IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 June 2024, accepted 8 July 2024, date of publication 11 July 2024, date of current version 22 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3426566

RESEARCH ARTICLE

Neural Nonparametric Stability Indicator for Self-Excited Dynamical Systems

KRZYSZTOF LALIK
Faculty of Mechanical Engineering and Robotics, AGH University of Krakow, 30-059 Kraków, Poland
e-mail: klalik@agh.edu.pl

This work was supported by the Research Project through the Program "Excellence Initiative-Research University" for the AGH University of Krakow.

ABSTRACT This article introduces a novel methodology employing deep-learning neural networks to estimate Lyapunov functions in dynamic systems accurately. Unlike traditional parametric approaches, our method is model-free, enabling adaptability to various system dynamics without prior assumptions. We also present a new strategy for generating Lyapunov functions using neural networks, enhancing stability assessments' precision and robustness. The effectiveness of this approach is validated through comparative analysis within a self-excited acoustical system (SAS) applied across diverse materials. This research demonstrates a new approach to differentiate between material stress states and the presence of defects, as evidenced by variations in the potential funnel's dimensions of the Lyapunov function and specific asymmetries indicative of defective states. Key contributions include the development of a flexible, neural network-based framework for stability assessment and a new application for structural health monitoring. By leveraging this model-free neural approach, we provide a powerful tool for determining the stability of nonlinear dynamical systems and enhancing defect detection processes, significantly advancing control theory and material science.

INDEX TERMS Auto-oscillators, deep learning, nonlinear stability, self-excited systems.

I. INTRODUCTION

Structural Health Monitoring (SHM) is increasingly recognized as a cornerstone of modern engineering practices, vital for the assessment and management of the integrity of structures under operational conditions. SHM systems aim to detect, localize, and quantify damages and stresses within materials in real-time, thus ensuring safety, durability, and efficient infrastructure maintenance. It is particularly crucial in sectors such as aerospace, civil engineering, and renewable energy, where the failure of key components can lead to significant economic loss and, more importantly, endanger human lives.

Advancements in sensor technologies, data analytics, and computational methods have significantly propelled the field of SHM. Reference [9] underscored the evolution of SHM systems capable of autonomously monitoring the structural integrity of wind turbines, demonstrating how early detection of defects can prevent costly downtime and catastrophic failures. This highlights the shift towards predictive maintenance strategies, optimizing repair operations and extending the lifespan of critical infrastructure.

Furthermore, integrating machine learning and deep learning techniques has revolutionized the analysis and interpretation of data obtained from SHM systems. Reference [3] showed the application of deep neural networks in identifying and quantifying damage in composite materials, illustrating the potential for significant improvements in accuracy and reliability over traditional methods. Such advancements underscore the growing ability of SHM systems to handle complex data sets and provide actionable insights into the health of structures.

The role of SHM in enhancing the resilience of urban infrastructure against natural disasters has also gained prominence. Reference [27] discussed the implementation of SHM in earthquake-prone areas, where real-time data collection

The associate editor coordinating the review of this manuscript and approving it for publication was Min Wang.

and analysis can inform emergency response strategies and post-disaster assessments. This application of SHM not only aids in immediate response efforts but also contributes to the design of more resilient future structures.

Moreover, the development of wireless sensor networks has expanded the applicability and efficiency of SHM systems. As noted by [13], wireless sensors offer a cost-effective and scalable solution for monitoring large-scale structures, enabling widespread adoption across various sectors. This evolution towards more accessible and robust SHM technologies promises to democratize the benefits of advanced monitoring capabilities.

The predictive power of SHM systems, enhanced by artificial intelligence, offers a promising avenue for identifying and predicting the onset of structural issues. Reference [2] emphasized the potential of integrating SHM data with predictive models to forecast material degradation and fatigue, facilitating proactive maintenance and resource allocation.

A. BACKGROUND

Recent research demonstrates that integrating artificial intelligence (AI) and modern sensory systems in defectoscopy offers promising perspectives for enhancing inspection systems' precision, speed, and adaptive capabilities.

In the context of detecting rail surface defects, [6] showcases the potential of deep convolutional neural networks (CNNs) in industrial applications, emphasizing that despite their high effectiveness, these methods may encounter difficulties in defect localization and real-time data processing. Meanwhile, [38] proposes using deep learning for detecting defects in composites using highly nonlinear solitary waves (HNSWs), pointing to new possibilities for improving detection accuracy while maintaining inference speed. It opens the path to real-time, non-invasive assessment of composite materials, which has previously been a challenge.

Simultaneously, [8] explores the application of AI algorithms in detecting defects in cold forging processes, indicating the effectiveness of CNNs in identifying delamination. Similarly, the study by [7] focuses on defectoscopy in metallurgy, using machine learning to detect defects in additive manufacturing, further highlighting the versatility and potential of AI in various industrial applications.

As introduced in [37], the application of deep neural networks for the automated detection of defects in sewer systems, utilizing CCTV visualization data and the YOLOv3 algorithm, achieves impressive detection accuracy. They underlined the importance of AI in modernizing monitoring methods of critical urban infrastructure elements.

In work by [15], attention is focused on tyre defect inspection, combining image processing with unsupervised learning and VGG-16 neural networks to meet the challenges associated with automatic damage classification. This work shows the potential for integrating various AI methods to improve the accuracy and efficiency of inspection processes in the automotive industry.

The authors in [32] proposed an innovative method for detecting defects on smooth surfaces, such as laptop panels, using phase maps and the Vovecwnet network, resulting in exceptional detection accuracy. This study opens new perspectives for using AI in quality control in producing electronic devices.

The research by [28] demonstrated the capabilities of integrating data from multiple sensors using deep learning to characterize defects in steel elements, marking a milestone in monitoring the condition of industrial components.

In [14], the use of machine learning for detecting wheel defects was presented, offering a perspective on enhancing the safety and reliability of railway transport through the automated analysis of wheel vertical forces.

The authors [35] proposed the ''StressNet'' model, utilizing deep learning to predict maximum internal stress based on crack propagation and initial stress data. Their approach, combining the Temporal Independent Convolutional Neural Network (TI-CNN) and Bidirectional Long Short-term Memory (Bi-LSTM) Network, allows for accurate prediction of the evolution of maximum internal stress, overcoming the limitations of high computational costs of traditional methods.

Authors of [31] presented the use of machine learning methods for predicting material properties, such as strength, hardness, or elasticity, based on data from small-scale strength tests. Their study emphasizes the potential of ML in generating material property information, offering an alternative to time-consuming and expensive material tests.

In [30], a deep learning methodology framework was introduced for predicting post-damage stress distribution and failure patterns in composite material microstructures. Their work demonstrates how advanced deep learning models can accurately predict some of the most complex phenomena in solid mechanics.

Deep neural network (DNN) for rapid prediction of the structure-property relationship in additively manufactured steels was discussed in [34]. Their DNN model, using data from crystal plasticity finite element (CPFE) modelling, proves how deep learning can serve as a quality control tool, predicting mechanical properties based on microstructure with significantly lower computational costs.

