European Journal of Control 74 (2023) 100856

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

Estimation of region of attraction with Gaussian process classification[☆]

Ke Wang[a,*], Prathyush P. Menon[a], Joost Veenman[b], Samir Bennani[c]

[a] Faculty of Environment, Science and Economy, University of Exeter, Exeter EX4 4PY, United Kingdom
[b] SENER Aeroespacial, Madrid 28760, Spain
[c] European Space Research and Technology Centre, Noordwijk, 2201, the Netherlands

ARTICLE INFO

Article history:
Received 15 May 2023
Accepted 8 June 2023
Available online 14 June 2023

Recommended by Prof. T Parisini

Keywords:
Region of attraction
Gaussian process classification
Active learning

ABSTRACT

This paper proposes a methodology for assessing the region of attraction (ROA) of stable equilibrium points, a challenging problem for a general nonlinear system, using binary Gaussian process classification (GPC). Interest in this method stems from the fact that an arbitrary point belonging to the system's state space can be classified in the region of attraction or not. Importantly the proposed GPC approach for determining ROA gives a minimum confidence level associated with the estimate. Moreover, the active learning scheme helps to update the GPC model and yield better predictions by selecting informative observations from the state space sequentially. The methodology is applied to several examples to illustrate the effectiveness of this approach.

© 2023 The Author(s). Published by Elsevier Ltd on behalf of European Control Association.
This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/4.0/)

1. Introduction

The Region of Attraction (ROA) of an asymptotically stable equilibrium point $\mathbf{x}^*$ of a dynamic system is the set of all the initial states space from which trajectories of the system converge to $\mathbf{x}^*$ as time approaches infinity [14]. The exact ROA is often difficult to compute, some prior knowledge of its size and shape is helpful. Moreover, for a general nonlinear system, the ROA could be a very complicated set, where analytical representation can be impossible.

Numerous methods have been proposed in the literature for estimating the ROA of an equilibrium point, most of them can be classified as Lyapunov based and non-Lyapunov based methods [8]. The Lyapunov based approaches focus on determining Lyapunov function level sets, including Zubov methods [10,34] and La Salle methods [17], either by putting conditions for the Lyapunov function or extending the Lyapunov theory. A line search involving the solution of Linear Matrix Inequalities (LMIs) is used to compute the Lyapunov matrix in Valmorbida et al. [29] in order to obtain the largest ellipsoid of polynomial systems when the Lyapunov function space is restricted to quadratic functions. In general cases, for higher order Lyapunov functions, Sum of Squares (SOS) methodology can be used to transfer the problem into a set of Semi-Definite Programs (SDPs) [6]. The second family, non-Lyapunov methods, does not explicitly employ Lyapunov functions. Typical method like the "tracking function" method [22] guarantees a practical stability region by using a consideration of La Salle [17] about the conditions for system trajectories so as not to cross a fixed surface. Using integral quadratic constraints (IQCs) [12,25,26] allows to estimate ROA for generic problems like hard-nonlinearities. A global optimization approach [18] is proposed for the estimation of the ROA based on maximal Lyapunov functions by finding the best level set of a Lyapunov function which is fully contained in the region of negative definiteness of its time derivative. Rather than focusing on learning invariant sets that require trajectories to always lie within the set, Shen et al. [27] learns sets that satisfy a more flexible notion of invariance, Recurrent Sets, with necessary and sufficient conditions, to obtain an inner approximations of the ROA.

Recently, many works have studied to compute the ROAs for systems with uncertainties. Novel statistical verification frameworks [23] can be used to estimate the ROA for both deterministic and stochastic systems by combining data-driven statistical learning techniques and control system verification. Similar statistical methodology is used in Tadiparthi and Bhattacharya [28] by combining efficient uncertainty sampling with a modified representative sampling technique to arrive at a formulation that fuses informativeness with represent ativeness in the learning

[☆] The authors are thankful for the financial support from the European Space Agency, contract no. 4000126163/18/NL/GLC.

[*] Corresponding author.

E-mail addresses: kw535@exeter.ac.uk (K. Wang), p.m.prathyush@exeter.ac.uk (P.P. Menon), joost.veenman@aeroespacial.sener (J. Veenman), samir.bennani@esa.int (S. Bennani).

https://doi.org/10.1016/j.ejcon.2023.100856

0947-3580/© 2023 The Author(s). Published by Elsevier Ltd on behalf of European Control Association. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

paradigm. Berkenkamp et al. [1] integrates ideas from Gaussian process regression (GPR) learning, safe Bayesian optimization and ROA computation based on Lyapunov functions for uncertain systems to provide an algorithm to actively and safely explore the state space in order to expand the ROA estimate. Chen et al. [5] proposes a sampling-based method for constructing robust Lyapunov functions for uncertain dynamical systems by neural networks to estimate the ROA.

Our contributions. Firstly, in this paper, we transfer the estimation of the ROA into a binary classification problem to overcome the difficulty in determining the inner approximation of ROA by constructing a Lyapunov function for complicated nonlinear systems, which is typically a complex problem. The statistical classification model used in this work is Gaussian process classification (GPC) [21,32]. The underlying concept captures complex relationships between the input and output data and provides variance and confidence level information about the estimated ROA. Secondly, active learning methodology [4,9] is adopted to update the GPC model sequentially by selecting informative observations. The proposed approach requires only a few initial training samples, and over iterations, the algorithm chooses more informative sampling locations to converge to the estimate of ROA ultimately. The resulting algorithm returns an approximate ROA and a minimum confidence level associated with the estimate.

