Formal Synthesis of Barrier Certificates Using Fourier Kolmogorov-Arnold Network

Xiongqi Zhang, Junwei Xu, Yang Wang, Dongming Xiang, Wang Lin*, Zuohua Ding

School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China
{xqiii_zhang, xjwblackaw, wangy4303}@163.com, {dmxiang, linwang}@zstu.edu.cn, zouhuading@hotmail.com

Abstract

Barrier certificate generation is an efficient and powerful technique for formally verifying safety properties of cyber-physical systems. Feed-forward neural networks (FNNs) are commonly used to synthesize barrier certificates, but the fixed activation functions limit their efficiency and scalability. In this paper, we propose a novel method for generating barrier certificates using Fourier Kolmogorov-Arnold Networks (KANs). Specifically, it utilizes Fourier KANs to replace FNNs as the template of barrier certificates. Since Fourier KAN has learnable activation functions and uses trigonometric functions as its basis functions, it can efficiently improve the representation power and is easy to train for neural barrier certificates. Then, it formally verifies the validity of the candidate Fourier KAN barrier certificates using both the Lipschitz method and the Satisfiability Modulo Theories, improving the efficiency and success rate of verification. We implement the tool KAN4BC, and evaluate its performance over a set of benchmarks. The experimental results demonstrate the effectiveness and efficiency of our method.

Introduction

Ensuring the safety of dynamical systems, which are modeled by ordinary differential equations (ODEs), is crucial for various applications, including autonomous vehicles, robotics, aerospace engineering, and industrial process control. These systems form the backbone of cyber-physical systems (Squires, Pierpaoli, and Egerstedt 2018), featuring interactions between discrete and continuous dynamics. Formal safety verification, the process of determining whether a system can avoid reaching dangerous or undesirable states, is of paramount importance. However, this task is inherently complex and poses a significant challenge.

Barrier certificate generation is an effective and powerful technique for formally demonstrating the safety of continuous systems and hybrid systems that include both continuous and discrete states. When a barrier certificate is identified, it divides the model's state space into two regions. This division ensures that any system trajectory starting from a given initial set on one side of the barrier certificate cannot reach a given unsafe set on the other side. Therefore, successfully synthesizing a barrier certificate, which is generally not unique, serves as a formal certificate for safety properties of the dynamical system.

Sum-of-squares (SOS) programming is a traditional method for synthesizing polynomial barrier certificates (Prajna 2006; Legat, Tabuada, and Jungers 2020). But this method lacks scalability and expressiveness. Based on the universal approximation theorem (Leshno et al. 1993), neural networks have the capability to approximate any arbitrary function, making them suitable for representing barrier certificates. Zhao et al. (2020) first synthesized barrier certificates via neural network training and verification. FOSSIL (Abate et al. 2021) is a tool for the automated formal synthesis of barrier certificates, using neural networks as templates and Satisfiability Modulo Theories (SMT) solvers as verification tools. However, traditional feed-forward neural networks (FNNs) often use fixed activation functions, which can significantly impact the efficiency of synthesizing barrier certificates.

Liu et al. (2024) proposed Kolmogorov-Arnold Networks (KANs) as a promising alternative to multi-layer Perceptrons. In contrast to traditional models, KANs use activation functions on the connections between nodes, which can also learn and adapt during the training process. This architectural breakthrough enables KANs to better capture complex, nonlinear relationships by directly optimizing univariate functions. Initially, these networks used B-spline functions as learnable activation functions. However, due to difficulties in implementation and smoothness concerns, alternative basis functions have been explored. To address the issues of efficiency and scalability in generating barrier certificates caused by the use of fixed activation functions in FNNs, in this paper, we propose a new framework using Fourier KANs instead of FNNs to formally synthesize barrier certificates. Specifically, our proposed framework consists of two parts. In the first part, it uses Fourier KAN to generate candidate barrier certificates. Fourier KAN uses Fourier Coefficients as learnable activation functions, and uses trigonometric functions as basis functions that consist of sines and cosines, can improve the representation power and are easy to train. In the second part, it checks whether the candidate barrier certificates, represented by Fourier KAN, satisfy all the requirements of barrier certificates through SMT-solver dReal and Lipschitz method. The

*Corresponding authors.

Lipschitz method establishes a validity condition by leveraging the Lipschitz continuity of the network, formally ensuring the network's correctness, but its efficiency is severely limited by the size of the dataset, thus often leading to longer training times. The SMT solver-based verification involves a counterexample-guided Inductive Synthesis (CEGIS) procedure (Peruffo, Ahmed, and Abate 2021). The structure of Fourier KAN makes the SMT solver-based verification easier, but the CEGIS procedure does not have termination guarantees and may suffer from a low success rate of synthesizing truly barrier certificates. Therefore, we combine these two methods to improve the efficiency of the verification process. Overall, the main contributions of this work are described as follows:

- We introduce a new network called Fourier KAN to synthesize barrier certificates, which enhances representation power and makes training easier.
- We deploy both the Lipschitz method and the SMT solver as verification engines to overcome the limitations that may arise with either in certain situations, thereby improving the efficiency and success rate of verification.
- We implement our tool KAN4BC and conduct a detailed experimental evaluation on a set of benchmarks, demonstrating that our approach is more effective in generating barrier certificates than the state-of-the-art tool FOOSIL2.0.

