ТЕХНИЧЕСКИЕ НАУКИ | TECHNICAL SCIENCE

ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ
INFORMATICS, COMPUTER ENGINEERING AND MANAGEMENT

С.А. Романов [S. A. Romanov]
С. Е. Душин [S. E. Dushin],
И. И. Шпаковская [I. I. Shpakovshaya],

УДК 004.942
DOI: 10.37493/2307-910X.2022.3.1

ПОСТРОЕНИЕ ОБЛАСТИ ПРИТЯЖЕНИЯ НА ОСНОВЕ ФУНКЦИЙ ЛЯПУНОВА ДЛЯ НЕЛИНЕЙНЫХ СИСТЕМ ОБЩЕГО ВИДА

CONSTRUCTION OF THE DOMAIN OF ATTRACTION BASED ON LYAPUNOV FUNCTIONS FOR GENERAL NONLINEAR SYSTEMS

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина) Россия, Санкт-Петербург,
/ Saint Petersburg Electrotechnical University "LETI", St. Petersburg, Russian Federation

Аннотация
В работе предложен алгоритм для построения нейросетевых функций-кандидатов Ляпунова с целью максимизации оценки области притяжения положения равновесия. Для этого необходимо, чтобы инвариантное подмножество, задаваемое множеством уровня, занимало как можно большую долю полученной симуляцией эмпирической оценки области притяжения. Реализация данной цели осуществляется путём введения дополнительного слагаемого в функцию потерь. Алгоритм позволяет строить функции-кандидаты для систем с нелинейностями достаточно общего вида. Работа алгоритма проиллюстрирована на примере.
Ключевые слова: функция Ляпунова, область притяжения, искусственная нейронная сеть.
Abstract
The paper proposes an algorithm for constructing Lyapunov candidate neural network functions in order to maximize the estimate of the area of attraction of the equilibrium position. To do this, it is necessary that the invariant subset, given by the level set, occupy the largest possible share of the empirical estimate of the attraction region obtained by the simulation. The implementation of this goal is carried out by introducing an additional term in the loss function. The algorithm makes it possible to construct candidate functions for systems with non-linearities of a rather general form. The operation of the algorithm is illustrated by an example.
Keywords: Lyapunov function, attraction domain, artificial neural network.

Introduction

One of the most important problems in the analysis of control systems is to determine the stability of the equilibrium position of a closed system. As a rule, in applied problems it should be asymptotically or exponentially stable. However, there are cases when the area of attraction of a stable equilibrium position is smaller than the area of operation that is of interest in practice. Thus, for the target equilibrium position, it is desirable to be able to determine not only stability, but also the area of attraction.

In engineering applications, stability is often determined by the first Lyapunov method for a linearized system, but this method is not applicable to non-hyperbolic equilibria, that is, equilibria with zero real part of at least one of the eigenvalues of a linearized system.

There are various methods for determining the area of attraction, but perhaps one of the simplest and most common methods in practice is numerical simulation from the assigned area at the current step. In addition, based on the simulation results, one can also judge the stability of the equilibrium position, even in the non-hyperbolic case. However, this method relies on the reliability of numerical methods for solving the Cauchy problem, that is, it does not prove the stability of the fact that the resulting region is a subset of the true attraction region.

These two problems can also be solved jointly using the second (direct) Lyapunov method. If it is possible to construct a Lyapunov function that satisfies certain properties, then the stability of the equilibrium position is proved. In addition, the resulting Lyapunov function can be used to estimate the invariant subset of the attraction domain, which is defined by the level set.

The difficulty of applying the second Lyapunov method in practice lies in the absence of algorithms for constructing Lyapunov functions. Such algorithms exist for linear and polynomial systems [1], but it is known that there is no rigorous formal algorithm for systems of general form, since the problem of checking the positivity of functions of general form is unsolvable [2

In recent years, methods have been developed for constructing Lyapunov functions represented as a neural network [3; 4; 5]. This approach has a number of advantages. First, neural networks are universal approximators , that is, they are potentially capable of approximating a suitable Lyapunov function for arbitrary systems. Secondly, methods, software and hardware for their training are currently well developed. Thirdly, in recent years, methods and software tools for verifying neural networks have been developed, that is, checking the fulfillment of certain properties for given input values, for example, positivity in a certain area [6].