The remaining of the paper is organised as follows. Section 2 introduces the definition of the ROA, followed by the details for GPC in Section 3. The methodology of estimation of the ROA with GPC is proposed in Section 4 and tested with several examples in Section 5. Finally, we conclude the paper in Section 6.

2. Problem statement

2.1. Region of attraction (ROA)

Consider an autonomous nonlinear dynamic system:

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

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector. The vector field $F(\mathbf{x})$: $\mathbb{R}^n \to \mathbb{R}^n$ is assumed to be globally Lipschitz continuous to guarantee the existence and uniqueness of the solution of the system. The vector $\mathbf{x}^\star$ is an equilibrium point of system (1), if it has the property that the state of the system starts at $\mathbf{x}^\star$, it will remain at $\mathbf{x}^\star$ for all future time, and are real roots of the equation $F(\mathbf{x}^\star) = \mathbf{0}$. Let $\phi(t; \mathbf{x}(t_0), t_0)$ denote the trajectory with initial state $\mathbf{x}(t_0)$ during time period $[t_0, t]$, formally the ROA of the equilibrium point $\mathbf{x}^\star$ is given by Khalil [14]:

$$\mathcal{R}(\mathbf{x}^\star) := \left\{ \mathbf{x}(t_0) \in \mathbb{R}^n : \lim_{t\to\infty} \phi(t; \mathbf{x}(t_0), t_0) = \mathbf{x}^\star \right\}. \tag{2}$$

Throughout this work, we assume, without loss of generality, that the equilibrium point is the origin of the state space $\mathbb{R}^n$, $\mathbf{x}^\star = \mathbf{0}$. The ROA of the equilibrium point $\mathbf{x}^\star = \mathbf{0}$ is given by

$$\mathcal{R}(\mathbf{0}) := \left\{ \mathbf{x}(t_0) \in \mathbb{R}^n : \lim_{t\to\infty} \phi(t; \mathbf{x}(t_0), t_0) = \mathbf{0} \right\}. \tag{3}$$

2.2. Binary classification

In machine learning, binary classification refers to a supervised learning algorithm [2], which categorizes new observations into one of two classes. The initial state $\mathbf{x}(t_0)$ can be classified either in the ROA $\mathcal{R}(\mathbf{0})$, or in its complementary region $\bar{\mathcal{R}}(\mathbf{0})$ such that $\mathcal{R}(\mathbf{0}) \cap \bar{\mathcal{R}}(\mathbf{0}) = \emptyset$ and $\mathcal{R}(\mathbf{0}) \cup \bar{\mathcal{R}}(\mathbf{0}) = \mathbb{R}^n$. Hence, the estimation of the ROA is posed as a binary classification problem, labelled as $+1$ for all the points in $\mathcal{R}(\mathbf{0})$, and -1 for all the points in $\bar{\mathcal{R}}(\mathbf{0})$. This paper uses GPC to design the scheme for determining ROA. The advantage of the GPC methodology is that it can capture complex relationships between the input and output data while providing confidence-level information about the estimated ROA. The details of GPC are introduced in the sequel. Other commonly used methods for binary classification are decision trees [24], support vector machines [7], and neural networks [11], among others.

3. Gaussian process classification

A GP is an infinite collection of random variables where any finite subset follows a joint multivariate Gaussian distribution. GP can be divided into GP regression (GPR) and classification problems as a supervised learning method. Here, we only introduce the GPC; more details on GPR can be found in Williams and Rasmussen [32].

Suppose a data set $\mathcal{L} = \{\mathbf{X}, \mathbf{y}\}$ with N input-output data pairs, consists of inputs $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$ and corresponding outputs $\mathbf{y} = [y_1, \dots, y_N]^T$, where $\mathbf{x}_i \in \mathbb{R}^n$, for the sake of brevity of the initial state $\mathbf{x}_i(t_0)$, is the input, and a label $y_i = y(\mathbf{x}_i) \in \{-1, +1\}$ is the binary class label, $i = 1, \dots, N$.

A Gaussian process prior over the latent function $f(\mathbf{x})$ is,

$$f(\mathbf{x}) \sim \mathcal{N}\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right). \tag{4}$$

In (4), $m(\mathbf{x})$ represents the mean function and $k\left(\mathbf{x}, \mathbf{x}'\right)$ is the covariance function. For simplicity, we choose a zero mean function $m(\mathbf{x}) = 0$. The covariance functions are required to be positive semi-definite functions and have the property that points closer in the input space are more strongly correlated [32]. A commonly used covariance function is the squared exponential covariance function with automatic relevance determination (ARD) distance measure. The covariance function is parameterized as

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_f^2 \exp\left(-\frac{1}{2} \left(\mathbf{x} - \mathbf{x}'\right)^T \mathbf{\Lambda} \left(\mathbf{x} - \mathbf{x}'\right) \right), \tag{5}$$

where σ_f^2 is the signal variance linked to the general function variance, $\mathbf{\Lambda} = diag(\boldsymbol{l})$ is a diagonal matrix with $\boldsymbol{l}$ defining the $n \times 1$ positive ARD characteristic legnth-scale vector, and n corresponds to the input state space dimension. Obviously, a short length scale hence corresponds to high relevance. Other mean and covariance functions can be found in Williams and Rasmussen [32].

We use a vector $\boldsymbol{\theta} = \left\{ \sigma_f^2, \boldsymbol{l} \right\}$ to gather all the parameters of covariance function, which are known as hyperparameters. Since $f(\mathbf{x})$ has a GP prior, for any N input vectors $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$, denote $\mathbf{f} = [f_1, \dots, f_N]^T$, where $f_i = f(\mathbf{x}_i)$, $i = 1, \dots, N$, the prior over $\mathbf{f}$ is jointly Gaussian,

$$p\left(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}\right) = \mathcal{N}(\mathbf{0}, K), \tag{6}$$

where $\mathbf{0}$ is the $N \times 1$ zero mean vector, and $K = K(\mathbf{X}, \mathbf{X})$ having elements $k_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$, $i, j = 1, \dots, N$, is the $N \times N$ covariance matrix.