Related Work

In the control field, extensive research has been conducted on the synthesis of barrier certificates over many years. Prajna and Jadbabaie (2004) generated barrier certificates using SOS and semidefinite programming. Kong et al. (2014) introduced a novel exponential condition barrier certificate that maintains convexity while reducing conservativeness, thereby providing a more precise and effective approach for safety verification of semialgebraic hybrid systems. Sloth, Pappas and Wisniewski (2012) presented a compositional approach to safety verification using barrier certificates, enabling the validation of higher-dimensional systems through coupled subproblems. Zeng et al. (2016) proposed a new verification condition and a novel computational method combining sampling-based relaxation with least-squares and quadratic programming(LS-QP) alternating projection to find Darboux-type barrier certificates, enhancing the verification of nonlinear hybrid systems. Platzer and Clarke (2008) developed a fixed-point algorithm for verifying hybrid systems with polynomial differential equations, using continuous induction, compositional logic, and a saturation procedure for efficient and robust verification. Sogokon et al. (2018) extended the concept of barrier certificates to a multi-dimensional framework, employing vector Lyapunov functions for safety verification and invariant synthesis in non-linear systems.

The emergence of neural networks has provided new possibilities for synthesizing barrier certificates. Researchers have explored learning neural network certificate functions and have achieved encouraging results (Zhang et al. 2023; Lindemann et al. 2021). However, to ensure deterministic guarantees, these methods must include a verification procedure to confirm the validity of the certificate. Some contributions have employed CEGIS procedure (Liu, Liu, and Dolan 2023), where a learner trains a neural network to meet barrier certificate requirements over a finite set of samples, and a verifier either proves validity of the barrier certificate or provides counterexamples using an SMT solver (Kapinski et al. 2014; Chang, Roohi, and Gao 2019). However, SMT-based verification becomes inefficient for high-dimensional systems. To address this issue, researchers have used the Lipschitz method (Chang, Roohi, and Gao 2019) to reduce extra verification time. By utilizing the powerful expressive capabilities of neural networks, these certificates can capture complex, non-linear safety constraints that traditional methods struggle to model. Neural barrier certificates represent a major advancement in control learning, offering robust safety assurances and improving the overall reliability of trained controllers. Additionally, in complex tasks, reinforcement learning algorithms are often employed to learn neural barrier certificates to achieve near-zero constraint violations within accurate environment models (Yang et al. 2023). There are still many new methods to be discovered for the synthesis of barrier certificates.

Preliminaries

This section introduces the basic concepts used in the paper.

Continuous Dynamical System

Consider a continuous dynamical system S:

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

where $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$ is a column vector representing the state of the system, $\dot{\mathbf{x}}$ denotes the derivative of $\mathbf{x}$ with respect to time t, and $\mathbf{f}(\mathbf{x}) : \Omega \to \mathbb{R}^n$ is a continuous vector field $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_n(\mathbf{x}))^T$ defined on an open subset $\Omega \subseteq \mathbb{R}^n$. We assume that $\mathbf{f}$ is Lipschitz continuous, which ensures the existence and uniqueness of solutions to the ODEs for a given initial state $\mathbf{x}_0$. Specifically, there exists a unique time trajectory $\mathbf{x}(t, \mathbf{x}_0)$ starting in $\mathbf{x}_0$ denotes the value of the state $\mathbf{x}$ at time t, where $t > 0$.

In this paper, we focus on the safety verification problem. Specially, given an initial state $\mathbf{x}(0) = \mathbf{x}_0 \in X_I$, where X_I represents the set of initial states, and an unsafe set $X_U \subseteq \mathbb{R}^n$, assuming that both X_I and X_U are subsets of the domain X_D. For safety verification, we consider if any trajectories starting from the initial region X_I can enter the unsafe region X_U.

Definition 1 (safety)**.** *For a constrained continuous dynamical system, a given initial region $X_I \subseteq X_D$ and a given unsafe region $X_U \subseteq X_D$, the system is safe if the following holds:*

$$\mathbf{x}(t, \mathbf{x}_0) \in X_D \Rightarrow \mathbf{x}(t, \mathbf{x}_0) \notin X_U, \forall \mathbf{x}_0 \in X_I, \forall t \geq 0. \tag{2}$$

Barrier Certificate

The barrier certificate plays a significant role in verifying the safety of the system. For a continuous dynamical system (1), given the sets X_D, X_U and X_I, if there exists a continuous real-valued function $B(\mathbf{x}) : X_D \to \mathbb{R}$ satisfying:

$$\begin{aligned} &B(\mathbf{x}) \leq 0 \quad \forall \mathbf{x} \in X_I \\ &B(\mathbf{x}) > 0 \quad \forall \mathbf{x} \in X_U \\ &\dot{B}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \in X_D \quad s.t. B(\mathbf{x}) = 0, \end{aligned} \tag{3}$$

then $B(\mathbf{x})$ is a barrier certificate, and the safety of the system is guaranteed. Here, $\dot{B}(\mathbf{x})$ is the Lie derivative of B with respect to a vector field $\mathbf{f}$, defined as follows:

$$\dot{B}(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial B}{\partial x_i} \frac{dx_i}{dt} = \sum_{i=1}^{n} \frac{\partial B}{\partial x_i} f_i(\mathbf{x}). \tag{4}$$

Consider a trajectory $\mathbf{x}(t, \mathbf{x}_0)$ and observe the evolution of $B(\mathbf{x}(t, \mathbf{x}_0))$ along this trajectory. The initial condition ensures $B(\mathbf{x}_0) \leq 0$, and the final condition requires $B(\mathbf{x}(t, \mathbf{x}_0))$ to decrease alone the trajectory $\mathbf{x}(t, \mathbf{x}_0)$. As a result, such a trajectory $\mathbf{x}(t, \mathbf{x}_0)$ is prevented from entering the unsafe region X_U, where $B(\mathbf{x}) > 0$, so the safety of the system is guaranteed.