In most works in this area, the Lyapunov function is sought to prove stability, without the requirement to obtain a maximum estimate of the attraction domain. In this work, a step is taken to eliminate this shortcoming - an algorithm for constructing Lyapunov candidate functions is proposed in order to maximize the estimate of the attraction domain for general nonlinear systems.

.

1. Methods for constructing Lyapunov functions for general systems

The paper considers nonlinear dynamical systems defined by differential

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

or difference equations

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k), \qquad (2)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $f\colon \mathbb{R}^n \to \mathbb{R}^n$ is an arbitrary differentiable function. We will assume that the equilibrium position of interest to us exists and is zero: $f(0) = 0$, which can always be achieved by changing the coordinates.

Solution of systems (1)and (2)with the initial condition $\mathbf{x}_0 \in \mathbb{R}^n$ as $\mathbf{x}(\mathbf{t};\ \mathbf{x}_k)$ and $\mathbf{x}(\mathbf{k};\ \mathbf{x}_k)$ respectively, where $\mathbf{t} \in \mathbb{R}$ is continuous and $\mathbf{k} \in \mathbb{Z}$ is discrete time. We will assume that the conditions for the existence and uniqueness of the solution are satisfied.

Since the conditions of the second Lyapunov method [7] are used below, we briefly recall its formulation.

Theorem 1. Consider system (1), where $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^n$ and the zero equilibrium position is stable. If there exists a continuously differentiable function $V\colon \mathcal{D} \to \mathbb{R}$ such that

$V(0) = 0$,

$V(\mathbf{x}) > 0$ for everyone $\mathbf{x} \in \mathcal{D} / \{0\}$,

$\dot{V}(\mathbf{x}) \leq 0$ for everyone $\mathbf{x} \in \mathcal{D}$,

then the zero equilibrium position is Lyapunov stable. If, moreover, $\dot{V}(\mathbf{x}) < 0$ for all $\mathbf{x} \in \mathcal{D} / \{0\}$, then the zero equilibrium is asymptotically stable.

Similar theorems are also valid for discrete systems (2). In this case , the first difference is used $\Delta V = V(f(\mathbf{x}_k)) - V(\mathbf{x}_k)$instead of the derivative $\dot{V} = \nabla V(\mathbf{x}) f(\mathbf{x})$, and it is enough for the function $V$to be continuous.

Knowing the Lyapunov function, one can try to find an invariant subset of the attraction domain. Namely, the set:

$$\Omega_c = \{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) \leq c\} \qquad (3)$$

is an invariant subset of the attraction domain for all $c > 0$such that $\Omega_c \subseteq \{\mathbf{x} \in \mathbb{R}^n \mid \dot{V}(\mathbf{x}) < 0\}$in the continuous case or $\Omega_c \subseteq \{\mathbf{x} \in \mathbb{R}^n \mid \Delta V < 0\}$in the discrete case.

Thus, in order to maximize the estimate of the attraction domain, it is necessary to be able to construct Lyapunov functions and find such $c$that the set $\Omega_c$has the maximum volume.

The main disadvantage of this approach is the need to search for the Lyapunov function. For general non-linear systems, the most promising is the construction of candidate functions based on machine learning methods, which allow solving problems of large dimensions, but may not give any formal guarantees.

The idea of using machine learning methods to solve the problem of constructing the Lyapunov function is not new. In particular, works [6; 9; 10]. Later works differ in the type of representation of the Lyapunov function in the form of a neural network , the architecture of the neural network , and the loss function used, which formalizes the Lyapunov conditions. Thus, in [11], the Lyapunov function is sought in the form of a multilayer neural network $V_\theta(\mathbf{x})$with the number of inputs equal to the dimension of the system and with one output. The region of interest is set D, and the points on which the network is trained are generated in it.