The likelihood $p(\mathbf{y}|\mathbf{f})$ is calculated as

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} p(y_i | f_i), \tag{7}$$

since the likelihood factorizes over the training samples. To find the posterior distribution over the latent variable $p\left(\mathbf{f} | \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}\right)$, using (7), according to the Bayesian rule,

$$p\left(\mathbf{f} | \mathbf{X}, \mathbf{y}, \boldsymbol{\theta}\right) = \frac{1}{Z} p\left(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}\right) \prod_{i=1}^{N} p(y_i | f_i), \tag{8}$$

where the prior $p\left(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}\right)$ is Gaussian, and the normalization term Z is the marginal likelihood

$$Z = p\left(\mathbf{y} | \mathbf{X}, \boldsymbol{\theta}\right) = \int p\left(\mathbf{f} | \mathbf{X}, \boldsymbol{\theta}\right) \prod_{i=1}^{N} p(y_i | f_i) d\mathbf{f}. \tag{9}$$

In this paper, we use the probit likelihood for binary classification [2]

$$p(y_i|f_i) = \Phi(y_i f_i), \tag{10}$$

where $\Phi(z)$ is the cumulative density function of a standard normal distribution $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(x|0,1)dx$. Other sigmoidal transformation including linear logistic regression is discussed in Williams and Rasmussen [32]. Using (10) in (9), the posterior distribution becomes

$$p\big(\mathbf{f}|\mathbf{X},\mathbf{y},\boldsymbol{\theta}\big) = \frac{1}{Z} p\big(\mathbf{f}|\mathbf{X},\boldsymbol{\theta}\big) \prod_{i=1}^{N} \Phi(y_i f_i). \tag{11}$$

Suppose we want to predict the class label y_* at a test point $\mathbf{x}_*$, from the definition of GP we know the joint prior over the training and test latent variable $\mathbf{f}$ and f_* (short for $f(\mathbf{x}_*)$), given the corresponding inputs, is Gaussian

$$p\big(f, f_*|\mathbf{X}, x_*, \boldsymbol{\theta}\big) = \mathcal{N}\left(0, \begin{bmatrix} K & K_* \\ K_*^T & k_{**} \end{bmatrix}\right), \tag{12}$$

where $K_* = [k(\mathbf{x}_1, \mathbf{x}_*), \dots, k(\mathbf{x}_N, \mathbf{x}_*)]^T$, $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$, and the conditional becomes

$$p\big(f_*|\mathbf{f}, \mathbf{X}, \mathbf{x}_*, \boldsymbol{\theta}\big) = \mathcal{N}\big(K_*^T K^{-1}\mathbf{f}, k_{**} - K_*^T K^{-1} K_*\big). \tag{13}$$

The prediction distribution can be computed as

$$p\big(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \boldsymbol{\theta}\big) = \int p\big(f_*|\mathbf{f}, \mathbf{X}, \mathbf{x}_*, \boldsymbol{\theta}\big) p\big(\mathbf{f}|\mathbf{X}, \mathbf{y}, \boldsymbol{\theta}\big) d\mathbf{f}. \tag{14}$$

However, the posterior $p\big(\mathbf{f}|\mathbf{X},\mathbf{y},\boldsymbol{\theta}\big)$ in (11) is analytically intractable because of the non-Gaussian likelihood in (10), which makes the evaluation of the integral on the right hand of (14) impossible.

To circumvent above difficulty, we follow the expectation propagation (EP) framework [15,16,19] and approximate the likelihood in (10) by a local likelihood approximation. By doing this, the posterior $p\big(\mathbf{f}|\mathbf{X},\mathbf{y},\boldsymbol{\theta}\big)$ in (11) becomes Gaussian. Meanwhile, as the normalization likelihood Z in (9) is conditional on the hyperparameters $\boldsymbol{\theta}$, which allows us to estimate the hyperparameters using maximum likelihood II (ML-II) type [16] hyperparameter estimation by setting the derivatives of the log marginal likelihood w.r.t to the hyperparameter $\boldsymbol{\theta}$ to zero. More methods to handle this problem include Laplace approximation method [31] and Markov chain Monte Carlo [20].

If we denote the learned hyperparameters $\boldsymbol{\theta} = \boldsymbol{\theta}_*$, the approximate predictive distribution $q\big(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \boldsymbol{\theta}_*\big)$ then becomes

$$q\big(f_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \boldsymbol{\theta}_*\big) = \mathcal{N}\big(\mu_*, \sigma_*^2\big), \tag{15}$$

with mean

$$\mu_* = K_*^T \big(K + \tilde{\Sigma}\big)^{-1} \tilde{\mu}, \tag{16}$$

and variance

$$\sigma_*^2 = k_{**} - K_*^T \big(K + \tilde{\Sigma}\big)^{-1} K_*, \tag{17}$$

where $\tilde{\mu}$ and $\tilde{\Sigma}$ are the mean vector and covariance matrix for the local likelihood approximation [19]. Using (15) in (10), the predicted probability of the class label y_* becomes

$$q\big(y_* = +1|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \boldsymbol{\theta}_*\big) = \Phi\left(\frac{\mu_*}{\sqrt{1 + \sigma_*^2}}\right). \tag{18}$$

