Received: 11 October 2019 | Revised: 15 January 2021 | Accepted: 21 February 2021

DOI: 10.1002/acs.3237

RESEARCH ARTICLE

WILEY

Dynamical system learning using extreme learning machines with safety and stability guarantees

Iman Salehi | **Ghananeel Rotithor** | **Gang Yao** | **Ashwin P. Dani**

Department of Electrical and Computer Engineering, University of Connecticut, Storrs, Connecticut, USA

Correspondence
Ashwin P. Dani, Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA.
Email: ashwin.dani@uconn.edu

Funding information
Space Technology Research Institutes from NASA Space Technology Research Grants Program, in part by the U.S., Grant/Award Number: 80NSSC19K1076; Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office, Grant/Award Number: DE-EE0007613; Advanced Robotics for Manufacturing ("ARM") Institute, Grant/Award Number: ARM-17-QS-F-04; Office of the Secretary of Defense, Grant/Award Number: W911 NF-17-3-0004

Summary

This article presents a continuous dynamical system model learning methodology that can be used to generate reference trajectories for the autonomous systems to follow, such that these trajectories are invariant to a given closed set and uniformly ultimately bounded with respect to an equilibrium point inside the closed set. The autonomous system dynamics are approximated using extreme learning machines (ELM), the parameters of which are learned subject to the safety constraints expressed using a reciprocal barrier function, and the stability constraints derived using a Lyapunov analysis in the presence of the ELM reconstruction error. This formulation leads to solving a constrained quadratic program (QP) that includes a finite number of decision variables with an infinite number of constraints. Theorems are developed to relax the QP with infinite number of constraints to a QP with a finite number of constraints which can be practically implemented using a QP solver. In addition, an active sampling methodology is developed that further reduced the number of required constraints for the QP by only evaluating the constraints at a smaller subset of points. The proposed method is validated using a motion reproduction task on a seven degree-of-freedom Baxter robot, where the task space position and velocity dynamics are learned using the presented methodology.

KEYWORDS
barrier certificates, model learning, robustness, safety

1 | INTRODUCTION

Learning desired system dynamics from recorded motion trajectory data is one of the methods used to produce desired motion paths that autonomous systems can follow in many manufacturing and space robotics applications to simplify robot programming.[1] Gaussian mixture models (GMM),[2] Gaussian process regression,[3] artificial neural networks,[4] and extreme learning machines (ELM)[5] are a few among many function approximation methods that can be used to encode the trajectory data in the form of a differential equation model to represent the system dynamics. Unlike classical system identification approaches[6,7] data-driven approaches create black box models that do not require much prior knowledge about

The article was intended for the *Adaptive Methods for Resilient Control Systems* Special Issue.
[Correction added on 4 June 2021, after first online publication: the article category was changed from 'SPECIAL ISSUE ARTICLE' to 'RESEARCH ARTICLE', and a footnote text was added in this version.]

the system.[8] While accurate encoding of the motion plans for autonomous systems is fundamental to robotics and control systems applications, preserving natural properties observed in these motion plans such as convergence to a goal location, avoiding prohibited areas, obstacle, or staying within given bounds (invariance), robustness to faults in the data-driven model is essential. The stability and robustness properties provide resiliency property to the learned system model.

Classical system identification literature has developed methods that learn the stable closed-loop transfer function models.[9] A concurrent learning is an adaptive update scheme, where the model parameters are identified correctly without persistency of excitation condition, is developed in Reference 10. In imitation learning literature, one of the commonly used dynamical system (DS)-based frameworks is called as a dynamic movement primitives.[11-13] In Reference 14, a nonlinear state space system is learned using a stable estimator of dynamical systems algorithm that preserves the property of a global convergence to an equilibrium point. The problem is formulated as a parameter learning problem of GMM subject to stability constraints derived from Lyapunov analysis. In Reference 15, a method called control Lyapunov function (LF)-based dynamic movements is introduced to stabilize the system. As it is mentioned in Reference 16, although the method can learn complex dynamics, it requires an online correction signal and does not provide mathematical guarantees. In Reference 17, a method called contracting dynamical system primitives (CDSP) that learns the parameters of GMM subject to partial contraction analysis is presented. The method uses a sum-of-squares based decomposition to eliminate the state dependence from the constraints, which overcomes the problem of infinite constraints. Nonlinear DS models that are learned using the CDSP method are shown to be capable of recreating complex trajectories and are robust to sudden target change and spatial perturbations. An algorithm to encode motion dynamics using ELM, called neurally imprinted vector fields, is presented in Reference 18. Stability of the learned system is verified using constraints derived through Lyapunov analysis. However, these methods do not consider safety property in the model learning problem. In this article, the model learning problem is formulated subject to the safety constraints encoded using reciprocal barrier functions (BFs) in addition to the stability constraints encoded using LFs.

In addition to the stability property of DSs, providing safety guarantees on the learned model is another critical property to be considered in many autonomous robotic systems such as robot manipulators, autonomous vehicles, and so forth.[19,20] In Reference 20, an active learning method is developed to learn the safe region of attraction set given the system dynamics represented using a Gaussian process (GP). In Reference 21, a supervised machine learning approach is used to estimate the control BFs from sensor data. In References 22,23, a reinforcement learning method that guarantees stability and safely explores the state space to collect new data for learning is presented. Barrier certificates is a commonly used approach to study forward invariance of the DS model in a given closed set, which can be used to examine the safety of the dynamic system model. Barrier certificates define a forward invariant safe region such that the solutions of the DS that start in that region remain in that region for all time.[24,25] In References 19,26-29, barrier certificates are successfully used for DSs where ensuring safety conditions is critical. In References 23,30 BF transform is used to constrain the state space in order to develop a controller using reinforcement learning. In Reference 31, algorithms for learning uncertainty represented as a GP in a nonlinear DS, as well as for maximal safe set estimation using BFs, are derived. In Reference 32, a cooperative control law is developed which is based on control barrier function for human–robotic network teaming. However, these methods focus on achieving safety property using a control design. By contrast, a model learning problem is solved in this article such that the learned model preserves stability and safety properties observed in the data. In Figure 1 trajectories that are generated using three different parameterization techniques, namely, GMM, ELM, and single layer feedforward network without any constraints are shown. As shown in Figure 1, models that are learned without constraints will result in trajectories leaving a prescribed set. The set can be regarded as a safe region for the robot to operate, for example, avoiding obstacles or in achieving safety in human–robot collaboration where the human is modeled as a moving obstacle. In Figure 2 a sequence of images is shown where the motion trajectories generated for the Baxter robot fail to remain inside the prescribed set.

The main contribution of this article is to design a model learning methodology for a continuous nonlinear DS model, which preserves the forward invariance property in a prescribed set and ensures the motion trajectories are uniformly ultimately bounded (UUB) with respect to an equilibrium point inside the invariant region. Due to the constraints developed using BF and Lyapunov analyses, the proposed method accurately reproduces the demonstrations from any initial conditions that start inside the invariant set. Moreover, it guarantees that the learned trajectories remain inside a closed set and converge to a bound. An ELM is used to approximate the unknown continuous nonlinear function of the DS. The learning problem is formulated as a parameter learning problem of ELM subject to the BF constraints along with the stability constraints derived using Lyapunov analysis in the presence of the function reconstruction error of the ELM. Moreover, the parameter learning problem is posed as an optimization problem where the safety and stability constraints

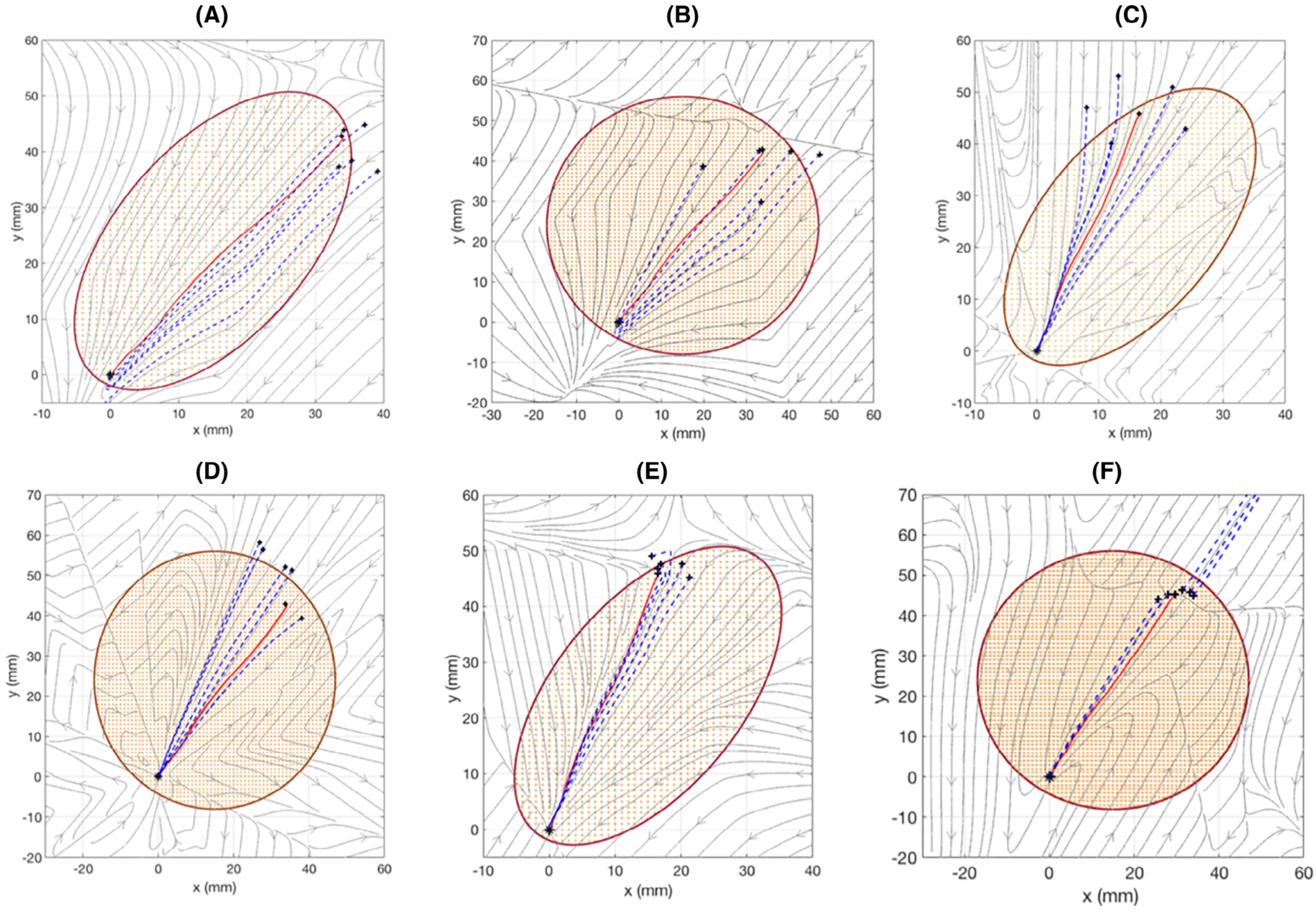

FIGURE 1 Results of model using Gaussian mixture model ((A) and (B)), extreme learning machines (C) and (D)), single layer feedforward network ((E) and (F)) without constraints. Ellipse and circle are used as invariant sets. (shown in dark red). The streamlines of the models learned *without* barrier function constraints (in gray) are shown along with the demonstrated data (in solid red lines) and the reproductions (in dashed blue lines) [Colour figure can be viewed at wileyonlinelibrary.com]

are enforced at all the points in the invariant set, which leads to a semiinfinite program, that is, optimization with a finite number of decision variables and an infinite number of constraints.[33] In this article, the challenge of an infinite number of constraints is overcome by drawing random samples from uniformly distributed points inside the invariant set of interest, and only a finite number of constraints are used in the optimization problem for ELM parameter learning. By exploiting the locally Lipschitz continuous assumption of the BF, Theorem 1 is developed, which states that enforcing a modified reciprocal barrier constraint at a finite number of points inside the invariant set is sufficient to ensure forward invariance of the entire set with respect to the trajectories of the nonlinear system. Similarly, it is proven in Theorem 2 that only a finite number of sample points from the inside of the invariant set suffices to ensure the trajectories of the nonlinear system are UUB. Furthermore, an active sampling strategy is used to select points that are most informative for enforcing the reciprocal barrier and Lyapunov constraints. The sampling strategy substantially reduces the number of points used for training the nonlinear DS model. Thus, scaling of the learning process to a higher state dimensions is possible. The Lyapunov condition provides exponential stability of the learned system model without the function reconstruction errors. It is shown that with the ELM function reconstruction error and external disturbances added to the system, the system trajectories remain within a ball of known radius with respect to the goal point. Thus, the learned system model is robust with respect to the external disturbances. Due to the ELM approximation of the system model, the optimization problem for NN training is convex. The reciprocal barrier constraints with ELM parameter learning can be reformulated as linear constraints. For tuning the input layer weights, slopes and biases of the ELM are obtained using the batch intrinsic plasticity (BIP) algorithm.[34]