These studies collectively showcase the growing interest in using AI and deep learning in defectoscopy and material characterization. However, the presented approaches only detect defects or stresses in the examined material. A system combining such measurements was presented in [16]. It is a self-excited system based on the elasto-acoustic effect. The influence of this effect has been described more broadly in [17]. The effectiveness of such a measurement system for various environments, such as mining [22], [24], the cement industry [12], [19], in bridges [4], or in heavy machinery [23], as well as in different materials [18], [21],

has been confirmed. However, the system also had two principal drawbacks. The first was related to the so-called frequency shift or the uncontrolled transition of the system to the next limit cycle. This problem was solved using an advanced filtration system [5] and implementing neural filters in the feedback loop [20]. The second problem was the interpretability of the results. Unfortunately, quite often, the change in frequency of the self-excited system had various causes, which were not entirely interpretable. It was mainly the case when there were defects and inclusions apart from stresses in the examined object. It could turn out that the read frequency did not result from a specific stress level but from defects. This article presents an innovative method to eliminate the inconvenience related to the lack of interpretability of the results, which uses deep learning neural networks.

B. CONTRIBUTIONS

This article describes an innovative approach to analyzing the stability of self-excited systems using neural networks and deep learning techniques. The novelty of this study lies in integrating algorithms and deep networks for modelling dynamic systems and estimating Lyapunov functions. This paper presents a new approach to a nonparametric neural method of assessing system stability over a wider area of state space, surpassing the capabilities of traditional deterministic methods.

Previously, self-excited systems were primarily analyzed using parametric approaches, relying mainly on mathematical models derived from physical laws. However, such methods are ineffective in complex, nonlinear dynamics, where precise mathematical relationships are not fully understood or are too complicated to model accurately. This article demonstrates using neural networks to learn these dynamics from data without explicit parametric models directly. As a data-based approach, it constitutes a novelty in the dynamic analysis of self-excited systems.

The most significant novelty of the solution is the development of a method for assessing the stability of a self-excited system using Lyapunov functions. The classical approach requires precise knowledge of the system's parameters and intuition to indicate a Candidate Lyapunov Function (CLF). In the context of stability analysis of dynamic systems, a candidate for a Lyapunov function is a proposed function that can demonstrate the system's stability according to Lyapunov's criteria. Unfortunately, in the case of nonlinear systems, which by definition are self-excited systems, there are significant difficulties in determining the CLF, and often, the knowledge of the system's parameters is unavailable. Moreover, if the dynamic model of the self-excited system is defined using an additional neural network, its parametric description is unavailable. Therefore, traditional stability analysis methods require knowledge of the system's dynamics and often involve simplifying assumptions that may not apply in complex or highly nonlinear systems. The proposed approach, in contrast, allows for determining the system's stability in the sense of Lyapunov in a specified state space in a nonparametric manner, which is a significant innovation.

II. METHODOLOGY

The key element of the presented approach is the application of deep learning networks for modelling and optimizing the operation of the self-excited system. This chapter describes the measurement system and the self-excitation phenomenon to demonstrate how experimental data were provided for training the neural network, which estimates the dynamic model of the system. An algorithm is presented that allows for the creation of the model, followed by a method for determining the dynamic gain of the feedback path using the deep network. Subsequently, a method is presented for using another neural network to determine the Lyapunov function, indicating the entire system's stability. We describe how this function was designed, trained, and verified to ensure a reliable stability assessment. The summary presents an overall framework that combines traditional stability analysis techniques with modern machine and deep learning methods, offering an innovative tool for researching and designing stable self-excited dynamic systems.

A. SELF-EXCITED ACOUSTICAL SYSTEM

The Self-Excited Acoustic System (SAS) enables the generation and measurement of self-excited vibrations in structures and materials. Utilizing a specific configuration in which the vibration emitter and receiver work in a feedback loop with an appropriate power amplifier, the SAS system allows for precise tracking of changes in resonance and the speed of acoustic wave propagation depending on the load. Such a structure of the SAS system makes it an excellent data source for analyzing the dynamics of self-excited dynamical systems. The schematic of the SAS system is shown in Fig. 1. The system can fundamentally be divided into two parts. The first part consists of the object under investigation. The second part of the system is the executive part, which has two essential elements: the exciter (E) - a piezoelectric actuator, and the receiver (R) - in this role, a piezoelectric accelerometer sensor is used - they are the essential components of the self-excited system. Additionally, a conditioner and amplifier match and appropriately amplify the signal. The amplifier, exciter (E), conditioner, and accelerometer sensor (R) realize positive feedback. Following the conditioner is a measurement card and an FPGA system with a Real-Time Operating System (RTOS), which measures the signal from the conditioner and its appropriate amplification and then sends it to the amplifier. Their task is to read the current frequency of the self-excited system and introduce controllable amplification to the signal. The main factor affecting the frequency of the self-excited system is the change in the wave propagation speed and, thereby, the change in the wave transit time through the object under investigation. Changes in the speed

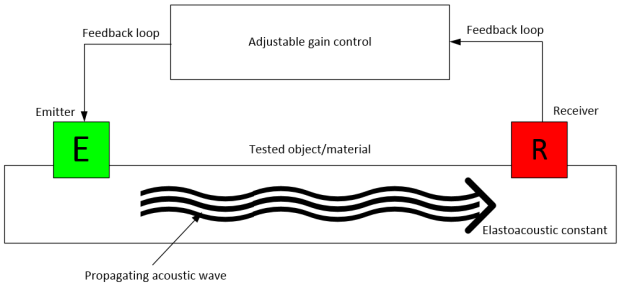


FIGURE 1. SAS system scheme.

of acoustic wave propagation caused by stress change are minor. This necessitates the measurement of wave pulse transit times with nanosecond accuracy. However, changes in this delay significantly affect the frequency of the SAS system's limit cycle, which is easily measurable.

B. DYNAMIC SYSTEM ESTIMATION

The data obtained through the SAS system are used as batch input for neural network modelling. The algorithm for identifying the dynamic model has been described in [33] and [36] and is the LTSM. The LSTM (Long Short-Term Memory) algorithm is a particular type of neural network capable of learning long-term dependencies in sequential data. The basic idea behind LSTM is its ability to selectively remember and forget information, allowing the model to store and utilize long-term dependencies in the data efficiently. This makes LSTM highly effective in many tasks related to data sequences, outperforming traditional RNN models. The outputs are produced in accordance with functions 1-6, defined in [36]:

$$i[n] = \sigma(W_{ii}x[n] + b_{ii} + W_{hi}h[n-1] + b_{hi}) \quad (1)$$
$$f[n] = \sigma(W_{if}x[n] + b_{if} + W_{hf}h[n-1] + b_{hf}) \quad (2)$$
$$\tilde{c}[n] = \tanh(W_{ic}x[n] + b_{ic} + W_{hc}h[n-1] + b_{hc}) \quad (3)$$
$$o[n] = \sigma(W_{io}x[n] + b_{io} + W_{ho}h[n-1] + b_{ho}) \quad (4)$$
$$c[n] = f[n]c[n-1] + i[n]\tilde{c}[n] \quad (5)$$
$$h[n] = o[n]\tanh(c[n]) \quad (6)$$

where:

- $i[n], f[n], \tilde{c}[n], o[n]$ are respectively the input gate, forget gate, cell state candidate, and output gate at the time step,
- σ denotes the sigmoid function,
- W, b are weights and biases respectively for different parts of the LSTM: ii, if, ic, io (for input weights) and hi,hf,hc,ho (for hidden state weights).
- $x[n]$ is the input, $c[n]$ is the cell state, and $h[n]$ is the hidden state at time step n.