Kolmogorov-Arnold Network (KAN)

Kolmogorov-Arnold Network (KAN) is a novel network architecture inspired by the Kolmogorov-Arnold representation theorem (Kolmogorov 1961).

Theorem 1 (Kolmogorov-Arnold representation theorem)**.** *For any continuous function f mapping from $[0, 1]^n$ to the real numbers $\mathbb{R}$, there exists a set of continuous functions $\phi_{i,j}$ (where i = 1, 2, . . . , 2n + 1 and j = 1, 2, . . . , n + 1) such that*

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{2n+1} \phi_i(\sum_{j=1}^{n} \phi_{i,j}(\mathbf{x}_j)). \tag{5}$$

According to Equation (5), KAN can be represented as a nested combination of two layers of univariate functions. In matrix form, KAN is defined as:

$$\text{KAN}(\mathbf{x}) = \Phi_{out} \circ \Phi_{in} \circ \mathbf{x}, \tag{6}$$

where Φ_{in} is a matrix composed of univariate functions, represented as:

$$\Phi_{in} = \begin{pmatrix} \phi_{1,1}(\cdot) & \cdots & \phi_{1,n}(\cdot) \\ \vdots & \ddots & \vdots \\ \phi_{2n+1,1}(\cdot) & \cdots & \phi_{2n+1,n}(\cdot) \end{pmatrix}. \tag{7}$$

and Φ_{out} is a row vector of univariate functions:

$$\Phi_{out} = (\phi_1(\cdot) \quad \ldots \quad \phi_{2n+1}(\cdot)). \tag{8}$$

So the KAN in Equation (6) are simply compositions of two KAN layers. In practical applications, we can achieve deeper KAN by simply stacking more KAN layers.

It is known that FNNs can be described as a sequence of alternating affine transformations $\mathbf{W}$ and fixed nonlinear functions σ:

$$\text{FNN}(\mathbf{x}) = (\mathbf{W}_{L-1} \circ \sigma \circ \cdots \circ \mathbf{W}_1 \circ \sigma \circ \mathbf{W}_0)\mathbf{x}. \tag{9}$$

Different from FNNs, KAN employs learnable activation functions $\phi(\mathbf{x})$ at the network's edges:

$$\phi(\mathbf{x}) = \mathbf{w}_b b(\mathbf{x}) + \mathbf{w}_s \text{spline}(\mathbf{x}), \tag{10}$$

which is the sum of the basis function $b(\mathbf{x})$:

$$b(\mathbf{x}) = \text{silu}(\mathbf{x}) = \frac{\mathbf{x}}{1 + e^{\mathbf{x}}}, \tag{11}$$

and the spline function $\text{spline}(\mathbf{x})$:

$$\text{spline}(\mathbf{x}) = \sum_i c_i B_i(\mathbf{x}), \tag{12}$$

where c_i is trainable, and $B_i(\mathbf{x})$ are B-spline basis functions, and $\mathbf{w}_b$ and $\mathbf{w}_s$ are trainable.

Problem Statement

Previous works for constructing barrier certificates often represent them in the form of FNNs, which require manual setting of activation functions. Additionally, different activation functions can have varying efficiencies in generating barrier certificates for different systems. Selecting an appropriate activation function for FNNs can be quite costly. Therefore, we attempt to replace FNNs with KANs, because they have learnable activation functions, which offer enhanced scalability and expressive power.

Thus, we consider the problem of synthesizing barrier certificates using an efficient KAN, called Fourier KAN.

Problem 1. *Given a continuous dynamical system S, we use Fourier KAN to generate a barrier certificate and verify whether it satisfies Equation* (3) *to ensure the safety of the system.*

Method

In this section, we introduce a novel framework for synthesizing Fourier KAN barrier certificates, which can guarantee the safety of the continuous dynamical systems. The complete process can be divided into the following two stages.

- We devise a Learner component to synthesize candidate barrier certificates. This component trains a Fourier KAN to fit on the sampled dataset and learns the network's activation functions by a special loss function.
- We design a Verifier component to formally verify the validity of the candidate Fourier KAN barrier certificates. This component includes two verification methods, the Lipschitz method and the SMT solver.