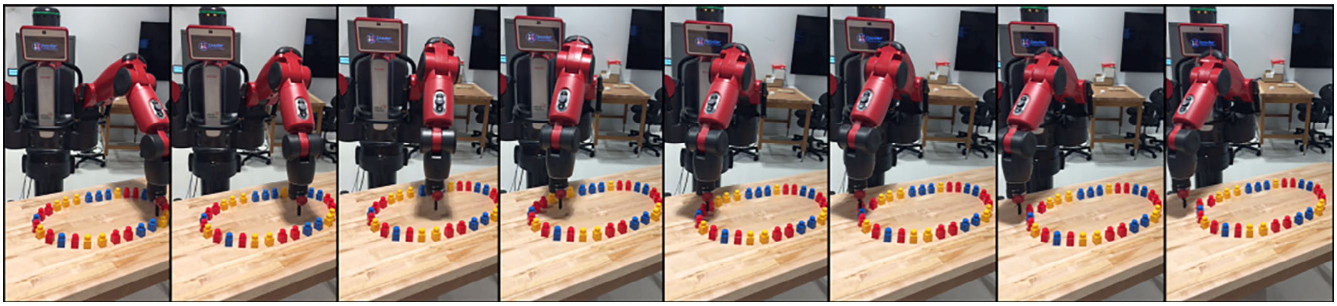

FIGURE 2 Sequence of images show that the robot end effector goes outside the ellipse when the model is trained *without* the barrier constraints [Colour figure can be viewed at wileyonlinelibrary.com]

Compared with our prior work published in Reference 35, this article presents a detailed literature review of the state-of-the-art. It develops a quadratic programming (QP) scheme based on BF and LF to achieve a safe and stable identification of the motion dynamics in the presence of the function reconstruction error of the ELM. Lemma 2 is added to prove that the derived barrier constraint is implementable, and it is robust to the parameter uncertainty associated with the ELM model. In addition, Lemma 4 is added, which is related to the robustness analysis of the learned model to external disturbances. Moreover, an active sampling method is proposed that selects the most informative points for enforcing both safety and stability constraints. The proposed method is evaluated through numerical simulations on multiple motion trajectories of various shapes from a publicly available dataset. The performance of the method is demonstrated on a Baxter robot, and the method's ability to learn and reproduce motions from real-world data is illustrated.

The rest of the article is organized as follows. In Section 2, a brief introduction to the BF and ELM is provided. The problem formalism and the proposed method are described in Section 3. In Section 4, a set of experiments illustrate the learning method's ability to learn the DS under barrier and Lyapunov constraints. Finally, in Section 5, discussion of results and conclusions of the work are given along with possible future works.

2 | PRELIMINARIES

In this section, preliminaries of BFs, ELM with BIP learning are presented. ELM is used to approximate the nonlinear function of the DS.

2.1 | Barrier function

Consider a continuous nonlinear DS of the form

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

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous nonlinear function and $x(t) \in \mathcal{S} \subseteq \mathbb{R}^n$ is the state of the system. A set $\mathcal{S} \in \mathbb{R}^n$ is called *(forward) invariant* with respect to (1) if for any initial condition $x(0) := x(t_0) \in \mathcal{S}$ implies that $x(t) \in \mathcal{S}$, $\forall t \in \mathbb{R}$.[36] Note that throughout this article, for ease of notation, we abbreviate $x(t)$ by x, unless necessary for clarity.

Assumption 1. The function $f : \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz continuous with Lipschitz constant L_f and bounded in $\mathcal{S}$ with $\|f(\cdot)\| \le \bar{f}$, where $\bar{f}$ is a positive scalar.

BFs define a forward invariant safe region, where the solutions of a DS in this region remain in the region for all time.[19,37]

2.1.1 | Constructing the BF

Given a closed set $\mathcal{S} \subset \mathbb{R}^n$, its interior and its boundary are defined as follows

$$\mathcal{S} = \{x \in \mathbb{R}^n : h(x) \geq 0\}, \tag{2}$$

$$\partial\mathcal{S} = \{x \in \mathbb{R}^n : h(x) = 0\}, \tag{3}$$

$$\mathrm{Int}(\mathcal{S}) = \{x \in \mathbb{R}^n : h(x) > 0\}, \tag{4}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function.

Definition 1 (19, definition 1). Given the continuous system (1), the closed set $\mathcal{S}$ defined by (2)–(4), and continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, a real-valued function $B : \mathrm{Int}(\mathcal{S}) \to \mathbb{R}$ that is differentiable with respect to its argument is said to be a reciprocal BF, if there exist class $\mathcal{K}$ functions η_1, η_2, η_3 such that for all $x \in \mathrm{Int}(\mathcal{S})$

$$\frac{1}{\eta_1(h(x))} \leq B(x) \leq \frac{1}{\eta_2(h(x))} \tag{5}$$

$$\frac{\partial B(x)}{\partial x} f(x) \leq \eta_3(h(x)). \tag{6}$$

Candidate reciprocal BFs are *inverse-type* and *logarithmic-type* BFs given by $B(x) = \frac{1}{h(x)}$ and $B(x) = -\log \frac{h(x)}{1+h(x)}$, respectively.[19] Note that the candidate is unbounded on the set boundary, that is, $B(x) \to \infty$ as $x \to \partial\mathcal{S}$. In this article, an *inverse-type* reciprocal BF is used.

Assumption 2. The real-valued function $B(x)$ is locally Lipschitz continuous with Lipschitz constant L_B such that $\left\| \frac{\partial B}{\partial x} \right\| \leq L_B$; $\frac{\partial B}{\partial x}$ is locally Lipschitz continuous with Lipschitz constant $L_{\partial B}$.

Conventionally safety verification in the form of trajectory invariance with respect to a given closed set relies on the existence of a BF satisfying conditions on both the function itself and its time derivative along solution trajectories of the DS, namely, $\dot{B}(x) \leq 0$.[24] However, as it is mentioned in Reference 38, the existence of a BF is just a sufficient condition to guarantee the safety property to be verified. In other words, the invariance verification of all sublevels of the closed set $\mathcal{S}$ may not be required. Therefore, authors in Reference 19 relaxed the condition to $\dot{B} \leq \frac{\gamma}{B}$ for Reciprocal BFs, where γ is a positive constant. The modification enables only a single sublevel set to be invariant.

2.2 | The ELM and BIP

ELM is a learning algorithm that uses least squares approach to estimate the parameters of the single layer neural network (SLNN). According to the ELM theory,[39] input weights and all the hidden node parameters are randomly assigned. ELM theory claims, unlike the conventional learning methods, that parameter (i.e., input weights, slopes, and biases) tuning is not required in the learning. The choice of ELM over traditional gradient-based methods for SLNN is primarily based on faster learning speed of ELM. This choice also avoids potential issues with the gradient-based methods, such as local minima and improper learning rate.[39] Consider the following multilayer perceptron with n_h hidden nodes

$$y = \sum_{i}^{n_h} G_i(x, U_i, a_i, b_i) \cdot W_i^T, \tag{7}$$

where $y \in \mathbb{R}^n$, $G_i(\cdot)$ is a scalar, which denotes the ith hidden node activation function whose range depends on the choice of the nonlinear function, $U_i \in \mathbb{R}^n$ is the input weight vector connecting the input layer to the ith hidden neuron, $a_i, b_i \in \mathbb{R}$ are slope and bias corresponding to the ith hidden neuron, and $W_i^T \in \mathbb{R}^n$ is the output weight vector connecting the output layer to the ith hidden neuron. The ELM is created by randomly initializing the input weight matrix $U \in \mathbb{R}^{n \times n_h}$, the slopes (usually are set to one) $a \in \mathbb{R}^{n_h}$, and the biases $b \in \mathbb{R}^{n_h}$. The sigmoid function is typically chosen for the activation function. Although the ELM algorithm is fast, random initialization may lead to having saturated or constant neurons, which are not desired while learning the model.[34] To circumvent this problem, an intrinsic plasticity (IP) learning rule can be used.[40] IP is an online learning method that optimizes the information transmission of a single neuron by adapting the slopes and biases such that the output of the activation function becomes exponentially distributed. In our article, a computationally efficient batch version of the IP method, called BIP (cf. Reference 34) is used.

3 | PROBLEM FORMULATION AND SOLUTION APPROACH

3.1 | Constrained ELM learning problem

Consider demonstrations of a motion governed by the first-order differential equation (1), and let a set of N demonstrations be solutions to the DS. The nth demonstration consists of trajectories of both the states $\{x(t)\}_{t=0}^{t=T_n}$ and its derivatives $\{\dot{x}(t)\}_{t=0}^{t=T_n}$. Assuming that the nonlinear function $f(\cdot)$ in (1) is deterministic, the system in (1) can be approximated by ELM given by

$$\dot{x}(t) = f(x(t)) = W^T\sigma(q \cdot s(t)) + \epsilon(x(t)), \tag{8}$$

where $W \in \mathbb{R}^{n_h \times n}$ is the bounded constant output layer weight matrix, $\sigma(\cdot) \in \mathbb{R}^{n_h}$ is the vector sigmoid activation function, n_h is the number of neurons in the hidden layer of the ELM, and $\epsilon(x) \in \mathbb{R}^n$ is the function reconstruction error. The vector sigmoid activation function is given by $\sigma(q \cdot s(t)) = [\frac{1}{1+\exp(-(q\cdot s(t))_1)}, \dots, \frac{1}{1+\exp(-(q\cdot s(t))_i)}, \dots, \frac{1}{1+\exp(-(q\cdot s(t))_{n_h})}]^T$, where $(q \cdot s(t))_i$ is the *ith* element of the vector $(q \cdot s(t))$, $q = [P,\ b_p] \in \mathbb{R}^{n_h \times n+1}$, $P = \mathrm{diag}(a_p) \cdot U^T \in \mathbb{R}^{n_h \times n}$, $a_p \in \mathbb{R}^{n_h}$ and $b_p \in \mathbb{R}^{n_h}$ are the internal slopes and biases vectors, respectively, $U \in \mathbb{R}^{n \times n_h}$ is the bounded input layer weight matrix, and $s(t) = [x(t)^T, 1]^T \in \mathbb{R}^{n+1}$ is the input vector to the ELM.

Remark 1. The nonlinear function $f(x)$ is within real and positive constant $\overline{\epsilon}$ of the ELM range if there exist a finite number of hidden neurons n_h and constant weights so that for all $x(t) \in \mathcal{S}$ the approximation in (8) holds with $\|\epsilon(x)\| \le \overline{\epsilon}$. The boundedness of the reconstruction error $\epsilon(x)$ follows from the *Universal Approximation Property* of SLNN[41] and the fact that for any randomly generated activation function the reconstruction error goes to zero,[42,43] that is, $\lim_{n_h\to\infty} \epsilon(x) = 0$.

