Synthesis and computer research of a belt conveyor models with intelligent control

Olga V. Druzhinina[1], Olga N. Masina[2], Elena V. Igonina[2] and Alexey A. Petrov[2]

[1]*Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, 44, building 2, Vavilov St., Moscow, 119333, Russian Federation*

[2]*Bunin Yelets State University, 28, Kommunarov St., Yelets, 399770, Russian Federation*

Abstract

The synthesis problems of intelligent control models for a belt conveyor with a variable lifting angle are considered. The models take into account the acting forces, uneven loading and control actions. A complex criterion of the control quality is developed. The graphs of trajectories are obtained taking into account the selected parameters of the models. An approach to finding the optimal parameters of a model based on artificial intelligence methods is proposed. The conditions for stabilization of the belt conveyor control systems are obtained. Control algorithms are developed based on the construction of fuzzy controllers. The qualitative effects in the models of the belt conveyor control are revealed. Algorithmic and instrumental support is developed for the implementation of software and hardware packages for intelligent control of a belt conveyor with a variable lifting angle. Modules of a software package for modeling of control systems for a belt conveyor using the Python3 language in combination with the Jupyter system and mathematical libraries Numpy, Scipy, Sympy, as well as using open libraries for construction of systems with artificial intelligence are created. Using the developed intelligent algorithms and a high-performance workstation, test examples are studied and a series of computational experiments are carried out. Computational experiments are aimed at finding of control laws and optimal modeling parameters. The obtained results can be used for solving of the enterprises production lines control problems for the innovation cluster of mechanical engineering.

Keywords

computer modeling, applied programming, belt conveyor model, control quality criterion, intelligent control, fuzzy controller, neural network controller, stabilization algorithms, software package

1. Introduction

Methods of mathematical modeling, artificial intelligence, fuzzy logic and intelligent control are widely used in the control of production technological processes. In particular, the problems of constructing and analyzing mathematical models of conveyor transport are of great theoretical and applied interest. Belt conveyors are one of the most efficient and high-performance types

Workshop on information technology and scientific computing in the framework of the XI International Conference Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems (ITTMM-2021), Moscow, Russian, April 19–23, 2021

ovdruzh@mail.ru (O. V. Druzhinina); olga121@inbox.ru (O. N. Masina); elenaigonina7@mail.ru (E. V. Igonina); xeal91@yandex.ru (A. A. Petrov)

0000-0002-9242-9730 (O. V. Druzhinina); 0000-0002-0934-7217 (O. N. Masina); 0000-0002-7369-6219 (E. V. Igonina); 0000-0002-3011-9278 (A. A. Petrov)

of conveyor transport in machine-building enterprises. Numerous papers are devoted to the design and improvement of conveyor transport systems, in particular, [1, 2, 3].

The basics of intelligent control based on fuzzy controllers are described in the papers. The application of fuzzy logic and artificial neural networks in the control of belt transport is studied, in particular, in [4, 5, 6]. These results are based on the use of various types of fuzzy regulators [7, 8, 9, 10, 11, 12, 13, 14].

One of the most effective methods of artificial intelligence is machine learning. Machine learning is a class of intelligent methods that allow you to improve the performance of computers by learning from known data. There are many models for machine learning, but they tend to fall into one of three types: learning with a teacher, learning without a teacher, and reinforcement learning. The latter type of machine learning is of considerable interest in the study of industrial facility control systems, since it combines the advantages of learning with a teacher and learning without a teacher. Machine learning with reinforcement is the subject of papers [15, 16].

This paper is devoted to the synthesis and computer research of a belt conveyor models with intelligent control. In section 2, we synthesize a model of a four-wheel drive conveyor with a dynamically variable belt lift angle. In addition, in section 2, we formulate an optimal control problem for a dynamic belt conveyor model. In section 3, we consider partial stabilization based on the use of a sliding mode. Section 4 deals with the control of the conveyor transport system based on artificial intelligence methods. Section 5 is dedicated to the original software developed by the team of authors. The developed software package is designed to modeling of conveyor transport systems.

2. Optimal control problem for a dynamic belt conveyor model

We synthesize a model of a four-wheel drive conveyor with a dynamically variable belt lifting angle. We accept the following conditions.

The phase space of the model is bounded by two coordinates. The specified coordinates specify the linear movement of the tape and the angle of elevation above the plane, respectively. The conveyor moves objects with a mass that can affect the characteristics of its movement. In the first approximation, we neglect the friction of the loads on the belt, the stretching of the belt. We will also consider the change in the speed of the ascent angle negligibly small due to the conservation of angular momentum.