4. Estimation of region of attraction with Gaussian process classification

For the rest of the paper, we omit the explicit dependency of the predictive probability of the class label on $\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \boldsymbol{\theta}_*$ for the sake of brevity. From (18), the approximate predictive probability of the class label $y_* = +1$ is $q(y_* = +1)$. Hence the approximate predictive probability of the class label $y_* = -1$ naturally becomes $q(y_* = -1) = 1 - q(y_* = +1)$. When $q(y_* = +1) > 0.5 \geq q(y_* = -1)$, we assign the class label $y_* = +1$, $\mathbf{x}_* \in \mathcal{R}(\mathbf{0})$. Contrary when $q(y_* = +1) < 0.5 \leq q(y_* = -1)$ the class label y_* is assigned value -1, and $\mathbf{x}_* \in \bar{\mathcal{R}}(\mathbf{0})$. In consequence, for the estimation of ROA framework, we have

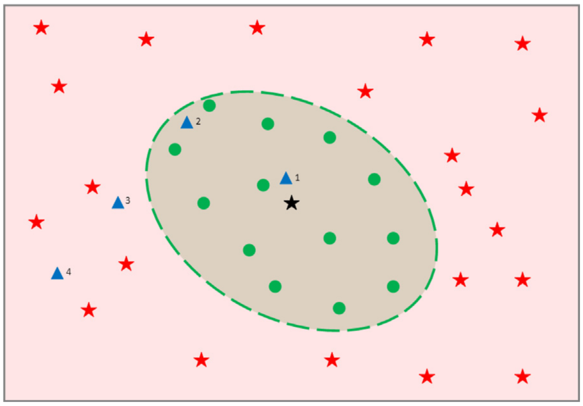

Fig. 1. Binary classification. In this two dimension scenario, the black star is the Equilibrium point $\mathbf{x}^{\star} = \mathbf{0}$, the green region is the $\mathcal{R}(\mathbf{0})$. Green dots are initial points in the $\mathcal{R}(\mathbf{0})$, and red stars are those in the $\bar{\mathcal{R}}(\mathbf{0})$. Four blue triangles $\mathbf{x}_*^i$, $i = 1, \dots, 4$, from right to left, are the test points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\mathbf{x}_* \in \begin{cases} \mathcal{R}(\mathbf{0}) & \text{for} \quad q(y_* = +1) > 0.5, \\ \bar{\mathcal{R}}(\mathbf{0}) & \text{for} \quad \text{otherwise.} \end{cases} \tag{19}$$

For the binary classification, as shown in the Fig. 1, since the GPC prediction is based on the location of the test data (see (5)), intuitively, $q_1 > q_2 > 0.5 > q_3 > q_4$, where $q_i = q\big(y_* = +1|\mathbf{x}_*^i\big)$.

Given the initial training data set $\mathcal{L}$, it is possible to have an initial rudimentary estimate of the ROA of the dynamical system. For a statistical classification model, GPC provides a more accurate estimation of the ROA with more data points. However, having as many data points is generally impossible due to either the fixed computational budget or the cost of more observations from the dynamic system simulations.

Inspired by Bayesian optimisation, we use active learning [4,9] to solve this problem by selecting informative initial conditions in the state space for subsequent observations over the iterations. To this end, a metric is required to measure the level of being '*informative*'. From (17) and (5), we know that the predictive variance depends on the location of the test data point and the training data points. In Figure1, points like $\mathbf{x}_*^1$ and $\mathbf{x}_*^4$ contain relatively less new helpful information of the ROA since they are far more likely (with probability close to 1 or 0) to be classified either in the $\mathcal{R}(\mathbf{0})$ or $\bar{\mathcal{R}}(\mathbf{0})$ with lower variance. It is because training points of a single label surround $\mathbf{x}_*^1$ and $\mathbf{x}_*^4$. By contrast, points like $\mathbf{x}_*^2$ and $\mathbf{x}_*^3$ have predictive probability close to 0.5 and higher variance, which makes them difficult to be classified. Points with predictive probability close to 0.5 (actually, the points on state space close to the boundary of the ROA) yield much more useful information about the shape of the ROA and more accurate estimated ROA.

The resulting metric to select the "*best*" or the most informative point and simulate the dynamic system to gather an observation becomes:

$$\bar{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathcal{D}_s} \big(|q(y_* = +1) - 0.5|\big),$$

$$
\begin{aligned}
&= \underset{\mathbf{x}\in\mathcal{D}_s}{\arg\min}\left(\left|\Phi\left(\frac{\mu_*}{\sqrt{1+\sigma_*^2}}\right)-\Phi(0)\right|\right),\\
&= \underset{\mathbf{x}\in\mathcal{D}_s}{\arg\min}\left(\left|\frac{\mu_*}{\sqrt{1+\sigma_*^2}}-0\right|\right),\\
&= \underset{\mathbf{x}\in\mathcal{D}_s}{\arg\max}\ \tilde{V}, \qquad (20)
\end{aligned}
$$

where $\tilde{V} = \sqrt{1+\sigma_*^2}/|\mu_*|$. The set $\mathcal{D}_s$ contains the available locations determined through random sampling. In this paper, we select the data as a grid over the state space. To quantity the confidence of the prediction, we introduce a probability