Assumption 3. The ideal weights of the ELM are bounded by known positive constants, that is, $\|W\|_F \le \overline{W}$, $\|U\|_F \le \bar{U}$, where $\|\cdot\|_F$ is the Frobenius norm.[44] In addition, it is assumed that $\|\epsilon'(x)\| \le \overline{\epsilon}'$, where prime denotes the derivation with respect to x.

Remark 2. The sigmoid function $\sigma_i(\cdot) \in [0, 1]$ and hence its derivative $\sigma_i(\cdot)(1 - \sigma_i(\cdot))$ has upper and lower bounds given by $0 \le \sigma_i(\cdot)(1 - \sigma_i(\cdot)) \le 0.25, \forall i = 1, \dots n_h$. Thus, $\|\sigma(\cdot)\| \le \sqrt{n_h}$ and $\|\sigma(\cdot)(1 - \sigma(\cdot))\| \le 0.25\sqrt{n_h}$. Given that the input and output weight matrices are assumed to be bounded, Assumption 1 is still valid for the ELM parameterization of $f(x)$.[45]

Given the trajectory data, this article addresses the problem of learning the nonlinear dynamic model $f(\cdot)$ in (8), which is approximated using ELM, such that the set bounded by the state predefined limits remains forward invariant and the trajectories are stable with respect to a specified equilibrium point inside the invariant region.

3.2 | Constrained learning methods for nonlinear ELM dynamics

In this subsection, the BF formulation is presented, which is used in the parameter learning algorithm of the ELM subject to the barrier constraints. The method is then extended to parameter learning of ELM subject to Lyapunov stability constraints followed by a method that learns the nonlinear model, which has both stability and safety constraints encoded using barrier and LFs formulation.

3.2.1 | Encoding safety constraints

To establish the safety of the learned model BFs are used to certify invariance of the DS defined in (8) with respect to a given closed set.

Reciprocal BF formulation

As mentioned in Subsection 2.1, BF defines a forward invariant region such that solutions of the DS that start in that region remain in that region for all time. Now, depending on the applications one could choose any arbitrary shape of the invariant regions, for which the BF can be designed, as long as it satisfies the conditions defined in (2)–(4). Without loss of generality, ellipse and circle are the two shapes chosen in this article to define the invariant regions. Such shapes are a commonly used approximation for representing obstacles, or the region of operation where the robot is allowed to move. To construct an inverse-type reciprocal BF, consider the convex connected set $\mathcal{X}_0 \subset \mathbb{R}^2$, which is defined according to (2) in Section 2.1.1. The function $h(x)$, where $x = [x_1, x_2]^T \in \mathcal{X}_0$ is selected as follows

$$h(x) = 1 - \frac{((x_1 - x_{1g})\cos\alpha + (x_2 - x_{2g})\sin\alpha)^2}{a_1^2} - \frac{((x_1 - x_{1g})\sin\alpha + ((x_2 - x_{2g})\cos\alpha)^2}{a_2^2}, \tag{9}$$

where x_{1g} and x_{2g} are the coordinates of the center of gravity of the ellipse, a_1 and a_2 are the major and minor axes of the ellipse, respectively, and α is the orientation of ellipse.

Remark 3. On defining the invariant region, the continuously differentiable function $h(x)$ in (9) can also be constructed for the higher dimensions.

Learning dynamics using ELM with BF constraints

Given the ELM parametrization and BF constraints, the ELM parameter learning problem with BF constraints is now discussed. The constrained optimization problem to be solved in order to train the ELM is given by

$$W^* = \arg\min_W \sum_{t=0}^{T_n} [\dot{x}(t) - \hat{\dot{x}}(t)]^T [\dot{x}(t) - \hat{\dot{x}}(t)] + \mu_W(tr(W^T W)) \tag{10}$$

$$\text{s.t.} \quad \dot{B}(x) \leq \frac{\gamma}{B(x)}, \quad \forall x \in \mathcal{X}_0, \tag{11}$$

where $\dot{x}(t) \in \mathbb{R}^n$ and $\hat{\dot{x}}(t) \in \mathbb{R}^n$ represent the target and the ELM's output, $\mu_W \in \mathbb{R}^+$ is the parameter of the regularization, $B(x) \in \mathbb{R}$ is a BF, and $\gamma \in \mathbb{R}^+$ is a positive constant. For the reciprocal BF $B(x) = \frac{1}{h(x)}$, the constraint in (11) can be rewritten as follows:

$$C_B = \left\{ x \in \mathcal{X}_0 \; : \; \frac{\partial B}{\partial x} f(x) - \frac{\gamma}{B(x)} \leq 0 \right\}. \tag{12}$$

Enforcing the constraint (12) in the optimization problem (10) and (11) for learning the DS is a difficult task because the constraint (12) has to be satisfied $\forall x \in \mathcal{X}_0$, which is an uncountably infinite set. To mitigate this problem, the Lipschitz continuity of B^{-1} is proven in Lemma 1.

Lemma 1. *The functions $B^{-1}(x)$ is Lipschitz continuous with Lipschitz constant $L_{B^{-1}}$.*

Proof. To prove $B^{-1}(x)$ is Lipschitz continuous, we know that $h(x)$ is a continuously differentiable function, and therefore the following function $\|B^{-1}(x) - B^{-1}(x')\|$ can be rewritten as $\|h(x) - h(x')\| \leq L_h \|x - x'\|$ with $\left\| \frac{\partial h(x)}{\partial x} \right\| \leq L_{h=B^{-1}}$, $\forall x, x' \in \mathcal{X}_0 \subset \mathbb{R}^n$, proving the result. ∎

Given that $B^{-1}(x)$ is Lipschitz continuous in x with Lipschitz constants $L_{B^{-1}}$, next it is proven in Theorem 1 that the constraint in (12) needs to be evaluated at only finite number of sampled points in $\mathcal{X}_0$. It is proven that enforcing a modified barrier constraint for a finite number of sampled points is sufficient for providing solution to the optimization problem in (10) and (11). Let $\mathcal{X}_\tau \subset \mathcal{X}_0$ be a discretization of the state space $\mathcal{X}_0$ with the closest point in $\mathcal{X}_\tau$ to $x \in \mathcal{X}_0$ denoted by $[x]_\tau$ such that $\|x - [x]_\tau\| \leq \frac{\tau}{2}$, and τ is the discretization resolution.

Remark 4. The discretized state space $\mathcal{X}_\tau$ can be obtained by a simple grid discretization, that is, for example to use uniform sampling strategy. Other examples for probability sampling are including but not limited to simple random sampling, systematic sampling, stratified sampling, and so forth.[46,47]

Theorem 1. *If the following condition holds*

$$\left[\frac{\partial B}{\partial x}f(x)\right]_\tau - \gamma B^{-1}([x]_\tau) \leq -\left(\mathcal{L}_{\dot{B}} + \gamma L_{B^{-1}}\frac{\tau}{2}\right), \quad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{13}$$

where $\mathcal{L}_{\dot{B}} \triangleq (L_{\partial B}(\overline{W}\sqrt{n_h} + \overline{\epsilon}) + L_B L_f)\frac{\tau}{2}$, *then the safety barrier constraint in (12) is satisfied* $\forall x \in \mathcal{X}_0$.

Proof. Using $\dot{B}(x) = \frac{\partial B(x)}{\partial x}f(x)$, (13) can be rewritten as $\dot{B}([x]_\tau) - \gamma B^{-1}([x]_\tau) \leq -\left(\mathcal{L}_{\dot{B}} + \gamma L_{B^{-1}}\frac{\tau}{2}\right)$, $\forall [x]_\tau \in \mathcal{X}_\tau$, which implies

$$\dot{B}([x]_\tau) + \mathcal{L}_{\dot{B}} - \gamma B^{-1}([x]_\tau) + \gamma L_{B^{-1}}\frac{\tau}{2} \leq 0, \quad \forall [x]_\tau \in \mathcal{X}_\tau. \tag{14}$$

If $\dot{B}(x) - \gamma B^{-1}(x)$ is a lower bound on $\dot{B}([x]_\tau) + \mathcal{L}_{\dot{B}} - \gamma B^{-1}([x]_\tau) + \gamma L_{B^{-1}}\frac{\tau}{2}$, then the barrier constraint in (12) will be satisfied. Consider

$$\mathcal{O} = \dot{B}(x) - \dot{B}([x]_\tau) - \gamma(B^{-1}(x) - B^{-1}([x]_\tau)), \tag{15}$$

which by using the triangle inequality, $\mathcal{O}$ in (15) can be upper bounded as

$$\mathcal{O} \leq |(\dot{B}(x) - \dot{B}([x]_\tau))| + \gamma|(B^{-1}(x) - B^{-1}([x]_\tau))|. \tag{16}$$

Adding and subtracting $\frac{\partial B([x]_\tau)}{\partial x}f(x)$ to the first term on the right-hand side of (16), that is, $|\dot{B}(x) - \dot{B}([x]_\tau)|$ and using Assumptions 1–3 we have

$$\begin{aligned}
|(\dot{B}(x) - \dot{B}([x]_\tau))| &= \left|\left(\frac{\partial B(x)}{\partial x} - \frac{\partial B([x]_\tau)}{\partial x}\right)f(x) + \frac{\partial B([x]_\tau)}{\partial x}(f(x) - f([x]_\tau))\right| \\
&\leq \left\|\left(\frac{\partial B(x)}{\partial x} - \frac{\partial B([x]_\tau)}{\partial x}\right)\right\| \|W^T\sigma(q\cdot s(t)) + \epsilon(x)\| + \left\|\frac{\partial B([x]_\tau)}{\partial x}\right\| \|(f(x) - f([x]_\tau))\| \\
&\leq (L_{\partial B}(\overline{W}\sqrt{n_h} + \overline{\epsilon}) + L_B L_f)\|x - [x]_\tau\| \\
&\leq (L_{\partial B}(\overline{W}\sqrt{n_h} + \overline{\epsilon}) + L_B L_f)\frac{\tau}{2} = \mathcal{L}_{\dot{B}}.
\end{aligned} \tag{17}$$

Furthermore, by using Lipschitz continuity of B^{-1}, $\forall x \in \mathcal{X}_0$, and thereby mean value theorem, (16) can be rewritten as follows

$$\begin{aligned}
\dot{B}(x) - \dot{B}([x]_\tau) - \gamma(B^{-1}(x) - B^{-1}([x]_\tau)) &\leq |(\dot{B}(x) - \dot{B}([x]_\tau))| + \gamma|(B^{-1}(x) - B^{-1}([x]_\tau))| \\
&\leq \mathcal{L}_{\dot{B}} + \gamma\left(\left\|\frac{\partial B^{-1}(\xi)}{\partial x}\right\| \|(x - [x]_\tau)\|\right),
\end{aligned} \tag{18}$$

where $\xi \in (x, [x]_\tau)$ is a point on the line segment connecting x to $[x]_\tau$. Using the results of Lemma 1, $\left\|\frac{\partial B^{-1}(\xi)}{\partial x}\right\|$ can be upper bounded by $L_{B^{-1}}$. Hence,

$$\dot{B}(x) - \dot{B}([x]_\tau) - \gamma(B^{-1}(x) - B^{-1}([x]_\tau)) \leq \mathcal{L}_{\dot{B}} + \gamma L_{B^{-1}}\frac{\tau}{2}, \tag{19}$$

which yields

$$\dot{B}(x) - \gamma B^{-1}(x) \leq \dot{B}([x]_\tau) + \mathcal{L}_{\dot{B}} - \gamma B^{-1}([x]_\tau) + \gamma L_{B^{-1}}\frac{\tau}{2} \quad \forall x \in \mathcal{X}_0 \text{ and } \forall [x]_\tau \in \mathcal{X}_\tau. \tag{20}$$

From (14) and (20), $\dot{B}(x) - \gamma B^{-1}(x) \leq 0, \quad \forall x \in \mathcal{X}_0$. ■