Under the conditions under consideration, the differential equations of the belt conveyor model have the form

$$\begin{aligned} \dot{x}_0 &= x_1, \\ \dot{x}_1 &= \frac{u_1(t) - kx_1 - m_1 gsin(\alpha_0)}{m_1 + m_0}, \\ \dot{\alpha}_0 &= \alpha_1, \\ \dot{\alpha}_1 &= \frac{u_2(t)}{s\varepsilon^2(m_0 + m_1)} - \frac{gcos(\alpha_0)}{\varepsilon}, \end{aligned} \tag{1}$$

where x_0 is moving the conveyor belt, m_0 is the conveyor belt weight, α_0 is the conveyor lifting angle, α_1 is the conveyor lifting speed, m_1 is total weight of loads on the conveyor, s is a coefficient that determines the moment of the conveyor inertia, ε is a position of the conveyor gravity center, k is a rolling friction coefficient, $u_1(t)$ is linear force of the conveyor forward motion, $u_2(t)$ is a torque value for controlling the lifting angle of the conveyor.

Let's assume that objects with different weights are loaded and unloaded on a moving conveyor at random times. Given these conditions, it is convenient to write the model (1) in the form of a model with switches of the form

$$\begin{aligned}
&\dot{x}_0 = x_1,\\
&\dot{x}_1 = \frac{u_1(t) - kx_1 - m_1 g sin(\alpha_0)}{m_1 + m_0},\\
&\dot{\alpha}_0 = \alpha_1,\\
&\dot{\alpha}_1 = \frac{u_2(t)}{s\varepsilon^2(m_0 + m_1)} - \frac{g cos(\alpha_0)}{\varepsilon},\\
&u_1(t) \in U_1, u_2(t) \in U_2, \varepsilon \in S, m_1 \in M,
\end{aligned} \tag{2}$$

where U_1, U_2 are multiple controls, S is multiple possible centers of gravity, M is the mass set of load on the belt.

We propose the following statement of the optimal control problem. Let us assume that we need to stabilize system (1) at a point in the phase space s over a conditionally infinite time interval. We assume that a sufficient condition for the stabilization of the system (1) is to obtain a stable solution of the equation (1) that satisfies the following conditions

$$X(0) \in E_1, X(t_n) \in E_2, \forall t_n \in (t_1, \infty), s \in E_2, \tag{3}$$

where X is partial phase vector x_1, α_0, α_1 systems (1), t_1 is time taken to stabilize the system (1), E_1 and E_2 are some subspaces of the solutions phase space. According to the fact that the domain of all permissible states of the system (1) is bounded by E_1, and the required state of the system is bounded by E_2, there is an inclusion $E_2 \subset E_1$.

We will consider the optimal control for which the volume of the phase space region bounded by the subspace E_2 is minimal. With this in mind, we formulate the following quality criterion:

$$\int_0^{t_n} f(Y(t) - s)dt \to \min, \tag{4}$$

where f is non-negative function. The quality criterion (4) is a generalized criterion. As special cases of criterion (4), we can consider the Euclidean norm of the deviation vector $(Y(t) - s)$, or the standard deviation $Y(t)$ from s.

The optimal control problem is to find such controls $u_1(t)$ and $u_2(t)$, that satisfy the quality criterion(4).

For technical systems with switching, the formulation of the optimal control problem and the construction of optimal trajectories are considered in [17, 18, 19]. In [17], we consider an algorithm for switching with increasing frequency for a model of a transport system with three phase variables. In [18], a switching algorithm is proposed for a generalized model of a technical system, and an example with two phase variables is considered. In [19], an algorithm using artificial neural networks to control the switching frequency in a technical system model is proposed.

For the belt conveyor model, taking into account criterion (4), we further propose to use three types of optimal control: sliding mode control, fuzzy controller-based control, and neural network controller-based control.

3. Partial stabilization based on the use of sliding mode

To control the linear speed of movement for the system (1), we use the introduction of the system in sliding mode. The control scheme has the form

$$\begin{aligned} &if x_1 > s_1 \; then \; u_1(t) = \hat{x}_1 - \delta, \\ &if x_1 < s_1 \; then \; u_1(t) = \hat{x}_1 + \delta, \end{aligned} \tag{5}$$

where $\hat{x}_1$ is the internal state of equilibrium, calculated from the equations of the system (2), δ is control panel, s_1 is expected equilibrium state for a phase variable x_1. The status check occurs at discrete time intervals Δt.

For conducting computational experiments, specialized software is developed in the language Python3 using libraries SciPy, NumPy, Scikit-fuzzy. The specified software includes a module for constructing the trajectories of the system (2) based on numerical methods for solving ordinary differential equations with switching, as well as modules for control based on artificial intelligence. Note that the authors proposed an original implementation of neural network algorithms in the language Python3.