In [12], a linear controller and a Lyapunov function of a closed system are simultaneously synthesized in a similar way. The loss function has the form (without taking into account the control for the article [12])

$$L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left(\max(0, -V_\theta(\mathbf{x}_i)) + \max\left(0, \dot{V}_\theta(\mathbf{x})\right)\right) + V_\theta^2(0) \qquad (4)$$

In [13], the construction of Lyapunov functions for high-dimensional systems is considered. The neural network has the form $\varphi_\theta : \mathbb{R}^n \to \mathbb{R}^m$, where m is the adjustable parameter. The Lyapunov function is sought in the form $V_\theta(x) = ||\varphi_\theta(x) - \varphi_\theta(0)||_2^2 + \delta \log(1 + ||x||^2)$. Here $\delta \log(1 + ||x||^2)$it can be replaced by another increasing function. Obviously, $V_\theta(0) = 0$for $V_\theta(x) > \delta \log(1 + ||x||^2) > 0$everyone $x \neq 0$. Hence, only one term is sufficient in the loss function:

$$L(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\max(0, \dot{V}_\theta(\mathbf{x}))^2 \qquad (5)$$

It is shown that for any $\varepsilon \in (0,1)$and $s \in \mathbb{N}$there exists a neural network of the $V_\theta(\mathbf{x})$above type with the ReLU activation function and size $O(\varepsilon^{-\frac{n}{s}})$such that for any Lyapunov function of $V \in C^s$the corresponding function $f$with zero equilibrium position

$$||V_\theta - V||_{L^\infty} := \max_{\mathbf{x}} |V_\theta(\mathbf{x}) - V(\mathbf{x})| \leq \varepsilon \qquad (6)$$

for the right parameters $\boldsymbol{\theta}$.

The work [4] is also devoted to the construction of Lyapunov functions for high-dimensional systems. It is assumed that the system admits the presence of a composite Lyapunov function, that is, there are functions V_i such that $V(\mathbf{x}) = \sum_{i=1}^{s} V_i(\mathbf{z}_i)$, where $\mathbf{z}_i$ is some subvector of the vector $\mathbf{x}$ that defines the i - th subsystem. The upper estimate for the number of subsystems is given as a parameter of the algorithm.

The work uses a single-layer network, divided into blocks, where each block is used for its subsystem. Let be $d_{\max}$ the maximum degree of the subsystem. It is shown that for single-layer networks with nonpolynomial infinitely differentiable activation functions, the (4)

$$N = O(n^{d_{\max}+1}\varepsilon^{-d_{\max}}), \quad (7)$$

that is, it grows exponentially not from the dimension of the system, but from the maximum dimension of the subsystem $d_{\max}$.

The loss function is given as:

$$L(\boldsymbol{\theta}) = \left(\left[\dot{V}_{\boldsymbol{\theta}}(\mathbf{x}) + ||\mathbf{x}||^2\right]_+\right)^2 + \upsilon\,(([V_{\boldsymbol{\theta}}(\mathbf{x}) + \alpha_1||\mathbf{x}||^2]_-)^2 + {}^2 \\ +([V_{\boldsymbol{\theta}}(\mathbf{x}) + \alpha_2||\mathbf{x}||^2]_+)^2) \quad (8)$$

and implements the requirements $\dot{V}_{\boldsymbol{\theta}}(\mathbf{x}) \le -||\mathbf{x}||^2$ and $\alpha_1(||\mathbf{x}||) \le V_{\boldsymbol{\theta}}(\mathbf{x}) \le \alpha_2(||\mathbf{x}||)$, where $\alpha_i(r) = c_i r^2$, $[\alpha]_- = \min(0, \alpha)$, $[\alpha]_+ = \max(0, \alpha)$.

