Python-based Fuzzy Control of Ore Feeding System

Yu Du (609028251@qq.com)
Kunming University of Science and Technology https://orcid.org/0000-0002-0798-4407
Yuebing Liu
Kunming University of Science and Technology
Dan Liu
Kunming University of Science and Technology
Ruitao Liu
Kunming University of Science and Technology
Wenkang Zhang
Kunming University of Science and Technology
Longzhou Yu
Kunming University of Science and Technology

Research Article

Keywords: Python, data processing, fuzzy control, beneficiation

Posted Date: May 25th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-419259/v1

Python-based fuzzy control of ore feeding system

Author information

Affiliations

Faculty of Land Resources Engineering,Kunming University of Science and Technology,Kunming 650093,China
Yu Du，Yuebing Liu，Dan Liu，Ruitao Liu&Wenkang Zhang

Yunnan Amade Electrical Engineering Company,Kunming 650033,China
Longzhou Yu

Corresponding author

Correspondence to Dan Liu
e-mail address of the corresponding author：18599278@qq.com

Abstract

Using Python as a programming language, this study investigates the problem of controlling the ore amount in the field of mineral processing. First, data on the influencing factors collected from a certain beneficiation mill in Yuanyang are quantitatively analyzed using the NumPy module library. Factors having a greater influence are screened out and then selected as the input while the motor frequency is generated as the output by the fuzzy control algorithm developed using the SciKit Fuzzy module library. The range of values of the fuzzy control variable is defined, the fuzzy membership function is generated, and the fuzzy control rule is established. Finally, the fuzzy controller is activated to realize fuzzy control of the mine. The NumPy algorithm can be effectively applied to the quantitative analysis of data, and the calculation results are reasonable and interpretable. The simulation results are as follows: the hardness of the raw ore and the weight of the belt scale are the key factors for controlling the ore amount, and they can be used as the input variables of the fuzzy control system. The fuzzy controller developed using the SciKit Fuzzy module library can be effectively applied to the field of mineral processing, and it overcomes the limitations of current computer technology in industrial mineral processing.

Keywords: Python; data processing; fuzzy control; beneficiation

Declarations:

Funding

No funding was received to assist with the preparation of this manuscript.

Conflicts of interest

The authors declare that they have no conflict of interest.

Availability of data and material

All data, models, and code generated or used during the study appear in the submitted article.

Ethical approval

This article does not contain any studies with human participants performed by any of the authors.

1.Introduction

Grinding and grading are critical processes in beneficiation. The quality of the process directly affects the final economic and technical indicators of the beneficiation plant. Therefore, whether the feed rate of the ore can be adjusted according to the nature of the ore and the realization of intelligent ore feed have a significant impact on the subsequent production. Although the pendulum feeder employed by most concentrators can realize continuous and even ore feed, it has poor operation accuracy and it cannot realize intelligent control of the feed quantity. The operation status of the equipment in the concentrator and identification and analysis of the nature of the ore are indispensable to the study of ore quantity control. To meet the requirements of mineral separation, researchers are committed to investigating and finding suitable data processing methods to analyze the influencing factors. Owing to

advancements in computer technology and artificial intelligence, Python is being increasingly used to facilitate automatic control in mineral processing. In particular, Python is easy to learn and use; moreover, it has an open-source code ecology and many mature integrated models. Control theory has achieved significant progress through the application of artificial intelligence to the industrial field, which is driven by programming languages such as Python and Java. However, these programming languages have rarely been used to solve control problems of key variables in the field of mineral processing.

When discussing how humans are better than the most complete machines, Wiener, the founder of cybernetics, said, “People have the ability to use fuzzy concepts.” The human brain can receive and process fuzzy information, make accurate recognition judgments based on a small amount of fuzzy information, and flexibly solve complex problems. Fuzzy control is a system that defines the input, output, and state variables on a fuzzy set. Fuzzy set theory, founded by Zadeh in 1965, is the mathematical foundation of fuzzy control theory, by which a computer programmer carries out algorithmic programming of the programming language on the control rules processed by the human brain using the fuzzy set based on the intuition and experience of simulating people. Thus, it is possible to realize controllability of artificial intelligence, which is the core of fuzzy control. Computer technology is widely used in automatic control and decision analysis in the industrial field because it can simulate the fuzziness of the human brain and realize the ability to deal with problems in a fuzzy manner.

Python program development mainly implements three functions: quantitative analysis, fuzzy controller, and 3D visualization. First, quantitative analysis is realized through the scientific computing library NumPy of Python. The collected data are quantitatively analyzed through a linear regression model, and the factors that have a greater impact on the ore feed volume of the ore feeding system are selected as the input values of the fuzzy controller. The fuzzy controller uses SciKit Fuzzy to establish fuzzy sets and fuzzy control rules, and defines membership functions and other steps. The pendulum feeder gives the miner motor frequency as the output value

to control the motor, visualizes the result in 3D, and finally displays it on the human–machine interface of the central control room.

The contributions of this study are as follows:

1. In traditional mining automation, the system does not have the ability to make independent decisions. To address this problem, an intelligent ore feeding system based on fuzzy control theory is proposed.

2. A quantitative analysis model suitable for the field of mineral processing is developed, which can fit regression lines and output the determination and correlation coefficients. The speed of screening the influencing factors in mineral processing technology is improved significantly.

3. To facilitate observation and recording by on-site employees, the output results are visualized in 3D, which further improves the efficiency and operation process of the entire concentrator.

4. New ideas for intelligent development in the field of mineral processing are presented.

2.Methods

The expected use of Python is shown in Fig. 1. The linear regression model established using the NumPy module library is employed to analyze the influence of various factors on the amount of ore, and the most influential factors are selected as the input variables of the fuzzy control process. Then, the SciKit Fuzzy module library is adopted to carry out the fuzzy inference process. Fuzzy rules and fuzzy sets are established and then defuzzied, and the control variables are generated as the output.

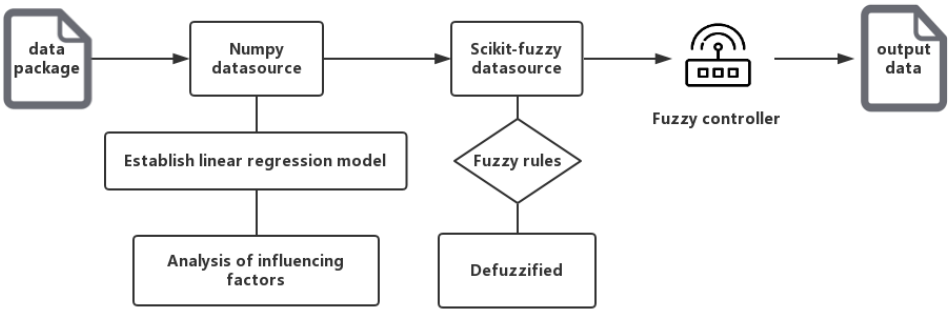

Fig. 1 Development of fuzzy control system for constant feeding by Python

2.1 Data analysis

The data involved in this study were collected at a gold mine beneficiation plant in Yunnan. The belt scale weight and motor frequency data of the selected belt are summarized in Table 1. The original data of the belt scale weight are displayed in the form of a line graph, which is not suitable for direct calculation. Hence, the belt scale weight and motor frequency were recorded every 3 min for a total of 30 min. The hardness of the ore can be categorized as soft, hard, and very hard according to the fragmentation distribution of the discharge port of the crushing section, and it is represented by numbers 1–6. The data of the ore hardness and motor frequency are summarized in Table 2.