$$\bar{p} = q(y_* = +1|\bar{\mathbf{x}}). \qquad (21)$$

The process starts with N_0 initial training set $\mathcal{L}$ of passively-selected samples (usually generated through random sampling and time simulations of the dynamics). Using $\mathcal{L}$, an initial GPC is trained, and using the GPC an initial ROA estimate $\mathcal{S}_0$ is found to initiate active sampling further. Suppose we have a computational budget that the number of observations (consequently simulations) is T_{total}. The sequential procedure then selects one point over the state space from the available sample locations set $\mathcal{D}_s$ according to (20). A simulation is carried out at the selected $\bar{\mathbf{x}}$ to obtain $y(\bar{\mathbf{x}})$, which is then used to update the GPC model and the estimated ROA $\mathcal{S}_i$ associated with the GPC. This iterative procedure repeats until either the number of remaining locations $T = T_{\text{total}} - N_0$ or the $p_{\min}$ (a minimum confidence level associated with the estimate) has been reached. The total number of observations T and the minimum probability $p_{\min}$ control the accuracy of the estimation, we will have further discussion in the numerical examples later. The algorithm returns an approximate ROA and a minimum confidence level associated with the estimate at the end of iteration. Algorithm 1 shows the analysis steps.

Algorithm 1 Estimation the ROA with GPC.

```
Require: initial training dataset $\mathcal{L} = \{X, \mathbf{y}\}$, available sample location $\mathcal{D}_s$, maximum number of additional samples $T$, train the GPC model with $\mathcal{L}$, initial ROA $\mathcal{S}_0$, minimum probability $p_{\min}$, $\bar{p} = 0$
while $i \le T$ and $\bar{p} < p_{\min}$ do
    Select $\bar{x} = \arg\max_{x\in\mathcal{D}_s} \tilde{V}$
    Perform simulation at $\bar{x}$, obtain $y(\bar{x})$ and $\bar{p}$
    Add $\{\bar{x}, y(\bar{x})\}$ to training dataset $\mathcal{L}$
    Retrain GP model with updated $\mathcal{L}$, updated ROA $\mathcal{S}_i$
end while
return $\mathcal{S}_i$ and $p_{\min}$
```

5. Experiments

In this section, the proposed algorithm is illustrated by three examples.

5.1. Van der Pol oscillator

Consider the dynamics of a Van der Pol oscillator, a non-conservative oscillator with nonlinear damping [13]:

$$
\begin{aligned}
\dot{x}_1 &= -x_2,\\
\dot{x}_2 &= x_1 - (1-x_1^2)x_2. \qquad (22)
\end{aligned}
$$

In this example, the initial training data set $\mathcal{L}$ consists of 36 (6×6) initial condition points X, which are selected as a grid over the state space $[-3, 3] \times [-3, 3]$ and shown in Fig. 3 as black points, and corresponding labels y, which is obtained by running the simulation for a sufficiently large time (e.g. 10 s) and checked if the state for a duration of last 5 s time interval is close to the equilibrium point, for example, the Euclid distance is less than 10^{-4}. Through this paper, we use the same way to generate training dataset, since we assume no other knowledge of the ROA is available in advance. The maximum number of additional samples $T = 64$, to make the total number of observations $T_{\text{total}} \le 100$. The available sample locations $\mathcal{D}_s$ is a 1,002,001 (1001×1001) data points grid over the state space $[-3, 3] \times [-3, 3]$, and the minimum probability is $p_{\min} = 99.999\%$.

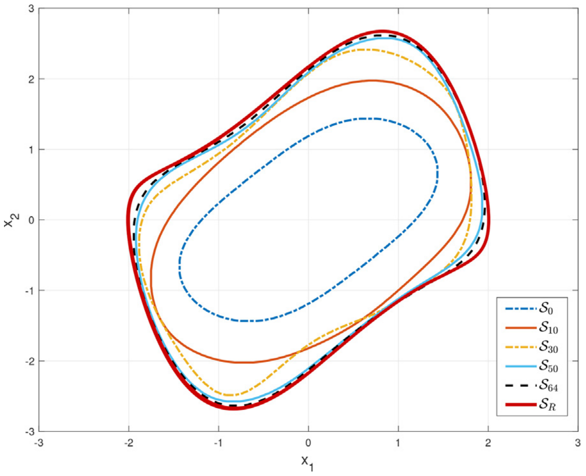

Fig. 2. Estimated ROAs for different steps.

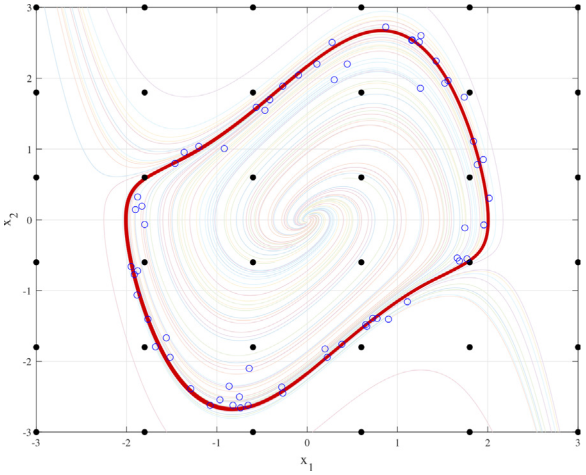

Fig. 3. Training and selected informative points. The black dots are the 36 initial training data points, the blue circles are the 64 selected informative points. Red solid line is the real ROA and the faded lines are the trajectories with the initial blue circles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 2 shows estimated ROAs at different ith steps. It is obvious with i increasing, the estimated ROAs are getting closer to the real one and end up with a conservative estimation. Comparison is given by the real ROA $\mathcal{S}_R$. Fig. 3 illustrates the training points (black dots) as a grid over the state space, selected informative points at each i^{th} step (indicated as blue circles) and trajectories (faded lines) starting from them.