Figure 1 shows a graph of the linear speed of movement for the model (2). The control parameters have the form $s_1 = 4$, $\delta = 5$, $\Delta t = 0.005$.

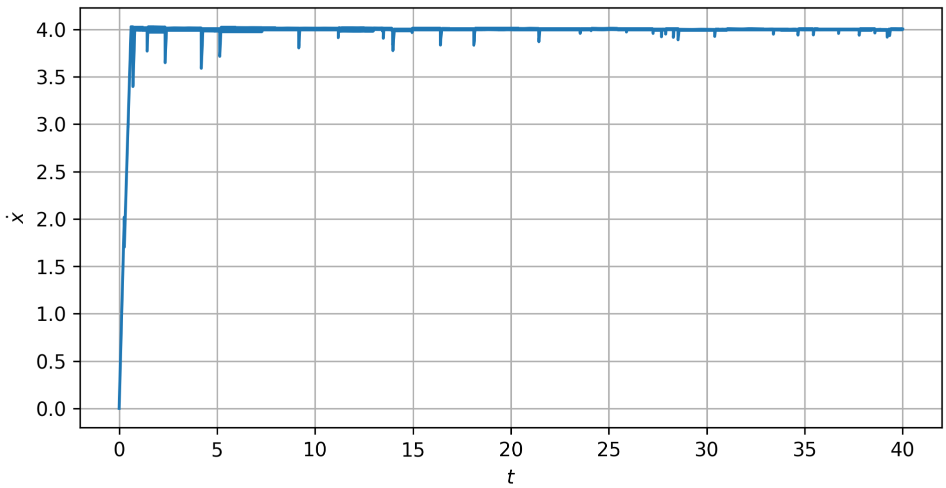

Figure 1: A graph of the linear speed of movement for the model (2) using a sliding mode control.

Table 1
Fuzzy Controller rule base

№	Antecedents	Consequents
1	IF α_0 IS MinusSmall OR P IS MinusSmall	TO $u_2(t)$ IS PositiveSmall
2	IF α_0 IS MinusSmall OR P IS MinusSmall	TO $u_2(t)$ IS PlusSmall
3	IF α_0 IS PlusSmall OR P IS MinusSmall	TO $u_2(t)$ IS PlusSmall
4	IF α_0 IS PlusSmall OR P IS PlusSmall	TO $u_2(t)$ IS MinusSmall
5	IF α_0 IS PlusLarge OR P IS PlusLarge	TO $u_2(t)$ IS MinusLarge
6	IF P IS PlusLarge	TO $u_2(t)$ IS MinusLarge
7	IF P IS MinusLarge	TO $u_2(t)$ IS PlusLarge

Taking into account the results presented in Figure 1, we note slight fluctuations associated with a large step of numerical integration. An increase in the mass of the loads on the conveyor leads to an abrupt change in the value of x_1, but the system quickly returns to the state of equilibrium $s_1 = 4$.

Based on the results obtained, it can be concluded that the sliding mode control is sufficient for partial stabilization with respect to the linear speed. Next, we consider the methods for controlling the angular position of the system (2).

4. System control based on intelligent methods

Computational experiments show that to control the angular speed does not lead to stability with a scheme of the form (5). Taking this fact into account, we use two types of intelligent control: control based on a fuzzy controller and control based on a neural network controller. To develop the program code of the fuzzy controller, the library is used Scikit-fuzzy.

A test example of the fuzzy controller rule base is shown in the table 1.

The rule base presented in Table 1 is formed on the basis of the expert's knowledge of the laws governing the lifting angle of the conveyor. When constructing the rule base, the need to meet the quality criterion (4) is indirectly taken into account.

The phase curve of the angular position of the system (2), constructed using a fuzzy controller, is shown in Fig. 2.

In Fig. 2, we take the lifting angle of the conveyor system (2) as α. The red plus sign in Fig. 2 we denote the expected equilibrium point on the phase plane. The trajectory shown in Fig. 2 is characterized by orbital stability in the vicinity of the equilibrium point s_1.

The graph of the conveyor lifting angle on the plane (α, t) is shown in Fig. 3. Taking into account the results presented in Fig. 3, it can be noted that the system (2) stabilizes in a relatively short time, but the angle of ascent fluctuates relative to the required value $\frac{\pi}{4}$.

The graph of such control $u_2(t)$, which is obtained using the control based on the fuzzy controller, is shown in Fig. 4. It should be noted that in this case, the control is oscillating in nature with an increasing average value.