A loss function was used to reconstruct the nonlinear dynamic system model based on the input and output signal from the SAS system. It minimizes the difference between the system's predicted and actual output values. In the case of nonlinear dynamic systems that can exhibit complex behaviours such as hysteresis, dynamics at different time scales, instabilities, and strong state dependencies, the standard MSE (Mean Squared Error) function is insufficient. Therefore, a loss function L, including the gradient, was used, which, in addition to minimizing the direct difference between outputs, pays attention to matching the time gradients of the outputs, defined by equation 7.

$$L = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \lambda \sum_{i=1}^{N-1}\left(\frac{d\hat{y}_i}{dt} - \frac{dy_i}{dt}\right)^2 \quad (7)$$

where: $\hat{y}_i$ and y_i are the predicted and actual output values of the system at the i-th time step, $\frac{d\hat{y}_i}{dt}$ and $\frac{dy_i}{dt}$ are the predicted and actual time gradients of the output, respectively, λ is a regularization parameter that balances the importance of the output value fitting term against the gradient matching term and N is the total number of time steps or data points in the dataset

The training process for the recurrent LSTM networks was based on data from 50 experiments conducted using the SAS system. Each experiment lasted 10 minutes and was carried out at a sampling frequency of 50 kHz, which allowed for collecting a large amount of data. For each experiment, the data were divided into vectors corresponding to 1 second, which enabled the creation of 30,000 data vectors intended for network training.

In preparation for the training process, the collected data were divided into three main sets: training, validation, and test sets. The training set, constituting about 70-80% of the total data, was used directly for training the model. The validation set (15% of the data) was used to adjust hyperparameters and monitor model performance during training to avoid overfitting. The test set was used for the final evaluation of the model, checking its ability to generalize on new data. A computer equipped with an Intel Core i9-10900K processor and 32 GB of RAM was used. The algorithm was computed using a GPU on an NVIDIA RTX 3090 card, which has 24 GB of GDDR6X memory, 10496 CUDA cores, and provides high memory throughput.

In this way, a model of the dynamic system was obtained, which could then be used for training subsequent components of the proposed system. In Fig. 2, an example phase portrait obtained for the neural network model of the dynamic system is presented.

C. LYAPUNOV FUNCTION DESIGN

The Lyapunov function is a key tool in the stability theory of dynamical systems. In classical approaches for linear systems, to utilize the Lyapunov function to determine the system's stability within a certain area of state space, it must meet certain conditions:

1) There must exist a function $V : \mathbb{R}^n \to \mathbb{R}$, where $\mathbb{R}^n$ is the state space of the system, such that V is continuous, differentiable concerning its arguments, and $V(\mathbf{x})$ is scalar.

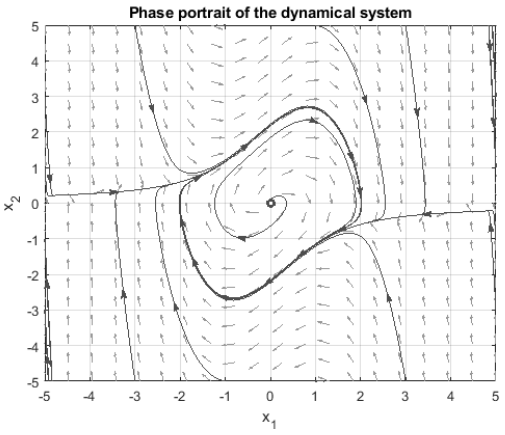


FIGURE 2. Phase portrait for a neural model of the dynamical system.

2) The Lyapunov function $V(\mathbf{x})$ must be positively defined in the area of state space of interest, meaning that $V(\mathbf{x}) > 0$ for all $\mathbf{x} \neq \mathbf{0}$ and $V(\mathbf{0}) = 0$.
3) The derivative of the Lyapunov function $V(\mathbf{x})$ over time, $\dot{V}(\mathbf{x})$, must be non-positive for all $\mathbf{x}$ in the area of interest, i.e., $\dot{V}(\mathbf{x}) \leq 0$. This means that the value of the Lyapunov function must either remain constant or decrease over time for every state $\mathbf{x}$ outside the equilibrium point.

If we have a system described by the differential equation (8):

$$\dot{\mathbf{x}} = f(\mathbf{x}) \tag{8}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is the function describing the dynamics of the system, and $\mathbf{x} \in \mathbb{R}^n$ is the state vector, then the derivative of V over time is given by equation (9).

$$\dot{V}(\mathbf{x}) = \nabla V(\mathbf{x}) \cdot f(\mathbf{x}) \tag{9}$$

where $\nabla V(\mathbf{x})$ is the gradient of the function V with respect to $\mathbf{x}$, and the $\cdot$ represents the dot product.

For linear systems, complete parametric knowledge of the system's dynamics is necessary [29]. On the other hand, for nonlinear systems, the approach proposed in [1], [11], and [26], based on the K_∞ method, is used. For a nonlinear system, defined by equation (10), it can be proven that there exists a trajectory that behaves asymptotically stable if it only meets the assumptions (11) and (12).

$$\dot{\mathbf{x}} = f(\mathbf{t}, \mathbf{x}), \mathbf{x}(\mathbf{t_0}) = \mathbf{x_0} \tag{10}$$

If $D \subset \mathbb{R}^n$ is the domain that includes the stable region B_μ, and the function $V(x)$ is continuously differentiable function, which satisfies:

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \tag{11}$$

$$\frac{\partial V}{\partial x} f(t, x) \leq -\mathscr{W}_3(x), \quad \forall x \in D \text{ with } \|x\| \geq \mu, \forall t \geq 0, \tag{12}$$

where α_1, α_2 are K-class functions, and $\mathscr{W}_3(x)$ is a continuous, positively defined function. In this case, there exists such $c > 0$ that $\Omega_c = V(x) \leq c$ is a compact set contained in D. Consequently, Ω_c is positively invariant for the system (10), and there exists a function β of class $\mathscr{KL}$ such that for every initial state $x(t_0) \in \Omega_c$, the solution (10) is asymptotically stable.

As proven in [11], there exist constants c_1, c_2, c_3, such that conditions (11) and (12) can be written in the form of (13) and (14). Therefore, the system is asymptotically stable for arbitrarily large μ.

$$c_1\|x\|^2 \leq V(x) \leq c_2\|x\|^2 \tag{13}$$

$$\frac{\partial V}{\partial x} f(t, x) \leq -c_3\|x\|^2, \quad \forall x \in D \text{ with } \|x\| \geq \mu, \ \forall t \geq 0 \tag{14}$$

The conditions defined for a candidate Lyapunov function, which can be considered as a solution to partial differential inequalities, will serve to determine the reward function for the neural network, which is to estimate in a nonparametric way an appropriate Lyapunov function that meets the conditions of asymptotic stability.