Table 1. Belt scale weight and motor frequency

Belt Scale Weight (T/H)	4.1	9.2	10.1	11.2	10.2	10.1	10.3	11.8	10.1	10.2
Motor Frequency (Hz)	50.0	45.2	44.6	43.5	45.1	44.8	45.1	45.1	45.2	45.1

Table 2. Ore hardness and motor frequency

Ore Hardness	1	2	3	4	5	6
Motor Frequency (Hz)	45.8	40.3	38.2	35.7	33.3	30.1

The best way to study the relationship between the variables is to establish a univariate linear regression model and analyze the influence of each independent variable on the dependent variable on the basis of the regression results. As a scientific computing tool, NumPy is an important component for building linear regression models, while SciKit Learn can independently train and evaluate the model, further optimize the model functions, and reduce the errors. A univariate linear regression model is established on the data, and 10000 simulation experiments are performed using the NumPy and SciKit Learn module libraries. The output model is shown in Figs. 2 and 3.

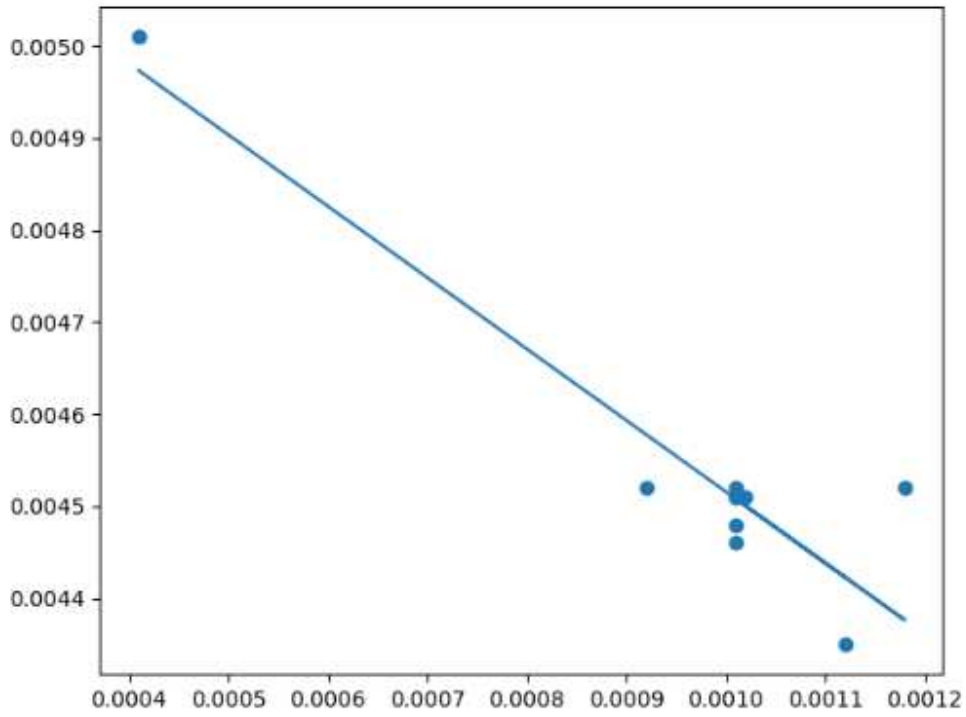

Fig. 2 Linear regression model of belt scale weight and motor frequency after 10000 simulation cycles using SciKit Learn

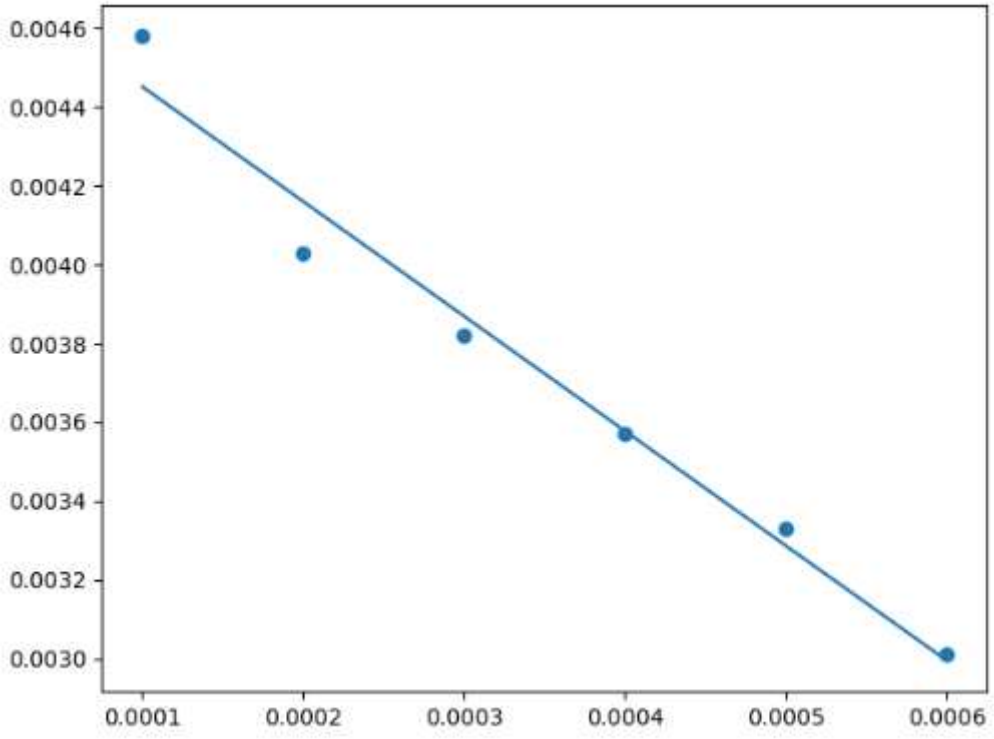

Fig. 3 Linear regression model of ore hardness and motor frequency after 10000 simulation cycles using SciKit Learn

It can be seen that there is a certain residual error between each data point and the regression line after the simulation experiment. In fact, the weight of the belt scale and the hardness of the ore are only two of the important factors affecting the frequency of the motor. Other factors such as the mill status and voltage fluctuations affect the motor frequency of the pendulum feeder. There is a deviation between the regression value and the actual value; hence, the linear regression model must be judged before studying the degree of linear correlation between the variables. The degree of fit can be judged by the determination coefficient, and the degree of linear correlation between the two variables is determined by the Pearson correlation coefficient. The determination and correlation coefficients can be calculated using NumPy. The calculated coefficients are listed in Table 3.