These papers are devoted to the problem of constructing the Lyapunov function, but do not aim at maximizing the attraction domain. Thus, even if the loss function is zero for the entire region D, then it is not guaranteed that D is an attraction region, since the inclusion is not guaranteed $\Omega_c = \{\mathbf{x} \in \mathbb{R}^n \mid \dot{V}(\mathbf{x}) < 0\}$.

An exception is the work [5]. It uses the following idea. Let's assume that we know exactly the area of attraction S. Then, one can try to find such parameters θ of the Lyapunov function so that the area $\{\mathbf{x} \mid V_\theta(\mathbf{x}) \le c\}$ coincides with S, that is, the level lines $V_\theta(\mathbf{x}) = c$ coincide with the boundary of the stability region. That is, you can set the decision rule

$$\hat{y}_{\boldsymbol{\theta}}(\mathbf{x}) = \operatorname{sign}(c - V_{\boldsymbol{\theta}}(\mathbf{x})) \quad (9)$$

and get the classification problem: for any , $\mathbf{x}$ the label is defined as $y = 1$ if $\mathbf{x} \in S$ and -1 otherwise. The condition for the negative of the derivative must also be satisfied:
$y = +1 \implies \dot{V}_{\boldsymbol{\theta}}(\mathbf{x}) < 0$

In this paper, the Lyapunov function is searched for in the form $V_{\boldsymbol{\theta}}(\mathbf{x}) = \varphi_{\boldsymbol{\theta}}(\mathbf{x})^{\mathrm{T}}\varphi_{\boldsymbol{\theta}}(\mathbf{x})$, where $\varphi_{\boldsymbol{\theta}}(\mathbf{x})$ is the neural network . It is obvious that $V_{\boldsymbol{\theta}}(\mathbf{x}) \ge 0$. The loss function has the form:

$$l(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta}) = \max\left(0, -\mathbf{y}\left(c - V_{\boldsymbol{\theta}}(\mathbf{x})\right)\right) + \\ + \lambda\left(\frac{\mathbf{y}+1}{2}\right)\max(0, \Delta V_{\boldsymbol{\theta}}(\mathbf{x})) \quad (10)$$

The definition of labels $\mathbf{y}$ is carried out through simulation. The quadratic Lyapunov function obtained from the linearized system is taken as the first approximation. For it, the maximum c_k and area are found S_k. Then, to increase the area through simulation, labels are obtained in the area with a radius αc_k where $\alpha > 1$. In general, the algorithm does not guarantee convergence to the true area of attraction and its increase with each iteration, but always gives some internal estimate of it.

Thus, the problem can be formulated as follows: finding an algorithm for constructing a candidate function $V_{\theta}(\mathbf{x})$ and a constant c such that the region Ω_c is an invariant subset of the attraction region of as large a volume as possible. Provided that the found algorithm works for general systems.

2. Main result

The work [5] considers an online statement of the problem of increasing the attraction area, so the simulation is carried out at each iteration to assign labels to points. At the same time, using simulation, one can directly estimate the area of attraction, for example, by simulating in reverse time from some neighborhood of the equilibrium position or from points distributed inside the area of interest to us.

Let us set the desired domain D, in which we will evaluate the invariant subset of the attraction domain. Let us generate N points in the given area D_N that serve as initial conditions. By simulation, we obtain a division D_N into a set of S converging and U non-converging initial conditions. The set S can be interpreted as an empirical estimate of the area of attraction.
However, this method relies on methods for solving differential equations and does not prove that the obtained points really belong to the attraction region. In addition, for systems of order greater than four, the question arises of characterizing the attraction domain. For example, belonging to some point not in the D_N area of attraction without using simulation.

The paper proposes to obtain a set of points B belonging to the boundary of the obtained estimate of the attraction domain S. For example, the vertices of the convex hull of the points can act as such points S. Since the attraction domain does not have to be convex, a more accurate boundary can be obtained using -forms [14]. However, in this case, it is required to select the parameter α, which is difficult if the system dimension is higher than three.