Network Architecture of Fourier KAN

The essence of KAN is to approximate arbitrary functions through the superposition of multiple nonlinear functions. Since B-spline functions are generated recursively, training KANs is more challenging than FNNs, making it difficult to achieve our goal of synthesizing barrier certificates efficiently. Additionally, spline functions cannot be formally expressed with explicit expressions, making verification challenging. Therefore, we replace B-spline functions in KAN with simpler and explicitly expressible Fourier Coefficients. The replaced activation function is represented as follows:

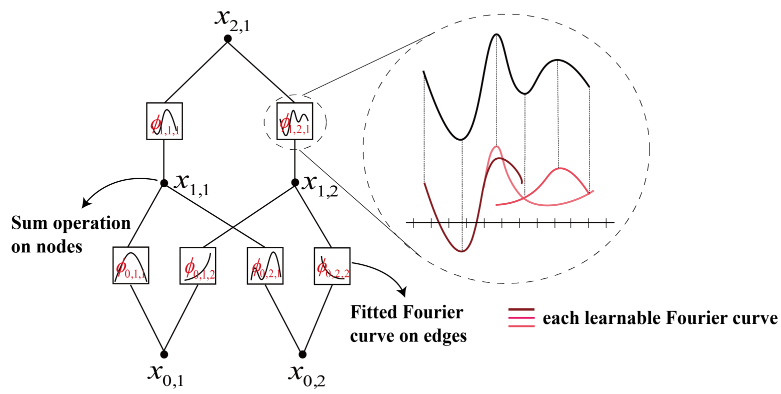

Figure 1: Left:A 2-Layer Fourier KAN with shape [2, 2, 1]. Right: an activation function is parameterized as a Fourier Curve.

$$\phi(\mathbf{x}) = \sum_{i=1}^{d}\sum_{k=1}^{g}(a_{ik}cos(k\mathbf{x}_i) + b_{ik}sin(k\mathbf{x}_i)), \quad (13)$$

where d is the dimension number of features, a_{ik} and b_{ik} are Fourier coefficients that can be trainable. The hyper-parameter g, known as the gridsize, is vital in influencing the number of frequencies in the Fourier series expansion. It specifically determines the variety of sine and cosine terms included in the Fourier Coefficients for each input dimension. The Fourier Coefficients play a crucial role in computational efficiency and address the training and verification challenges posed by spline functions.

The shape of a Fourier KAN can be represented as an integer array $[n_0, n_1, \cdots, n_L]$, where n_i denotes the number of nodes in the i-th layer. Let $x_{l,i}$ denote the i^{th} neuron in the l^{th} layer. There are $n_l n_{l+1}$ activation functions between the l-th layer and the $(l+1)$-th layer. Let $\phi_{l,i,j}$ denote the activation function that connects $x_{l,i}$ and $x_{l+1,j}$. The value of $x_{l+1,j}$ can be calculated as follows:

$$x_{l+1,j} = \sum_{i=1}^{n_l}\phi_{l,i,j}(x_{l,i}) \qquad j = 1, \cdots, n_{l+1}. \quad (14)$$

In matrix notation, this is expressed as:

$$\mathbf{x}_{l+1} = \begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,2,1}(\cdot) & \cdots & \phi_{l,n_l,1}(\cdot) \\ \phi_{l,1,2}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,n_l,2}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,1,n_{l+1}}(\cdot) & \phi_{l,2,n_{l+1}}(\cdot) & \cdots & \phi_{l,n_l,n_{l+1}}(\cdot) \end{pmatrix}\mathbf{x}_l. \quad (15)$$

Thus, a L-Layer Fourier KAN can be constructed as follows:

$$\text{Fourier KAN}(\mathbf{x}) = (\Phi_{L-1}\circ\Phi_{L-2}\circ\cdots\circ\Phi_1\circ\Phi_0)\mathbf{x}, \quad (16)$$

where Φ_l is the matrix of activation functions corresponding to the l^{th} Fourier KAN layer.

A simple Fourier KAN with two-dimensional input and one-dimensional output is illustrated in Figure 1, consists of two Fourier KAN layers where learnable activation functions are placed on edges rather than nodes. Assuming grid-size = 1, we take node $x_{1,1}$ as an example, the calculation formula for $x_{1,1}$ is as follows:

$$\begin{cases} \phi_{0,1,1}(x_{0,1}) = a_1\cos(x_{0,1}) + b_1\sin(x_{0,1}) \\ \phi_{0,2,1}(x_{0,2}) = a_2\cos(x_{0,2}) + b_2\sin(x_{0,2}) \\ x_{1,1} = \phi_{0,1,1}(x_{0,1}) + \phi_{0,2,1}(x_{0,2}) \end{cases} \quad (17)$$

where a_1, a_2, b_1, b_2 are Fourier coefficients.

By observing the network structure of Fourier KAN, the learnable activation function parameterized through Fourier coefficients enable Fourier KAN to represent complex functions with fewer parameters. Since Fourier KAN uses sine and cosine functions as basis functions, these functions naturally exhibit smoothness, so the training speed is faster compared to the original KAN. The number of sine and cosine functions can be controlled through the hyper-parameter gridsize. This not only reduces the model's complexity but also improves computational efficiency. Fourier KAN has an advantage in versatility because it can automatically learn activation functions to adapt to different systems.

Therefore, it is reasonable to use Fourier KAN as the network for synthesizing barrier certificates. For a continuous dynamical system S, the input dimension of the Fourier KAN matches the dimension of the system, while the output dimension is set to one.

Training of the Barrier Fourier Kolmogorov-Arnold Network

In this subsection, we will introduce the method for synthesizing Fourier KAN barrier certificates. Based on the barrier certificate conditions, we design a loss function for the training dataset. During the training process, the Fourier coefficients in the Fourier KAN are updated by minimizing the loss function value of sampled data points.

Training Dataset Construction. Without loss of generality, consider a state set X. We divide X into a finite number of cells $X_1, X_2, \ldots, X_N$ by choosing a discretization parameter ϵ. From each of these cells, we then select sample points $\mathbf{x}_i \in X_i$ such that $\|\mathbf{x} - \mathbf{x}_i\| \leq \epsilon$, for all $\mathbf{x} \in X_i$. Let D denote the set of all these sampled points. Using this method, we can construct the dataset D_I, D_U and D_D, with each batch being sampled from X_I, X_U and X_D.