The results obtained indicate that the stabilization conditions (3) of the system (2) can be considered fulfilled. However, the fuzzy controller does not directly take into account the

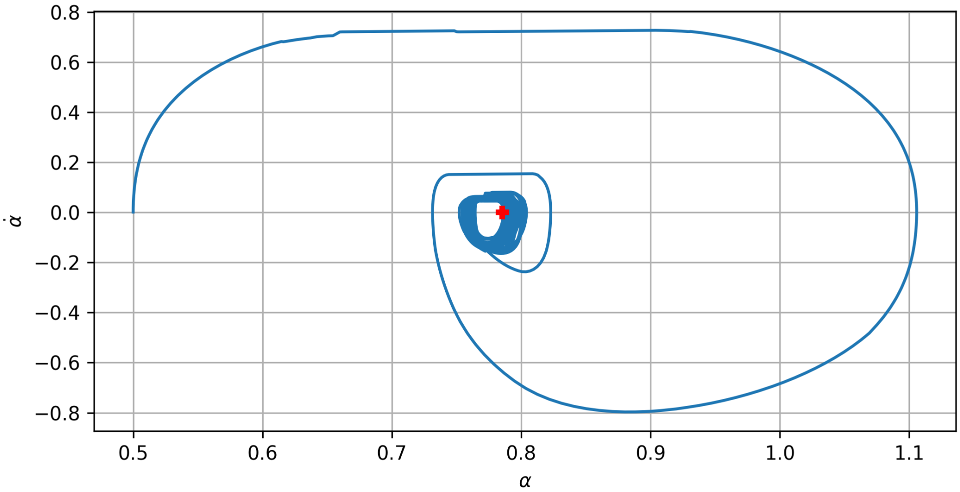

Figure 2: Phase curve of the angular position of the system (2).

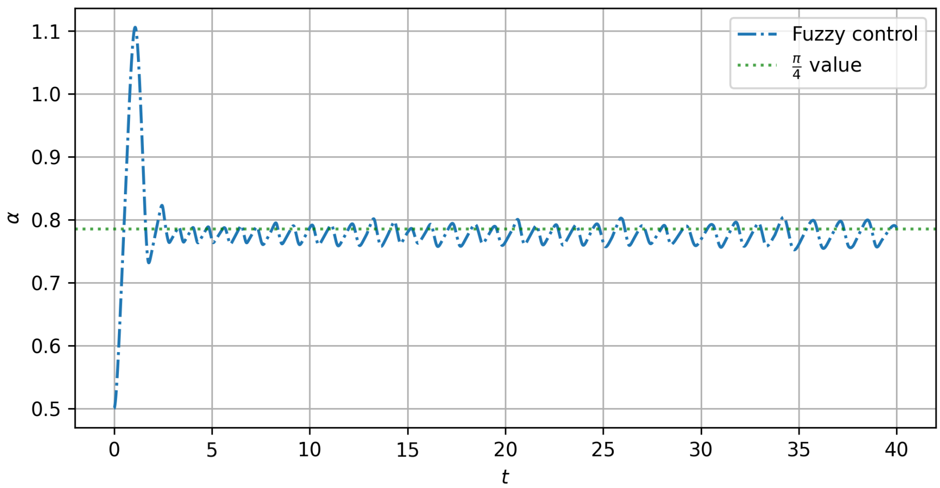

Figure 3: Graph of the lifting angle of the conveyor for the system (2), taking into account the fuzzy control.

quality criterion (4), which makes it difficult to quantify the quality of control.

In addition to the fuzzy controller for the system (2), we are developing a neural network controller for the system (2). We use a direct propagation perceptron with 8 neurons in a hidden layer and a tangential activation function. The neural network diagram is shown in Fig. 5.

On neural network inputs $\hat{i_1}$ and $\hat{i_2}$ the values α_1 and α_{0-s_1} are supplied, respectively. The output layer contains one neuron, the output value of which lies in the interval (-1, 1). The

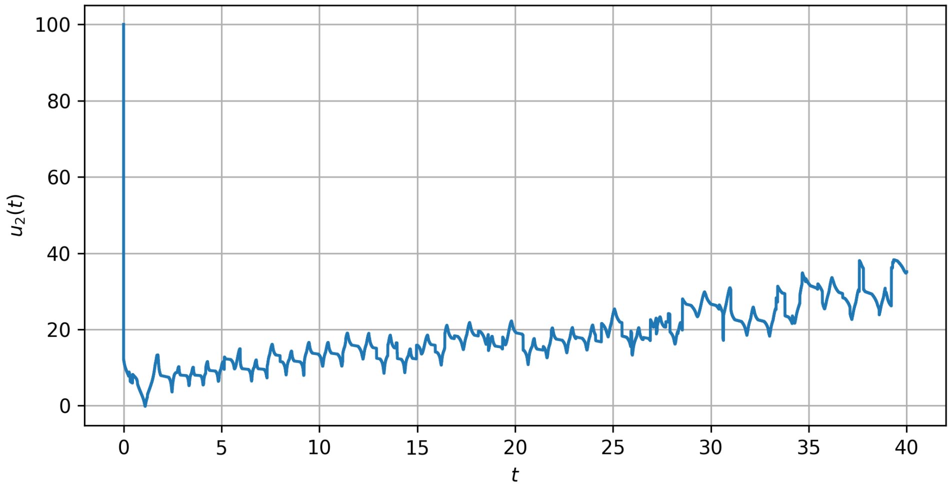

Figure 4: Control graph $u_2(t)$ using a fuzzy controller.