To maximize the attraction region, we will look for such a candidate function $V_{\theta}(\mathbf{x})$ that, in addition to the Lyapunov conditions, the level lines $V_{\theta}(\mathbf{x}) = c$ coincide with the boundary of the stability region, that is, $V_{\theta}(\mathbf{x}_b) = c$ for all $\mathbf{x}_b \in B$. To do this, it is proposed to introduce an additional term into the loss function:

$$\frac{1}{N_B}\sum_{i-1}^{N_B}(c - V_{\theta}(\mathbf{y}_i))^2,$$

where $\mathbf{y}_i \in B, N_B = |B|$.

Since the number of boundary points in the general case is much less than the number of points from S, then during training it is proposed to form a batch from the union of two subbatches : from the set S and from the set B. Thus, at each training iteration, both internal and boundary points will be present.

The constant c is set during training, let's denote it c_t, for example, you can always put $c_t = 1$. However, after the completion of training, switching on $\Omega_{c_t} \subseteq \{\mathbf{x} \in \mathbb{R}^n \mid \dot{V}(\mathbf{x}) < 0\}$ (or its discrete analog) is guaranteed only when the loss function reaches zero. Otherwise, the problem arises of finding the maximum value of the constant c for which the specified inclusion is performed. In addition, in both cases, it is necessary to verify the obtained candidate function V_{θ}.

The obvious way is to jointly solve both problems, that is, to find the maximum value c for which it is possible to verify the Lyapunov conditions. However, since verification generally takes a long time, this approach seems inappropriate.

It seems more productive to test on a large validation set, also generated in the domain D. In this case, points are sought $\mathbf{x}_i$ from the set of initial conditions $S_V \cup S$ converging to zero for which at least one of the Lyapunov conditions is violated. The set of such points will be denoted as P. If it is non- empty , then the constant c can be taken equal to the minimum value of the candidate function on it:

$$c = \min_{\mathbf{x}_i \in P} V_{\boldsymbol{\theta}}(\mathbf{x}_i) - \varepsilon \tag{11}$$

where ε is some small constant. If the set is P empty, then points are generated from some extension of the area D, for which the minimum value is also searched. This procedure is especially simple if D is a hypercube. Then $P = 1.01 \cdot D \setminus D$ the search for a constant c is carried out according to the formula above.

Note that the set P of points for which at least one of the Lyapunov conditions is violated can be added to the training set and further training of the neural network can be carried out.

Thus, the algorithm for constructing a candidate function V_{θ} and evaluating a constant c is as follows:

Set the area of interest D. We generate in it evenly distributed points for training D_T.

We run the simulation for all points from D_T., we get a set of S_T. convergent to zero initial conditions.

We find the set of boundary points B_T.

We train a neural network $V_{\boldsymbol{\theta}}: \mathbb{R}^n \to \mathbb{R}$ with a loss function:

$$L(\boldsymbol{\theta}) = \frac{1}{N_s}\sum_{i=1}^{N_S}\Big(\max\big(0, -V_{\boldsymbol{\theta}}(\mathbf{x}_i)\big) + \max\big(0, \dot{V}_{\boldsymbol{\theta}}(\mathbf{x}_i)\big)\Big) +$$
$$+\alpha_1 \frac{1}{N_B}\sum_{i=1}^{N_B}(c_t - V_{\boldsymbol{\theta}}(\mathbf{y}_i))^2 + \alpha_2 V_{\boldsymbol{\theta}}^2(0),$$

where $N_S = |S_T|,\ N_B = |B_T|,\ \alpha_1, \alpha_2 > 0$ are positive weights, c_t is a constant during training.

We generate D evenly distributed points in the area for validation D_V and find the set S_V of initial conditions converging to zero.

We obtain a set P of points from $S_T \cup S_V$ for which at least one of the Lyapunov conditions is not satisfied.

