Separation process control system in the cement production facility

E A Muravyova\(^1\), R F Gabitov and P A Sabanov

Ufa State Petroleum Technological University, Branch in Sterlitamak, Sterlitamak, Russia

\(^1\)E-mail: muraveva_ea@mail.ru

Abstract. The article developed a system for controlling the process of cement separation using neural networks. The control object is a separator. The task is to improve the cement production process control system by installing a frequency converter and an electric air damper. In order for the current frequency and the degree of opening of the damper to be set automatically, a neural network was developed for the separation process, which has two input and two output parameters. The neural network is able to adjust the rotor speed of the separator, relying on sensors that control the speed of the separator rotor and the size of the cement particles. If the speed of rotation of the rotor of the separator becomes low, artificial intelligence will adjust the fineness of grinding, changing the degree of opening of the damper for air supply to the separator. If the speed of rotation of the separator rotor approaches the maximum, and the grinding fineness has not reached the specified values, then the grinding fineness is controlled by changing the degree of opening of the damper for air supply to the separator.

1. Data collection, processing and preparation

For the development of an intelligent control system (ICS) using a neural network, it is necessary to collect data for its training. A training data set is a set of values of input and output variables of an object necessary for its training [1].

Before creating a table with a training set of samples, we determine which input variables influence the process of cement separation and have a mutual influence.

In a multiply connected cement separation process, it is necessary to take into account the input and output variables for controlling the cement separation process, which are shown in tables 1 and 2.

| Input variables | Valid values | Normal process values | Unit |
|-----------------|--------------|-----------------------|------|
| 1) Separator rotor speed | 200…575 | 275…500 | rpm |
| 2) Cement particle size | 0,01…50 | 10…40 | micron |
Table 2. Output variables.

| Output variables                                         | Valid values | Normal process values | Unit |
|----------------------------------------------------------|--------------|-----------------------|------|
| 1) Current frequency of the frequency converter of the separator motor | 20…60        | 27…57                 | Hz   |
| 2) The degree of opening of the damper for air supply to the separator | 20…80        | 50                    | %    |

To determine the current frequency of the frequency converter inverter of the separator and the degree of opening of the air damper, we need to know: the rotor speed in the separator (1-70a) and the size of the cement particles (1-83a). To do this, it is necessary to send signals to the control system from a speed sensor and a laser particle size analyzer located on the separator (1-70a) and in the production line after the separator (1-83a), respectively (figure 1) [2].

The size of cement particles depends on rotor speed of separator, the higher the speed, the smaller the particle size and vice versa, with a decrease in speed, the size of cement particles increases. If the speed of the separator rotor is critically low or critically high, then the specified size of cement particles is achieved by changing the degree of valve opening of air supply to separator. The speed of rotation of separator rotor changes in direct proportion to current frequency of a frequency converter of separator motor.

Figure 1 shows the location of the sensors of the cement separation unit. This node circuit of separator has 2 input (table 1) and 2 output variables (table 2). ICS based on a neural network will control the fineness of grinding, given the interconnection of parameters.

We will form a training sample for a neural network. We write the data obtained in table 3, which consists of 940 examples, which takes into account all the variables that affect the fineness of the cement and are involved in the process of cement separation [3].

Figure 1. Diagram of the separation unit.
After creating a training sample, we proceed directly to the development of a neural network. To do this, choose the type and architecture of the neural network, which is most suitable for solving the problem of controlling the process of cement separation [4].

2. The choice of the type and architecture of neural networks
The values of the output variables are largely formed by the chosen neural network architecture: by the number of hidden layers and neurons in them, by the nature of interneural connections in the hidden layers. The setting of the specified parameters of the neural network is performed during training of artificial neural networks. Today, there is no definite way to set the number of neurons in hidden layers, at the same time, if the number of neurons is insufficient, the network will be undertrained (it will not produce acceptable results), if the number of neurons is excessive, then the time spent on training the network increases. The only way to determine the number of neurons in a network is the experimental data on training and testing of artificial neural networks.

For the process under study according to the results of the performed experiments, it was revealed that the required number of neurons in the hidden layer is 12. There is one hidden layer used in the neural network architecture (figure 2).

