle maneuvers using the developed fast-slow learning-based real-time reliable control method. The objective is to maintain optimal performance and safety while reaching a predefined destination. The locations of the vehicle are shown at different time instants: $t = 0s$ and $t = 20s$ in Figures 2(a) and 2(b) respectively. From the figure, it is clear that the AV avoids unsafe maneuvers and safely reaches the destination without approaching the boundary. Figure 3 illustrates the control input from both fast and slow learning components. In Figure 3(a), the optimal control input u_k from the slow learning is shown. This

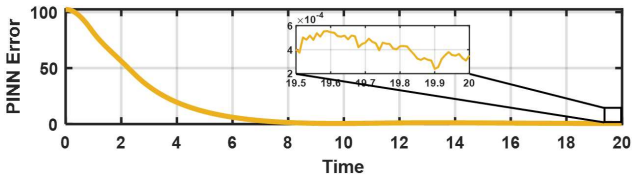


Fig. 4: Error convergence of the physics-informed based neural network.

optimal control steadily converges over time. A smaller window provides a closer look at the details. The figure demonstrates that the control input from the Receding Horizon Control (RHC) updates periodically over time, which is then transferred to the fast learning part. However, relying solely on this control can potentially lead to unsafe control strategies. The subsequent illustration showcases the control mechanism from fast learning, as referenced in Figure 3(b). This safety-oriented control aids the AV in quickly making safe decisions. Variations observed at different time points signify the timely adjustments made by the control to maintain a safe distance from the safety boundary Figure 4 illustrates the convergence of the objective function error for the SA-PINNs employed in learning the HJB partial differential equation (PDE), leading to the determination of the optimal control.

V. Conclusion

This paper introduces a biologically inspired dual learning approach, combining real-time optimal receding horizon control (RHC) strategy with integrated real-time safety measures for a complex nonlinear system subject to uncertainties. The developed algorithm aims to strike a balance between the agent's optimal performance and safety. The framework incorporates a novel dual learning algorithm, where the slow learning component involves training neural networks to model the uncertain system and implementing optimal RHC control using a situation-aware physics-informed neural network (SA-PINN). In addition, the fast learning component addresses this unsafety challenge from uncertain environments by designing a fast adaptive Control Barrier Function (FA-CBF) along with adaptive boundary updates. The resulting safe control from the fast learning part, combined with the optimal control from the slow learning aspect, offers a real-time, reliable control strategy that ensures both optimal performance and safety. Moreover, the paper includes Lyapunov stability analysis to demonstrate the convergence of the dual learning process and closed-loop stability. Eventually, numerical experiments are conducted to illustrate the effectiveness of the developed approach.

References

[1] J. Russell Carpenter, "Decentralized control of satellite formations," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 12, no. 2-3, pp. 141–161, 2002.

[2] S. Milani, H. Khayyam, H. Marzbani, W. Melek, N. L. Azad, and R. N. Jazar, "Smart autodriver algorithm for real-time autonomous vehicle trajectory control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1984–1995, 2020.

[3] S. Dey and H. Xu, "Distributed adaptive flocking control for large-scale multiagent systems," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[4] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 269–296, 2020.

[5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[6] Z. Marvi and B. Kiumarsi, "Safe reinforcement learning: A control barrier function optimization approach," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, 2021.

[7] Y. Zhang, S. S. Ge, X. Liang, B. V. E. How, and H. Chen, "Safety-critical automated surface vessels mimo control with adaptive control barrier functions under model uncertainties," *IEEE Transactions on Automation Science and Engineering*, pp. 1–13, 2024.

[8] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory." *Psychological review*, vol. 102, no. 3, p. 419, 1995.

[9] M. Bardi, I. C. Dolcetta *et al.*, *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Springer, 1997, vol. 12.

[10] S. Dey and H. Xu, "Hierarchical game theoretical distributed adaptive control for large scale multi-group multi-agent system," *IET Control Theory & Applications*, vol. 17, no. 17, pp. 2332–2352, 2023.

[11] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[12] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[13] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 108–113, 2019.

[14] R. C. O'Reilly and K. A. Norman, "Hippocampal and neocortical contributions to memory: Advances in the complementary learning systems framework," *Trends in cognitive sciences*, vol. 6, no. 12, pp. 505–510, 2002.

[15] T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE transactions on neural networks*, vol. 6, no. 4, pp. 911–917, 1995.

[16] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[17] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.