Loss Function. We formally define the loss function below:

$$L_B = \sum_{\mathbf{x}\in D_I} \max\{B(\mathbf{x}) + \tau_I\} + \sum_{\mathbf{x}\in D_U} \max\{-B(\mathbf{x}) + \tau_U\} + \sum_{\mathbf{x}\in D_D, B(\mathbf{x})=0} \max\{\dot{B}(\mathbf{x}) + \tau_D\}, \tag{18}$$

where τ_I, τ_U, τ_D serve as offsets that are included to enhance the numerical stability during training. Note that the terms in (18) correspond to the three barrier certificate conditions, respectively.

To generate a Fourier KAN barrier certificate candidate $B(\mathbf{x})$, we employ gradient descent techniques to minimize L_B. When the loss decreases to 0, it indicates that the learned Fourier KAN can be a candidate barrier certificate.

Fourier KAN Barrier Verification

In this subsection, we will introduce two methods for formally verifying candidate barrier certificates represented by Fourier KAN, as those generated through gradient descent techniques do not provide formal safety guarantees. The first method is based on using an SMT solver to search for counterexamples that violate the barrier certificate properties. The second method is the Lipschitz approach, which ensures that our trained network provides a formal guarantee of correctness. Our main algorithm is illustrated in Algorithm 1. Note that a candidate barrier certificate will first be verified by SMT solver. If there exist counterexamples within a maximum number of SMT iterations (e.g., MaxIter =10) , then it will be verified using the Lipschitz method.

Verification based on SMT solvers. We design a counterexample-guided framework based on the SMT solver for verification, which is designed to find states that violate the barrier conditions in (3). To achieve this, we formulate the negation of the three conditions and verify the conditions are unsatisfiable, as follows:

$$\begin{aligned} &x \in X_I \wedge B(\mathbf{x}) > 0 \\ &x \in X_U \wedge B(\mathbf{x}) \leq 0 \\ &x \in X_D \wedge \dot{B}(\mathbf{x}) \geq 0 \wedge B(\mathbf{x}) = 0. \end{aligned} \tag{19}$$

We choose dReal as our SMT verifier, which ensures the correctness of its unsat decisions. Therefore, when dReal returns unsat for the given formula (19), it confirms that no solution exists within the specified precision, and the candidate barrier certificate $B(\mathbf{x})$ is valid. Otherwise, it means that dReal has found a counterexample that violates the safety properties of the barrier certificate.

During the verification process, dReal can only find one counterexample at a time. To improve the efficiency of verification, points around the counterexample will be sampled and added to the dataset to help the learner refine the barrier certificate. We repeat this process of refinement and verification until no counterexamples are found.

It should be pointed out that the structure of Fourier KAN makes the SMT solver based verification much easier.

Algorithm 1: Learning Formally Verified Barrier Certificate

Input: System $S = (\mathbf{f}, X_D, X_I, X_U)$, a discretization parameter ϵ
Output: Barrier Certificates B

```
function LEARNER(S, ε)
  repeat
    D ← SampleData(S, ε)
    compute loss L_B and η, update Fourier KAN
  until L_B = 0
  return candidate Fourier KAN barrier B, η
end function

function VERIFIER(B, η, iter)
  if iter < MaxIter then
    B ← encoding_dReal(B)
    Cex ← verify_dReal(B)
    if Cex = None then
      return True
    else
      D ← D ∪ Cex
    end if
  else
    ℒ ← calLipschitz(B)
    if ℒε + η ≤ 0 then
      return True
    end if
  end if
  return False
end function

function CEGIS(S)
  initialise Fourier KAN, S, ε
  repeat
    iter ← iter + 1
    B, η ← LEARNER(S, ε)
    Flag ← VERIFIER(B, η, iter)
  until Flag = True or iter = MaxIter + 1
  return B
end function
```

Verification based on Lipschitz method. Generally, the SMT-based verification method lacks termination guarantees and may have a low success rate in synthesizing valid barrier certificates. Therefore we additionally implement a validity condition within the training of Fourier KAN to ensure the correctness of the barrier certificates without the need for post-facto verification.

Theorem 2 (Correctness of barrier certificates)**.** *Consider a system S with initial region X_I, unsafe region X_U and domain region X. Let D denote the set of points sampled from X. Assume B is a Lipschitz continuous barrier for the* *given system S learned from sampled data D. If*

$$\mathcal{L}\epsilon + \eta \leq 0, \tag{20}$$

where $\mathcal{L}$ is the Lipschitz constant of B, η is a negative number and ϵ is the grid density of sample, then B is a valid barrier certificate that satisfies Equation (3).