Fig. 4 shows estimated ROAs given different $p_{\min}$. It can be seen that a larger $p_{\min}$ gives a smaller ROA, which means a higher confidence level of the estimate, a smaller approximate ROA. In Fig. 5, different training datasets are used to obtain the estimated ROAs. The black dashed line shows an estimated ROA with the initial training dataset containing 36 training points as a grid (the same

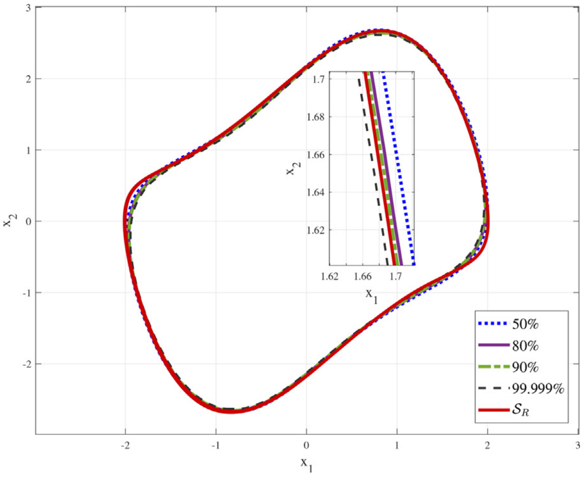

Fig. 4. Estimated ROAs for different minimum probability p_{min}.

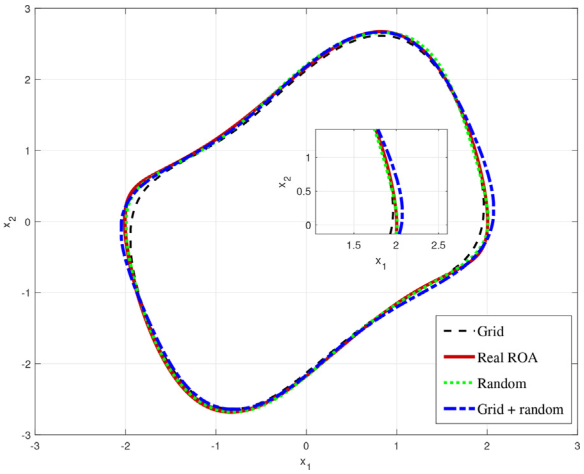

Fig. 5. Estimated ROAs with different initial training datasets.

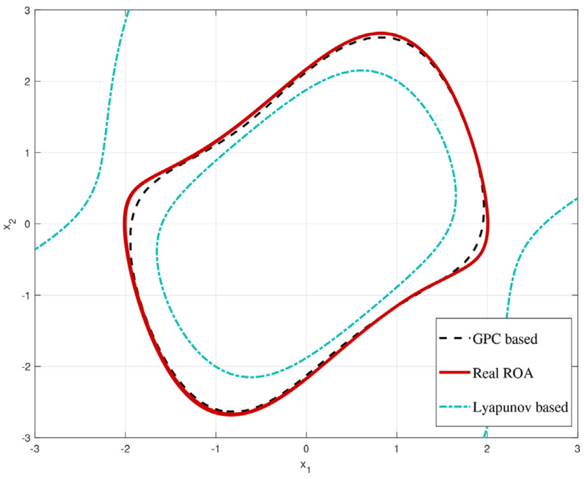

Fig. 6. Comparison of different ROAs. The black dash line shows the estimated ROA using the proposed method in this paper, red solid line is the real ROA and blue sash-dot line give the estimated ROA using Lyapunov based method introduced in Vannelli and Vidyasagar [30]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

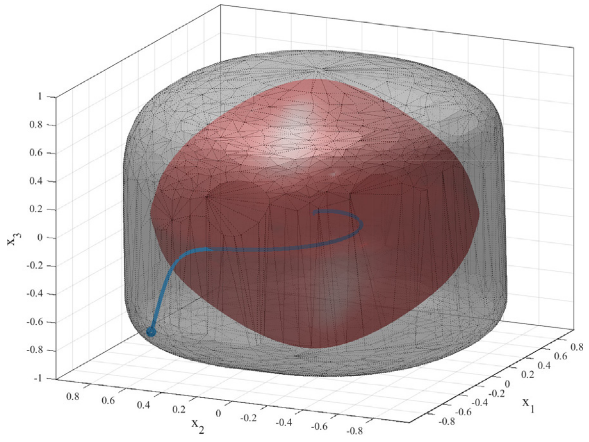

Fig. 7. Comparison of estimated ROA by proposed algorithm (black dashed) and the one in Björnsson et al. [3] (red solid). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

as in the previous result), the green dot line gives the estimated ROA with the initial training dataset containing 36 random selected training points, and the blue dash-dot line depicts the result with 72 training points which is a union of those two training datasets. In all three situations $T_{\text{total}} = 100$. An observation can be made from the Fig. 5 that different initial training datasets affect the result of the estimated ROA, but all over iterations promisingly converge towards the real ROA.

Fig. 6 depicts a comparison of different ROAs obtained by the proposed method in this paper (black dash line), the real ROA (red solid line) and a Lyapunov based method introduced in Vannelli and Vidyasagar [30] (blue sash-dot line), respectively. It is evident that the proposed algorithm gives a good estimation of inner approximation of the real ROA. Please bear in mind the algorithm introduced in Vannelli and Vidyasagar [30] gives a closed form of the ROA which contains several boundaries when plotted the ROA, as you can see in Fig. 6, however, only the closed boundary around the origin is the estimated ROA. We refer the interested reader once more to Vannelli and Vidyasagar [30] for a more in-depth discussion.