Table 3. Determination and correlation coefficients of belt scale weight, ore hardness, and motor frequency

	Belt scale weight and motor frequency	Ore hardness and motor frequency
Determination coefficient	0.8745	0.9543
Correlation coefficient	-0.9874	-0.9351

It can be seen that the determination coefficients are greater than 0.8, indicating that the two models are well fitted, the error between the predicted value of the simulation experiment and the actual collected data is not significant, and the linear regression model is of good quality. The correlation coefficients are -0.9874 and -0.9351, respectively, indicating that there is a negative correlation between x (belt scale weight and ore hardness) and y (motor frequency), and the correlation is extremely high, i.e., the belt scale weight and ore hardness are two important variables that affect the motor frequency. The core code for establishing a univariate linear regression model and outputting the determination and correlation coefficients is presented below.

Code	Note
import numpy *as* np *from* sklearn.linear_model *import* LinearRegression *import* matplotlib.pyplot *as* plt *import* pandas *as* pd *from* scipy *import* stats *import* pylab	Call module
x = np.array ([ ])	Input data
y = np.array ([ ])	
slope, intercept, r_value = stats.linregress (x,y)	Calculate slope, intercept, and correlation coefficient
lr = LinearRegression () lr.fit (x,y) lr.score (x,y) y_hat = lr.predict (x)	Training model

print (lr.score (x,y)) print (r_value) plt.scatter (x,y) plt.plot (x, y_hat) plt.show ()	Output determination coefficient and correlation coefficient

2.2 Fuzzy control system based on SciKit fuzzification

According to current research, factors such as voltage fluctuations, ore properties, mill operating status, and ore weight of the conveyor belt affect the operating frequency of the motor. In the data analysis, the linear correlation degree between the ore hardness and the weight displayed by the belt scale and the motor frequency is greater than 90%. Therefore, the ore hardness and the weight displayed by the belt scale can be selected as the input variables of the fuzzy control system, while the motor frequency is the output variable. The expected use of SciKit Fuzzy in the fuzzy control system is shown in Fig. 4. First, for the fuzzy input variables, the range and membership function are defined. The input will be decomposed into different fuzzy sets when the fuzzy control rules are established. Then, the data are defuzzified and the motor frequency is output to achieve the purpose of controlling the motor for the mining machine.

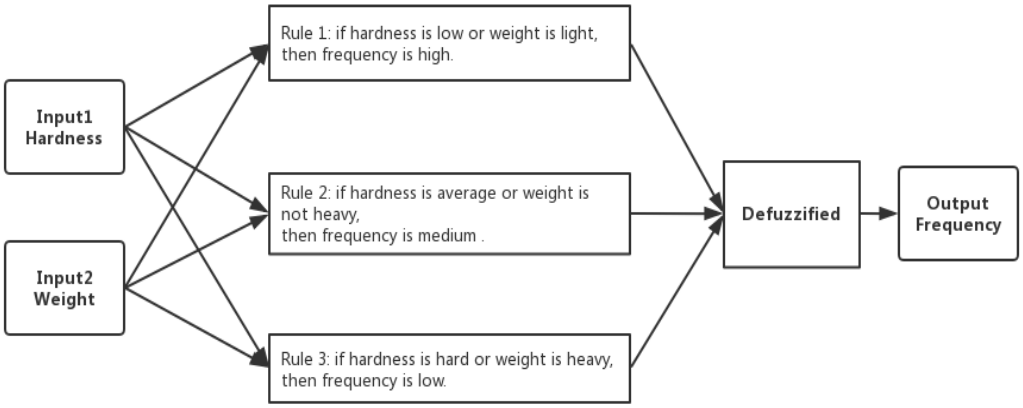

Fig. 4 Fuzzy reasoning process

2.2.1 Fuzzification of input variables and establishment of fuzzy set

The fuzzification of input variables refers to obtaining the membership value of each input variable by comparing the input variable with the membership function. The degree of membership is defined as the membership function. There are two main types of membership functions:

(1) Triangular membership function

$$f(x) = \begin{cases} 1 - \dfrac{|x-m|}{\sigma}, |x-m| < \sigma \\ 0, \qquad\qquad |x-m| \geq 0 \end{cases}$$

where m is the center of the fuzzy set and σ is the width of the fuzzy set.

(2) Gaussian membership function

$$f(x) = e^{-\frac{(x-c)^2}{\sigma^2}}$$

where c is the center of the Gaussian fuzzy set and σ is the width of the Gaussian fuzzy set.

The value range of the y-axis of the membership function is [0–1], i.e., the closer the membership function value of the object is to 1, the higher is the degree of the object belonging to the fuzzy set; otherwise, the degree of the object belonging to the fuzzy set becomes lower. According to the data in Tables 1 and 2, the range of the ore hardness is set to [0–7], the range of the ore weight is set to [0–25], and the range of the output motor frequency is set to [0–51]. After the range setting is completed, the triangular membership function is selected as the built-in function of the fuzzy controller.

Fuzzy sets are created after the fuzzification is completed, which is the extension of traditional set theory and includes all the objects described by the fuzzy concepts. The specific contents of the fuzzy sets are summarized in Table 4.

Table 4. Fuzzy set of the feeding system

hardness (small) = L	hardness (medium) = M	hardness (large) = H
weight (small) = L	weight (medium) = M	weight (large) = H
frequency (small) = L	frequency (medium) = M	frequency (large) = H

According to the fuzzy sets listed in Table 4, the input variables "hardness" and

"weight" and the output variable "frequency" are defined in the membership ranges of "L", "M", and "H", respectively. The centroid method is selected for defuzzification after the fuzzy membership function is defined. Finally, the membership functions of the two input variables are visualized, as shown in Figs. 5 and 6.

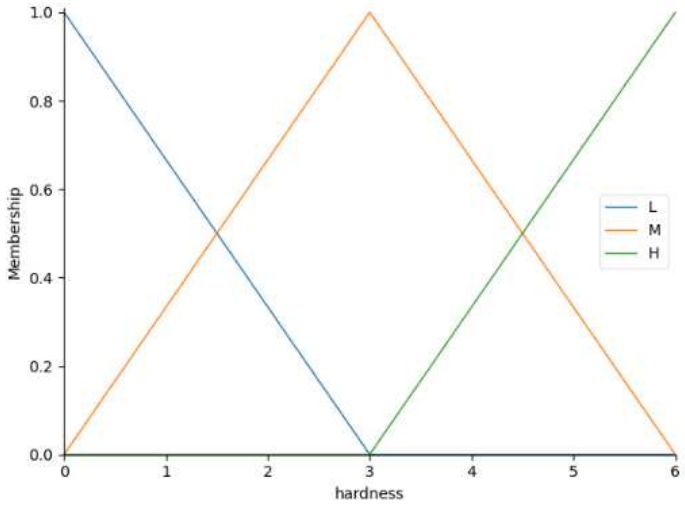

Fig. 5 Membership function of ore hardness

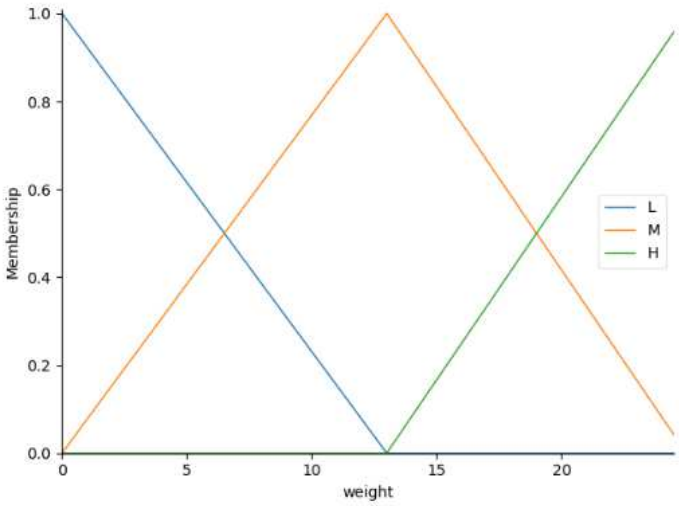

Fig. 6 Membership function of ore weight

2.2.2 Development of fuzzy rules

Fuzzy rules are established to describe the relationship among the three variables of hardness, weight, and frequency. The two most common fuzzy rule models are the IF THEN model and the Takagi–Sugeno–Kang (TSK) model. The TSK model is more