If the set is P empty, then we generate points in the neighborhood and $D: P = D_{new} \setminus D$ find a constant c:

$$c = \min_{\mathbf{x}_i \in P} V_{\boldsymbol{\theta}}(\mathbf{x}_i) - \varepsilon \tag{12}$$

and complete the algorithm.

If the number of iterations is less than the specified one, then we add the set P to the training sample S_T and go to step 4. Otherwise, we find the constant c using formula (12)and complete the algorithm.

Example

To illustrate the method, let us estimate the area of attraction of the zero equilibrium position of system (12):

$$\begin{cases} \dot{x}_1 = -\frac{1}{4}x_1 + \log(1 + x_2); \\ \dot{x}_2 = -\frac{3}{8}x_1 - \frac{1}{5}x_1x_2 + \left(\frac{1}{8}x_1 - x_2\right)\cos x_1 \end{cases} \qquad (13)$$

generate $N = 16000$ points in the range $x_1 \in [-7;\ 4],\ x_2 \in [-1{,}2;\ 11{,}5]$. The estimate of the area of attraction obtained using the simulation is shown in Figure 1 by a solid line, the boundary points obtained by constructing the convex hull are shown by filled circles.

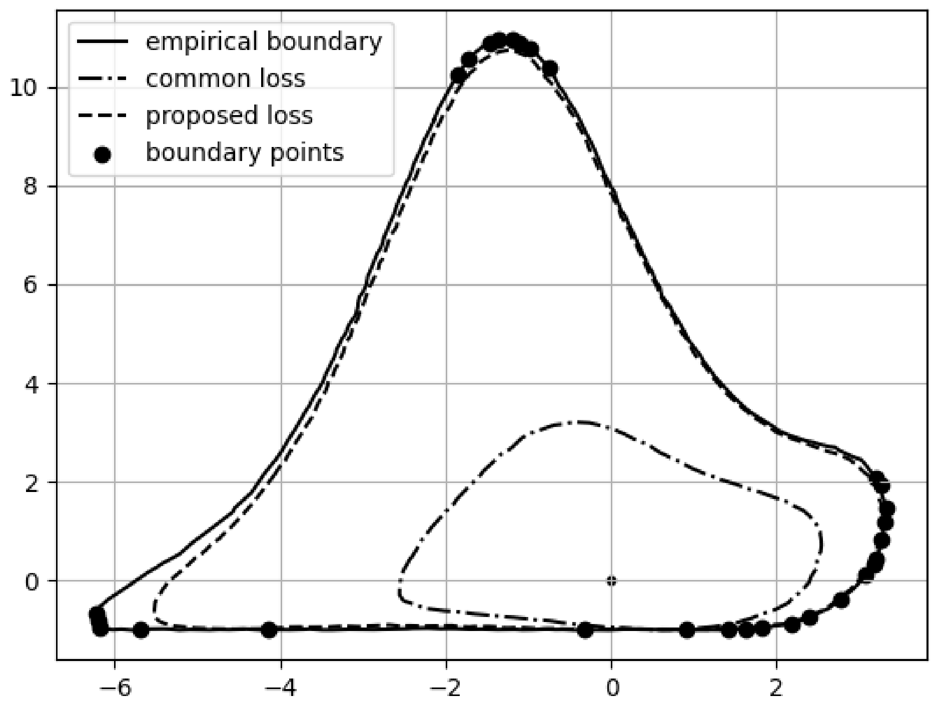

Figure 1 - Evaluation of the area of attraction

Let 's define $\alpha_1 = 1,\ \alpha_2 = 0.1,\ c_t = 1$, and train a neural network with 3 hidden layers of 128 neurons with the softplus activation function . For validation , we will generate 128000 points and we will carry out the training- validation procedure 10 times. The resulting area of attraction for the loss function of the form (4)is shown in Figure 1 by a dash-dotted line, for the proposed loss function it is a dashed line.

Note that in [15] the area of attraction proved for this system is a circle with a radius of 0.51, and the learning process with the loss function from [5] diverges.