Using the result of Theorem 1, the optimization problem (10) and (11) can be written in its modified form, given by

$$W^* = \arg\min_{W} \sum_{t=0}^{T_n} [\dot{x}(t) - \hat{\dot{x}}(t)]^T [\dot{x}(t) - \hat{\dot{x}}(t)] + \mu_W (tr(W^T W)) \tag{21}$$

$$\text{s.t.} \quad \dot{B}([x]_\tau) - \gamma B^{-1}([x]_\tau) \leq -\left(\mathcal{L}_{\dot{B}} + \gamma L_{B^{-1}} \frac{\tau}{2}\right), \ \forall [x]_\tau \in \mathcal{X}_\tau. \tag{22}$$

Substituting (8) in (13), the optimization problem in (21) and (22) can be written in terms of $h(x)$ as follows

$$W^* = \arg\min_{W} \sum_{t=0}^{T_n} [\dot{x}(t) - \hat{\dot{x}}(t)]^T [\dot{x}(t) - \hat{\dot{x}}(t)] + \mu_W (tr(W^T W))$$
$$\text{s.t.} \quad -J_x([x]_\tau)(W^T \sigma(q \cdot [s]_\tau) + \epsilon([x]_\tau)) - \gamma h^3([x]_\tau) \leq -h^2([x]_\tau) \left(\mathcal{L}_{\dot{B}} + \gamma L_{B^{-1}} \frac{\tau}{2}\right) \quad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{23}$$

where $J_x \in \mathbb{R}^{1\times n}$ is the gradient of $h(x)$. Moreover, the constraint in (23) can be reformulated as a linear constraint in decision variable θ, given by

$$-J_x([x]_\tau)(\Sigma\theta + \epsilon([x]_\tau)) - \gamma h^3([x]_\tau) \leq -h^2([x]_\tau) \left(\mathcal{L}_{\dot{B}} + \gamma L_{B^{-1}} \frac{\tau}{2}\right), \quad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{24}$$

where $\theta = [w_1^T, \ldots, w_n^T]^T \in \mathbb{R}^{(n \cdot n_h)}$ and $w_i \in \mathbb{R}^{n_h}$, $\forall i = 1, \ldots, n$ are the columns of the output weight matrix $W \in \mathbb{R}^{n_h \times n}$, and $\Sigma \in \mathbb{R}^{n \times (n \cdot n_h)}$ is given by

$$\Sigma = \begin{bmatrix} \sigma(\cdot)^T & & 0_{1\times(n-1)\cdot n_h} \\ & \ddots & \\ 0_{1\times(n-1)\cdot n_h} & & \sigma(\cdot)^T \end{bmatrix}. \tag{25}$$

In practice the reconstruction error is not measurable, and therefore enforcing BF constraint in (24) is not feasible. The following lemma shows that BF constraint is robust to parameter uncertainty and proves that while (24) is not implementable the new modified BF constraint in (26) can be employed instead.

Lemma 2. *If the following constraint is satisfied for the DS in (8),*

$$-J_x([x]_\tau)\Sigma\theta - \gamma h^3([x]_\tau) \leq -h^2([x]_\tau)\mathcal{E} \quad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{26}$$

where

$$\mathcal{E} = (L_{\partial B}(\overline{W}\sqrt{n_h} + \overline{\epsilon}) + L_B L_f + \gamma L_{B^{-1}})\frac{\tau}{2} + L_B \overline{\epsilon} > 0, \tag{27}$$

then system trajectories are guaranteed to be safe with respect to the invariant region defined by $h([x]_\tau)$*, but also robust to parameter uncertainty, that is,* $\epsilon([x]_\tau)$*.*

Proof. Consider the constraint (24), which guarantees the safety of the DS in (8) in the form of trajectory invariance with respect to the compact set $\mathcal{X}_0$. It can be rewritten as

$$\frac{\partial B}{\partial x}\Sigma\theta + \frac{\partial B}{\partial x}\epsilon([x]_\tau) - \gamma h([x]_\tau) \leq -\mathcal{L}_{\dot{B}} - \gamma L_{B^{-1}}\frac{\tau}{2}, \quad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{28}$$

which after substituting (17) in (28), upper bounding $\frac{\partial B}{\partial x}\epsilon([x]_\tau)$ once it is taken to the right-hand side, and simple algebraic manipulation it yields

$$\frac{\partial B}{\partial x}\Sigma\theta - \gamma h([x]_\tau) \leq -(L_{\partial B}(\overline{W}\sqrt{n_h} + \overline{\epsilon}) + L_B L_f + \gamma L_{B^{-1}})\frac{\tau}{2} - L_B\overline{\epsilon}$$
$$-J_x([x]_\tau)\Sigma\theta - \gamma h^3([x]_\tau) \leq -h^2([x]_\tau)\mathcal{E}, \quad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{29}$$

where $\mathcal{E}$ is defined in (27). ∎

3.2.2 | Encoding stability constraints

The boundedness solutions of the nonlinear DS defined in (8) can be ensured using Lemma 3.

Lemma 3. *The nonlinear system defined in (8) is said to be UUB if*

$$C_L = \{x \in \mathcal{X}_0 : \ (x - x^*)^T \Sigma\theta \leq -\beta(x)\}, \tag{30}$$

where $\beta(x) = \rho(x - x^*)^T(x - x^*)$, x^* is the equilibrium point, and $\rho \in \mathbb{R}^+$ is a positive constant. Furthermore, the ultimate bound is given by

$$\|x - x^*\| \leq \frac{\overline{\epsilon}}{\rho}, \quad \forall t > t_0 + T \text{ and } T \geq 0. \tag{31}$$

Proof. Consider a quadratic Lyapunov candidate function $V(x) = \frac{1}{2}(x - x^*)^T(x - x^*)$, $\forall x \in \mathcal{X}_0$. Taking the time derivative of the LF along the trajectories of (8) yields

$$\dot{V}(x) = (x - x^*)^T\dot{x} = (x - x^*)^T(W^T\sigma(q \cdot s) + \epsilon(x)). \tag{32}$$

Given $W = [w_1, \ \ldots \ , w_n]$, where $w_i \in \mathbb{R}^{n_h}$ for all $i = 1, \ \ldots \ , n$, and also the fact that $w_i^T\sigma(\cdot)$ is a scalar, (32) can be rewritten as a linear constraint in decision variable $\theta = [w_1^T, \ \ldots \ , w_n^T]^T \in \mathbb{R}^{(n \cdot n_h)}$ given by

$$\dot{V}(x) = (x - x^*)^T(\Sigma\theta + \epsilon(x)), \tag{33}$$

where Σ is defined in (25). Using the definition of $\dot{V}(x)$ in (33), the constraint in (30) can be written as

$$\begin{aligned} \dot{V}(x) - (x - x^*)^T\epsilon(x) &\leq -\rho\|x - x^*\|^2 \\ \dot{V}(x) &\leq -\rho\|x - x^*\|^2 + (x - x^*)^T\epsilon(x), \end{aligned} \tag{34}$$

which, after upper bounding the second term on the right-hand side of (34) and completing the squares, can be simplified further and it is given by

$$\dot{V}(x) \leq -\rho V(x) + \frac{\overline{\epsilon}^2}{2\rho}. \tag{35}$$

Thus, invoking theorem 4.18 in Reference 36 the ultimate bound is given by $\|x - x^*\| \leq \frac{\overline{\epsilon}}{\rho}$, $\forall t > t_0 + T$, which $T \geq 0$. ■

Similar to Theorem 1, the LF constraint defined in (30) needs to be evaluated at only finite number of sampled points in $\mathcal{X}_0$. To this end, Theorem 2 proves that enforcing a modified Lyapunov constraint for a finite number of sampled points is sufficient to obtain an optimal solution of the QP defined in (10).

Theorem 2. *Given Assumption 3, if the following condition holds*

$$([x]_\tau - x^*)^T\Sigma\theta \leq -\beta([x]_\tau) - \mathcal{L}_V\frac{\tau}{2}, \qquad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{36}$$

where $\mathcal{L}_V = L_{\dot{V}} + 2\rho L_V + \sqrt{2V([x]_\tau)}\overline{\epsilon}' + \overline{\epsilon}$, L_V is a Lipschitz constant of the selected LF,

$$L_{\dot{V}} = \overline{W}\sqrt{n_h} + \overline{\epsilon} + \|\xi - x^*\|\left(\frac{\bar{a}_p\sqrt{n_h}\overline{W}\bar{U}}{4} + \overline{\epsilon}'\right), \tag{37}$$

$\xi \in (x, [x]_\tau)$ is a point on the line segment connecting x to $[x]_\tau$, and$\bar{a}_p = \|\text{diag}(a_p)\|_F$, then the Lyapunov constraint in (30) is satisfied $\forall x \in \mathcal{X}_0$.

Proof. The constraint in (36) can be rewritten as

$$\dot{V}([x]_\tau) - ([x]_\tau - x^*)^T \epsilon([x]_\tau) + \mathcal{L}_V \frac{\tau}{2} + \beta([x]_\tau) \leq 0, \tag{38}$$

for all $[x]_\tau \in \mathcal{X}_\tau$. If $\dot{V}(x) + \beta(x) - (x - x^*)^T \epsilon(x)$ is an upper bound on (38) then the Lyapunov constraint given in (30) is satisfied. Using the mean value theorem for $\dot{V}(x)$ yields

$$\dot{V}(x) - \dot{V}([x]_\tau) = \frac{\partial \dot{V}(\xi)}{\partial x}(x - [x]_\tau), \tag{39}$$

The LHS of Equation (39) can be upper bounded by

$$\dot{V}(x) - \dot{V}([x]_\tau) \leq \left\| \frac{\partial \dot{V}(\xi)}{\partial x} \right\| \|(x - [x]_\tau)\|. \tag{40}$$

The upper bound on $\frac{\partial \dot{V}(\xi)}{\partial x}$ can be derived as follows

$$\begin{aligned}
\frac{\partial \dot{V}(\xi)}{\partial x} &= \frac{\partial}{\partial x}(x - x^*)^T (W^T \sigma(q \cdot s) + \epsilon(x)) \\
\left\| \frac{\partial \dot{V}(\xi)}{\partial x} \right\| &\leq \|W^T\| \|\sigma(q \cdot [\xi^T,\ 1]^T)\| + \|(\xi - x^*)^T\| \|W^T\| \|\sigma'(q \cdot [\xi^T,\ 1]^T)\| \|\mathrm{diag}(a_p)\| \|U^T\| \\
&\quad + \|\epsilon(x)\| + \|(\xi - x^*)^T\| \|\epsilon'(x)\| \\
&= \overline{W}\sqrt{n_h} + \overline{\epsilon} + \|\xi - x^*\|(0.25\sqrt{n_h}\bar{a}_p \overline{W}\bar{U} + \overline{\epsilon}') = L_{\dot{V}},
\end{aligned} \tag{41}$$

where Remarks 1 and 2 are used. Now, consider $\mathcal{O}_v = (\dot{V}(x) - \dot{V}([x]_\tau)) + (\beta(x) - \beta([x]_\tau)) + [([x]_\tau - x^*)^T \epsilon([x]_\tau) - (x - x^*)^T \epsilon(x)]$, which it can be upper bounded as

$$\mathcal{O}_v \leq \|(\dot{V}(x) - \dot{V}([x]_\tau))\| + 2\rho \|(V(x) - V([x]_\tau))\| + \|[([x]_\tau - x^*)^T \epsilon([x]_\tau) - (x - x^*)^T \epsilon(x)]\|. \tag{42}$$

Adding and subtracting $[x]_\tau^T \epsilon(x)$ to the third term on the RHS of (42), using Assumption 3 and the triangle and Cauchy-Schwartz inequalities we have

$$\mathcal{O}_v \leq (L_{\dot{V}} + 2\rho L_V + \sqrt{2V([x]_\tau)}\overline{\epsilon}' + \overline{\epsilon})\frac{\tau}{2} = \mathcal{L}_V \frac{\tau}{2}. \tag{43}$$