suitable as there are two input variables. The main contents of the TKS model are as follows:

$$R^i: IF\ \ x_1\ \ is\ \ F_1^i\ \ and\ \ \cdots\cdots and\ \ x_r\ \ is\ \ F_r^i\ \ THEN\ \ y^i\ \ = \alpha_0^i + \alpha_1^i x_1 + \cdots + \alpha_r^i x_r$$

where $F_j^i\ \ (j = 1,2,\cdots,r)$ is a fuzzy set, $\alpha_j^i\ \ (i = 1,2,\cdots,r, j = 1,2,\cdots,r)$ is a real-valued parameter, and y^i is the system output under the i-th rule.

The IF part of the TSK model is fuzzy, while the THEN part is clear, and the output is a linear combination of all the input variables. The specific contents of the fuzzy rules for the mining system are summarized in Table 5

Table 5. Fuzzy rules of feeding system

Weight, motor frequency, ore hardness	L	M	H
L	H	H	M
M	H	M	L
H	M	L	L

2.2.3 Simulation of production conditions

An ore from a gold concentrator in Yunnan was selected. The ore hardness was 5 (H) and the belt scale display weight was 13 (M). These parameters were input into the fuzzy controller to simulate the production conditions, and the output result was 31.33, i.e., the motor frequency was 31.33 when the ore hardness was 3 and the display weight of the belt scale was 18. The result is shown in Fig. 4 after visualizing the membership function of the output frequency.

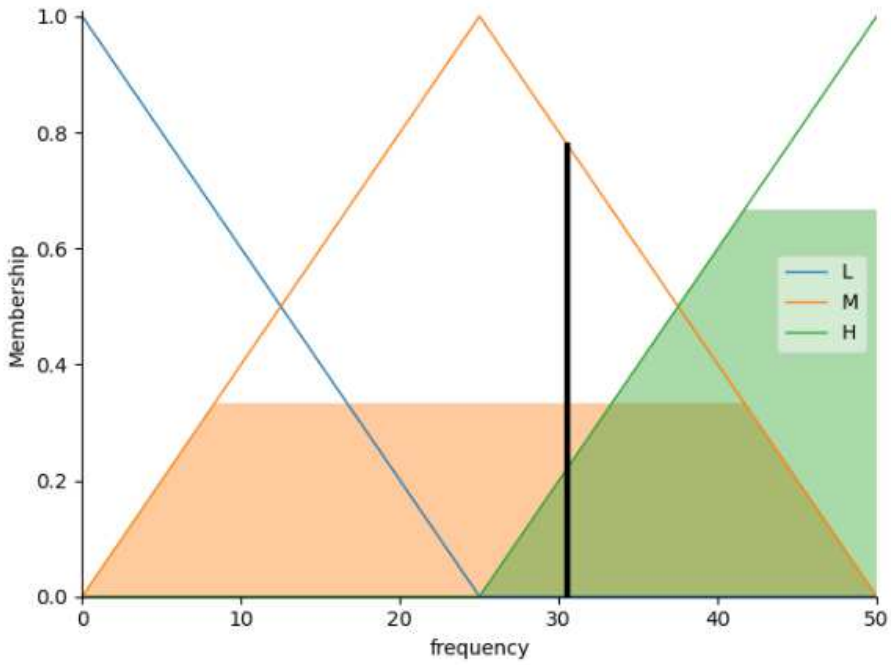

Fig. 7 Membership function of motor frequency

The core code of the fuzzy control system developed using SciKit Fuzzy is as follows:

Code	Note
import numpy *as* np *import* skfuzzy *as* fuzz *from* skfuzzy *import* control *as* ctrl *import* math	Call module
x_hardness = np.arange (0,7,1) x_weight = np.arange (0,25,0.5) x_frequency = np.arange (0,51,1)	Input variable range define pre- and post-conditions
hardness = ctrl.Antecedent (x_hardness,'hardness') weight = ctrl.Antecedent (x_weight,'weight') frequency = ctrl.Consequent (x_frequency, 'frequency')	Define the membership function and value range of hardness, weight, and frequency
hardness['L'] = fuzz.trimf (x_hardness, [0, 0, 3]) hardness['M'] = fuzz.trimf (x_hardness, [0, 3, 6]) hardness['H'] = fuzz.trimf (x_hardness, [3, 6, 6])	
weight['L'] = fuzz.trimf (x_weight, [0, 0, 13]) weight['M'] = fuzz.trimf (x_weight, [0, 13, 24]) weight['H'] = fuzz.trimf (x_weight, [13, 24, 24])	

frequency['L'] = fuzz.trimf (x_frequency, [0, 0, 25]) frequency['M'] = fuzz.trimf (x_frequency, [0, 25, 50]) frequency['H'] = fuzz.trimf (x_frequency, [25, 50, 50]) frequency.defuzzify_method='centroid' rule1=ctrl.Rule (antecedent= ((hardness['L'] & weight['L']) | (hardness['L'] & weight['M']) | (hardness['M'] & weight['L'])),consequent=frequency['H'],label='High') rule2=ctrl.Rule (antecedent= ((hardness['M'] & weight['M']) | (hardness['L'] & weight['H']) | (hardness['H'] & weight['L'])),consequent=frequency['M'],label='Medium') rule3=ctrl.Rule (antecedent= ((hardness['M'] & weight['H']) | (hardness['H'] & weight['M']) | (hardness['H'] & weight['H'])),consequent=frequency['L'],label='Low') tipping_ctrl = ctrl.ControlSystem ([rule1, rule2, rule3]) tipping = ctrl.ControlSystemSimulation (tipping_ctrl) tipping.input['hardness'] = 3 tipping.input['weight'] = 18 tipping.compute () print (tipping.output['frequency'])	Solve the fuzzy problem using the centroid method Establish fuzzy rules Simulate production conditions

2.3 Results of 3D visualization

To facilitate observation and recording by the staff of the concentrator, the output of the motor frequency can be visualized in 3D as shown in Fig. 5 using Python, and the operating frequency of the motor can be indicated by different colors.