5.2. 3-dimensional example

Here we give a 3-dimensional system example from Björnsson et al. [3]:

$$\begin{aligned}
\dot{x}_1 &= x_1(x_1^2 + x_2^2 - 1) - x_2(x_3^2 + 1),\\
\dot{x}_2 &= x_2(x_1^2 + x_2^2 - 1) + x_1(x_3^2 + 1),\\
\dot{x}_3 &= 10x_3(x_3^2 - 1).
\end{aligned} \tag{23}$$

In this example, a 216 ($6 \times 6 \times 6$) grid points over the state space $[-1, 1] \times [-1, 1] \times [-1, 1]$ are used as training data set, and the maximum number of additional samples $T = 84$, to make the total number of observations $T_{\text{total}} \leq 300$. The available sample location $\mathcal{D}_s$ is a 1,030,301 ($101 \times 101 \times 101$) data points grid over the state space $[-1, 1] \times [-1, 1] \times [-1, 1]$, and the minimum probability is $p_{\text{min}} = 99.999\%$.

In Fig. 7, the red plot is the ROA obtained by the algorithm provided in Björnsson et al. [3] (available freely at www.ru.is/kennarar/sigurdurh/MICNON2015CPP.rar) and the transparent grey volume is the estimated ROA obtained with alg:EROAGPC. A blue solid trajectory corresponding to an initial point (marked as blue circle) between two volumes is also given to show it is in the ROA.

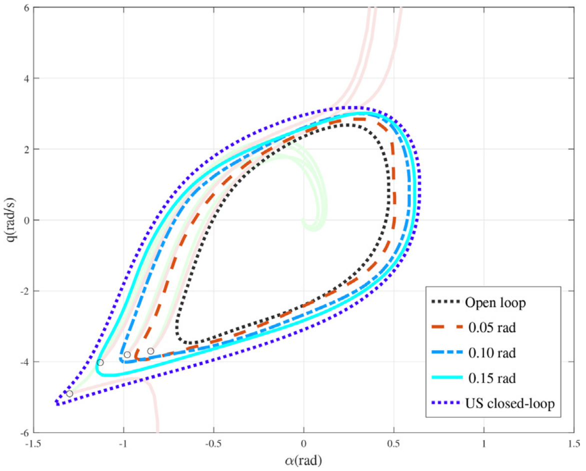

Fig. 8. Estimations of the ROA for different saturation levels.

5.3. Closed-loop short period GTM aircraft

The closed-loop short period motion of the NASA's Generic Transport Model (GTM) can be approximated as a two states polynomial system [12]:

$$\begin{aligned}\dot{\alpha} &= -1.492\alpha^3+4.239\alpha^2+0.2402\alpha\delta+0.003063\alpha q\\ &\quad -0.0649\delta^2+0.006226q^2-3.236\alpha-0.3166\delta+0.9227q,\\ \dot{q} &= -7.228\alpha^3+18.36\alpha^2+41.5\alpha\delta-45.34\alpha-59.99\delta\\ &\quad -4.372q+1.103q^3,\\ \delta_{\text{CMD}} &= Kq, \quad K=\frac{4\pi}{180},\end{aligned} \tag{24}$$

where α is the angle of attack, q is the pitch rate, and δ is the elevator deflection, which is assumed to be subject to actuator magnitude saturation:

$$\delta = \begin{cases} sgn(\delta_{\text{CMD}})\delta^{sat} & \text{for} \quad |\delta_{\text{CMD}}| > \delta^{sat}, \\ \delta_{\text{CMD}} & \text{for} \quad \text{otherwise}. \end{cases} \tag{25}$$

where $sgn(\cdot)$ is the sign function and δ^{sat} is the saturation level. The GTM steady-state solution consists of a locally stable equilibrium point at the origin $\mathbf{x}^\star = \mathbf{0}$. In this example, 36 (6×6) training data points are selected as a grid over the state space $[-1.5, 1.5]\times[-6, 6]$, and the maximum number of additional samples $T = 64$, to make the total number of observations $T_{\text{total}} \le 100$. The available sample location $\mathcal{D}_s$ is a 1,002,001 (1001×1001) data points grid over the state space $[-1.5, 1.5]\times[-6, 6]$, and the minimum probability is $p_{\min} = 99.999\%$.

Fig. 8 gives the estimations of the ROA of the saturated GTM model obtained with alg:EROAGPC. The estimation of ROAs with the open loop and unsaturated (US) closed-loop and three levels of saturation $\delta^{sat} = 0.05$ rad, 0.10 rad and 0.15 rad are given for comparison. It can be seen the estimated ROAs with δ^{sat} are larger than the open loop ROA but smaller than the unsaturated closed-loop ROA. What is more, as the value of δ^{sat} increases, the corresponding ROAs get closer to the unsaturated closed-loop ROA, which is an expected result. Fig. 8 also gives trajectories with black circles marks corresponding to four initial conditions. Four faded pink solid lines are unstable trajectories without enough saturation levels ($\delta^{sat} = [\text{open loop}, 0.05\text{ rad}, 0.10\text{ rad}, 0.15\text{ rad}]$), and four faded green solid lines are stable trajectories with enough saturation levels ($\delta^{sat} = [0.05\text{ rad}, 0.10\text{ rad}, 0.15\text{ rad}, \text{unsaturated}]$).

6. Conclusion

Estimating the region of attraction for a general nonlinear system is a hard problem. In this paper, we formulate whether or not an initial state belongs to the region of attraction of the system into a binary classification problem and leverage machine learning principles. Here we use a Gaussian Process Classification framework. Our method starts with building a GPC model with a few selected training data points and, subsequently, updating the GPC model over iterations with more informative points using active learning, given a designed metric. The noteworthy feature of the proposed method is the provision of a minimum confidence level associated with the estimation of the ROA. The efficacy of the proposed methodology is demonstrated using multiple examples from the literature, and the comparison of the new results with existing solutions is promising. Further research will explore developing more computationally efficient ways to estimate the ROA for higher dimension systems with multi-output sparse GP [33] using inducing points. In the future, we aim to apply the techniques to various industrial problems.