Hence $\dot{V}(x) + \beta(x) - (x - x^*)^T \epsilon(x) \leq \dot{V}([x]_\tau) + \beta([x]_\tau) - ([x]_\tau - x^*)^T \epsilon([x]_\tau) + \mathcal{L}_V \frac{\tau}{2} \leq 0$, which after rearrangement the constraint in (36) is obtained. ■

We can formulate the Lyapunov constraint in (36) into the QP in (10) and (11) to solve for the parameters of the ELM. A relaxation variable δ can also be introduced in the Lyapunov constraint (30) to maintain the feasibility of the QP by softening the Lyapunov constraint. The following barrier and Lyapunov function-based quadratic program (BFLF-QP) is formulated as follows:

BFLF-QP

$$\begin{aligned}
&[W^{*^T},\ \delta^*]^T = \arg \min_{(W,\ \delta)\ \in\ \mathbb{R}^{n_h+1}} E_D \\
&\text{s.t.}\quad [\mathcal{C}_B]_\tau, \qquad \text{(Safety-Constraint)} \\
&\qquad\ \ [\mathcal{C}_L]_\tau, \qquad \text{(Stability-Constraint)},
\end{aligned} \tag{44}$$

where

$$E_D = \sum_{t=0}^{T_n} [\dot{x}(t) - \hat{\dot{x}}(t)]^T [\dot{x}(t) - \hat{\dot{x}}(t)] + \mu_W (tr(W^T W)) + p\delta^2 \tag{45}$$

$$[\mathcal{C}_{\mathcal{B}}]_\tau = -J_x([x]_\tau)\Sigma\theta - \gamma h^3([x]_\tau) \leq -h^2([x]_\tau)\mathcal{E} \quad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{46}$$

$$[\mathcal{C}_L]_\tau = ([x]_\tau - x^*)^T \Sigma\theta \leq -\beta([x]_\tau) - \mathcal{L}_V \frac{\tau}{2} + \delta, \quad \forall [x]_\tau \in \mathcal{X}_\tau, \tag{47}$$

where p is a positive constant.

Remark 5. The unconstrained learning for ELM can be used to obtain $\bar{\epsilon}$ and $\bar{\epsilon}'$ to be used in constraints (46) and (47).

The following Lemma shows that the trajectory of the system (8) generated using W^* in (44) is forward invariant with respect to the safe set $\mathcal{X}_0$, and it remains UUB in the presence of an external disturbance. Consider the perturbed system of (8)

$$\dot{x} = f(x(t)) + d(x(t)), \tag{48}$$

where $d(x)$ is the deterministic disturbance, and it is assumed to be bounded, that is, $\|d(x)\| \leq \bar{d}$.

Lemma 4. *(Robustness analysis) The trajectories of the system in (48) are robust to external disturbances $d(x)$ such that for a bounded disturbance $\|d(x)\| \leq \bar{d}$, the following inequality holds $\|x - x^*\| \leq \frac{\bar{\epsilon}+\bar{d}}{\rho}$, $\forall t > t_0 + T$, which $T \geq 0$.*

Proof. Since the reciprocal BF formulation is used in this article, the application of perturbation/disturbances to the system dynamics in (8) will only keep the trajectories of (8) inside the set $\mathcal{X}_0$ if the perturbation/disturbance is small enough such that the trajectories do not leave the set $\mathcal{X}_0$.[37]

Differentiating the Lyapunov candidate function $V(x)$ selected in the proof of Lemma 3 along the solution of (48) gives

$$\dot{V} = (x - x^*)^T(\Sigma\theta + \epsilon(x)) + (x - x^*)^T d(x), \tag{49}$$

which can be utilized to rewrite the constraint in (30) as follows

$$\dot{V}(x) \leq -\rho\|x - x^*\|^2 + \|x - x^*\|(\epsilon(x) + d(x)). \tag{50}$$

After upper bounding and completing the squares, (50) can be further simplified, and it is given by

$$\dot{V}(x) \leq -\rho V(x) + \frac{(\bar{\epsilon} + \bar{d})^2}{2\rho}. \tag{51}$$

Thus, invoking theorem 4.18 in Reference 36 the ultimate bound is given by $\|x - x^*\| \leq \frac{\bar{\epsilon}+\bar{d}}{\rho}$, $\forall t > t_0 + T$, and $T \geq 0$, which implies that the state trajectory of (8) in the presence of an external disturbance remains UUB. The provided bound on the error between the system trajectories and the equilibrium point is a measure of the error of the reproduced trajectories as a function of the external disturbance acting on the system. The parameter ρ can be selected to reduce the bound on the error. ■

3.2.3 | Encoding safety and stability constraints

Consider the BFLF-QP developed in (45), the objective is to minimize the sum of squared error and of an relaxation variable δ, while satisfying the $[\mathcal{C}_{\mathcal{B}}]_\tau$ and $[\mathcal{C}_L]_\tau$ constraints in (46) and (47), respectively. The $[\mathcal{C}_{\mathcal{B}}]_\tau$ constraint guarantees to keep the system trajectories invariant with respect to the safe set $\mathcal{X}_0$, and the $[\mathcal{C}_L]_\tau$ constraint satisfies the stabilization objective. While the result of Theorem 1 enables the constraints in (46) and (47) to be evaluated at a

Algorithm 1. Active sampling strategy for the computation of the barrier and Lyapunov constraints

1. Obtain the discretized region $\mathcal{X}_0$ using a uniform sampling strategy, an empty sample pool set $\hat{\Psi}$, and an initial guess of $\hat{W}_0$ using standard learning scheme for ELM.
2. Draw N_s random samples without replacement from the uniformly distributed points inside the invariant region $\mathcal{X}_\tau$ to be stored inside a new set Ψ.
3. Use $\hat{W}_0$ to compute the barrier and Lyapunov constraints in (46) and (47) on the points inside the set Ψ.
4. Sort constraints that violate $[\mathcal{C}_\mathcal{B}]_\tau$ and $[\mathcal{C}_L]_\tau$ based on their values.
5. Select top $N = p \cdot N_s$ violators from the set Ψ such that $p \leq \frac{N}{N_s}$; and if $p>\frac{N}{N_s}$ then resample the set Ψ and follow the procedure again.
6. Add a few more points from the area close to the boundary set $\partial\mathcal{X}_\tau$ to make sure enough points are selected for the computation of the constraints.

finite number of points, the number of sampling points can still be very large and grows even larger with respect to the dimension of the state space. To further reduce the sampling space and overcome the challenge of validating the BF and Lyapunov constraints on a large number of points, an active sampling strategy is introduced in the following section.

3.2.4 | Active sampling strategy

This section describes an active sampling strategy to sample efficiently from the discretized constraint set $\mathcal{X}_\tau$. Given the sample trajectory data as well as the discretized region, in the first step the standard learning scheme for ELM, namely, ridge regression is used without any constraints to learn the parameters of the network. Next, N_s random samples $\Psi = \{\psi_1, \ldots, \psi_{N_s}\} \subset \mathcal{X}_\tau$ are drawn at random from the uniformly distributed points inside the invariant region $\mathcal{X}_\tau$ to verify whether the selected points violate either or both of the constraints in (46) and (47). The violators are then added to the set $\hat{\Psi}$ given by

$$\hat{\Psi} = ([\overline{\mathcal{C}}_\mathcal{B}]_\tau \cup [\overline{\mathcal{C}}_L]_\tau), \tag{52}$$

where $[\overline{\mathcal{C}}_\mathcal{B}]_\tau$ and $[\overline{\mathcal{C}}_L]_\tau$ are the complements of the sets $[\mathcal{C}_\mathcal{B}]_\tau$ and $[\mathcal{C}_L]_\tau$, respectively. Note that initially the sample pool $\hat{\Psi} = \{\emptyset\}$ is an empty set.

The choice of $\hat{\Psi}$ stems from the fact that if one were to solve the BFLF-QP in (45) analytically using, for example, the Lagrangian relaxation method with Karush–Kuhn–Tucker multipliers,[33] there would be four different cases to consider depending on the activation of each constraint. The four cases are (1) when both constraints are inactive, (2, 3) when only one of the two constraints is active, and (4) when both constraints are active. The set $\hat{\Psi}$ in (52) is a complement of a set whose element jointly satisfies the two constraints in (46) and (47). Therefore, selecting points from $\hat{\Psi}$ implies that the last three cases satisfied. Note that the first case is satisfied when the set is empty, which is already considered when an initial estimate of the ELM parameters is obtained. The termination criterion of the sampling procedure is obtained using the sorted lists of $[\overline{\mathcal{C}}_\mathcal{B}]_\tau$ and $[\overline{\mathcal{C}}_L]_\tau$ based on their quantitative values such that p percent or less of N_s are selected. For example, let $N = N_\mathcal{B} + N_L - N_{[\overline{\mathcal{C}}_{BL}]_\tau}$ be the violators from the set Ψ, where $N_\mathcal{B}$, N_L, are the top suitably selected points from $[\overline{\mathcal{C}}_\mathcal{B}]_\tau$ and $[\overline{\mathcal{C}}_L]_\tau$, respectively. Moreover, $N_{[\overline{\mathcal{C}}_{BL}]_\tau}$ denotes the points that are accounted in both sets $[\overline{\mathcal{C}}_\mathcal{B}]_\tau$ and $[\overline{\mathcal{C}}_L]_\tau$, that is, $[\overline{\mathcal{C}}_\mathcal{B}]_\tau \cap [\overline{\mathcal{C}}_L]_\tau$. If $p > \frac{N}{N_s}$, then the algorithm starts over from resampling the set Ψ and follow the procedure again.

Finally, a few more points from the area close to the boundary set $\partial\mathcal{X}_\tau$ are added to ensure that enough points close to the boundary are selected to compute the constraints. The algorithmic view of the active sampling strategy is given in Algorithm 1.

Using the active sampling strategy in Algorithm 1, the modified optimization problem is given by

BFLF-QP

$$[W^{*^T},\ \delta^*]^T = \arg \min_{(W,\ \delta)\ \in\ \mathbb{R}^{n_h+1}} E_D$$
$$\text{s.t.} \quad \hat{\Psi}. \qquad \text{(Safety and Stability Constraints).} \qquad (53)$$

4 | EXPERIMENTAL VALIDATION

In this section, first the proposed method is evaluated in simulated settings and then the effectiveness of the method using a Baxter robot is demonstrated.

4.1 | Simulation

The experiments are conducted using a MacBook Pro with an Intel i7 processor and 16 Gigabytes of memory. The method is coded using MATLAB 2018b, and it is tested on the shapes from the LASA human handwriting dataset.[14] The chosen dataset consists of 7 demonstrations where the handwriting motions were collected from a pen input using a Tablet PC. The estimates of the slopes and biases are obtained using the BIP algorithm presented in Reference 34. In the simulation, MATLAB is used to solve QP problem with linear constraints. The design parameters of the QP problem in (53) are selected as follows: $\gamma = 10$ $\rho = 0.01$, $\tau = 1$, and $\mu_W = 0.01$. The upper bounds $\overline{\epsilon}$ and $\overline{\epsilon}'$ are selected empirically.

4.1.1 | Evaluation of model with safety constraints

The objective is to design a barrier constraint to be enforced while learning the parameters, that is, the output weights of the ELM, in order to guarantee the solutions of the underlying DS stay within the selected barrier certificate. The barrier certificate is selected such that it encloses all the demonstrations. Two shapes for the barrier certificate are chosen, an ellipse and a circle. Once the model is learned, the parameters are used to forward propagate the dynamics from various initial conditions inside and outside the invariant set. Figure 3(A,B) illustrate reproduction of trajectories using a model that uses only reciprocal BF $B(x) = \frac{1}{h(x)}$. As it can be seen from Figure 3(A,B), trajectories that are initialized outside the safe region do not reenter the barrier region and those trajectories that are initialized inside the barrier region remain inside the invariant set. From Figure 3(A,B), it is seen that the trajectories initialized just outside the barrier region and neighboring initial points just inside the barrier region have completely different behavior which is to be expected from enforcing the barrier constraints in the learning process.