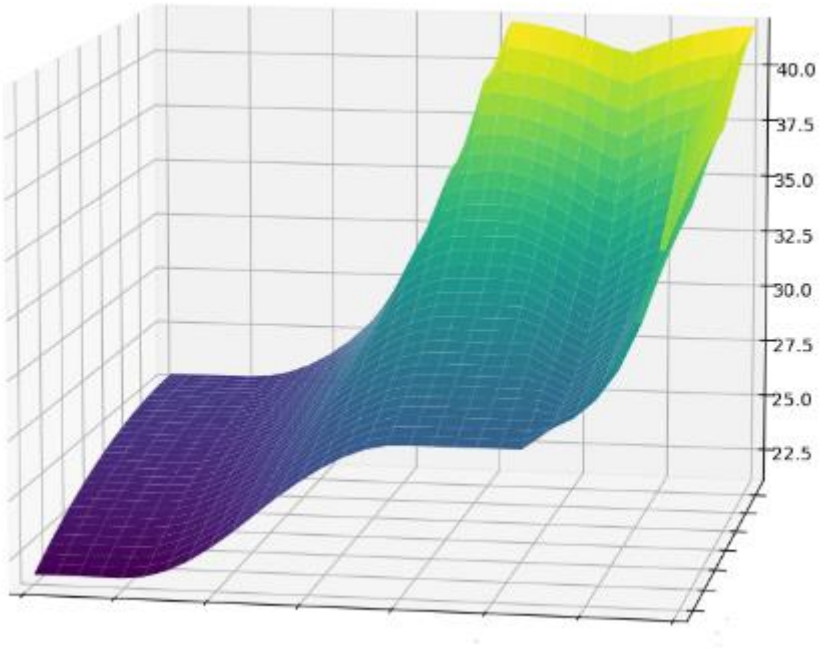

Fig. 8 3D visualized simulation results of constant feeding system

The core code for the 3D visualization of the variable motor frequency is as follows:

Code	Note
import matplotlib.pyplot *as* plt *from* mpl_toolkits.mplot3d *import* Axes3D	Call module
upsampled = np.linspace (0, 15, 30) x,y = np.meshgrid (upsampled, upsampled) z = np.zeros_like (x)	Produce grid matrix Input data

pp=[]
for i *in* range (0,30) :
  for j *in* range (0,30) :
  tipping.input['hardness'] = x[i, j]
  tipping.input['weight'] = y[i, j]
  tipping.compute ()
  z[i, j] = tipping.output['frequency']
  pp.append (z[i,j])
print ('max:',max (pp))
print ('min:',min (pp))
	Define the size of the canvas

fig = plt.figure (figsize= (8, 8)) ax = fig.add_subplot (111, projection='3d')	

surf = ax.plot_surface (x, y, z, rstride=1, cstride=1, cmap='viridis',linewidth=0.4, antialiased=*True*) ax.view_init (10, 100) plt.xlim (0,7) plt.ylim (0.25) plt.show ()	Set the viewing angle

3.Conclusions

In this study, the SciKit Fuzzy module library was employed to solve the problem of instability of multiple input and output systems in the beneficiation process. However, the use of Python in the beneficiation industry remains relatively scarce. In fact, Python has been applied to various industrial intelligence fields and achieved promising results. It is a powerful open-source language that is compatible with many industrial control systems, such as PLC control systems, and its use reduces the costs associated with research and development considerably. Moreover, it is easy to use and maintain; thus, most employees can get started immediately after training. The code of all Python module libraries is available on the github official website, and it can be used as reference and supplementary material in actual production. In addition, Python can achieve 3D visualization of running data, which is convenient for recording and observation by the staff of the concentrator. The intelligent fuzzy control constant feeding system proposed in this paper can select the appropriate feeding rate according to the different ore properties and automatically adjust the frequency of the feeding machine motor according to the weight of the ore conveyed, further improving the operating efficiency of the lower mill. This has a positive impact on the production and economic indicators of the factory. In summary, research on the application of Python can bridge the gap in the intelligentization of mineral processing. Module libraries such as SciKit Fuzzy facilitate the development of intelligent systems suitable for mineral processing and further improve the degree of intelligence in the mineral processing industry.

References

Wong, C. M., Vong, C. M., & Wong, P. K. (2016). Kernel-based multilayer extreme learning machines for representation learning. IEEE Transactions on Neural Networks and Learning Systems, 29(3), 757-762.https://doi.org/10.1016/j.neucom.2018.05.032

Yilmaz, S., & Oysal, Y. (2010b). Fuzzy wavelet neural network models for prediction and identification of dynamical systems. IEEE Transactions on Neural Networks, 21(10), 1599 – 1609.https://doi.org/10.1109/TNN.2010.2066285

Ganjefar, S., Tofighi, M., & Karami, H. (2015). Fuzzy wavelet plus a quantum neural network as a design base for power system stability enhancement. Neural Networks, 71.https://doi.org/10.1016/j.neunet.2015.07.010

Yuan, X., Qi, S., Wang, Y., et al. (2020). A dynamic CNN for nonlinear dynamic feature learning in soft sensor modeling of industrial process data. Control Engineering Practice, 104, Article 104614. https://doi.org/10.1016/j.conengprac.2020.104614

Park, J., & Sandberg, I. W. (1993). Approximation and radial-basis-function networks.Neural Computation, 5(2), 305 – 316. https://doi.org/10.1162/neco.1993.5.2.305

Fleming, P.J, and R.C Purshouse. (2002). Evolutionary algorithms in control systems engineering: a survey. Control Engineering Practice, 10(11), 1223–1241. https://doi.org/10.1016/S0967-0661(02)00081-3

Lei, Y., Karimi, H. R., Cen, L., Chen, X., & Xie, Y. (2021). Processes soft modeling based on stacked autoencoders and wavelet extreme learning machine for aluminum plant-wide application. Control Engineering Practice, 108, 104706.

https://doi.org/10.1016/j.conengprac.2020.104706

Jrvisalo, M., Ahonen, T., & Ahola, J. (2016). Soft-sensor-based flow rate and specific energy estimation of industrial variable-speed-driven twin rotary screw compressor. IEEE Transactions on Industrial Electronics, 63(5), https://doi.org/3282–3289.10.1109/TIE.2016.2527621

Huang, W., Oh, S. K., & Pedrycz, W. (2017). Hybrid fuzzy wavelet neural networks architecture based on polynomial neural networks and fuzzy set/relation inferencebased wavelet neurons. IEEE Transactions on Neural Networks and Learning Systems, 29(8), 3452–3462.10.1109/tnnls.https://doi.org/2017.2729589

Roman, R.-C., Precup, R.-E., & Petriu, E. M. (2020). Hybrid data-driven fuzzy active disturbance rejection control for tower crane systems. European Journal of Control. https://doi.org/10.1016/j.ejcon.2020.08.001

Tanaka, K., & Sugeno, M. (1992). Stability analysis and design of fuzzy control systems. Fuzzy Sets and Systems.45(2), 135–156. https://doi.org/10.1016/0165-0114(92)90113-I

Chen, Y., Liu, Z., Chen, C. L. P., & Zhang, Y. (2021). Adaptive fuzzy control of switched nonlinear systems with uncertain dead-zone: A mode-dependent fuzzy dead-zone model. Neurocomputing, 432, 133–144. https://doi.org/10.1016/j.neucom.2020.12.044

Feng, G. (2006). A Survey on Analysis and Design of Model-Based Fuzzy Control Systems. IEEE Transactions on Fuzzy Systems, 14(5), 676–697.

