

Machine learning for rotating machines: simulation, diagnosis and control

Alexey Kornae¹, Leonid Savin², Nickolay Kornae³, Roman Zaretsky⁴, Elena Kornae⁵, Alexander Babin⁶, Ivan Stebakov⁷

Department of Mechatronics, Mechanics and Robotics, Orel State University named after I.S. Turgenyev, Orel, 302026, Russian Federation

¹Corresponding author

E-mail: ¹rusakor@inbox.ru, ²savin@ostu.ru, ³rusnkor@bk.ru, ⁴kogots@bk.ru, ⁵lenoks_box@mail.ru, ⁶alex.mech.osu@gmail.com, ⁷chester50796@yandex.ru

Received 11 June 2020; accepted 18 June 2020

DOI <https://doi.org/10.21595/vp.2020.21549>



Copyright © 2020 Alexey Kornae, et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract. The goal of this work is association of several machine learning methods in a study of rotating machines with fluid-film bearings. A fitting method is applied to fit a non-linear reaction force in a bearing and solve a rotor dynamics problem. The solution in the form of a simulation model of a rotor machine has become a part of a control system based on reinforcement learning and the policy gradient method. Experimental part of the paper deals with a pattern recognition and fault diagnosis problem. All the methods are effective and accurate enough.

Keywords: supervised learning, reinforcement learning, rotor dynamics, fault diagnosis, control.

1. Introduction

The main tool in modern machine learning is an artificial neural network (ANN) [1]. This work deals with applications of machine learning to rotating machines with fluid-film bearings. A rotor rotation is usually accompanied by lateral vibrations [2, 3]. Rotor trajectory contains information about the condition of the bearings and the rotor machine at a whole. Rotor dynamics modeling is a difficult task especially when the rotor has fluid-film bearings [2]. Hydrodynamic calculations are computationally expensive. Therefore, this part of the rotor dynamics problem can be implemented using ANNs [4]. Modern rotating machines can be equipped with a number of sensors. Analysis of their measurements can be automated using specialized ANNs. These ANNs implement logistic regression [1]. Both shallow learning and deep learning are used in pattern recognition. Deep convolutional neural networks are widely used in fault diagnosis [5, 6]. Deep learning is emerging in reinforcement learning and continuous control systems [7, 8].

This paper unites theoretical and experimental results achieved by the authors in applications of machine learning to simulation, diagnosis and control of rotating machines with fluid-film bearings.

2. Shallow learning for rotor dynamics simulation and fault diagnosis

The goal of supervised learning is to determine relationship between two sets: an input set \mathbf{X} and a target set \mathbf{Y} . The difference between predictions \mathbf{H} and targets \mathbf{Y} is minimized in training process. This error function is called a target function, a cost function, or a loss function.

The scheme of a simple feed forward ANN called multilayer perceptron is represented in Fig. 1. The l -layer ANN has an input layer, $l - 2$ hidden layers and an output layer. A feature of the network architecture is that each neuron of the previous layer transmits its signal to each neuron of the next layer [4].

Input data in the form of n_1 numbers can be represented as a one-dimensional matrix. The input layer receives this matrix $\mathbf{X} = ((x_i^{(1)}))$ ($i = 1 \dots n_1$) and simply transmits it to the second layer with an additional unit element. The matrix $\mathbf{A}^{(1)} = ((1 \quad \mathbf{X}))$ is the output of the first layer.

In the second (hidden) layer data from the first layer is multiplied by the weights matrix $\Theta^{(1)}$ and added $\mathbf{Z}^{(2)} = \mathbf{A}^{(1)}\Theta^{(1)}$. An activation function is applied to the result $\mathbf{Z}^{(2)}$, and resulting matrix with additional unit element is the output of the second layer $\mathbf{A}^{(2)} = \left(\left(1 \quad activation^{(2)}(\mathbf{Z}^{(2)}) \right) \right)$. The same actions take place on an arbitrary $k - 1$ hidden layer:

$$\mathbf{Z}^{(k-1)} = \mathbf{A}^{(k-2)}\Theta^{(k-2)}, \quad \mathbf{A}^{(k-1)} = \left(\left(1 \quad activation^{(k-1)}(\mathbf{Z}^{(k-1)}) \right) \right), \tag{1}$$

where $\mathbf{Z}^{(k-1)}$ is matrix with results of multiplication by weights and adding in the current layer, $\mathbf{A}^{(k-2)}, \mathbf{A}^{(k-1)}$ are matrices of outputs in the previous and the current layers respectively, $\Theta^{(k-2)}$ is the previous layer weights matrix, $activation^{(k-1)}$ is an activation function.