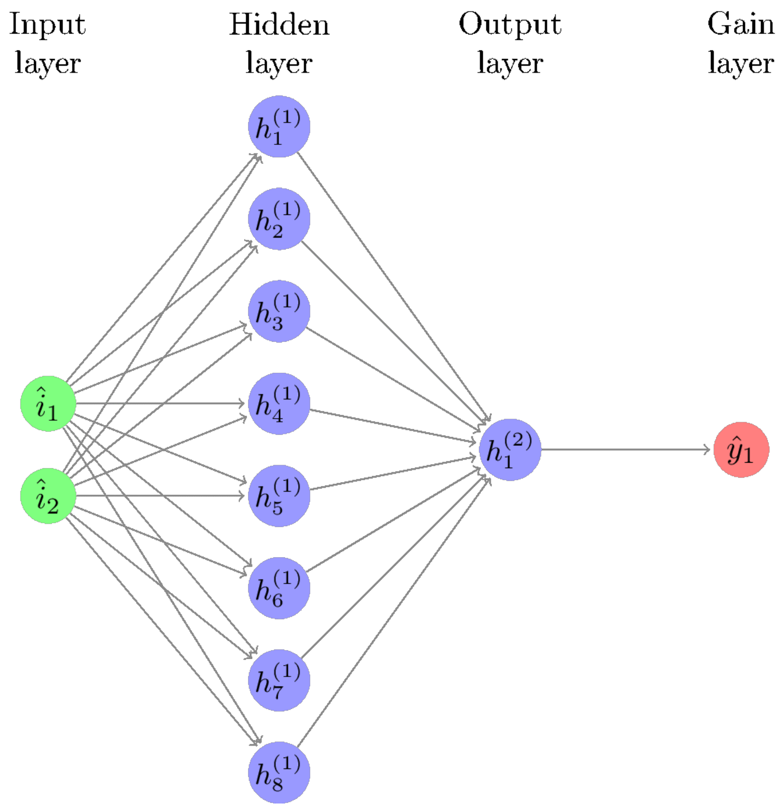

Figure 5: Topology of the neural network controller for the system (2).

specified output value is multiplied in the gain layer by a fixed positive value.

Neural network training is based on reinforcement learning. As a penalty, the quality criterion is used, written in the following form

$$\int_0^{t_n} \|\alpha_0(t) - s_1\| \to \min .$$

To train the neural network, we use heuristic search engine optimization algorithms. A number of effective search engine optimization algorithms are considered in [20, 21, 22]. In this paper, differential evolution from the SciPy mathematical library is used for optimization. A characteristic feature of this training method application to the model (2) is that the training is performed for the case ε = const. With this fact in mind, we arrive at incomplete conditions for a computational experiment compared to a control based on a fuzzy controller. Next, using the developed software, we check how the change in the parameter ε affects the stability of the system (2) with respect to α_0, α_1 with the resulting neural network control.

The results of computational experiments are shown in Fig. 6, 7.

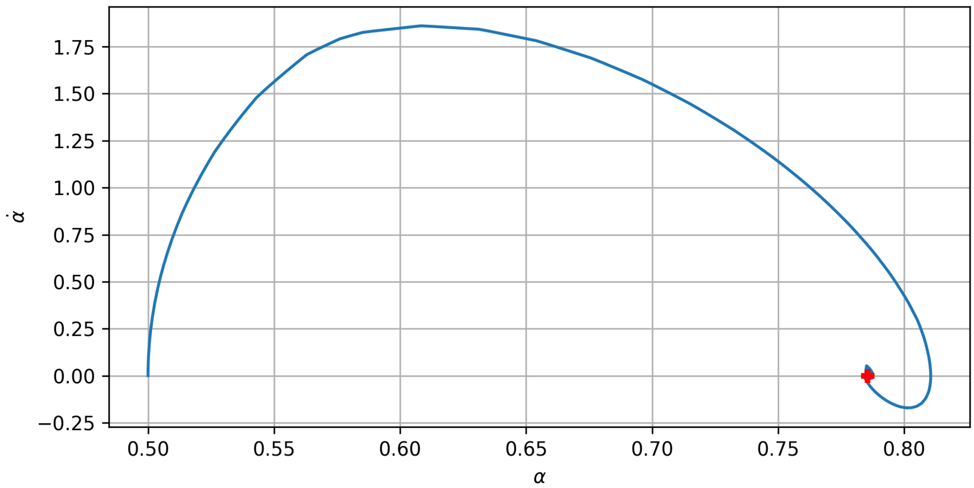

Figure 6: Phase curve of the angular position of the system (2) with a neural network controller.