Sala, A., & Ariño, C. (2007). Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: Applications of Polya's theorem. Fuzzy Sets and Systems, 158(24), 2671–2686.

https://doi.org/10.1016/j.fss.2007.06.016

Amutha, J., Sharma, S., & Sharma, S. K. (2021). Strategies based on various aspects of clustering in wireless sensor networks using classical, optimization and machine learning techniques: Review, taxonomy, research findings, challenges and future directions.

Wang, Y., Song, Y., Hill, D. J., & Krstic, M. (2018). Prescribed-time consensus and containment control of networked multiagent systems. IEEE Transactions on Cybernetics, 49(4), 1138 – 1147.https://doi.org/10.1109/TCYB.2017.2788874

Hong, H., Yu, W., Wen, G., & Yu, X. (2017). Distributed robust fixed-time consensus for nonlinear and disturbed multiagent systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(7), 1464-1473. https://doi.org/10.1109/TSMC.2016.2623634

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-end memory networks. In Advances in Neural Information Processing Systems (pp. 2440 – 2448).

Zou, Q., Ni, L., Zhang, T., & Wang, Q. (2015). Deep Learning Based Feature Selection for Remote Sensing Scene Classification. IEEE Geoscience and Remote Sensing Letters, 12(11), 2321 – 2325.https://doi.org/10.1109/LGRS.2015.2475299

Angelov, P. (1995). A growing neural gas network learns topologies. Advances in Neural Information Processing System,7,845 – 865.http://dx.doi.org/

Kim, S. W., Kim, E. T., & Park, M. (1996). A new adaptive controller using parallel structure of fuzzy controller and its application. Fuzzy Sets and Systems, 81, 205 – 226.https://doi.org/10.1016/0165-0114(95)00206-5

Tzafestas, S. G. . (1996). Neural fuzzy control systems with structure and parameter

learning. Journal of Intelligent & Robotic Systems, 17(4). https://doi.org/1429-430.10.1007/BF00571702

Phan, P. A., & Gale, T. J. (2008). Direct adaptive fuzzy control with a self-structuring algorithm. Fuzzy Sets and Systems, 159, 871 – 899. https://doi.org/10.1016/j.fss.2007.09.012

Fritzke, B. (2004). An approach for fuzzy rule-base adaptation using on-line clustering. Integration of Methods and Hybrid Systems, 35(3), 275 – 298. https://doi.org/10.1016/j.ijar.2003.08.006

Chen, L., Bastin, G., & Van Breusegam, V. (1995). A case study of adaptive nonlinear regulation of fed-batch reactors. Automatica, 31, 55 – 65. https://doi.org/10.1016/0005-1098(94)00068-T

Chan, P. T., Rad, A. B., & Wang, J. (2001). Indirect adaptive fuzzy sliding mode control: Part II: Parameter projection and supervisory control. Fuzzy Sets and Systems, 122, 31 – 43.https://doi.org/10.1016/S0165-0114(99)00180-3

Li, Y., Shen, J., Lee, K. Y., & Liu, X. (2012). Offset-free fuzzy model predictive control of a boiler – turbine system based on genetic algorithm.Simulation Modelling Practice and Theory, 26, 77 – 95. https://doi.org/10.1016/j.simpat.2012.04.002

Wang, Y., Sun, Z. Q. & Sun, F. C. (2004). Stability analysis and control of discrete-time fuzzy systems: A fuzzy Lyapunov function approach. In Proceedings of 2004 5th Asian control conference.

Zhang, S., Taft, C. W., Bentsman, J., Hussey, A., & Petrus, B. (2012). Simultaneous gains tuning in boiler/turbine PID-based controller clusters using iterative feedback

tuning methodology. ISA Transactions, 51, 609 – 621. https://doi.org/10.1016/j.isatra.2012.04.003

Feng, G. (2006). A survey on analysis and design of model-based fuzzy control systems. IEEE Transactions on Fuzzy Systems, 14(5), 676 – https://doi.org/697.10.1109/TFUZZ.2006.883415

Alhoniemi, E., Hollmén, J., Olli, S., & Vesanto, J. (1999). Process monitoring and modeling using the self-organizing map. Integrated Computer-Aided Engineering, 6, 3 – 14.

Shang, C., Yang, F., Huang, D. X., & Lyu, W. (2014). Data-driven soft sensor development based on deep learning technique. Journal of Process Control, 24, 223 – 233.https://doi.org/10.1016/j.jprocont.2014.01.012

Soria-Olivas, E., Góez-Sanchis, J., Martín, J. D., Vila-Francés, J., et al. (2011). BELM:Bayesian extreme learning machine. IEEE Transactions on Neural Networks Learing Systems, 22(3), 505 – 509.https://doi.org/10.1109/TNN.2010.2103956

Zhang, M., Liu, X., & Zhang, Z. (2016). A soft sensor for industrial melt index prediction based on evolutionary extreme learning machine. Chinese Journal of Chemical Engineering, 24, 1013 – 1019.https://doi.org/10.1016/j.cjche.2016.05.030

Geng, Z., Dong, J., Chen, J., & Han, Y. (2017). A new self-organizing extreme learning machine soft sensor model and its applications in complicated chemical processes. Engineering Application of Artificial Intelligence, 62, 38 – 50.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521, 436 – 444.

Zhong, R. Y., Xu, X., Eberhard, K., & Newman, S. T. (2017). Intelligent

manufacturing in the context of industry 4.0: a review. Engineering, 3(5), 616 – https://doi.org/630.10.1016/J.ENG.2017.05.015

Maddalena, E. T., Lian, Y. Z., & Jones, C. N. (2020). Data-driven methods for building control — A review and promising future directions. Control Engineering Practice, 95, Article 104211.https://doi.org/10.1016/j.conengprac.2019.104211

Jahedsaravani, A., Marhaban, M. H., Massinaei, M., Saripan, M. I., & Noor, S. B. M. (2015). Froth-based modeling and control of a batch flotation process.International Journal of Mineral Processing, 146, 90 – 96. https://doi.org/10.1016/j.minpro.2015.12.002