The structural plan of the network (figure 2) includes 2 input variables (table 4) and 2 output variables (table 5).

| Separator rotor speed, rpm, R | The particle size of the cement, microns, G | Frequency of the frequency converter, Hz, N | Degree of opening, %, Z |
|-----------------------------|-----------------------------------|----------------------------------------|----------------------|
| 200.00                     | 50.00                             | 20.13                                  | 80.00                |
| 201.00                     | 49.87                             | 20.23                                  | 79.60                |
| 202.00                     | 49.73                             | 20.33                                  | 79.20                |
| 203.00                     | 49.60                             | 20.44                                  | 78.80                |
| 204.00                     | 49.47                             | 20.54                                  | 78.40                |
| 205.00                     | 49.34                             | 20.64                                  | 78.00                |
| 206.00                     | 49.20                             | 20.74                                  | 77.60                |
| 207.00                     | 49.07                             | 20.84                                  | 77.20                |
| 208.00                     | 48.94                             | 20.94                                  | 76.80                |
| 209.00                     | 48.80                             | 21.04                                  | 76.40                |
| 210.00                     | 48.67                             | 21.14                                  | 76.00                |
| 211.00                     | 48.54                             | 21.24                                  | 75.60                |
| 212.00                     | 48.40                             | 21.34                                  | 75.20                |
| 213.00                     | 48.27                             | 21.44                                  | 74.80                |
| 214.00                     | 48.14                             | 21.54                                  | 74.40                |
| 215.00                     | 48.01                             | 21.64                                  | 74.00                |
| ...                        | ...                               | ...                                    | ...                  |
| 570.00                     | 0.66                              | 57.38                                  | 12.00                |
| 571.00                     | 0.53                              | 57.48                                  | 11.60                |
| 572.00                     | 0.40                              | 57.58                                  | 11.20                |
| 573.00                     | 0.27                              | 57.68                                  | 10.80                |
| 574.00                     | 0.13                              | 57.78                                  | 10.40                |
| 575.00                     | 0.03                              | 57.88                                  | 10.00                |
Table 4. Input variables.

| Variable | Parameter |
|----------|-----------|
| I        | 2         |
| R        | Separator rotor speed |
| G        | Cement particle size |

Table 5. Output variables.

| Variable | Parameter |
|----------|-----------|
| I        | 2         |
| N        | Separator rotor motor frequency inverter current frequency |
| Z        | The degree of opening of the damper for air supply to the separator |

Figure 2. Block diagram of neural network.

The Levenberg-Marquart optimization algorithm is used as a neural network learning algorithm. The chosen algorithm minimizes the combinations of squared errors and weights, after which determines their corrected combination which provides an increase in the training ability of the network.

The next step after defining the architecture of a neural network is its construction and training. For these purposes, we will collect a training set.

3. Building and training a neural network in Matlab

Variables for training of neural networks are imported and then displayed in the Matlab environment into datasets: input and output.

For the work with neural networks, a command nnstart is used, which opens the “Neural Network Start” window.

To set the input and output values for the neural network, let us press the "Fitting app" button. Next, the "Neural Fitting (nftool)" window will be opened, where the neural network will be described.

Let us set 70% in "training" section, 15% in "NN check" section, and 15% in "testing". To determine the number of neurons in the "Number of Hidden Neurons" section, let us set the number to 12.

Let us select "Levenberg-Marquardt" in the "Choose a training algorithm" section of the neural network training algorithm. The NN training is started with the “Train” button.

Figure 3 shows the given architecture of the neural network, which consists of 2 inputs, 2 outputs, the hidden layer has 12 neurons specified, the totality of which performs a linear transformation of the "weight" matrix.
The result of training of the neural network can be seen in the window of completion of the training process, which displays the number of iterations performed – 1000, time spent on training – 00:00:04 h, mean square error – 0.00580, gradient value – 0.00604, regularization value is 0.00100 and frequency of deviations of the obtained values from a given error – 1.

Being in the window of completion of the training process, after clicking on "Performance" button, the network training graph can be seen (figure 4). Here the training error (its dynamics) is presented, which shows the deviation of a real value from the given one. The graph shows that within 999 epochs the value of the root mean square error of 0.0058605 was reached, which is less than the specified one – 0.315.

It is then seen how the gradient and training coefficient changed during the process of neural network training (figure 5). The gradient graph shows how the gradient of the learning error functional by the neural network weights changed. The weights of the network are considered as arguments of the function, where, in order to find the minimum, there is a sequential advance to lower and lower points in the search space. The gradient graph clearly shows the jumps in the search for the best solution for adjusting the values of the training sample. The gradient value 0.0060351 is the sum of all adjustments for the variables that are above the specified error rate.

The Mu graph (figure 5) shows how the regularization variables of the method we have chosen – Levenberg-Marquardt method – changed. Regularization is the range of numerical values needed to adjust values of the training selected subset and retrain the neural network. On the obtained Mu graph (figure 5) it can be seen that such numbers are in the range $[10^{-6}; 10^{-1}]$, and since the magnitude of the values in this range is negligible, therefore, these values may be neglected [9]. The Validation checks graph (figure 5) shows the frequency of deviation of the received values from the given error. Figure 5 shows that:

- at 75th epoch of training, the frequency of deviations increases to 1, after which in next epoch the neural network eliminated the deviation;
- at 165th epoch of training, the frequency of deviation increases once again to 1, after which the neural network eliminated the deviation in the next epoch;
- further, single deviations appear at 210, 220, 230, 240 epochs, after which the neural network in 211,221,231,241 erases eliminates the deviations;
• on 310, 320, 330, 340, 350 epochs, the frequency of deviations increases to one, after the NA in 311, 321, 331, 341, 351 eras, eliminates deviations and training continues;
• further, the frequency of deviations reaches unity at 410, 430, 490, 520, 560, 640 eras [10].