Proof. First, let us consider the initial constraint. Assuming η such that $B(x_i) \leq \eta$ holds for all $x_i \in D_I$. Since B is Lipschitz continuous, we obtain:

$$B(x_j) - B(x_k) \leq \mathcal{L}\|x_j - x_k\|, \tag{21}$$

where $x_j, x_k \in X$. Besides, for all $x \in X_I$ there exists x_i such that $\|x - x_i\| \leq \epsilon$. So, we can derive:

$$B(x) - B(x_i) \leq \mathcal{L}\|x - x_i\| \leq \mathcal{L}\epsilon, \tag{22}$$

where x_i denotes the center of a hyper-rectangle. By adjusting the equation:

$$B(x) \leq B(x_i) + \mathcal{L}\epsilon. \tag{23}$$

Since $\mathcal{L}\epsilon \leq -\eta$ according to (20), it follows that:

$$B(x) \leq B(x_i) - \eta. \tag{24}$$

We can get $B(x_i) \leq \eta$, and by merging these two inequalities, we obtain: $B(x) \leq 0$ for all $x \in X_I$. A similar reasoning can be used to prove the other two conditions. □

Experiments

We have implemented a tool named KAN4BC using PyTorch platform for synthesizing Fourier KAN barrier certificates. We compare our tool against the latest version of FOSSIL2.0 (Edwards, Peruffo, and Abate 2024).

Implementation Details

To compare and illustrate the performance differences between the automatic learning of activation functions by KAN4BC and the manual selection of activation functions in FOSSIL2.0, we choose sigmoid, softplus, tanh, poly_2, cosh functions as the activation functions for FNNs respectively. For each case, we ensure that the number of sampling points is consistent between the two tools. Besides, we use AdamW as the optimization algorithm whose learning rate is 0.001 with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$, and the loss function in Equation (18) with $\tau_I = \tau_U = \tau_D = 0$. We set the hyper-parameter gridsize in Fourier KAN to 1. The timeout is set to 1 hour, meaning that if a real barrier certificate is not synthesized within 1 hour, the attempt is considered a failure. All experiments are performed on a machine running Ubuntu 24.04 with AMD Ryzen 5 5600G and 16GB memory.

Case Studies

We evaluate the effectiveness of our tool with examples that include both polynomial and non-polynomial cases. The cases from $barr_1$ to $barr_4$ are taken from FOSSIL2.0's own benchmarks, $barr_1$ originally presented in (Zeng et al. 2016), $barr_2$ (Liu et al. 2015) includes non-polynomial terms in the dynamics, $barr_3$ (Prajna 2006) uses non-convex domains, $barr_4$ (Barry, Majumdar, and Tedrake 2012) represents a robotics application, specifically the control of a plane's angular velocity, $barr_5$ and $barr_6$ (Sogokon, Ghorbal, and Johnson 2016) are 2-dimensional models, $barr_7$ (Sankaranarayanan, Chen et al. 2013) is a 4-dimensional model.

Example 1 ($barr_3$). *The continuous dynamical system is as follows:*

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + \frac{1}{3}x_1^3 - x_2\end{bmatrix}.$$

The domain is $X_D = \{\mathbf{x} \in \mathbb{R}^2 \mid -3 \leq x_1 \leq 2.5, -2 \leq x_2 \leq 1\}$. Our goal is to generate a barrier certificate that ensures all trajectories starting from the initial region $X_I = \{\mathbf{x} \in \mathbb{R}^2 \mid (x_1 - 1.5)^2 + x_2^2 \leq 0.25 \ \vee \ (-1.8 \leq x_1 \leq -1.2 \wedge -0.1 \leq x_2 \leq 0.1) \vee (-1.4 \leq x_1 \leq -1.2 \wedge -0.5 \leq x_2 \leq 0.1)\}$ will never enter the unsafe region $X_U = \{\mathbf{x} \in \mathbb{R}^2 \mid (x_1+1)^2 + (x_2+1)^2 \leq 0.16 \vee (0.4 \leq x_1 \leq 0.6 \wedge 0.1 \leq x_2 \leq 0.5) \ \vee \ (0.4 \leq x_1 \leq 0.8 \wedge 0.1 \leq x_2 \leq 0.3)\}$.

We sample 256 points from the initial region,1024 points from unsafe region, and 16384 points from domain region. Using a two-layer Fourier KAN with shape [2, 15, 1], KAN4BC successfully synthesized the Fourier KAN barrier certificate, as illustrated in Figure 2, in which initial regions X_I and unsafe sets X_U are represented in green and red respectively, and the blue line outlines the level curve $B(\mathbf{x}) = 0$. Note that FOSSIL2.0 failed to synthesize barrier certificate with all five activation functions. B-spline-based KAN failed to synthesize barrier certificate.

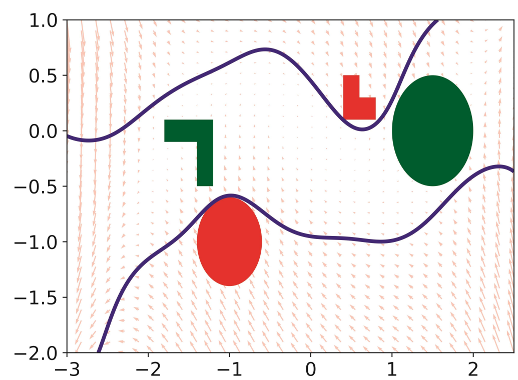

Figure 2: The learned barrier for Example 1

Example 2 ($barr_4$). *The continuous dynamical system is as follow:*

$$\begin{bmatrix}\dot{x} \\ \dot{y} \\ \dot{\phi}\end{bmatrix} = \begin{bmatrix} vsin(\phi) \\ vcos(\phi) \\ -sin(\phi) + \frac{3xsin(\phi)+3ycos(\phi)}{0.5+x^2+y^2}\end{bmatrix}.$$