Similar calculations occur in the output layer. The result of the calculation in the output layer is the matrix of predictions $\mathbf{H} = \left(\left(activation^{(l)}(\mathbf{Z}^{(l)}) \right) \right)$.

The unknown weights matrices $\Theta^{(k)}$ are determined by minimizing the objective function $J(\Theta^{(k)}) \Rightarrow min$. The type of objective function depends on the type of a given problem.

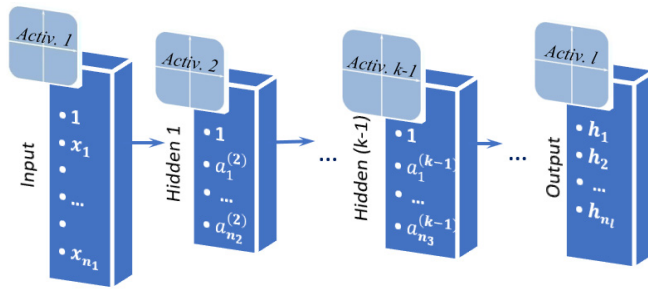


Fig. 1. An l -layer feed forward neural network with n_1 inputs and n_l outputs

2.1. Multi-dimensional mapping for rotor dynamics simulation

A rigid unbalanced rotor with a gear coupling and a fluid-film bearing at the opposite tips is considered (see Fig. 2). The equations of the rotor’s motion can be represented as follows [3]:

$$\begin{cases} m d^2\mathbf{X}/dt^2 = \mathbf{C} + \mathbf{B} + \mathbf{F}, \\ J_d d^2\boldsymbol{\Psi}/dt^2 + J_p d\boldsymbol{\Psi}/dt = \mathbf{C} \cdot \mathbf{GC} + \mathbf{B} \cdot \mathbf{GB}, \end{cases} \tag{2}$$

where m is rotor mass, $\mathbf{X}, \boldsymbol{\Psi}$ are coordinates of the center of mass and angles of the rotor rotation respectively, t is time, \mathbf{C}, \mathbf{B} are reaction forces of the gear coupling and the bearing respectively, $\mathbf{F} = m\Delta\omega^2 \begin{bmatrix} \cos(\omega t + \varphi) \\ \sin(\omega t + \varphi) \end{bmatrix}$ is inertia force, $m\Delta$ is the rotor unbalance, φ is phase of unbalance, J_p, J_d are polar and diametral moments of inertia respectively, \mathbf{GC}, \mathbf{GB} are radius-vectors to the coupling and to the bearing respectively.

It is assumed that reaction forces are equivalent to springs and dampers: $\mathbf{C} = \mathbf{C}(\mathbf{X}, d\mathbf{X}/dt)$, $\mathbf{B} = \mathbf{B}(\mathbf{X}, d\mathbf{X}/dt)$. Also, it is assumed that reaction in a coupling is linear with given linear coefficients and reaction in a bearing is non-linear. Calculation of the bearing reaction is connected with solution of the Reynolds equation [2, 4]:

$$\frac{\partial}{\partial x_1} \left[\frac{h^3}{\mu} \frac{\partial p}{\partial x_1} \right] + \frac{\partial}{\partial x_3} \left[\frac{h^3}{\mu} \frac{\partial p}{\partial x_3} \right] = 6 \frac{\partial}{\partial x_1} (u_1 h) - 12u_2, \tag{3}$$

where x_i are coordinates connected with the thin oil film, $p = p(x_1, x_3)$ is the unknown pressure function, $h = h(x_1)$ is the oil film thickness, μ is viscosity, $u_j = u_j(x_1)$ are the tangential and normal components of the journal surface velocity, where $i = 1, 2, 3, j = 1, 2$.

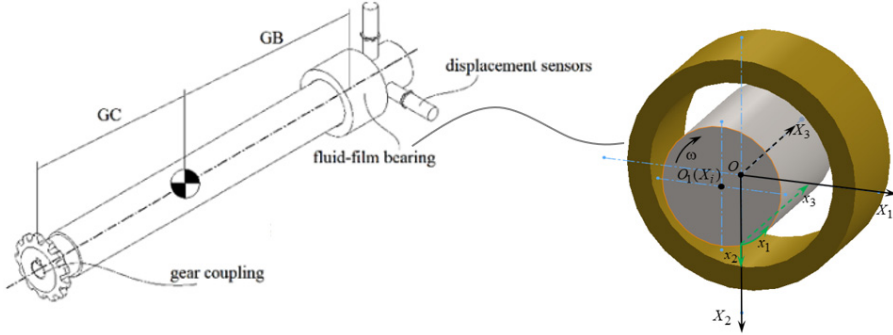


Fig. 2. Calculation schematic of a rotor-bearing test rig with a fluid-film bearing

Given the pressure function, the components of reaction can be calculated by integration:

$$B_1 = - \int_0^L \int_0^{\pi D} p(x_1, x_3) \cos(\alpha) dx_1 dx_3, \quad B_2 = - \int_0^L \int_0^{\pi D} p(x_1, x_3) \sin(\alpha) dx_1 dx_3, \quad (4)$$

where L, D are the bearing length and diameter respectively, $\alpha = 2x_1/D$.