Conditions (11) and (12) demonstrate asymptotic stability for the Lyapunov function. The next step is to confirm the convergence of the function estimated by the neural network, which can be proven with the Universal Approximation Theorem [10], [25]. Assuming that there exists any continuous Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$, , then there exists a neural network $W(x)$ for every $\epsilon > 0$, that the inequality 15 is satisfied.

$$\sup_{x \in K} \|V(x) - W(x)\| < \epsilon \tag{15}$$

where:

- $K \subset \mathbb{R}^n$ is a closed and bounded subset,
- $W(x)$ is the function realized by the network, consisting of a linear combination of activation functions g, i.e.,

$$W(x) = \sum_{i=1}^{N} w_i g(v_i^T x + b_i),$$

- N is the number of neurons in the hidden layer,
- w_i, v_i, and b_i are the weights and bias, respectively, which need to be adjusted so that $F(x)$ can approximate $f(x)$ with an error less than ϵ,
- g is the nonlinear activation function applied in the neurons.

Directly from this theorem, it can be concluded that a neural network can obtain an arbitrarily accurate estimate of the Lyapunov function for a given system. However, to ensure that the neural network seeks this specific function, it is necessary to apply a reward function that fulfils the assumptions specified in conditions (13) and (14).

To estimate the function $W(x)$, a network with 2 hidden layers, each with 256 neurons and using the Softmax

activation function, was used. The cost function used to train this network consists of three components:

$$L = h_1 L_1^2 + h_2 L_2^2 + h_3 L_3^2 \tag{16}$$

where h_1, h_2 and h_3 are the prelimenary user-defined hyperparameters and L_1, L_2 and L_3 are the terms inducted from stability conditions (11) and (12).

A discussion on the physical interpretation of each factor is necessary to understand the roles of hyperparameters. Hyperparameter h_1 is responsible for the reward for the network's performance near the equilibrium point. One frequently observed effect is that the Lyapunov function estimate achieves many local minima that are very close to the equilibrium point. Such an estimate obviously does not meet the CLF (Candidate Lyapunov Function) criteria, even though it is correct in most of the RoA (Region of Attraction). Increasing the ratio of hyperparameter h_1 relative to the other parameters forces the neural network to pay more attention to estimating the Lyapunov function near the equilibrium. The physical interpretation involves stabilizing the system around the equilibrium point, which is crucial for ensuring the global stability of the system. Hyperparameter h_2 represents the weight of the lower bounding function of the CLF. By increasing its value, the reward for the network for not exceeding the lower bound increases. The physical interpretation is that this expands the RoA for the estimated function. If the CLF does not exceed the lower bound, it means that the function $V(x)$ does not have inflexions, and thus, there is no change in the gradient's sign, which indirectly results from the requirements for the Lyapunov function. Increasing the value of h_2 allows for more global estimation of the Lyapunov function, which is crucial for the stability of the entire system over a broader range of state variables. Hyperparameter h_3 is responsible for the reward for the speed at which the dynamic system is brought to the equilibrium point, i.e., for the gradient value, which, the greater it is throughout the RoA, the faster the system is brought to equilibrium. A high value of h_3 causes the neural network to emphasise quickly achieving stability, which is essential in applications requiring a rapid system response.

Each of these parameters is important because they affect the system's dynamics and the size of the RoA, which, the larger it is, the better, as they indicate that the system is stable over a broader range of state variables. In our case, it was decided that the ratio of the parameter values $h_1 : h_2 : h_3$ would be 5:2:1. For the network designer, the most critical aspect was maintaining the estimated function around the operating point, followed by the broadest possible RoA. The speed of reaching the equilibrium point was a less critical feature (though not negligible) of the function.

The condition for L_1 is specified based on the relationship (17). The physical interpretation of this condition can be understood as a penalty for the system if its trajectory does not head towards the decreasing of the Lyapunov function $V(x(t))$. It can be proven that such a function, if it only meets this condition, then the function's trajectory will reach the area Ω_c (positively invariant) in finite time and remain therein after that.

$$L_1 = \frac{\partial W}{\partial x} f(t, x) + c_3 \|x\|^2 \tag{17}$$

The condition for L_2 is defined based on relationship (18). In this case, it is interpreted so that the network receives a penalty only and exclusively when it does not meet the left boundary condition from equation 13. The greater the penalty, the more the condition is exceeded.

$$L_2 = \min \begin{cases} W - c_1 \|x\|^2, \\ 0 \end{cases} \tag{18}$$

The condition for L_3 is defined based on relationship (19). In this case, it is interpreted so that the network receives a penalty only and exclusively when it does not meet the right boundary condition from equation 13. The greater the penalty, the more the condition is exceeded.

$$L_3 = \max \begin{cases} W - c_2 \|x\|^2, \\ 0 \end{cases} \tag{19}$$

Such a defined reward function should lead to the estimated function $W(x)$ to follow the Lyapunov function $V(x)$ in accordance with theorem 15 and fulfilling the conditions of asymptotic stability (11) and (12).

D. FRAMEWORK

The concept of the presented method is contained in three consecutive stages. They are presented in Fig. 3. In the first stage, an experiment using the SAS system is conducted on a real system. The data obtained from this system, x, v, that is, displacement and vibration velocity, are then the training data for the first neural network. This network aims to learn how a dynamic system processes state variables. Hence, the result of training this network is a specific nonparametric filter, which is supposed to reflect the operation of the real system. The nonparametric nature of this system is based on the fact that standard equations for state spaces do not define it, but the network structure is supposed to reflect the hyperplane of the real system. It is evident that the network will have its parameters, such as weights and biases. Still, it is treated as a ''black box'' model, without analyzing these parameters, but with analyzing its mapping. The exact parameters of this stage are presented in Section II-B. This includes the network architecture, description of measurement data, training procedure, and computational unit parameters.

The performance of the first neural network in terms of classification metrics is summarized in Table 1. These metrics include accuracy, precision, recall, and F1 score, which comprehensively evaluate the network's effectiveness in capturing the dynamics of the real system.

The trained network, as a nonparametric model, is then used in the second stage of the method. It serves another network for the estimation of the Lyapunov function. The second

TABLE 1. Performance metrics of the first neural network.

Metric	Value
Accuracy	0.95
Precision	0.94
Recall	0.93
F1 Score	0.93

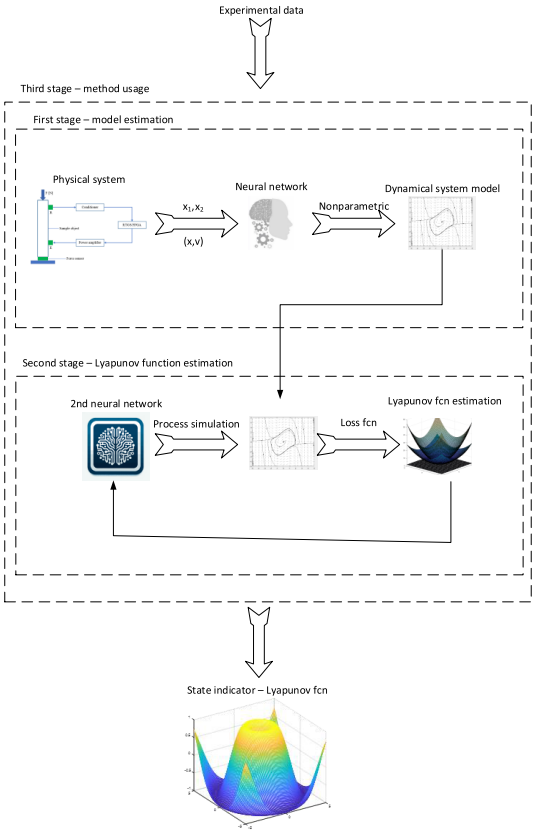


FIGURE 3. Conception scheme of the proposed method.

neural network starts with a random Lyapunov function V in a specified domain D. The designer must predefine the variability range for the state variables x_1 and x_2. In each iteration i, the network processes the function V_{i-1} for each state by simulating the operation of the dynamic system and taking into account the optimization conditions and loss function into the function V_i. According to equation (15), achieving any level of accuracy for networks with two hidden layers is possible. Therefore, in each iteration, the network approaches obtaining a correct estimate of the Lyapunov function. A conceptual diagram of this part of the method has been presented in Fig.4. During the network training, it was assumed that the network could stop learning only when the total value of the loss function for all points in the state space is less than 10^{-7}. The trained network, as a nonparametric model, is then used in the second stage of the method. It serves another network for the estimation of the Lyapunov function. The second neural network starts with a random Lyapunov function V in a specified domain D.