To showcase the significance of using a sampling strategy in order to achieve efficiency in learning, the active sampling method described in Section 3.2.4 is implemented on another set of experiments with a similar objective. That is, to learn a nonlinear function $f(x)$ subject to the barrier constraints such that the solution trajectories of DS in (1) are invariant in a closed set. Figure 5 shows that boundary of the invariant regions are effectively blocking any solution trajectories to crossover, while only a few samples are selected for learning. The results of the experiments for training ELM networks with and without the active sampling method are computed over five independent runs and the average of each measure, that is, the CPU time and the number of sampling points, are summarized in Table 1. As it can be seen from the table, when the active sampling method is used the CPU time is significantly reduced for both shapes. Therefore, the claim that a fewer number of points translate to a fewer number of constraints, and thus faster convergence to a minimum point in terms of the computation time, is verified by the results. The average of CPU time for the optimization problem computed over 5 different runs is reported in Table 1. To illustrate the robustness of the learned model, a perturbation is applied to the position updates at random time instance t with amplitude a and direction v according to the systematic

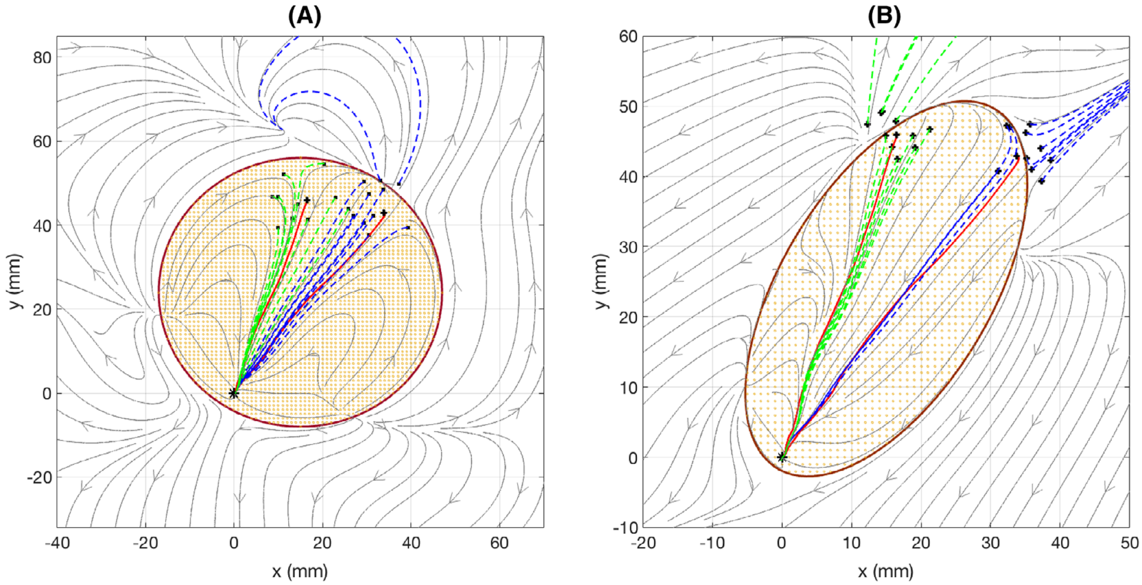

FIGURE 3 The streamlines of the solutions of dynamical system models (in gray) are shown along with the demonstrated data (in solid red lines) and the reproductions (in dashed blue and green lines) for circular barrier function (A) and elliptical barrier function (B) [Colour figure can be viewed at wileyonlinelibrary.com]

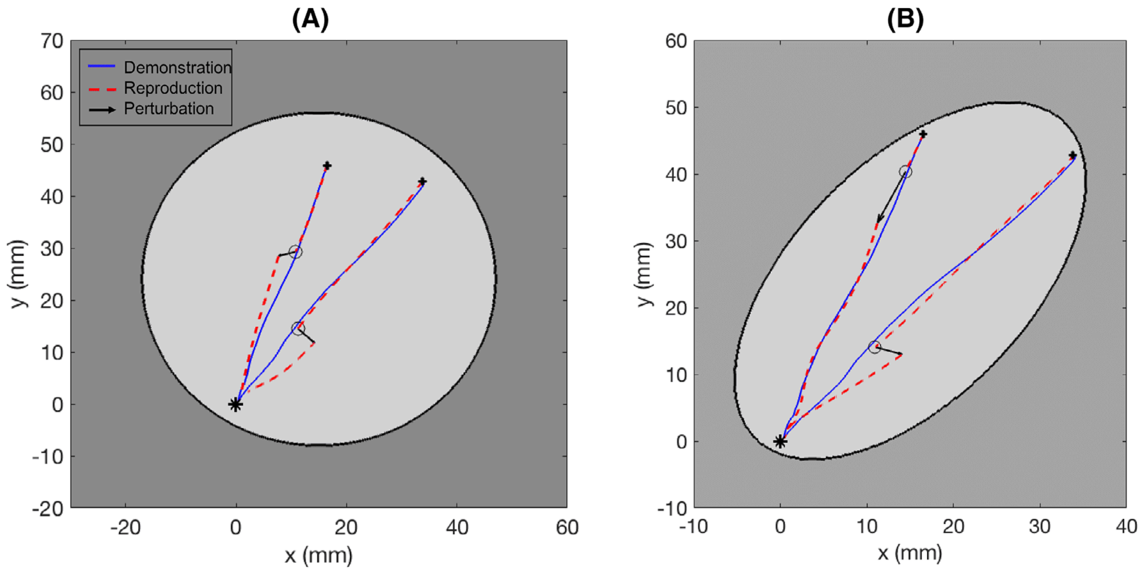

FIGURE 4 Ellipse and circle are used as smooth functions (black) to construct a barrier certificate that splits the state space into safe region (light gray) and unsafe region (dark gray). Illustration of the learned model's ability to handle perturbations [Colour figure can be viewed at wileyonlinelibrary.com]

method described in Reference 48. Figure 4 shows the results of the sudden perturbation experiment during the motion reproduction. Note that the black arrows in Figure 4 shows the direction and amplitude of the applied perturbation.

4.1.2 | Evaluation of model with safety and stability constraints

In learning complex motion dynamics, it is essential to provide stability and safety guarantees on the learned model. In other words, it is important to ensure that the motion reproductions converge to the desired goal location while remaining

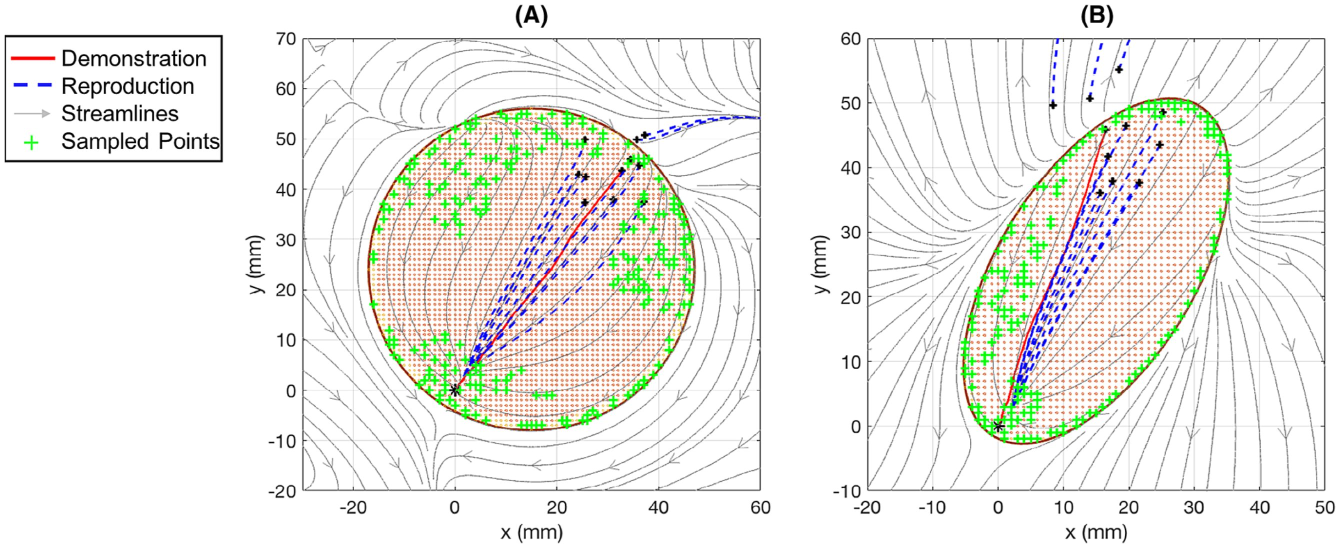

FIGURE 5 Illustration of the model learning using active sampling strategy that selected most informative points (green dots) for training [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Network learning with and without active sampling

	Without active sampling		**With active sampling**	
	No. of points used for learning	**CPU time (s)**	**No. of points used for learning**	**CPU time (s)**
Circle	3209	192.38	519	3.40
Ellipse	1411	109.03	441	3.63

inside a predefined region known to be a safe operating zone for robots. Therefore, the objective in this section is to design a barrier constraint to be enforced while learning the parameters of the ELM, in order to guarantee the solutions of the underlying DS stay within the selected barrier certificate while maintaining the shape of the demonstrations and converging to the target location.

Similar to the previous section, ellipse and circle shapes are used to express the safe regions, and reciprocal BF $B(x) = \frac{1}{h(x)}$ is used to enforce the positive forward invariance of the selected regions. In Figure 6, exemplary complex motions *"Snake"*, *"Sshape"*, and *"WShape"* from the LASA handwriting dataset are used to show that models that are learned solely with stability constraints are only able to converge to the target location without any safety guarantees. As can be seen, for models in Figure 6 the safety of the DS with respect to the ellipse or circle are not preserved, and clearly, the vector field (gray streamlines) crosses the boundary. On the other hand, models that are learned with both safety and stability constraints, the reproductions are guaranteed to not only converge to the target location but also remain inside the predefined safe region (ellipse and circle) for all time. Figure 7 shows models of a few complex shapes from the dataset, learned with both barrier and Lyapunov constraints. The streamlines of the models are shown to remain inside the safe regions while converging to the target location, and the streamlines that are outside of the safe region will bounce off the virtual wall (barrier) if they get close. The models are trained with the most informative points selected using the active sampling strategy given in Algorithm 1.