Approximation of Eq. (4) with the function $\mathbf{B} = \mathbf{B}(\mathbf{X}, \mathbf{V} = d\mathbf{X}/dt)$ of four arguments can be implemented by the ANN represented in Fig. 1. The input layer receives a matrix with components of rotors position and its velocity of lateral vibrations $((X_1 \ X_2 \ V_1 \ V_2))$. The output of the ANN is the fluid-film reaction force $((B_1 \ B_2))$. A 3-layer feed-forward ANN with sigmoid activation function $a_j^{(2)} = 1/1 + e^{-z_j^{(2)}}$ ($j = 1, \dots, n_2$) in the hidden layer and linear function in the output layer $B_j = h_j = a_j^{(3)} = z_j^{(3)}$ is used to solve multi-dimensional mapping problem. Forward propagation in the ANN includes following calculations:

$$\mathbf{A}^{(1)} = ((1 \ X_1 \ X_2 \ V_1 \ V_2)), \quad \mathbf{Z}^{(2)} = \mathbf{A}^{(1)}\boldsymbol{\Theta}^{(1)}, \quad \mathbf{A}^{(2)} = ((1 \ \text{sigmoid}(\mathbf{Z}^{(2)}))), \quad (5)$$

$$\mathbf{Z}^{(3)} = \mathbf{A}^{(2)}\boldsymbol{\Theta}^{(2)}, \quad \mathbf{B} = \mathbf{H} = \mathbf{A}^{(3)} = \mathbf{Z}^{(3)}.$$

The number of hidden neurons n_2 is arbitrary. Network training takes place on a large number of samples, i.e. pairs of input $((X_1 \ X_2 \ V_1 \ V_2))$ and output $((B_1 \ B_2))$ matrices. In mapping problems, the cost function has the following form [1]:

$$J(\boldsymbol{\Theta}^{(k)}) = \frac{1}{2m} \sum_{i=1}^m \sum_{j=1}^{n_l} (h_j^{(i)} - y_j^{(i)})^2 + \frac{\lambda}{2m} \sum_{k=1}^{l-1} \sum_{i=1}^{n_k} \sum_{j=1}^{n_{k+1}} (\theta_{ij}^{(k)})^2 \Rightarrow \min, \quad (6)$$

here m is number of samples in a dataset, $h_j^{(i)}, y_j^{(i)}$ are predicted and target values of j -th output value calculated for the i -th sample, n_k, n_l are the numbers of neurons in the k -th layer and in the output layer respectively, λ is a regularization parameter.

In the training process the values of weights $\boldsymbol{\Theta}^{(k)}$ and the regularization parameter λ are calculated. The network is trained with Levenberg-Marquardt backpropagation algorithm. The training process is implemented in one of the specialized programming environments [9, 10].

2.2. Classification and pattern recognition tools for rotating machine diagnosis

The main idea is the same: to determine relationship between inputs \mathbf{X} and targets \mathbf{Y} . The main difference is that the targets values are discrete and equal to 0 or 1, and predictions \mathbf{H} approximated by logistic function has continuous values in the interval (0 1) [1].

Sensor measurements are recorded during the tests under various conditions of a rotating machine. Given conditions are needed to be recognized by ANNs. The data from different types of sensors can be normalized [1] and merged into an input matrix \mathbf{X} . The number of classes in a target matrix \mathbf{Y} is equal to the number of observed conditions of a rotating machine. A 3-layer feed-forward ANN with a sigmoid activation function in the hidden layer (see section 2.1) and a softmax function in the output layer $h_j = a_j^{(3)} = e^{z_j^{(3)}} / \sum_{i=1}^{n_3+1} e^{z_i^{(3)}} (j = 1, \dots, n_3)$ is used to solve pattern recognition problem. Forward propagation includes following calculations:

$$\begin{aligned} \mathbf{A}^{(1)} &= ((1 \ \mathbf{X})), \quad \mathbf{Z}^{(2)} = \mathbf{A}^{(1)}\boldsymbol{\Theta}^{(1)}, \quad \mathbf{A}^{(2)} = ((1 \ \text{sigmoid}(\mathbf{Z}^{(2)}))), \\ \mathbf{Z}^{(3)} &= \mathbf{A}^{(2)}\boldsymbol{\Theta}^{(2)}, \quad \mathbf{A}^{(3)} = \mathbf{H} = ((1 \ \text{softmax}(\mathbf{Z}^{(3)}))). \end{aligned} \quad (7)$$