Figures

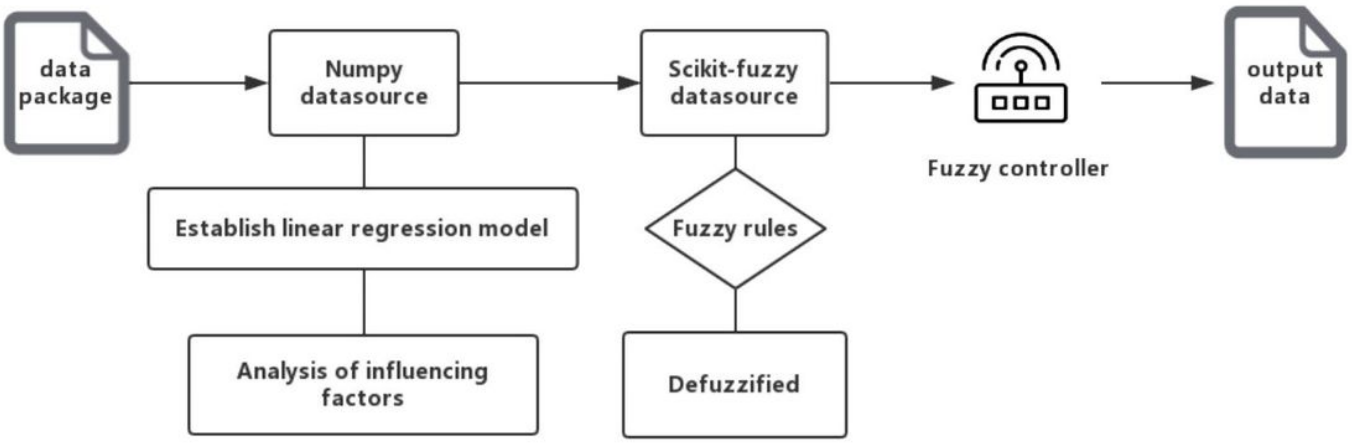

Figure 1

Development of fuzzy control system for constant feeding by Python

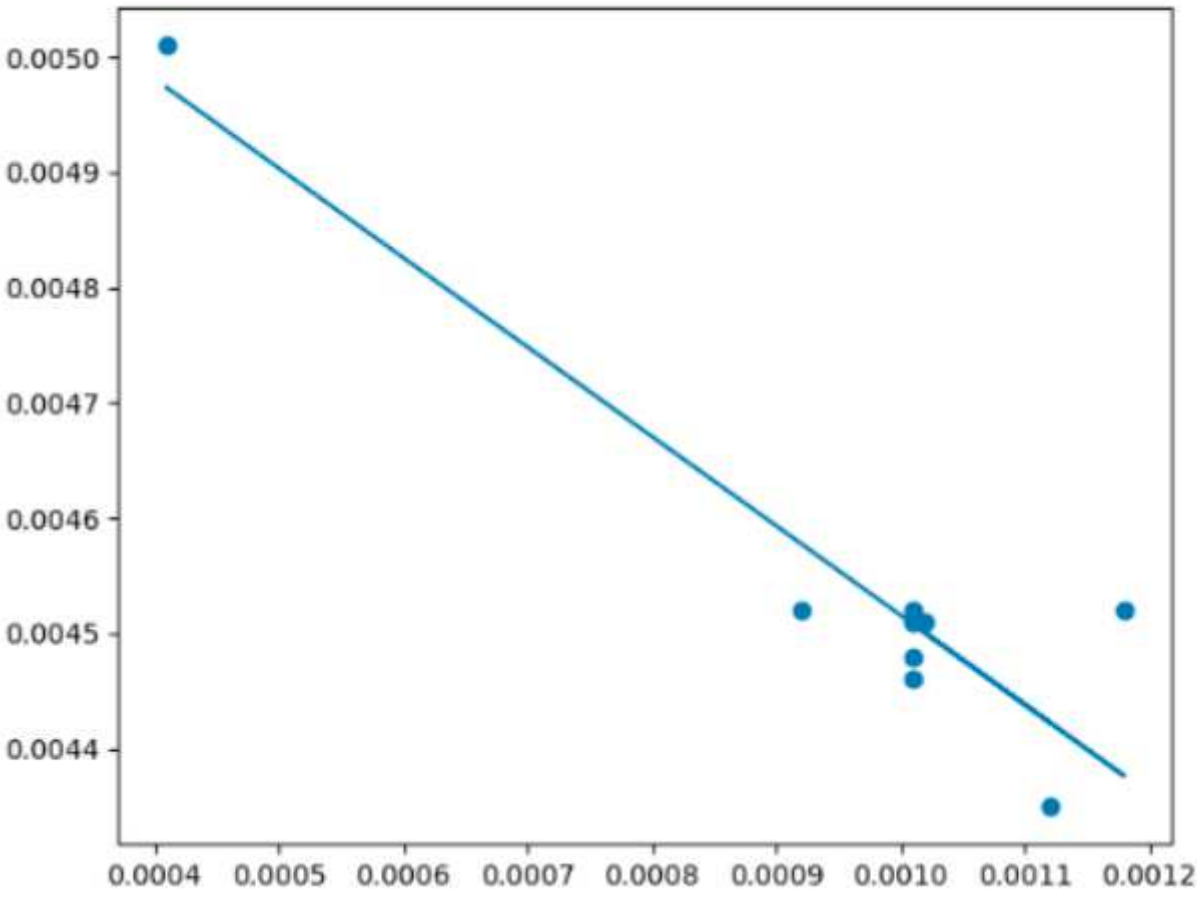

Figure 2

Linear regression model of belt scale weight and motor frequency after 10000 simulation cycles using SciKit Learn

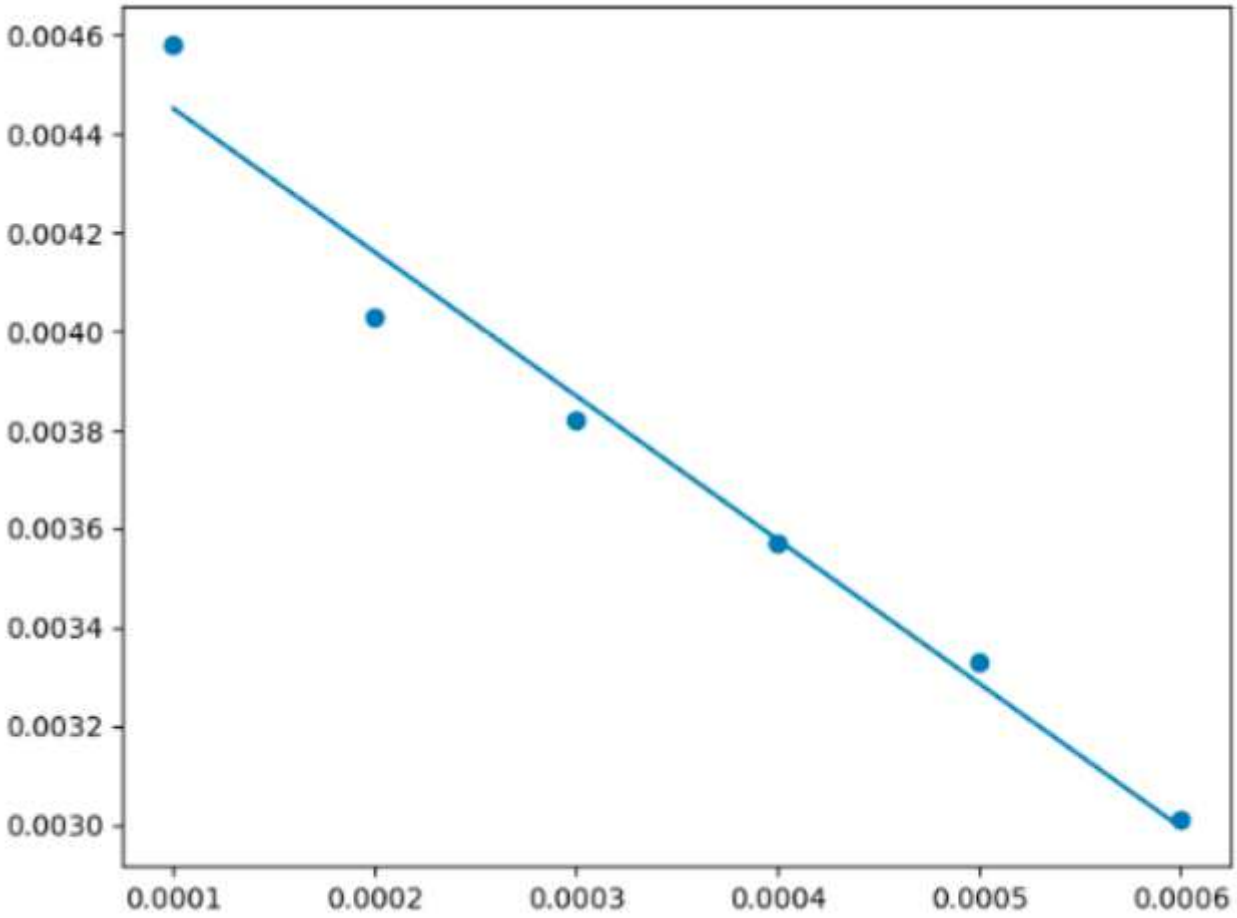

Figure 3

Linear regression model of ore hardness and motor frequency after 10000 simulation cycles using SciKit Learn