Conclusion

In this paper, an algorithm is obtained for constructing Lyapunov candidate functions in order to maximize the attraction domain. Although the proposed algorithm can be used for general nonlinear systems, it only allows one to construct candidate functions for which the Lyapunov conditions are satisfied at some finite set of points. Thus, in the future, it is necessary to carry out verification, that is, to prove that these conditions are met for all points from the estimated area. In the current work, verification issues are not considered, it is proposed to use the dReal solver . These issues will be the subject of further research.

ЛИТЕРАТУРА

1. Pablo Parrilo. "Structured Semidenite Programs and Semialgebraic Geometry Methods in Robustness and Optimization". В: PhD thesis, 2000

2. Daniel Richardson. "Some Undecidable Problems Involving Elementary Functions of a Real Variable". В: The Journal of Symbolic Logic 33.4 (1968), с. 514— 520. issn: 00224812. url: http://www.jstor.org/stable/2271358 (дата обр. 08.04.2022)

3. Hongkai Dai и др. Lyapunov-stable neural-network control. 2021. arXiv: 2109.14152

4. Lars Gr¨une. "Computing Lyapunov functions using deep neural networks". В: Journal of Computational Dynamics 8.2 (2021), с. 131—152.

5. Spencer M. Richards, Felix Berkenkamp и Andreas Krause. The Lyapunov Neural Network: Adaptive Stability Certification for Safe Learning of Dynamical Systems. 2018. arXiv: 1808.00924

6. Guy Katz и др. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. 2017. doi: 10.48550/ARXIV.1702.01135. url: https://arxiv.org/abs/1702.01135

7. Ляпунов А.М. Общая задача об устойчивости движения. Москва: Гостехиздат; 1950. 472 с

8. Y. Long и M.M. Bayoumi. "Feedback stabilization: Control Lyapunov functions modelled by neural networks". В: Proceedings of 32nd IEEE Conference on Decision and Control. IEEE. 1993, с. 2812—2814

9. D.V. Prokhorov. "A Lyapunov machine for stability analysis of nonlinear systems". В: Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94). Т. 2. 1994, 1028—1031 vol.2. doi: 10.1109/ICNN. 1994.374324

10. Gursel Serpen. "Empirical approximation for Lyapunov functions with artificial neural nets". В: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. Т. 2. IEEE. 2005, с. 735—740

11. Alessandro Abate и др. "Formal synthesis of lyapunov neural networks". В: IEEE Control Systems Letters 5.3 (2020), с. 773—778

12. Ya-Chien Chang, Nima Roohi и Sicun Gao. Neural Lyapunov Control. 2020. arXiv: 2005.00611

13. Nathan Gaby, Fumin Zhang и Xiaojing Ye. "Lyapunov-net: A deep neural network architecture for Lyapunov function approximation". В: arXiv preprint arXiv:2109.13359, 2021

14. H. Edelsbrunner, D. Kirkpatrick и R. Seidel. "On the shape of a set of points in the plane". В: IEEE Transactions on Information Theory 29.4 (1983), с. 551—559. doi: 10.1109/TIT.1983.1056714

15. Graziano Chesi. "Domain of attraction: estimates for non-polynomial systems via LMIs". В: 2005

REFERENCES

1. Pablo Parrilo. "Structured Semidenite Programs and Semialgebraic Geometry Methods in Robustness and Optimization". V: PhD thesis, 2000

2. Daniel Richardson. “Some Undecidable Problems Involving Elementary Functions of a Real Variable”. V: The Journal of Symbolic Logic 33.4 (1968), s. 514— 520. issn: 00224812. url: http://www.jstor.org/stable/2271358 (data obr. 08.04.2022)

3. Hongkai Dai i dr. Lyapunov-stable neural-network control. 2021. arXiv: 2109.14152

4. Lars Gr¨une. “Computing Lyapunov functions using deep neural networks”. V: Journal of Computational Dynamics 8.2 (2021), s. 131—152.