As for the previous ANNs architecture (see subsection 2.1) the number of hidden neurons n_2 is arbitrary and training process needs a number of training samples. In pattern recognition problems, the cost function has the following form [1]:

$$J(\boldsymbol{\Theta}^{(k)}) = \frac{-1}{m} \sum_{i=1}^m \sum_{j=1}^{n_l} y_j^{(i)} \ln(h_j^{(i)}) + \frac{\lambda}{2m} \sum_{k=1}^{l-1} \sum_{i=1}^{n_k} \sum_{j=1}^{n_{k+1}} \theta_{ij}^{(k)} \Rightarrow \min. \quad (8)$$

The network is trained with scaled conjugate gradient using functions of specialized programming environments [9, 10].

3. Deep reinforcement learning for rotating machine control

The main objective of the control system under study is minimization of energy consumption. The fluid-film reaction and the friction torque in a bearing are non-linear functions depending on pressure distribution (see Eq. (3)). It is assumed that the rotor vibrations described with Eq. (2), the coupling reaction \mathbf{C} is linear and the bearing reaction \mathbf{B} is simulated by the ANN described in Subsection 2.1 (see Fig. 2). In terms of reinforcement learning the rotor-bearing simulation model is an environment and the control system is an agent. At each time step t the agent receives feedback from the environment \mathbf{S}_t in form of a matrix of simulation results and takes an action a_t in response in form of pressure supply or any other parameter of a fluid-film bearing. The main idea is to train agent after the event giving him higher reward r_t for better actions.

The deep deterministic policy gradient (DDPG) is used. The algorithm of DDPG agents is represented in [8]. At each time step the value function v_t is calculated as follows [8]:

$$v_t = r_t + \gamma q'(S_{t+1}, \mu'(S_{t+1} | \boldsymbol{\Theta}^\mu), \boldsymbol{\Theta}^q), \quad (9)$$

where γ is a discount factor, q' is a q -function calculated by a target critic, μ' is a policy function of the action by a target agent, $\boldsymbol{\Theta}^\mu$, $\boldsymbol{\Theta}^q$ are unknown parameters of the actor and the critic ANNs respectively.

The architectures of the networks will be represented in the next section of the paper. The unknown parameters are calculated by minimizing the loss function [8]:

$$J(\Theta^\mu, \Theta^q) = \frac{1}{m} \sum_{j=1}^m (v_i - q(S_i, a_i | \Theta^q))^2 \Rightarrow \min. \quad (10)$$

The networks are trained with stochastic gradient descent method using functions of specialized programming environments [9, 10].

4. Simulation and experimental results

The first series of simulation tests was performed with a model of the test rig based on Eqs. (2-4). A set of rotor trajectories was calculated. Then the ANN described by Eqs. (5-6) was trained and tested in comparison with a known linear model characterized by the spring and damper matrices [4]. The results demonstrated that the rotor dynamics simulation program with the ANN module allows calculation rotor trajectory two times faster than a real time process. It was demonstrated also that the ANN allows simulation of non-linear transient processes with variable rotor speed and high vibrations [4].

The second series of tests was performed using the test rig with a multi-sensor measurement system. Two displacement sensors measured the rotor vibrations, three vibroaccelerometers measured the rotor and the electromotor housings vibroaccelerations, a microphone measured the operating rotor machine noise. Inside the bearing the pressure sensor measured pressure supply and the contact resistance sensor measured the fluid-film thickness. Six conditions of the test rig were studied, including the normal condition, the conditions with loosened bolts and the rotor unbalance condition. The general classification problem for six classes recognition and the simplified classification problem for two classes recognition (normal or abnormal) were solved. Several random samples of two classes dataset with 800 data points each are shown in Fig. 3

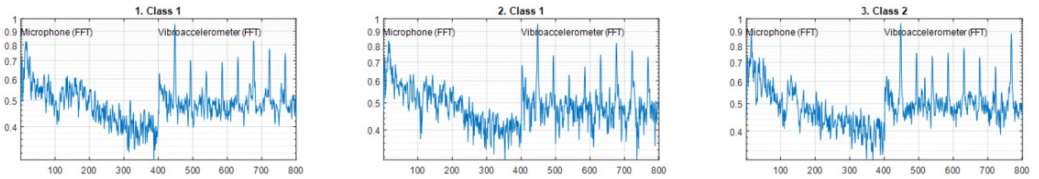


Fig. 3. Random samples of two classes dataset with normalized measurements results from the microphone and from one of the vibroaccelerometers

Then the ANN described by Eqs. (7-8) was trained and tested to solve the classification problems. The accuracy of the two classes and the six classes recognition were up to 80 % and 90 % respectively. It should be noted, that the accuracy of the two classes classification by experts was up to 70 %. The value of accuracy means that classification process was close to random.