Declaration of Competing Interest

Authors declare that they have no conflict of interest.

References

[1] F. Berkenkamp, R. Moriconi, A.P. Schoellig, A. Krause, Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes, in: 2016 IEEE 55th Conference on Decision and Control (CDC), 2016, pp. 4661–4666.
[2] C.M. Bishop, N.M. Nasrabadi, Pattern Recognition and Machine Learning, springer, New York, 2006.
[3] J. Bj, S. Gudmundsson, S. Hafstein, Class library in C++ to compute Lyapunov functions for nonlinear systems, 48, 2015, pp. 778–783.
[4] G. Chen, Z. Sabato, Z. Kong, Active learning based requirement mining for cyber-physical systems, in: IEEE Conference on Decision and Control, 2016.
[5] S. Chen, M. Fazlyab, M. Morari, G.J. Pappas, V.M. Preciado, Learning region of attraction for nonlinear systems, in: 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 6477–6484.
[6] G. Chesi, Domain of Attraction: Analysis and Control via SOS Programming, Springer, 2011.
[7] C. Cortes, V. Vapnik, Support-vector networksMach. Learn., 20, 31995, 273–297.
[8] R. Genesio, M. Tartaglia, A. Vicino, On the estimation of asymptotic stability regions: state of the art and new proposals, IEEE Trans. Autom. Control 30 (1985) 747–755.
[9] A. Gotovos, N. Casati, G. Hitz, Active learning for level set estimation, in: International Joint Conference on Artificial Intelligence, 2013, pp. 1344–1350.
[10] W. Hahn, Stability of Motion, Springer Berlin, Heidelberg, 1967.
[11] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. 79 (8) (1982) 2554–2558.
[12] A. Iannelli, P. Seiler, A. Marcos, Region of attraction analysis with integral quadratic constraints, Automatica 109 (2019) 108543.
[13] T. Kanamaru, Van der Pol oscillator, Scholarpedia, 2 (1) 2007, 2202.
[14] H.K. Khalil., Nonlinear Systems, Prentice Hall, 1996.
[15] H.C. Kim, Z. Ghahramani, Bayesian Gaussian process classification with the EM-EP algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 1948–1959.
[16] M. Kuss, C.E. Rasmussen, R. Herbrich, Assessing approximate inference for binary Gaussianprocess classification, J. Mach. Learn. Res. 6 (10) (2005) 1679–1704.
[17] J. LaSalle, Some extensions of Liapunov's second method, IRE Trans. Circuit Theory 7 (4) (1960) 520–527.
[18] L.G. Matallana, A.M. Blanco, J.A. Bandoni, Estimation of domains of attraction: a global optimization approach, Math. Comput. Model. 52 (3–4) (2010) 574–585.
[19] T.P. Minka, A Family of Algorithms for Approximate Bayesian Inference, Massachusetts Institute of Technology, 2001 Ph.D. dissertation.
[20] R.M. Neal, Regression and Classification using Gaussian Process Priors, Oxford University Press, 1999, pp. 475–501.
[21] H. Nickisch, C.E. Rasmussen, Approximations for binary Gaussian process classification, J. Mach. Learn. Res. 9 (2008) 2035–2078.
[22] W.O. Paradis, D.D. Perlmutter, Tracking function approach to practical stability and ultimate boundedness, AIChE J. 12 (1) (1966) 130–136.
[23] J.F. Quindlen, Data-Driven Methods for Statistical Verification of Uncertain Nonlinear Systems, Massachusetts Institute of Technology, 2018 Doctoral dissertation.
[24] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[25] C.W. Scherer, Dissipativity and integral quadratic constraints: tailored computational robustness tests for complex interconnections, IEEE Control Syst. Mag. 42 (3) (2022) 115–139.
[26] C.W. Scherer, J. Veenman, Stability analysis by dynamic dissipation inequalities: on merging frequency-domain techniques with time-domain conditions, Syst. Control Lett. 121 (2018) 7–15.

[27] Y. Shen, M. Bichuch, E. Mallada, Model-free learning of regions of attraction via recurrent sets, 2022, arXiv:2204.10372.
[28] V.V. Tadiparthi, R. Bhattacharya, Data-driven verification using efficient active learning, in: AIAA SCITECH 2022 Forum, 2022, p. 2086.
[29] G. Valmorbida, S. Tarbouriech, G. Garcia, Region of attraction estimates for polynomial systems, in: Proceedings of the 48th IEEE Conference on Decision and Control (CDC), 2009, pp. 5947–5952.
[30] A. Vannelli, M. Vidyasagar, Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems, Automatica 21 (1) (1985) 69–80.
[31] C.K. Williams, D. Barber, Bayesian classification with Gaussian processes, IEEE Trans. Pattern Anal. Mach. intell. 20 (1998) 1342–1351.
[32] C.K. Williams, C.E. Rasmussen, Gaussian Processes for Machine Learning, MIT Press, Cambridge, MA, 2006.
[33] L. Yang, K. Wang, L. Mihaylova, Online sparse multi-output Gaussian process regression and learning, IEEE Trans. Signal Inf. Process. Netw. 5 (2018) 258–272.
[34] V.I. Zubov, Mehtods of A. M. Lyaponuv and Their Application, Noordhoff, Groningen, 1964.