5. Spencer M. Richards, Felix Berkenkamp i Andreas Krause. The Lyapunov Neural Network: Adaptive Stability Certification for Safe Learning of Dynamical Systems. 2018. arXiv: 1808.00924

6. Guy Katz i dr. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. 2017. doi: 10.48550/ARXIV.1702.01135. url: https://arxiv.org/abs/1702.01135

7. Lyapunov A.M. Obshchaya zadacha ob ustoichivosti dvizheniya. Moskva: Gostekhizdat; 1950. 472 s

8. Y. Long i M.M. Bayoumi. “Feedback stabilization: Control Lyapunov functions modelled by neural networks”. V: Proceedings of 32nd IEEE Conference on Decision and Control. IEEE. 1993, s. 2812—2814

9. D.V. Prokhorov. “A Lyapunov machine for stability analysis of nonlinear systems”. V: Pro-ceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94). T. 2. 1994, 1028—1031 vol.2. doi: 10.1109/ICNN. 1994.374324

10. Gursel Serpen. “Empirical approximation for Lyapunov functions with artificial neural nets”. V: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. T. 2. IEEE. 2005, s. 735—740

11. Alessandro Abate i dr. “Formal synthesis of lyapunov neural networks”. V: IEEE Con-trol Systems Letters 5.3 (2020), s. 773—778

12. Ya-Chien Chang, Nima Roohi i Sicun Gao. Neural Lyapunov Control. 2020. arXiv: 2005.00611

13. Nathan Gaby, Fumin Zhang i Xiaojing Ye. “Lyapunov-net: A deep neural network archi-tecture for Lyapunov function approximation”. V: arXiv preprint arXiv:2109.13359, 2021

14. H. Edelsbrunner, D. Kirkpatrick i R. Seidel. “On the shape of a set of points in the plane”. V: IEEE Transactions on Information Theory 29.4 (1983), s. 551—559. doi: 10.1109/TIT.1983.1056714

15. Graziano Chesi. “Domain of attraction: estimates for non-polynomial systems via LMIs”. V: 2005

ОБ АВТОРАХ / ABOUT THE AUTHORS

Романов Сергей Алексеевич, научный сотрудник кафедры автоматики и процессов управления Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» имени В. И. Ульянова (Ленина); ORCID 0000-0002-8647-9096; e-mail: sarmnv@gmail.com

Romanov Sergey Alekseevich, Researcher at the Department of Automation and Control Processes of the St. Petersburg State Electrotechnical University "LETI" named after V.I. Ulyanov (Lenin); ORCID 0000-0002-8647-9096; e-mail: sarmnv@gmail.com

Душин Сергей Евгеньевич, доктор технических наук, профессор кафедры автоматики и процессов управления Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» имени В. И. Ульянова (Ленина), Санкт-Петербург, ул. Профессора Попова, дом 5, e-mail: dushins@yandex.ru, +7 (921) 970-46-31

Dushin Sergey Evgenievich, Doctor of Technical Sciences, Professor of the Department of Automation and Control Processes of the St. Petersburg State Electrotechnical University "LETI" named after V.I. Ulyanov (Lenin), St. Petersburg, Professora Popova str., house 5, e-mail: dushins@yandex.ru , +7 (921) 970-46-31

Шпаковская Ирина Игоревна, ассистент кафедры автоматики и процессов управления Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» имени В. И. Ульянова (Ленина); ORCID 0000-0002-5851-6160; e-mail: iishpakovskaia@etu.ru, Санкт-Петербург, ул. Профессора Попова, дом 5, +7 (950) 049-71-40

Shpakovskaya Irina Igorevna, Assistant of the Department of Automation and Control Processes of the St. Petersburg State Electrotechnical University "LETI" named after V. I. Ulyanov (Lenin); ORCID 0000-0002-5851-6160; e-mail: iishpakovskaia@etu.ru , St. Petersburg, Professora Popova str., 5, +7 (950) 049-71-40

Дата поступления в редакцию: 12.06.2022
После рецензирования:13.08.2022
Дата принятия к публикации:07.09.2022