Having analyzed the Validation checks graph, we conclude that it is advisable to stop the process of training a neural network in the period from the 640th to the 1000th era. During this period, the neural network learned to eliminate previously appeared deviations and has no further development, in view of the absence of deviations [11].

A value of 1 assigned to the Validation checks graph indicates the latest frequency of deviation from the given error. Neural network training ceased due to the achievement of the maximum number of eras.

Figure 5. Neural network training graphs.

Figure 6 shows a histogram of errors. It displays on how many examples the neural network gives one or another error. This error is calculated as the difference between the target value and the output of the neural network. The graph shows that the errors lie in the range [-0.1765; 0.1939], with initial ≈250 values. Further, the error range has changed from [-0.01765; 0.1939] to [-0.1181; 0.1155] when the number of values has reached ≈120. After ≈120 values, the error range has changed from [-0.1181; 0.1155] to [-0.05974; 0.05706]. Further, when the value of epochs reached ≈210, the error range is fixed by one value close to zero and equals -0.0034.

Figure 6. Histogram of errors.
The given histogram shows that the neural network when increasing training data set, produces a smaller value of data error at all stages of training and testing. The resulting error margin is within acceptable ranges.

Figures 7–10 show graphs of error regression in the “training”, “verification”, “testing” modes and a graph of error regression in all modes that display a linear regression of the results of training a neural network on subsets. Also, for each result, the correlation coefficient R is calculated, the regression equation is derived in the general form: Output=a·target+b, and a regression graph is constructed using this equation. R = 1 - with full coincidence of the target values and outputs of the neural network. As shown in (figures 7–10), the regression of the “training” mode is R = 0.99999, the regression of the regime is “test” R = 0.99999, the regression of the regime is “test” R = 0.9998, and the regression of the totality of all modes is R = 0.99999 [13].

In percentage terms, the coincidence of the target values and outputs of the neural network is: for the “training” mode = 99.99%; for the “check” mode = 99.99%; for the test mode = 99.98% for all modes = 99.99%.

The above values allows to conclude that the developed neural network approximates this function quite qualitatively [14].

![Figure 7](image1.png)  **Figure 7.** Schedule regression mode “training”.

![Figure 8](image2.png)  **Figure 8.** Regression graph of the “check” mode.

![Figure 9](image3.png)  **Figure 9.** Regression graph of the test mode.

![Figure 10](image4.png)  **Figure 10.** Schedule of regression of all modes.

The program code of the developed neural network is shown in (figure 11).
4. Neural network testing

Testing has shown that the developed neural network without reconfiguration of link weights, gives a tolerance of the output variable within a specified range when any set of input variables from the test set is given to input of the neural network. The selected neural network "Neural fitting" trained for 1000 epochs has a minimum mean square error of 0.005865 at 999 epoch. Such values indicate that the trained neural network gives a high accuracy in controlling the speed of the separator motor and the degree of valve opening.

To check the results in network training, we should enter the command: `sim (net, [R, G])`. Then the input parameters are entered and the output parameters are displayed in the following form:

```
ans =
N, Z.
```

Test result when the rotor speed of the separator is critically low and the cement particle size is large

```
w sim (net, [205; 49.3351])
ans =
20.6375
21.9874
```

In this case, the current frequency of the frequency converter is set in accordance with the speed of rotation of separator rotor and, due to critically low speed, the air supply damper to the separator smoothly closes to achieve the specified grinding fineness and to avoid material flow jam [16].

The result of the check when the values of the rotor speed of the separator rotor and the particle size of cement are average;

```
>> sim (net, [333; 32.3138])
```

![Neural network program code.](image)
In this case, the current frequency of the frequency converter is set in accordance with the speed of rotation of the separator rotor, and the valve for air supply to the separator is maintained half-opened.

Test result when the rotor speed of the separator is critically high, and the particle size of the cement is small:

\[
\text{ans} = \\
33.5200 \\
50.0671
\]

In this case, the current frequency of frequency converter is set in accordance with the speed of rotation of the separator rotor and, due to the critically high speed, air supply valve opens smoothly to achieve the specified grinding fineness and to avoid breakage and rapid wear of the separator [17].