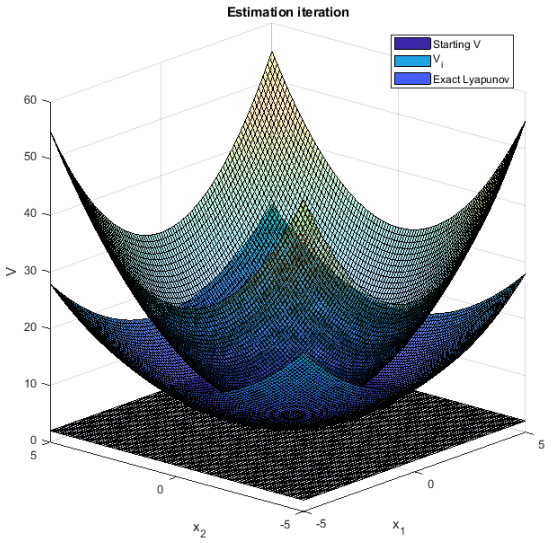


FIGURE 4. Iteration of the Lyapunov function estimation.

The designer must predefine the variability range for the state variables x_1, x_2. In each iteration i, the network processes the function V_{i-1} for each state by simulating the operation of the dynamic system and taking into account the optimization conditions and loss function, transforming it into the function V_i. According to equation (15), achieving any level of accuracy for networks with two hidden layers is possible. Therefore, in each iteration, the network approaches obtaining a correct estimate of the Lyapunov function. A conceptual diagram of this part of the method has been presented in Fig. 4. During the network training, it was assumed that the network could stop learning only when the total value of the loss function for all points in the state space is less than 10^{-7}.

The exact parameters of this stage are presented in Section II-C. The second neural network is a deep, two-layer network. The architecture consists of two hidden layers, each comprising 128 neurons. The activation functions used are ReLU (Rectified Linear Unit) for the hidden layers, which is defined as $f(x) = \max(0, x)$, providing non-linearity to the model and allowing it to learn complex patterns. The output layer uses a linear activation function to produce the continuous output required for the Lyapunov function.

The training process involves the Adam optimizer, an adaptive learning rate optimization algorithm designed to train deep neural networks efficiently. The learning rate was set to 0.001. The loss function used is the Mean Squared Error (MSE), which measures the average squared difference between the estimated and actual values. The network was trained using a batch size of 32, and the training continued until the total value of the loss function for all points in the state space was less than 10^{-7}.

The second-stage networks were trained on the same computer as the first, ensuring consistency in computational resources and environment.

The detailed architecture and parameters are as follows:

- **Input Layer:** 2 neurons (representing state variables x_1 and x_2)
- **Hidden Layer 1:** 128 neurons, ReLU activation
- **Hidden Layer 2:** 128 neurons, ReLU activation
- **Output Layer:** 1 neuron, linear activation
- **Optimizer:** Adam, learning rate 0.001
- **Loss Function:** Mean Squared Error (MSE)
- **Batch Size:** 32
- **Stopping Criterion:** Loss function value $< 10^{-7}$

These parameters were chosen to balance the complexity of the network with the need for accurate and efficient training, ensuring that the Lyapunov function is correctly estimated across the defined state space.

The third stage of the method occurs only after reaching the condition for the loss function. It means that a satisfactory error in the estimate of the Lyapunov function has been achieved. In this case, the method can be applied to further measurements. The first two stages are treated somewhat like constructing two serial filters, which aim to process the state variables from the experiment into an estimate of the Lyapunov function.

The detailed steps of the proposed method are encapsulated in a pseudocode format, as shown in Algorithm 1. This pseudocode provides a clear and structured outline of the three-stage neural network method for Lyapunov function estimation.

The primary motivation for proposing this method is the inconvenience of using the SAS system. Generally, this system makes it easy to differentiate between unstressed and inclusion-free material from material with defects. For these two cases, different frequencies of the limit cycle are obtained. Nevertheless, this limit cycle's frequency change can occur for two different reasons. Changes in stress in the examined material change this frequency, but defects can affect it simultaneously. Standard frequency measurement is unable to distinguish between these two causes. In turn, the Lyapunov function is a mathematical tool for analyzing the stability of equilibrium states in dynamical systems. From this perspective, it helps determine whether and how, after a disturbance occurs, the system will return to its initial state. The fundamental assumption of the method is that the system will have different phase trajectories for each type of disturbance in the examined material. Hence, three different types of materials were studied: A system without inclusions and stresses, a system without defects but with stresses, and an unstressed system with inclusions. The results are presented in the next section.

Algorithm 1 Three-Stage Neural Network Method for Lyapunov Function Estimation

function TRAINFIRSTNETWORK(x, v)
  Collect data from the SAS system: displacement x and vibration velocity v
  Preprocess the collected data
  Initialize the first neural network (NN1)
  repeat
    Train NN1 using data (x, v)
    Compute MSE L(NN1)
    Update weights $\theta_{\text{NN1}} \leftarrow \theta_{\text{NN1}} - \alpha \nabla_\theta L(\text{NN1})$ using SGD
  until convergence
  Evaluate performance of NN1
  return trained NN1 model
end function
function TRAINLYAPUNOVNETWORK(V, D, x_1, x_2)
  Initialize NN2 with random Lyapunov function V in domain D
  Predefine variability range for state variables x_1, x_2
  repeat
    $V_i \leftarrow \text{NN2}(V_{i-1})$
    Simulate operation of the dynamic system
    Compute loss L(NN2)
    Update weights $\theta_{\text{NN2}} \leftarrow \theta_{\text{NN2}} - \alpha \nabla_\theta L(\text{NN2})$ using SGD
  until total loss $< 10^{-7}$
  return trained NN2 model
end function
function MAIN
  Collect data x, v using SAS system
  NN1 $\leftarrow$ TRAINFIRSTNETWORK(x, v)
  Initialize Lyapunov function V and domain D
  Predefine variability range for x_1, x_2
  NN2 $\leftarrow$ TRAINLYAPUNOVNETWORK(V, D, x_1, x_2)
  Apply NN1 and NN2 to further measurements
  Use NN1 and NN2 to process state variables and estimate the Lyapunov function
end function

III. RESULTS

After conducting the learning process by the framework presented in Fig.3, a series of Lyapunov function estimates were obtained for three states: standard, meaning without load on the examined material and with confirmed absence of defects; loaded (but with confirmed absence of inclusions); and for unloaded material but with confirmed defects such as inclusions and cracks. Typical forms of graphs for a given state have been presented in Figs.5-7.