Fig. 6 shows the phase curve of the angular position of the system (2) with the neural network controller. According to Fig. 6, the neural network algorithm stabilizes the system (2) faster and more accurately with respect to α_0, α_1 compared to the algorithm based on the fuzzy controller.

The graph of the control $u_2(t)$, obtained using the control based on the neural network controller is shown in Fig. 7.

According to Fig. 7, we note the step-like form $u_2(t)$, which is caused by the abrupt change in the mass of loads on the conveyor. The average moving value of $u_2(t)$ tends to increase due to an increase in the average weight of loads on the conveyor.

The developed program for neural network control of model (2) is based on the use and development of proprietary software models and algorithms for neural network computing.

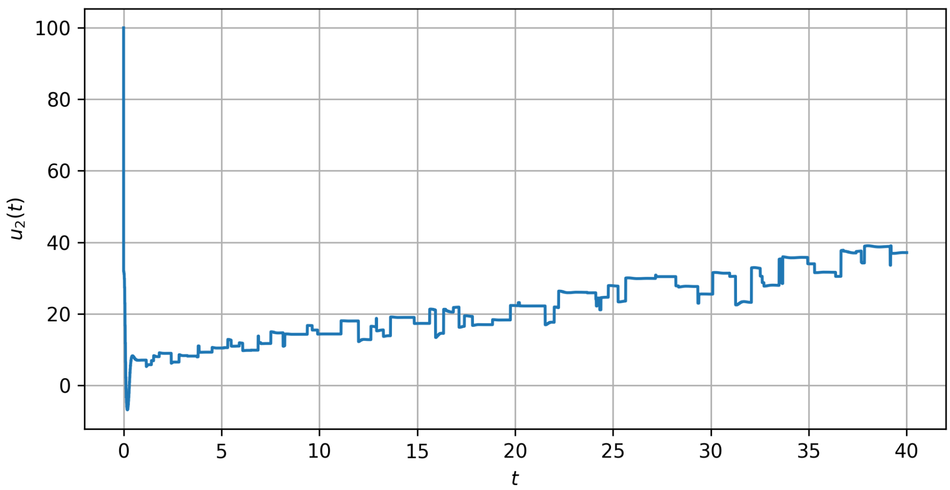

Figure 7: Control graph $u_2(t)$ using a fuzzy controller.

5. Developed software for modelling of conveyor transport systems

To implement neural network algorithms, the author's team developed an original module for neural network computing. This module is developed using the Python 3 language and the Numpy computing library. The basic structure of the neural network computing module is shown in Fig. 8.

For the developed module, the logical organization of an artificial neural network is a directed connected list with the possibility of branches and cycles. The neural network outputs are calculated by traversing the linked list. In this case, the neural network can be represented as a directed graph. Information between the nodes of such a graph is transmitted in the form of vectors. An example of a neural network graph is shown in Fig. 9.

As for the current software, the neural network computing module has achieved the implementation of two types of layers (nodes). The specified types include the fully connected layer and the input layer. In addition, we achieved the implementation of the possibility of recurrent node traversal. To activate layers, the neural network module provides the following functions: tangential function, Heaviside function, linear function.

Listing 1 shows a part of the code for initializing a neural network with a 2-8-1 topology (see Fig. 5) using the developed module. The *dummy_layer*() class creates an input layer that does not have activation functions. The *set_child*() method is responsible for linking layers to each other. The weights are set using the *set_weights*() method. The *calculate*() method calculates the neural network outputs and returns the response as a vector column. Since the column vector in this case actually consists of a single element, it is necessary to extract it by zero indexes. It can be noted that the API of the developed neural network module is minimalistic and easy to use.

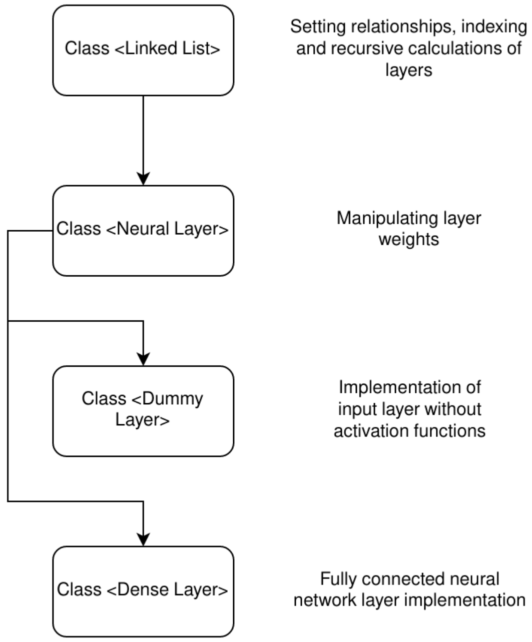

Figure 8: Basic structure of the neural network computing module.