4.2 | Robot implementation

To demonstrate the utility of the proposed method in robot motion generation, four robot experiments are carried out. All experiments involved the use of the ELM, trained on the demonstrations of the "Line" and "Angle" shapes, to generate

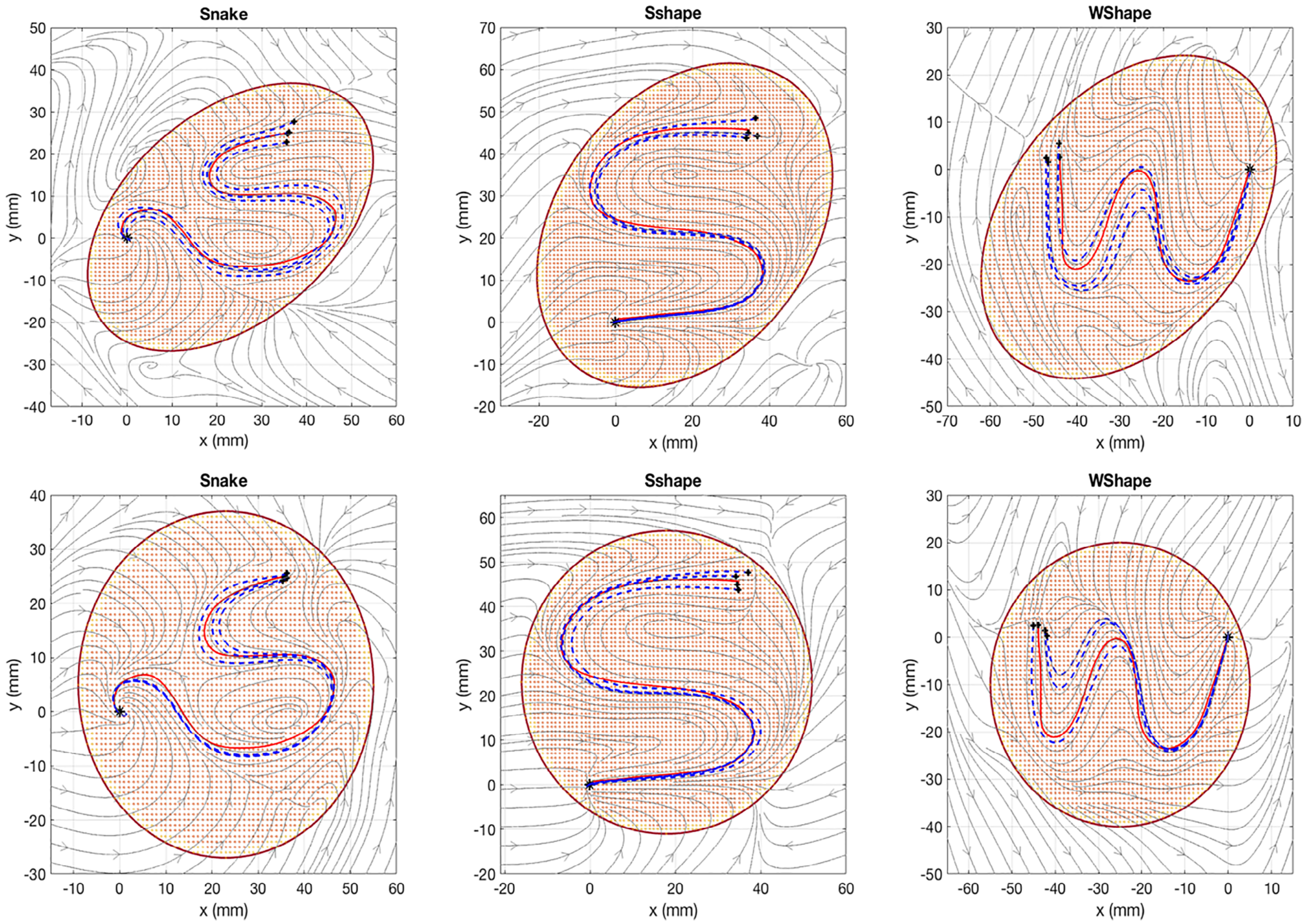

FIGURE 6 Models of *Snake*, *Sshape*, and *WShape* from the LASA handwriting dataset, learned only with Lyapunov stability constraints. The streamlines of the models (in gray) are shown along with the demonstrated data (in solid red lines) and the reproductions (in broken blue lines) [Colour figure can be viewed at wileyonlinelibrary.com]

reference trajectories for a seven degree-of-freedom Baxter research robot to reproduce the demonstration. The demonstrations are in two dimensions x and y with a fixed height z, which are in Cartesian space. The built-in low-level controller and inverse kinematics engine, IKFast[49] on Baxter robot are used to convert the demonstrations from Cartesian space to joint space suitable to execute the reproduced trajectories.

The first experiment is conducted to show that the learned model is capable of generating trajectories inside the invariant set from different initial conditions. Note that the experiments are carried out for a longer duration of time than the original demonstrations to show that the trajectories always remain inside the safe region. In every trial the robot successfully remained inside the safe region location. An example sequence of images showing the Baxter robot moving along the reproduced waypoints inside the safe region without hitting the ellipse shape boundary is shown in Figure 8.

In the second robot experiment, the robustness of the learned model to sudden perturbation is tested to support the claim made in Lemma 4. The perturbations were chosen according to the guidelines in Reference 48 and were applied using the control software. To implement the spatial perturbations on the robot, sudden perturbations are introduced during reproduction at random instances by a human operator.

In the third and fourth experiments, a more realistic application of the proposed method is tested. A round table was chosen as a desired working space for Baxter robot to perform a simple pick-and-place task while following the "Angle" shape trajectory and converge to the target location. Figure 9 shows sequence of images where Baxter robot is following a trajectory generated from a model that uses merely stability constraints. As it can be seen in Figure 9, it is very likely that the robot's end-effector leaves the safe zone (its working space) particularly for the cases where the motion starts at a different initial condition other than the one used for training. On the other hand, if the model is trained with both safety and stability considerations, as it is seen in Figure 10, the robot's end effector remains inside the working space regardless

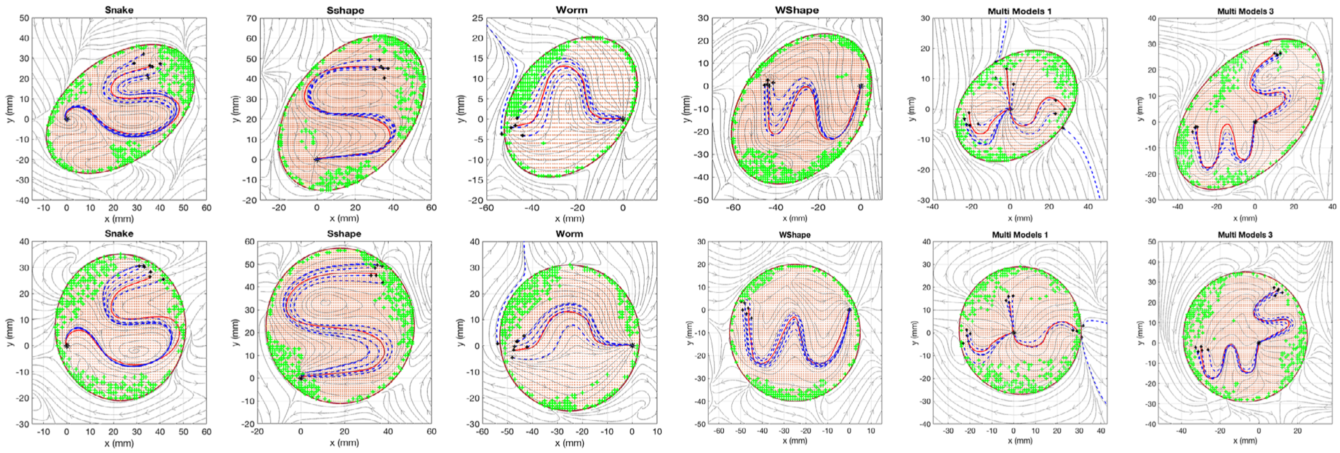

FIGURE 7 Models of *Snake*, *Sshape*, *Worm*,*WShape, MultiModels_1,* and *MutltiModels_3* from the LASA handwriting dataset, learned with both barrier and Lyapunov constraints. The streamlines of the models (in gray) are shown along with the demonstrated data (in solid red lines) and the reproductions (in broken blue lines). Green dots are the selected points for training the models using the active sampling strategy [Colour figure can be viewed at wileyonlinelibrary.com]

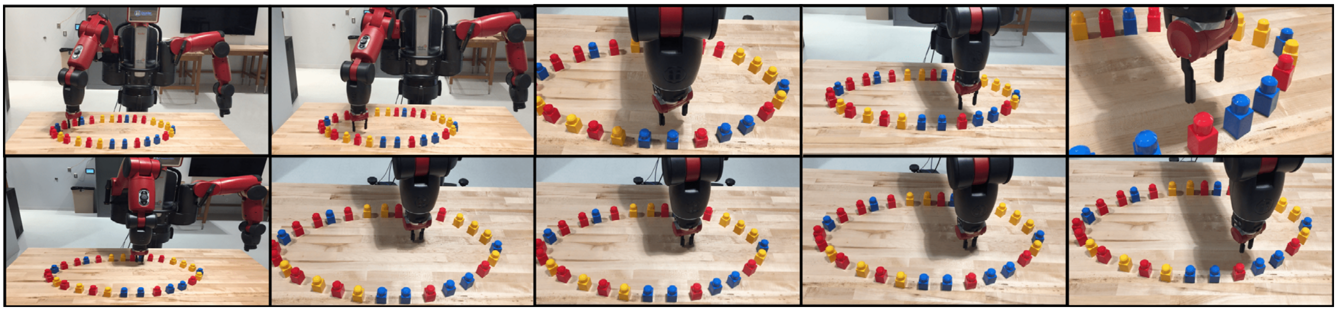

FIGURE 8 Sequence of images show Baxter robot moving along the *Line* shape without hitting the ellipse boundary (top row). The bottom row of images is showing Baxter robot moving along the *Line* shape but starting at a different initial condition without hitting the obstacles kept on the boundary [Colour figure can be viewed at wileyonlinelibrary.com]

of its initial condition. Finally, Figure 11 shows the results of the Baxter implementation of the learning DSs algorithm that keeps the end effector generated trajectories in a confined space and converges to the desired target location. Moreover, the result of the learning algorithm that only ensures stability in terms of convergence to the target location is overlaid on the same figure.

5 | CONCLUSION AND FUTURE WORK

A learning method using ELM algorithm with barrier and Lyapunov constraints is presented. To ensure that the trajectories generated by the learned model remain inside a safe region, and at the same time motion reproductions converge to the desired goal, BF constraints along with constraints derived using Lyapunov analysis are enforced while learning the parameters of the model. Analysis of the results in Section 4 reveals that enforcing the constraints discussed in Section 3.2 while learning the parameters of the model guarantees that the approximated solutions of the DSs remain inside the safe region and converge to the target location. The former statement is true as long as the initial condition belongs to the forward invariant set, which is defined by the BF to be enforced during the optimization. Section 4.2 validates the proposed method's ability to learn models from demonstrations and to provide guarantees such that the learned model remain inside the safe region defined by the BF. Future work will explore the use of the algorithm presented in this article for the obstacle avoidance in the trajectory learning methods.

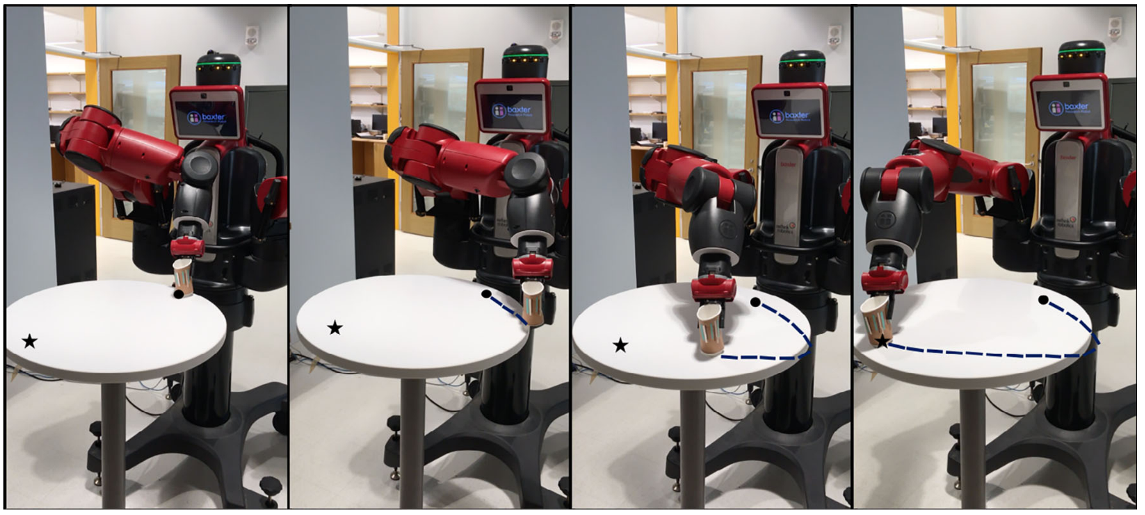

FIGURE 9 Sequence of images show that Baxter robot performing a simple pick-and-place task while moving along the *Angle* shape trajectory to converge to the target location (black star). While stability constraints ensure the robot end effector converges to the goal location during reproductions (dashed blue line), depending on the initial condition (black circle), it is very likely that the robot end effector goes outside of the working space when the model is trained without any safety constraints [Colour figure can be viewed at wileyonlinelibrary.com]