5. Conclusion

We have developed, trained and tested an artificial neural network aimed to control the process of cement separation which also takes into account mutual influence of parameters. The simulation of neural network was performed in Matlab program using Matlab Neural Network Toolbox package.

Training and test selected subset of the network includes 940 examples. Using these, a neural network was developed that gives minimum error of output parameters when any set of input data from the training and test sets is given to network input. The resulting neural network has a minimum root-mean-square error of 0.0058605 obtained over 999 epochs.

As a result of testing, the network has shown output values with a maximum absolute error of ± 0.06, and a reduced relative error of 0.01%, which is less than the specified 0.315%.

The ICS developed for separation process has the ability to independently analyze data on speed of rotation of separator rotor and the size of cement particles, and after that make decisions on regulating current frequency of frequency converter and the degree of valve opening, which will lead to:

- increasing the quality of cement grade, due to timely response and monitoring of the parameters of grinding fineness control process;
- will reduce construction costs due to an increase in the quality of budget cement grades [20].

References

[1] Spitsyn V G and Tsoi Yu R 2007 The use of artificial neural networks for information processing: guidelines for laboratory work (the use of artificial neural networks for information processing: guidelines for laboratory work) (Tomsk: TPU Publishing House) pp 128-34

[2] Muravyova E A and Popkov V V 2019 Development and Research of a Dynamic Flow Laboratory Bench Model (Ufa: 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS)) pp 117–2

[3] Muravyova E A and Gabitov R F 2018 Economic Features to Optimize the Catalyst Calcinations Process (Vladivostok: International Scientific Multi-Conference on Industrial Engineering and Modern Technologies FarEastCon) p 178

[4] Muravyova E A, Sharipov M I and Gabitov R F 2018 SCADA-System Based on Multidimensional Precise Logic Controller for the Control of a Cement Kiln (Vladivostok: International Scientific Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon) p 128

[5] Muravyova E A and Sharipov M I 2018 Method of optimal parameters control in three-phase separator using fuzzy controller (Novosibirsk: International Conference on Actual Issues of Mechanical Engineering (AIME)) pp 419-23
[6] Muravyova E A and Uspenskaya N N 2018 *Development of a Neural Network for a Boiler Unit Generating Water Vapour Control* vol. 27 (Moscow: Optical Memory and Neural Networks) pp 297-307

[7] Haikin S 2006 *Neural Networks: a Complete Course* (Moscow: Publishing house «Williams») p 1104

[8] Kriesel D 2014 *Brief Introduction to Neural Networks* (Peru: «Autoedicin») p 244

[9] Callan R 2003 *Basic concepts of neural networks* (Moscow: Publishing house «Williams») p 291

[10] Nikolenko S and Kadurin A 2018 *Deep learning. Immersion in the world of neural networks* (SPb.: Peter) p 477

[11] Alekseev B V 1980 *Cement Production Technology* (Moscow: Higher School) p 260

[12] Haykin S 2006 *Neural Networks: A Comprehensive Foundation* (New Jersey: Prentice Hall PTR Upper Saddle River) p 1104

[13] Muravyova E A and Azanov A N and Enikeeva E R 2018 *Modeling power consumption by pump station control systems based on fuzzy controllers with discrete terms* (Irkutsk: International Conference on Aviamechanical Engineering and Transport (AVENT)) pp 429-34

[14] Cu Nguyen Giap and Le Hoang Son Francisco Chiclana 2018 *Dynamic structural neural network* (Shanghai: Journal of Intelligent & Fuzzy Systems) pp 137-41

[15] Dr G Suresh Kumar 2017 *Cellular Neural Network* (India: Fuzzy Cellular Neural Network and its Applications ) p 228

[16] Evmenov V 2009 *Intelligent Control Systems* (Moscow, Russia: book house "Librokom") p 290

[17] Sherfey Jason S, Soplata Austin E, Ardid Salva, Roberts Erik A, Stanley David, Pittman-Polletta Benjamin R and Kopell Nancy J DynaSim 2018 *A MATLAB Toolbox for Neural Modeling and Simulation* (Oslo: Frontiers in Neuroinformatics) p 310

[18] Anghel Daniel-Constantin, Ene Alexandru and Belu Nadia 2014 *A matlab neural network application for the study of working conditions* (Zurich: Advanced Materials Research) pp 197-203

[19] Arturo J Mendez and Maria J Lado 2014 *Integrating Matlab Neural Networks Toolbox functionality in a fully reusable software component library* (Germany: Neural Computing and Applications) pp 147-149

[20] Lunan Zheng 2016 *Matlab Simulink of Varying-Parameter Convergent-Differential Neural-Network for Solving Online Time-Varying Matrix Inverse* (Stockholm: 8th International symposium on computational intelligence and design (ISCID)) p 187