In the case of an unloaded and defect-free material (Fig. 5), a characteristic potential funnel is visible, indicating the presence of the Region of Attraction. The function reaches a minimum at the system's equilibrium point and assumes non-negative values. Additionally, an area in the state space $x_1 - x_2$ can be distinguished, in which equilibrium states are preserved. This confirms the effectiveness of estimating the Lyapunov function using Deep Learning networks.

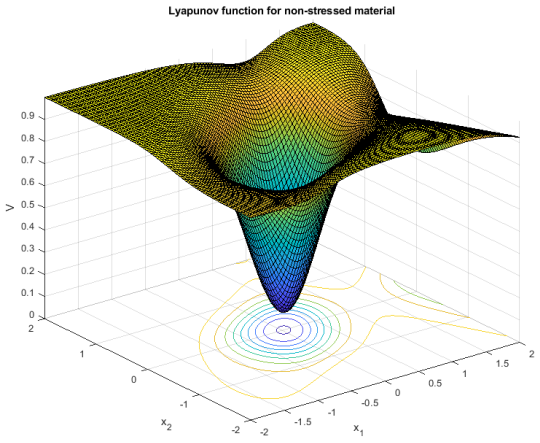


FIGURE 5. Lyapunov function estimation for non-stressed and no-defect material.

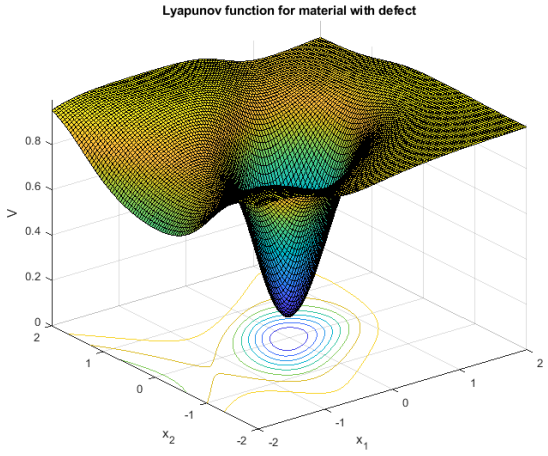


FIGURE 7. Lyapunov function estimation for material with defect.

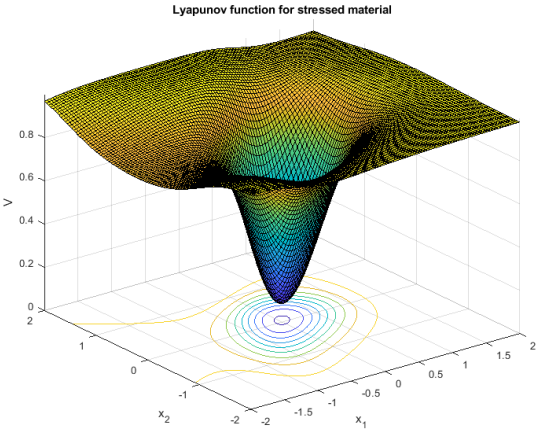


FIGURE 6. Lyapunov function estimation for stressed material.

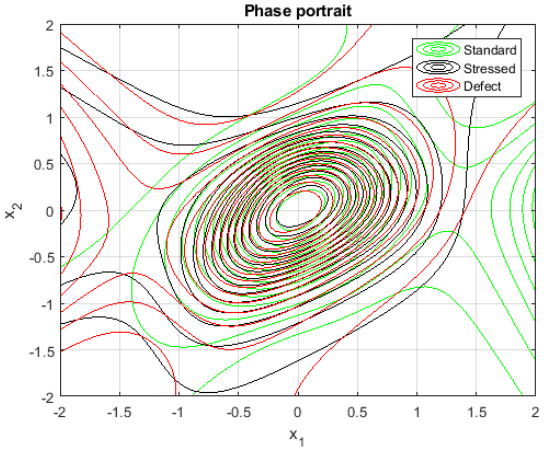


FIGURE 8. Phase portrait for each state.

Some differences are observable in the case of the study using the SAS system with a loaded material (Fig. 6). Although on both graphs, the Lyapunov function creates a shape resembling a funnel, which reaches a minimum at the point (0, 0) on the $x_1 - x_2$ plane, for the loaded material, the funnel is shallower compared to the graph for the unloaded material. This means that the energy needed for the system to exit from dynamic equilibrium is lower for the loaded material. The funnel on the graph for the loaded material is slightly wider at the base. This may indicate that the attraction area is more extensive, but the funnel's less ''steep'' walls suggest that the system may need more time to return to equilibrium after a disturbance. This confirms empirical experiences with working with the SAS system. This setup switched more frequently to other limit cycles when the material was loaded than when it was unloaded. A shallower potential well means that a lower level of disturbance can lead the dynamic system to another local minimum.

The Lyapunov function on the graph for the material with defects (Fig. 7) reveals certain distortions and asymmetries compared to the symmetric funnel visible for the unloaded material. These asymmetries are evident for state parameters distant from the equilibrium point. This confirms the hypothesis that a defect in the material affects the SAS system's dynamic properties, changing the system's local stability. As a result, the system reacts differently to perturbations depending on their direction. This demonstrates the validity of the proposed methodology. Indeed, the defect causes the energy used for control, represented by the Lyapunov function, to depend on the presence of the defect in the examined material. Further research may assess the impact of the defect's location in the material on the shape of the Lyapunov function. This could create a neural defect locator in the examined material based on the estimated function V.

The discussion is confirmed by the phase portrait shown in Fig. 8. In the case of the standard material scenario, trajectories around the central equilibrium point are closed and regular. This indicates that for this material, the system behaves predictably. Closed trajectories suggest stable limit cycles, meaning the system regularly oscillates around the

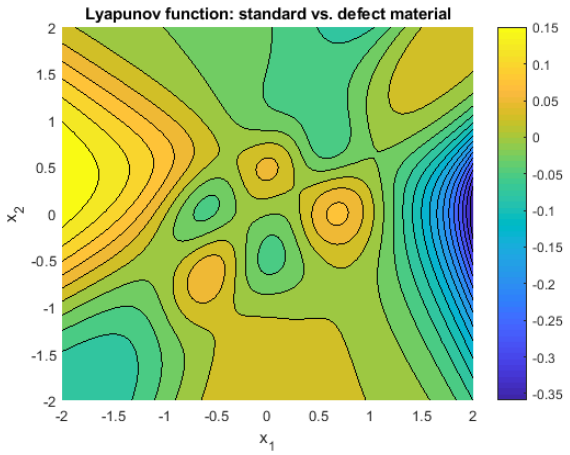


FIGURE 9. Lyapunov function subtraction for standard and defected material.