The domain is $X_D = \{\mathbf{x} \in \mathbb{R}^3 \mid -2 \leq x \leq 2, -2 \leq y \leq 2, -1.57 \leq \phi \leq 1.57\}$. Our goal is to generate a barrier certificate that ensures all trajectories starting from the initial region $X_I = \{\mathbf{x} \in \mathbb{R}^3 \mid -0.1 \leq x \leq 0.1, -2 \leq y \leq 1.8, -0.52 \leq \phi \leq 0.52\}$ will never enter the unsafe region $X_U = \{\mathbf{x} \in \mathbb{R}^3 \mid -0.2 \leq x \leq -0.2, -0.2 \leq y \leq 0.2, -0.52 \leq \phi \leq -0.52, x^2 + y^2 <= 0.04\}$.

examples	shape	KAN4BC		FOSSIL2.0									
		iter	time (s)	σ_{sig}		σ_{soft}		σ_t		σ_2		σ_c	
				iter	time (s)	iter	time (s)	iter	time (s)	iter	time (s)	iter	time (s)
$barr_1$	[2,10,1]	2	**5.14**	7	10.50	34	49.99	10	7.96	11	17.15	6	9.15
$barr_2$	[2,15,1]	3	**13.46**	58	17.14	/	×	20	14.46	/	×	/	×
$barr_3$	[2,15,1]	3	**66.34**	/	×	/	×	/	×	/	×	/	×
$barr_4$	[3,10,1]	1	265.77	12	385.19	40	1166.11	28	**124.85**	/	×	/	×
$barr_5$	[2,5,1]	4	**1.37**	3	2.45	2	2.51	1	1.51	43	57.24	14	15.03
$barr_6$	[2,5,1]	3	**1.31**	1	1.33	2	1.87	2	2.14	1	1.65	1	1.38
$barr_7$	[4,5,1]	2	300.30	/	×	3	302.17	73	2036.87	16	594.39	2	**7.33**

Table 1: Comparative results of the solution in different cases

We sample 512 points each from the initial region and the unsafe region, and 32768 points from the domain region. Using a two-layer Fourier KAN with shape [3, 10, 1], KAN4BC successfully synthesized the Fourier KAN barrier certificate, as illustrated in Figure 3, in which the zero level set of learned barrier certificate (the yellow surface) separates X_U (the red cylinder) from simulated trajectories starting from X_I (the green cuboid). Note that the performance of FOSSIL2.0 is limited by the choice of activation functions, and FOSSIL2.0 failed to synthesize barrier certificate with poly_2 and cosh functions. B-spline-based KAN failed to synthesize barrier certificate.

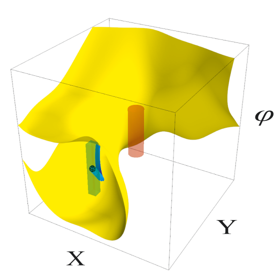

Figure 3: The learned barrier for Example 2

Results Analysis

Table 1 summarizes the results of our KAN4BC compared to FOSSIL2.0 which used five different activation functions. The symbol σ_{sig}, σ_{soft}, σ_t, σ_2 and σ_c indicate sigmoid, softplus, tanh, poly_2 and cosh functions, respectively. The $shape$ represents the Fourier KAN structures corresponding to the barrier certificate. To compare the performance between learnable activation functions and fixed activation functions, the number of layers and nodes per layer in the neural network within FOSSIL2.0 are kept consistent with those in Fourier KAN. The $iter$ and $time$ respectively represent the time and the number of iterations required for the method to generate a real barrier certificate in the corresponding case. And the symbol × indicates the synthesis of the barrier certificate failed within one hour.

From Table 1, we conclude that our tool KAN4BC can successfully synthesize real barrier certificates for all cases, but FOSSIL2.0 fails to synthesize a barrier certificate for $barr_3$ with all five activation functions. For $barr_2$ and $barr_7$, FOSSIL2.0 also fails to synthesize correct barrier certificates when using some activation functions. Our tool is more scalable because Fourier KAN can fit any function using learnable activation functions to adapt to different systems, while FOSSIL2.0's performance is limited by specific activation functions.

Considering the efficiency, our tool generally outperforms in the majority of cases, because it can control network complexity to achieve higher computational efficiency. On average, KAN4BC has a longer training time per iteration, but a shorter verification time, and the total number of iterations is fewer than FOSSIL2.0, so its total synthesis time ends up being shorter than FOSSIL2.0. Only for $barr_4$ with the tanh function and for $barr_7$ with the cosh function, FOSSIL2.0 synthesizes barrier certificates faster than our tool, which may be attributed to various factors, including system characteristics, network structure, and optimization strategies.

In summary, Table 1 indicates that our method successfully synthesizes real barrier certificates to ensure the system's safety across all seven benchmark experiments, but FOSSIL2.0 fails. Besides, KAN4BC requires fewer iterations and less time compared to FOSSIL2.0 in most cases. The results demonstrate that our method exhibits superior efficiency and versatility in synthesizing barrier certificates.

During our experiments, we find that the speed of SMT verification is consistently faster than the Lipschitz method, possibly due to the large size of the dataset, which will be an area for future improvement.