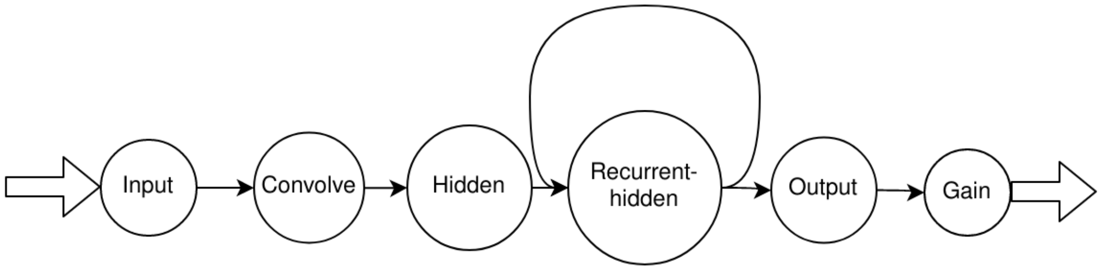

Figure 9: Example of a neural network graph.

```
network = dummy_layer(size = 2)
network.set_child(new_dense_layer(size = 8, activation=tanhl))
network.set_child(new_dense_layer(size = 1, activation=tanhl))
network.set_weights(weights)
network.calculate(np.vstack([model.pos, model.speed]))[0][0]
```

Listing 1: Neural network initialization code part

Listing 2 shows is the main code for neural control of model (2). In the specified program code, you can select the frequency of switching, control methods, and the probability function of object loading/unloading.

```python
def evaluate(state = pi/2.5, control = "fuzzy", weights = None):
    model = model1(args = {"s": 2,
                       "mass": 1,
                       "u": 100,
                       "eps": 1,
                       "mass_vec": np.array([1,0,0,0])})
            #^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
            # model description, special class "model1"

    x = ode(model, 0, [0,0,.5,0], 40, max_step=0.0025)

    plane = []; time = []; U = []; S = []
    for step,phase in enumerate(x): # ODE solver steps
        plane.append(phase)
        time.append(x.t)
        U.append(model.args["u"])
        S.append(model.args["s"])

        if step % 2 == 0: #every 2 steps
            sliding_ctrl(4, model, x)

        if step % 5 == 0: #every 5 steps
            if control == "fuzzy":
                model.args["u"] = fuzzy_ctrl(state, model, x)
            if control == "neural":
                model.args["u"] = neural_ctrl(state, model,
                                              x, weights)

        if uniform(0,1) > .995: #random action
            load_event(x, model, uniform(-.25,.25),
                       uniform(.8,1.2))

    return plane, time, U, S
```

Listing 2: Main code for control of model (2)

Note that the developed neural network computing module has the properties of versatility, extensibility and flexibility when used within various software and hardware platforms.

6. Conclusion

The analysis of the intelligent control models of a belt conveyor with a variable lifting angle carried out in this paper shows sufficient efficiency of the methods used. It is important to emphasize that the developed model takes into account the acting forces, uneven loading of the load and the ability to control the lifting angle of the conveyor belt. The developed complex control quality criterion is used to solve the optimal control problem. The graphs of the trajectories are obtained taking into account the selected parameters of the models. The conditions for the stabilization of the belt conveyor control systems are obtained. Control algorithms based on the construction of fuzzy controllers are developed. The proposed approach to the search for optimal control based on artificial intelligence methods demonstrates high efficiency in stabilizing the lifting angle for the belt conveyor model. Qualitative effects in belt conveyor control models are revealed. The prospects for the application of the developed algorithmic and instrumental software for the implementation of software and hardware complexes for intelligent control of a belt conveyor with a variable lifting angle are that they can be used at enterprises of the innovative cluster of mechanical engineering. The comparative analysis of control based on the synthesis of a fuzzy controller and control using artificial neural networks showed a higher efficiency of neural network control, but the results of control based on the synthesis of a fuzzy controller demonstrated the convenience of using the knowledge base expert assessments. The use of the Python3 language in combination with the Jupyter system and the Numpy, Scipy, and Sympy mathematical libraries allowed us to achieve high rates of development speed and program code quality.