The third series of the tests was simulation. The rotor dynamics simulation model became an environment observed and controlled by an agent. The agent model was based on the DDPG algorithm (see section 2.3). The ANNs architectures are shown in Fig. 4.

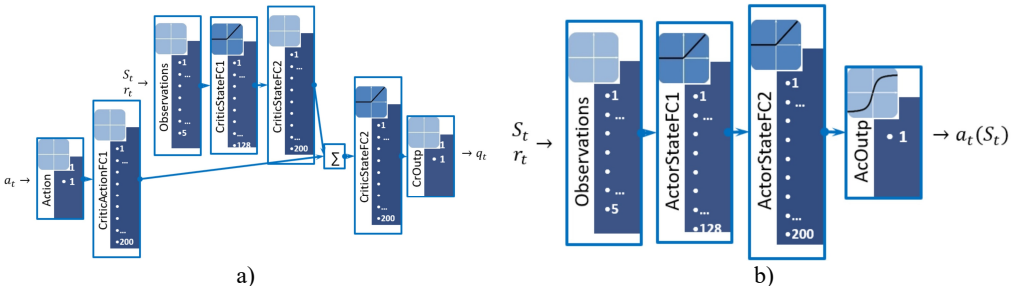


Fig. 4. The DDPG networks architectures: a) the actor network and b) the critic network [2]

The agent controlled the bearing clearance size h (see Eq. (3)) directly, and indirectly the pressure distribution and the reaction forces in the bearing. The goal of the control system was minimization of power loss due to friction and vibration in the bearing. The ANNs were trained and tested. The results demonstrated decreasing the power loss up to 17 % by the DDPG agent.

5. Conclusions

Suggested tools of rotor dynamics simulation, condition classification and control based on artificial neural networks allow development of predictive modeling systems, fault diagnosis and control of energy efficient operation to design intellectual rotating machines. All the developed systems can be combined in one device. The following study is connected with design of the device which will be able to combine multiple functions of predictive modeling, fault diagnosis and control in interaction with a rotating machine.

Acknowledgements

This work was supported by the Russian Science Foundation under the Project No. 16-19-00186. The authors gratefully acknowledge this support. Authors would also like to thank A. Rodichev, A. Fetisov, Y. Kazakov and S. Popov for the multi-sensory test rig development.

References

- [1] **Goodfellow Y., Bengio Y, Courville A.** Deep Learning. MIT Press, 2016.
- [2] **Hori Y.** Hydrodynamic Lubrication. Yokendo Ltd, Tokyo, 2006.
- [3] **Friswell M. I.** Dynamics of Rotating Machines. Cambridge University Press, 2010.
- [4] **Kornaev A. V., Kornaev N. V., Kornaeva E. P., Savin L. A.** Application of artificial neural networks to calculation of oil film reaction forces and dynamics of rotors on journal bearings. International Journal of Rotating Machinery, Vol. 2017, 2017, p. 9196701.
- [5] **Lei Y., Yang B., Jiang X., Jia F., Li N., Nandi A. K.** Applications of machine learning to machine fault diagnosis: a review and roadmap. Mechanical Systems and Signal Processing, Vol. 138, 2020, p. 106587.
- [6] **Liu R., Yang B., Zio E., Chen X.** Artificial intelligence for fault diagnosis of rotating machinery: A review. Mechanical Systems and Signal Processing, Vol. 108, 2018, p. 33-47.
- [7] **Busoniu L., Bruin T., Tolic D., Kober J., Palunko I.** Reinforcement learning for control: Performance, stability, and deep approximators. Annual Review in Control, Vol. 46, 2018, p. 8-28.
- [8] **Lillicrap T. P., Hunt J. J., Pritzel A., Heess N., Erez T., Tassa Y., Silver D., Wierstra D.** Continuous control with deep reinforcement learning. International Conference on Learning Representations, 2016.
- [9] Mathworks: help center. Nnstart tool, <https://www.mathworks.com>.
- [10] Keras API, <https://keras.io/api/>.