Conclusion

In this paper, we have proposed a novel and efficient method for formally synthesizing barrier certificates to verify the safety of general continuous dynamical systems. Our synthesis process consists of two stages: First, it generated candidate barrier certificates using Fourier KAN which can improve representation power and training speed. Second, it used the Lipschitz method or SMT algorithm to verify the validity of candidate Fourier KAN barrier certificates. We demonstrated the effectiveness of our tool using several case studies. A future research direction is improving the efficiency of the Lipschitz-based verification method.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62272416, 62472388 and 62132014, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LD24F020006, in part by the Zhejiang Provincial Key Research and Development Program of China under Grant 2022C01045, and in part by the Fundamental Research Funds of Zhejiang Sci-Tech University under Grant 24232090-Y.

References

Abate, A.; Ahmed, D.; Edwards, A.; Giacobbe, M.; and Peruffo, A. 2021. FOSSIL: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks. In *Proceedings of the 24th international conference on hybrid systems: computation and control*, 1–11.

Barry, A. J.; Majumdar, A.; and Tedrake, R. 2012. Safety verification of reactive controllers for UAV flight in cluttered environments using barrier certificates. In *2012 IEEE International Conference on Robotics and Automation*, 484–490. IEEE.

Chang, Y.-C.; Roohi, N.; and Gao, S. 2019. Neural lyapunov control. *Advances in neural information processing systems*, 32.

Edwards, A.; Peruffo, A.; and Abate, A. 2024. Fossil 2.0: Formal Certificate Synthesis for the Verification and Control of Dynamical Models. In *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, 1–10.

Kapinski, J.; Deshmukh, J. V.; Sankaranarayanan, S.; and Arechiga, N. 2014. Simulation-guided Lyapunov analysis for hybrid dynamical systems. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, 133–142.

Kolmogorov, A. 1961. *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables*. American Mathematical Society.

Kong, H.; Song, X.; Han, D.; Gu, M.; and Sun, J. 2014. A new barrier certificate for safety verification of hybrid systems. *The Computer Journal*, 57(7): 1033–1045.

Legat, B.; Tabuada, P.; and Jungers, R. M. 2020. Sum-of-squares methods for controlled invariant sets with applications to model-predictive control. *Nonlinear Analysis: Hybrid Systems*, 36: 100858.

Leshno, M.; Lin, V. Y.; Pinkus, A.; and Schocken, S. 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6): 861–867.

Lindemann, L.; Hu, H.; Robey, A.; Zhang, H.; Dimarogonas, D.; Tu, S.; and Matni, N. 2021. Learning hybrid control barrier functions from data. In *Conference on robot learning*, 1351–1370. PMLR.

Liu, J.; Zhan, N.; Zhao, H.; and Zou, L. 2015. Abstraction of elementary hybrid systems by variable transformation. In *International Symposium on Formal Methods*, 360–377. Springer.

Liu, S.; Liu, C.; and Dolan, J. 2023. Safe control under input limits with neural control barrier functions. In *Conference on Robot Learning*, 1970–1980. PMLR.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2024. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

Peruffo, A.; Ahmed, D.; and Abate, A. 2021. Automated and formal synthesis of neural barrier certificates for dynamical models. In *International conference on tools and algorithms for the construction and analysis of systems*, 370–388. Springer.

Platzer, A.; and Clarke, E. M. 2008. Computing differential invariants of hybrid systems as fixedpoints. In *Computer Aided Verification: 20th International Conference, CAV 2008 Princeton, NJ, USA, July 7-14, 2008 Proceedings 20*, 176–189. Springer.

Prajna, S. 2006. Barrier certificates for nonlinear model validation. *Automatica*, 42(1): 117–126.

Prajna, S.; and Jadbabaie, A. 2004. Safety verification of hybrid systems using barrier certificates. In *International Workshop on Hybrid Systems: Computation and Control*, 477–492. Springer.

Sankaranarayanan, S.; Chen, X.; et al. 2013. Lyapunov Function Synthesis using Handelman Representations. *IFAC Proceedings Volumes*, 46(23): 576–581.

Sloth, C.; Pappas, G. J.; and Wisniewski, R. 2012. Compositional safety analysis using barrier certificates. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, 15–24.

Sogokon, A.; Ghorbal, K.; and Johnson, T. T. 2016. Nonlinear continuous systems for safety verification (benchmark proposal). In *ARCH@ CPSWeek 2016-3rd International Workshop on Applied Verification for Continuous and Hybrid Systems*, volume 43, 42–51. EasyChair.

Sogokon, A.; Ghorbal, K.; Tan, Y. K.; and Platzer, A. 2018. Vector barrier certificates and comparison systems. In *International Symposium on Formal Methods*, 418–437. Springer.

Squires, E.; Pierpaoli, P.; and Egerstedt, M. 2018. Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 1656–1661. IEEE.

Yang, Y.; Jiang, Y.; Liu, Y.; Chen, J.; and Li, S. E. 2023. Model-free safe reinforcement learning through neural barrier certificate. *IEEE Robotics and Automation Letters*, 8(3): 1295–1302.

Zeng, X.; Lin, W.; Yang, Z.; Chen, X.; and Wang, L. 2016. Darboux-type barrier certificates for safety verification of nonlinear hybrid systems. In *Proceedings of the 13th International Conference on Embedded Software*, 1–10.

Zhang, S.; Xiu, Y.; Qu, G.; and Fan, C. 2023. Compositional neural certificates for networked dynamical systems. In *Learning for Dynamics and Control Conference*, 272–285. PMLR.

Zhao, H.; Zeng, X.; Chen, T.; and Liu, Z. 2020. Synthesizing barrier certificates using neural networks. In *Proceedings of the 23rd international conference on hybrid systems: Computation and control*, 1–11.