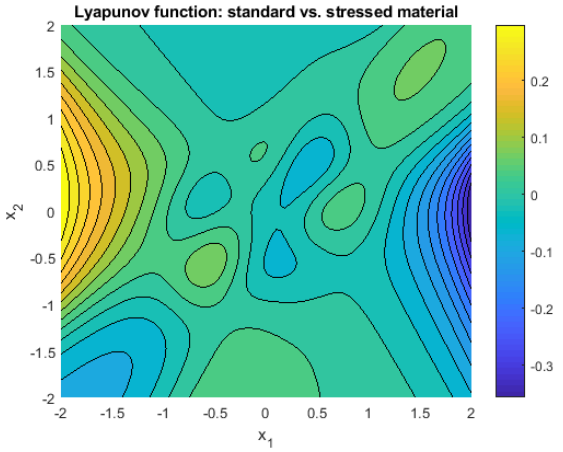


FIGURE 10. Lyapunov function subtraction for standard and stressed material.

equilibrium point. For studies on the loaded sample, the trajectories are also closed but occupy a larger area in the phase space than in the standard case. This indicates a greater amplitude of oscillations around the equilibrium point, which is typical for a system under the influence of greater loads and confirmed during empirical research. Although the system still achieves stable limit cycles, the larger space it occupies may indicate a lower resilience of the system. Trajectories for the state with a defect seem more irregular and scattered than in the two previous states.

Fig.9 and Fig.10 present the results for the simple subtraction of Lyapunov functions for the standard case of two consecutive ones being the subject of this article. By analyzing these graphs, it can be demonstrated that the two non-standard cases are distinguishable, even when they show similar limit frequencies. Initially, one should examine the state variable area within the potential funnel of the Lyapunov function. Here, fundamental differences between the loaded and defective states are visible. Compared to the standard state, the presence of a defect reveals itself in substantial

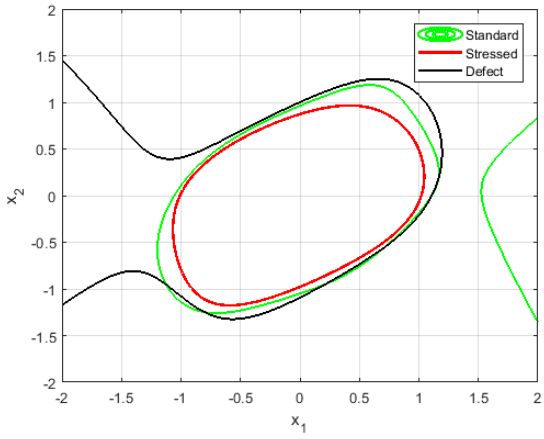


FIGURE 11. Region of attraction for each state's Lyapunov function visualisation.

point differences in the funnel area. It is caused by the presence of inclusions or obstacles to the dynamic system. The broadly understood control of the system encounters an unconventional obstacle at specific state points, which it must actively manage. Therefore, in the case of a defect, the limit cycle of the SAS system is noticeably different. Conducting a similar analysis for a system without defects but with an introduced load (Fig. 10), it can be seen that such an effect does not occur. This is one of the fundamental goals of the presented method. Thanks to such an approach, it is possible to distinguish the state of the examined material already within the potential funnel of the estimated Lyapunov functions.

Another method to distinguish between a loaded state and the presence of a defect can be the method of intersecting Lyapunov functionals using a plane parallel to the x_1, x_2 axes. As such a plane is moved, Regions of Attraction become visible (Fig. 11). These regions are closed and have regular shapes for both cases without defects. In the case of a defect occurring in the examined material, an asymmetry of the Lyapunov function is revealed, causing the Region of Attraction to open up and have a more irregular shape. All this confirms the method's utility in recognizing the state of the examined material using the SAS system.

IV. CONCLUSION

The research presented in this article demonstrates the capability of deep learning neural networks to estimate Lyapunov functions for dynamical systems accurately. By comparing the estimations of this function for the self-excited SAS system used for different materials, it is confirmed that the proposed method allows for the recognition of the material state in terms of both stress and the presence of defects. Differences in the depth and width of the potential funnel of the Lyapunov function estimates, as well as the presence of asymmetries in the case of defects, reveal how the responses of the dynamical system can be used to identify the condition of the material under study.

The methodology presented in this study offers a new approach to differentiating between the effects of material loads and the presence of defects. This distinction is crucial for materials science and engineering applications, where the integrity and performance of materials under load are critical. By analyzing the shape and behaviour of the estimated Lyapunov functions for the SAS system, it is possible to effectively determine not only stress or the presence of defects in the material under study but also potentially their location.

In addition to its significance in structural health monitoring, this article also contributes to the theory of controlling self-excited systems. The methodology presented here provides a new tool for determining the stability of such systems. The neural approach, which is fundamentally model-free, allows for the determination of stability for nonlinear dynamical systems. Previous methods of such assessment required knowledge of the exact physical model of the dynamical system. The presented methodology, however, allows for overcoming this inconvenience. It will be of key importance in many areas of engineering where determining the region of attraction to an equilibrium point plays a fundamental role.

The results suggest several paths for future research, especially in refining the estimation of Lyapunov functions for varying complex dynamical systems. Further research into the impact of defect location and nature on the shape of the Lyapunov function will lead to advanced diagnostic tools for materials science. Moreover, extending the application of this method to other types of dynamical systems will broaden our understanding of stability and equilibrium in complex nonlinear systems.

REFERENCES

[1] A. Chaillet, G. Göksu, and P. Pepe, "Lyapunov-krasovskii characterizations of integral input-to-state stability of delay systems with nonstrict dissipation rates," *IEEE Trans. Autom. Control*, vol. 67, no. 7, pp. 3259–3272, Jul. 2022.

[2] Q. Chen, J. Cao, and S. Zhu, "Data-driven monitoring and predictive maintenance for engineering structures: Technologies, implementation challenges, and future directions," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14527–14551, 2023.

[3] S. Dabetwar, S. Ekwaro-Osire, and J. P. Dias, "Damage detection of composite materials using data fusion with deep neural networks," *Turbo Expo. Power Land, Sea, Air*, vol. 84225, Nov. 2020, Art. no. V10BT27A019.

[4] I. Dominik, K. Lalik, and R. Dwornicka, "Preliminary research on applying self-excited system for bridge stress measurement," *Appl. Mech. Mater.*, vol. 712, pp. 75–80, Jan. 2015.

[5] I. Dominik, K. Lalik, and S. Flaga, "Modeling of self-excited stress measurement system," *J. Low Freq. Noise, Vibrat. Act. Control*, vol. 40, no. 2, pp. 852–866, Jun. 2021.

[6] J. H. Feng, H. Yuan, Y. Q. Hu, J. Lin, S. W. Liu, and X. Luo, "Research on deep learning method for rail surface defect detection," *IET Electr. Syst. Transp.*, vol. 10, no. 4, pp. 436–442, Dec. 2020.

[7] Y. Fu, A. R. J. Downey, L. Yuan, T. Zhang, A. Pratt, and Y. Balogun, "Machine learning algorithms for defect detection in metal laser-based additive manufacturing: A review," *J. Manuf. Processes*, vol. 75, pp. 693–710, Mar. 2022.

[8] A. Glaeser, V. Selvaraj, S. Lee, Y. Hwang, K. Lee, N. Lee, S. Lee, and S. Min, "Applications of deep learning for fault detection in industrial cold forging," *Int. J. Prod. Res.*, vol. 59, no. 16, pp. 4826–4835, 2021.

[9] E. Grundkötter and J. Melbert, "Adaptive power management of energy autonomous structural health monitoring systems for wind turbines," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2021, pp. 1–6.

[10] B. Hanin, "Universal function approximation by deep neural nets with bounded width and ReLU activations," *Mathematics*, vol. 7, no. 10, p. 992, Oct. 2019.