Acknowledgments

The research was funded by RFBR and Lipetsk Region, project number 20-47-480003.

References

[1] V. K. Dyachkov, M. A. Rikman, Overhead pushing conveyors with automatic addressing, 1964.

[2] V. V. Dmitrieva, Modern problems of automation of conveyor belt transport, mining information and analytical bulletin (scientific and technical journal (2013). URL: https://cyberleninka.ru/article/n/sovremennye-zadachi-avtomatizatsii-lentochnogo-konveyernogo-transporta.

[3] M. Andrejiova, A. Grincova, D. Marasova, Measurement and simulation of impact wear damage to industrial conveyor belts, Wear 12 (2016) 368–369. doi:10.1016/j.wear.2016.10.010.

[4] Y. N. Kozhubaev, I. M. Semenov, Belt conveyor control systems, scientific and technical bulletin of the st, Petersburg State Polytechnic University 195, volume 2 (2014) 181–186.

[5] Y. E. Zyuzicheva, Model of a belt conveyor located at an angle to the horizon. determination of the optimal inclination angle for the transition process (2006). URL: https://cyberleninka.ru/article/n/model-lentochnogo-konveyera-raspolozhennogo-pod-uglom-k-gorizontu-opredelenie-optimalnogo-dlya-perehodnogo-protsessa-ugla-naklona.

[6] O. F. Lutfy, H. Selamat, S. B. Noor, Mohd, Intelligent modeling and control of a conveyor belt grain dryer using a simplified type 2 neuro-fuzzy controller, Drying Technology 33 (2015) 1210–1222. doi:10.1080/07373937.2015.1021007.

[7] S. N. Vasiliev, To intelligent control, Nonlinear control theory and its applications, Moscow, Fizmatlit, 2000.
[8] V. I. Gostev, Fuzzy controllers in automatic control systems, Kiev, Radiomator, 2008.
[9] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst., Man and Cybernetics 15 (1985) 116–132.
[10] M. Alattar, A. Alsammak, Speed control of asymmetrical six phase induction motor based fuzzy logic controller, International Research Journal of Engineering and Technology 7 (2020) 213–217.
[11] Y. I. Kudinov, A. Y. Kelina, A. V. Bukhtiyarov, F. F. Pashchenko, A. F. Pashchenko, Classification of fuzzy pid controllers, in: Proceedings of the Tenth International Conference «Large-scale systems development control», Publishing house of V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia, 2017, p. 444–446.
[12] E. V. Igonina, O. V. Masina, O. V. Druzhinina, Stability analysis of dynamic systems based on intelligent control methods and properties of linear matrix inequalities, Bunin Yelets State University, Yelets, 2020.
[13] O. V. Masina, O. V. Druzhinina, Modeling and analysis of stability of some classes of control systems, Dorodnitsyn Computing Center of RAS, Moscow, 2011.
[14] O. V. Druzhinina, O. V. Masina, Stability analysis methods of dynamic intelligent control systems, URSS Group Publishing House, Moscow, 2016.
[15] A. Sundas, A. Bhatia, M. Saggi, Reinforcement learning, machine learning and big data: Concepts, algorithms, Tools and Applications (2020) 281–303.
[16] Z. Ding, Y. Huang, H. Yuan, H. Dong, Introduction to reinforcement learning, Deep Reinforcement Learning (2020) 47–123.
[17] O. V. Masina, O. V. Druzhinina, A. A. Petrov, Models for the control of technical systems motion taking into account optimality conditions, in: Proceedings of the Selected Papers of the 8th International Conference on Optimization Methods and Applications «Optimization and Applications», volume 1987, Petrovac, Montenegro, 2017, p. 386–391. URL: http://CEUR-WS.org.
[18] O. V. Druzhinina, O. V. Masina, A. A. Petrov, The synthesis of the switching systems optimal parameters search algorithms, Communications in Computer and Information Science 974 (2019) 306–320. doi:10.1007/978-3-030-10934-9_22.
[19] O. V. Druzhinina, O. V. Masina, A. A. Petrov, E. V. Lisovsky, M. A. Lyudagovskaya, Neural network optimization algorithms for controlled switching systems, Advances in Intelligent Systems and Computing 1225 (2020) 470–483.
[20] A. P. Karpenko, Modern search engine optimization algorithms. Algorithms inspired by nature, 2nd ed., n.e ed., Bauman MSTU, Moscow, 2016.
[21] M. Sakharov, A. Karpenko, Meta-optimization of mind evolutionary computation algorithm using design of experiments (2018) 473–482.
[22] O. Nelles, Nonlinear Global Optimization, Nonlinear System Identification, Springer, in, 2001. doi:10.1007/978-3-662-04323-3_5.