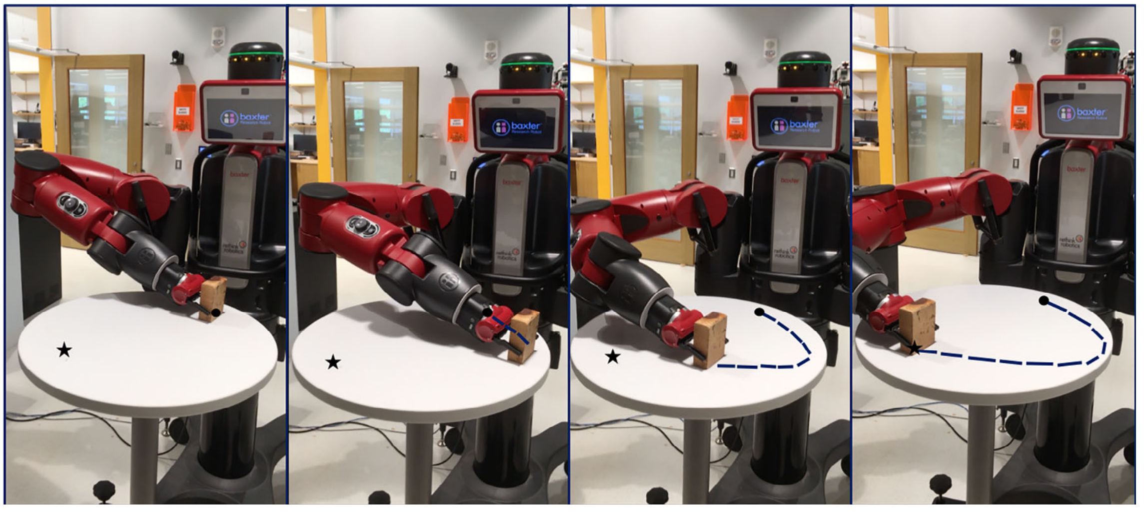

FIGURE 10 Sequence of images show the robot's end effector converges to the target location (black star) from any initial condition (black circle) inside its working space when the model is trained both with safety and stability constraints. The reproduction is shown in a blue dashed line [Colour figure can be viewed at wileyonlinelibrary.com]

ACKNOWLEDGMENTS

This work was in part by a Space Technology Research Institutes grant (number 80NSSC19K1076) from NASA Space Technology Research Grants Program, in part by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007613, and in part supported by the Subaward No. ARM-17-QS-F-04 from the Advanced Robotics for Manufacturing ("ARM") Institute under Agreement Number W911NF-17-3-0004 sponsored by the Office of the Secretary of Defense.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

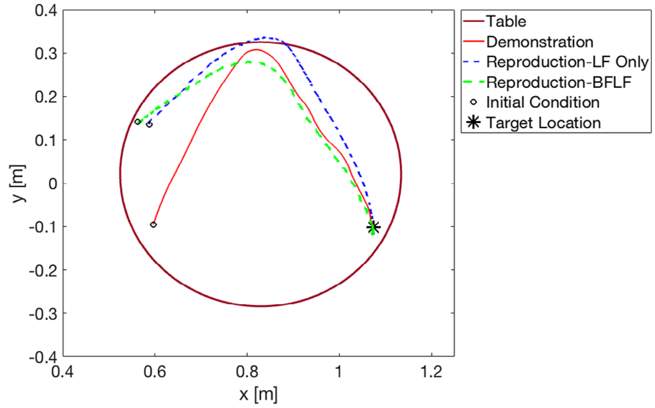

FIGURE 11 Baxter implementation results showing reproductions using a model trained with stability and safety constraints along with a model trained merely with a stability constraint [Colour figure can be viewed at wileyonlinelibrary.com]

ORCID

Iman Salehi https://orcid.org/0000-0002-9144-3573
Ashwin P. Dani https://orcid.org/0000-0002-7091-5607

REFERENCES

1. Giusti A, Zeestraten MJA, Icer E, et al. Flexible automation driven by demonstration: leveraging strategies that simplify robotics. *IEEE Robot Automat Mag*. 2018;25(2):18-27.
2. McLachlan G, Peel D. Mixtures of factor analyzers. *Finite Mixture Models*. New York: John Wiley & Sons; 2000;238-256.
3. Rasmussen CE. Gaussian Processes for Machine Learning. *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, T{\"u}bingen, Germany, August 4 - 16, 2003, Revised Lectures*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004:63–71.
4. Bishop CM. *Neural Networks for Pattern Recognition*. New York: Oxford University Press; 1995.
5. Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feedforward neural networks. Paper presented at: Proceedings 2004 IEEE International Joint Conference on Neural Networks 2004, Budapest, Hungary; vol. 2, 2004:985-990.
6. Ljung L. System Identification. *Wiley Encyclopedia of Electrical and Electronics Engineering*; 1999:1-19.
7. Goodwin GC, Payne RL. *Dynamic System Identification: Experiment Design and Data Analysis*. Mathematics in Science and Engineering. New York, NY: Academic Press; 1977;136.
8. Ghahramani Z, Roweis ST. Learning Nonlinear Dynamical Systems Using an EM Algorithm. *Advances in Neural Information Processing Systems*. 11. Cambridge, MA: MIT Press; 1999;431-437.
9. Pillonetto G, Dinuzzo F, Chen T, De Nicolao G, Ljung L. Kernel methods in system identification, machine learning and function estimation: a survey. *Automatica*. 2014;50(3):657-682.
10. Parikh A, Kamalapurkar R, Dixon WE. Integral concurrent learning: adaptive control with parameter convergence using finite excitation. *Int J Adapt Control Signal Process*. 2019;33(12):1775-1787.
11. Schaal S. Is imitation learning the route to humanoid robots? *Trends Cogn Sci*. 1999;3(6):233-242.
12. Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput*. 2013;25(2):328-373.
13. Rai A, Meier F, Ijspeert A, Schaal S. "Learning coupling terms for obstacle avoidance," 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain; 2014:512-518.
14. Khansari-Zadeh SM, Billard A. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Trans Robot*. 2011;27(5):943-957.
15. Khansari-Zadeh SM, Billard A. Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robot Auton Syst*. 2014;62(6):752-765.
16. Neumann K, Steil JJ. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robot Auton Syst*. 2015;70:1-15.
17. Ravichandar H, Salehi I, Dani A. "Learning Partially Contracting Dynamical Systems from Demonstrations," Proceedings of the 1st Annual Conference on Robot Learning, PMLR, Mountain View, CA; 2017:369-378.
18. Lemme A, Neumann K, Reinhart F, Steil JJ. "Neurally imprinted stable vector fields," 21st European Symposium on Artificial Neural Networks, Bruges, Belgium; 2013:327-332.
19. Ames AD, Xu X, Grizzle JW, Tabuada P. Control barrier function based quadratic programs for safety critical systems. *IEEE Trans Autom Control*. 2017;62(8):3861-3876.
20. Berkenkamp F, Moriconi R, Schoellig AP, Krause A. "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV; 2016:4661-4666.

21. Srinivasan M, Dabholkar A, Coogan S, Vela P. "Synthesis of Control Barrier Functions Using a Supervised Machine Learning Approach," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV; 2020;7139-7145.
22. Berkenkamp F, Turchetta M, Schoellig A, Krause A. Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems*. 2017;2:908-919.
23. Yang Y, Yin Y, He W, Vamvoudakis KG, Modares H, Wunsch DC. "Safety-Aware Reinforcement Learning Framework with an Actor-Critic-Barrier Structure," American Control Conference (ACC), Philadelphia, PA; 2019:2352-2358.
24. Stephen P, Jadbabaie A. Safety verification of hybrid systems using barrier certificates. International Workshop on Hybrid Systems: Computation and Control, Springer, Berlin, Heidelberg; 2004:477-492.
25. Wieland P, Allgöwer F. Constructive safety using control barrier functions. *IFAC Proc Vols*. 2007;40(12):462-467.
26. Nguyen Q, Hereid A, Grizzle JW, Ames AD, Sreenath K. "3D dynamic walking on stepping stones with control barrier functions," IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV; 2016:827-834.
27. Wu G, Sreenath K. "Safety-critical control of a planar quadrotor," American Control Conference (ACC), Boston, MA; 2016:2252-2258.
28. Wang L, Ames AD, Egerstedt M. "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, 2016:2659-2664.
29. Wang L, Ames AD, Egerstedt M. "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017:3293-3298.
30. Greene ML, Deptula P, Nivison S, Dixon WE. Sparse learning-based approximate dynamic programming with barrier constraints. *IEEE Control Syst Lett*. 2020;4(3):743-748.
31. Wang L, Theodorou EA, Egerstedt M. "Safe Learning of Quadrotor Dynamics Using Barrier Certificates," IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia; 2018;2460-2465.
32. Farras AW, Yamauchi J, Hatanaka T, Fujita M. "Safe Cooperative Control of Human-Robotic Network Teaming with Control Barrier Function,". SICE International Symposium on Control Systems (SICE ISCS), Tokushima, Japan; 2020:33-39.
33. Bertsekas D. *Nonlinear Programming*. Athena Scientific Optimization and Computation Series Athena Scientific. 2016.
34. Neumann K, Steil JJ. Optimizing extreme learning machines via ridge regression and batch intrinsic plasticity. *Neurocomputing*. 2013;102:23-30.
35. Salehi I, Yao G, P. Dani "Active sampling based safe identification of dynamical systems using extreme learning machines and barrier certificates," International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada; 2019;22-28.
36. Khalil HK. *Nonlinear Systems*. 3rd ed. Upper Saddle River, NJ: Prentice Hall; 2002.
37. Xu X, Tabuada P, Grizzle JW, Ames AD. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*. 2015;48(27):54-61.
38. Dai L, Gan T, Xia B, Zhan N. Barrier certificates revisited. *J Symb Comput*. 2017;80:62-86.
39. Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. *Neurocomputing*. 2006;70(1-3):489-501.
40. Triesch J. "A gradient rule for the plasticity of a Neuron's intrinsic excitability," International Conference on Artificial Neural Networks, Springer, Berlin, Heidelberg; 2005:65-70.
41. Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Netw*. 1989;2(5):359-366.
42. Huang GB, Chen L, Siew CK. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans Neural Netw*. 2006;17(4):879-892.
43. Kosmatopoulos EB, Polycarpou MM, Christodoulou MA, Ioannou PA. High-order neural network structures for identification of dynamical systems. *IEEE Trans Neural Netw*. 1995;6(2):422-431.
44. Lewis FL, Campos J, Selmic R. *Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities*. Philadelphia, PA: ; Society for Industrial and Applied Mathematics; 2002.: 1–244.
45. Lewis FL, Liu K, Yesildirek A. Neural net robot controller with guaranteed tracking performance. *IEEE Trans Neural Netw*. 1995;6(3):703-715.
46. Govindarajulu Z. *Elements of Sampling Theory and Methods*. Upper Saddle River, NJ: Prentice Hall; 1999.
47. Cochran WG. *Sampling Techniques*. Hoboken, NJ: John Wiley & Sons; 2007.
48. Lemme A, Meirovitch Y, Khansari-Zadeh SM, Flash T, Billard A, Steil JJ. Open-source benchmarking for learned reaching motion generation in robotics. *Paladyn J Behav Robot*. 2015;6:30-41.
49. Diankov R. *Automated Construction of Robotic Manipulation Programs* [Ph.D thesis]. Pittsburgh, PA: Carnegie Mellon University; 2010.

How to cite this article: Salehi I, Rotithor G, Yao G, Dani AP. Dynamical system learning using extreme learning machines with safety and stability guarantees. *Int J Adapt Control Signal Process*. 2021;35:894–914. https://doi.org/10.1002/acs.3237