[11] H. K. Khalil, *Nonlinear Control*. London, U.K.: Pearson, 2015.

[12] I. Kieliba, I. Dominik, K. Lalik, T. Tonnesen, J. Szczerba, and R. Telle, "Self-excited acoustical system frequency monitoring for refractory concrete under uniaxial compression," *Energies*, vol. 14, no. 8, p. 2222, Apr. 2021.

[13] T. Kijewski-Correa, M. Haenggi, and P. Antsaklis, "Wireless sensor networks for structural health monitoring: A multi-scale approach," in *Proc. Struct. Congr.*, Oct. 2006, pp. 1–16.

[14] G. Krummenacher, C. S. Ong, S. Koller, S. Kobayashi, and J. M. Buhmann, "Wheel defect detection with machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1176–1187, Apr. 2018.

[15] I. Kuric, J. Klarák, M. Sága, M. Císar, A. Hajdučík, and D. Wiecek, "Analysis of the possibilities of tire-defect inspection based on unsupervised learning and deep learning," *Sensors*, vol. 21, no. 21, p. 7073, Oct. 2021.

[16] J. Kwaśniewki, I. Dominik, and K. Lalik, "Application of self-oscillating system for stress measurement in metal," *J. Vibroeng.*, vol. 14, no. 1, pp. 1–19, 2012.

[17] J. Kwasniewki, I. Dominik, K. Lalik, and K. Holewa, "Influence of acoustoelastic coefficient on wave time of flight in stress measurement in piezoelectric self-excited system," *Mech. Syst. Signal Process.*, vol. 78, pp. 143–155, Oct. 2016.

[18] J. Kwaśniewski, I. Dominik, J. Konieczny, K. Lalik, and A. Sakeb, "Application of self-excited acoustical system for stress changes measurement in sandstone bar," *J. Theor. Appl. Mech.*, vol. 49, no. 4, pp. 1049–1058, 2011.

[19] J. Kwaśniewski, I. Dominik, and K. Lalik, "A self-excited acoustical system for stress measurement in a cement plant," *Mech. Control*, vol. 31, no. 1, p. 29, 2012.

[20] J. Kwasniewski, I. Dominik, and K. Lalik, "Real-time system based on FPGA applied to self-excited acoustical system for stress change measurement," *Czasopismo Techniczne*, vol. 1, no. 5, pp. 205–2012, 2013.

[21] J. Kwaśniewski, I. Dominik, K. Lalik, and R. Szymański, "Harmonic analysis of self-excited acoustical system for stress changes measurement in compressed steel structural section," *Solid State Phenomena*, vol. 198, pp. 639–644, 2013.

[22] J. Kwasniewski, Y. Kravtsov, I. Dominik, L. Dorobczynski, and K. Lalik, "Self-excited acoustical system for stress measurement in mass rocks," *J. Low Freq. Noise, Vibrat. Act. Control*, vol. 32, nos. 1–2, pp. 133–144, Mar. 2013.

[23] K. Lalik, I. Dominik, P. Cwiąkała, and J. Kwasniewski, "Integrated stress measurement system in tower crane mast," *Measurement*, vol. 102, pp. 47–56, May 2017.

[24] K. Lalik, I. Dominik, P. Gut, K. Skrzypkowski, W. Korzeniowski, and K. Zagórski, "Non-destructive acoustical rock bolt testing system with intelligent filtering in salt mine 'Wieliczka,'" *Energies*, vol. 14, no. 17, p. 5522, Sep. 2021.

[25] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Mach. Intell.*, vol. 3, no. 3, pp. 218–229, Mar. 2021.

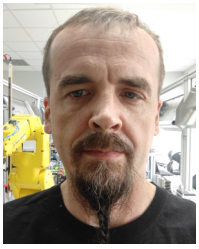
[26] R. D. McAllister and J. B. Rawlings, "Stochastic Lyapunov functions and asymptotic stability in probability," *Texas-Wisconsin-California Control Consortium*, vol. 1, no. 2, pp. 1–26, 2020.

[27] S. A. Mitoulis, "Challenges and opportunities for the application of integral abutment bridges in earthquake-prone areas: A review," *Soil Dyn. Earth Eng.*, vol. 135, Aug. 2020, Art. no. 106183.

[28] G. Psuj, "Multi-sensor data integration using deep learning for characterization of defects in steel elements," *Sensors*, vol. 18, no. 1, p. 292, Jan. 2018.

[29] A. Scampicchio, A. Aravkin, and G. Pillonetto, "Stable and robust LQR design via scenario approach," *Automatica*, vol. 129, Jul. 2021, Art. no. 109571.

[30] R. Sepasdar, A. Karpatne, and M. Shakiba, "A data-driven approach to full-field nonlinear stress distribution and failure pattern prediction in composites using deep learning," *Comput. Methods Appl. Mech. Eng.*, vol. 397, Jul. 2022, Art. no. 115126.
[31] A. Stoll and P. Benner, "Machine learning for material characterization with an application for predicting mechanical properties," *GAMM-Mitteilungen*, vol. 44, no. 1, Mar. 2021, Art. no. e202100003.
[32] J. Tao, Y. Zhu, W. Liu, F. Jiang, and H. Liu, "Smooth surface defect detection by deep learning based on wrapped phase map," *IEEE Sensors J.*, vol. 21, no. 14, pp. 16236–16244, Jul. 2021.
[33] Y. Tiumentsev and M. Egorchev, *Neural Network Modeling and Identification of Dynamical Systems*. New York, NY, USA: Academic, 2019.
[34] Y. Tu, Z. Liu, L. Carneiro, C. M. Ryan, A. C. Parnell, S. B. Leen, and N. M. Harrison, "Towards an instant structure-property prediction quality control tool for additive manufactured steel using a crystal plasticity trained deep learning surrogate," *Mater. Des.*, vol. 213, Jan. 2022, Art. no. 110345.
[35] Y. Wang, D. Oyen, W. Guo, A. Mehta, C. B. Scott, N. Panda, M. G. Fernández-Godino, G. Srinivasan, and X. Yue, "StressNet—Deep learning to predict stress with fracture propagation in brittle materials," *Npj Mater. Degradation*, vol. 5, no. 1, p. 6, Feb. 2021.
[36] A. Wright, E.-P. Damskägg, and V. Valimaki, "Real-time black-box modelling with recurrent neural networks," in *Proc. 22nd Int. Conf. Digit. Audio Effects*, Sep. 2019, pp. 1–8.
[37] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, "A deep learning-based framework for an automated defect detection system for sewer pipes," *Autom. Construction*, vol. 109, Jan. 2020, Art. no. 102967.
[38] S. Yoon, A. S.-K. Kim, W. J. Cantwell, C. Y. Yeun, C.-S. Cho, Y.-J. Byon, and T.-Y. Kim, "Defect detection in composites by deep learning using solitary waves," *Int. J. Mech. Sci.*, vol. 239, Feb. 2023, Art. no. 107882.



KRZYSZTOF LALIK was born in Kraków, Poland, in 1985. He received the B.S. and Master of Science degrees in automatics and robotics and the Ph.D. degree in mechanical engineering from the University of Science and Technology AGH, Kraków, in 2009 and 2014, respectively.

His current research interests include artificial intelligence, machine learning, predictive maintenance, and industry 4.0 standards.