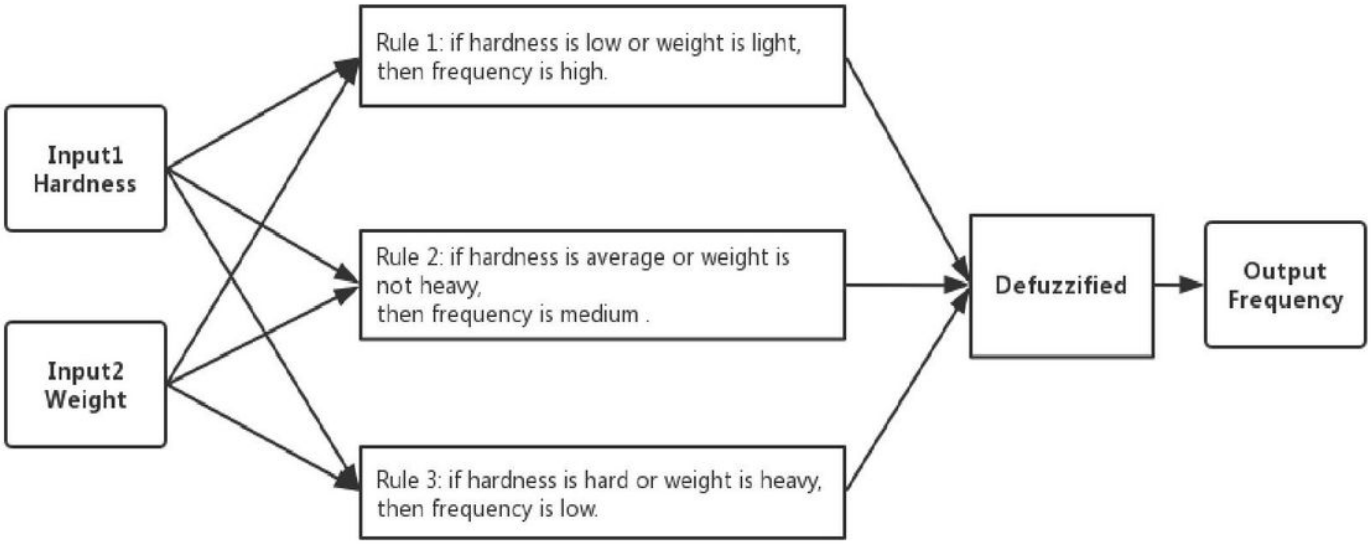

Figure 4

Fuzzy reasoning process

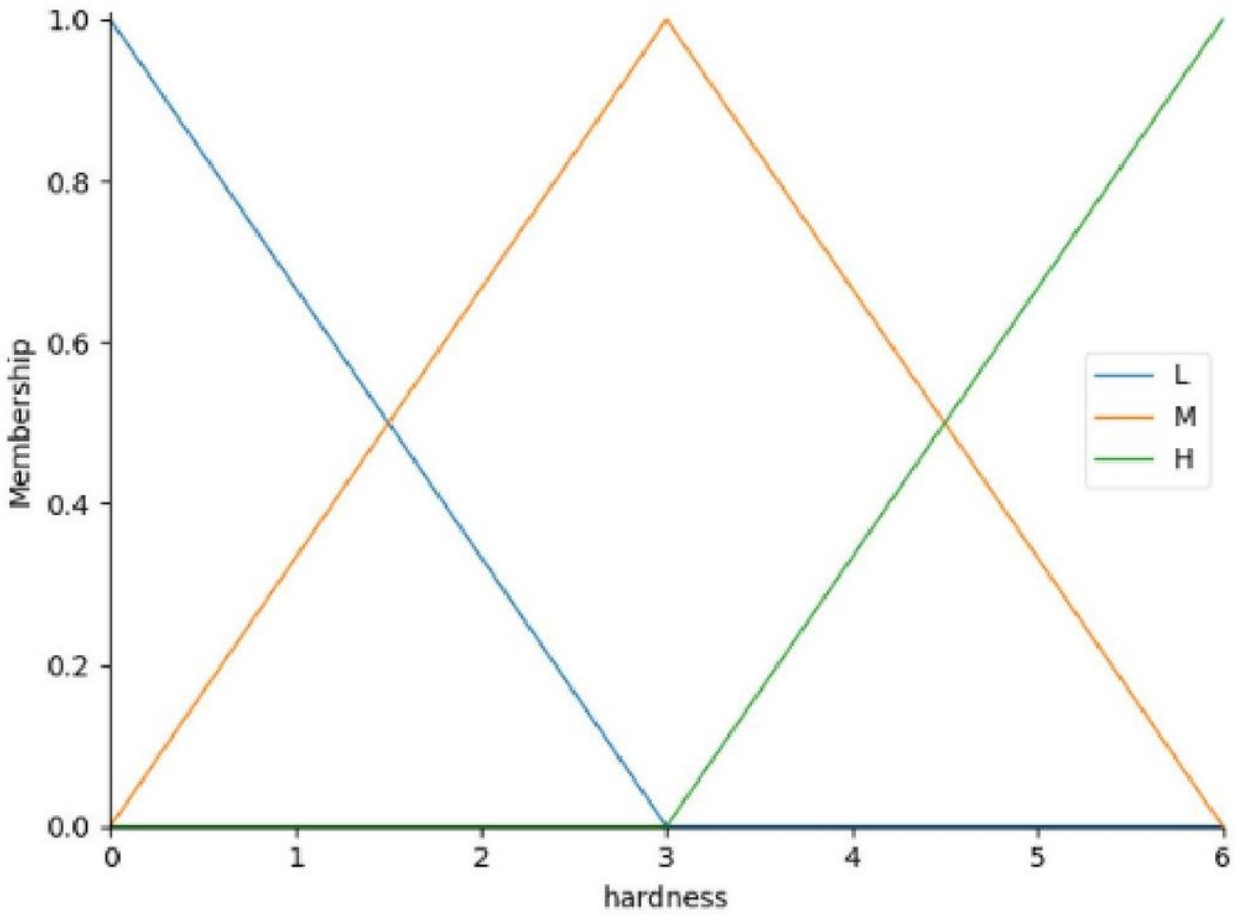

Figure 5

Membership function of ore hardness

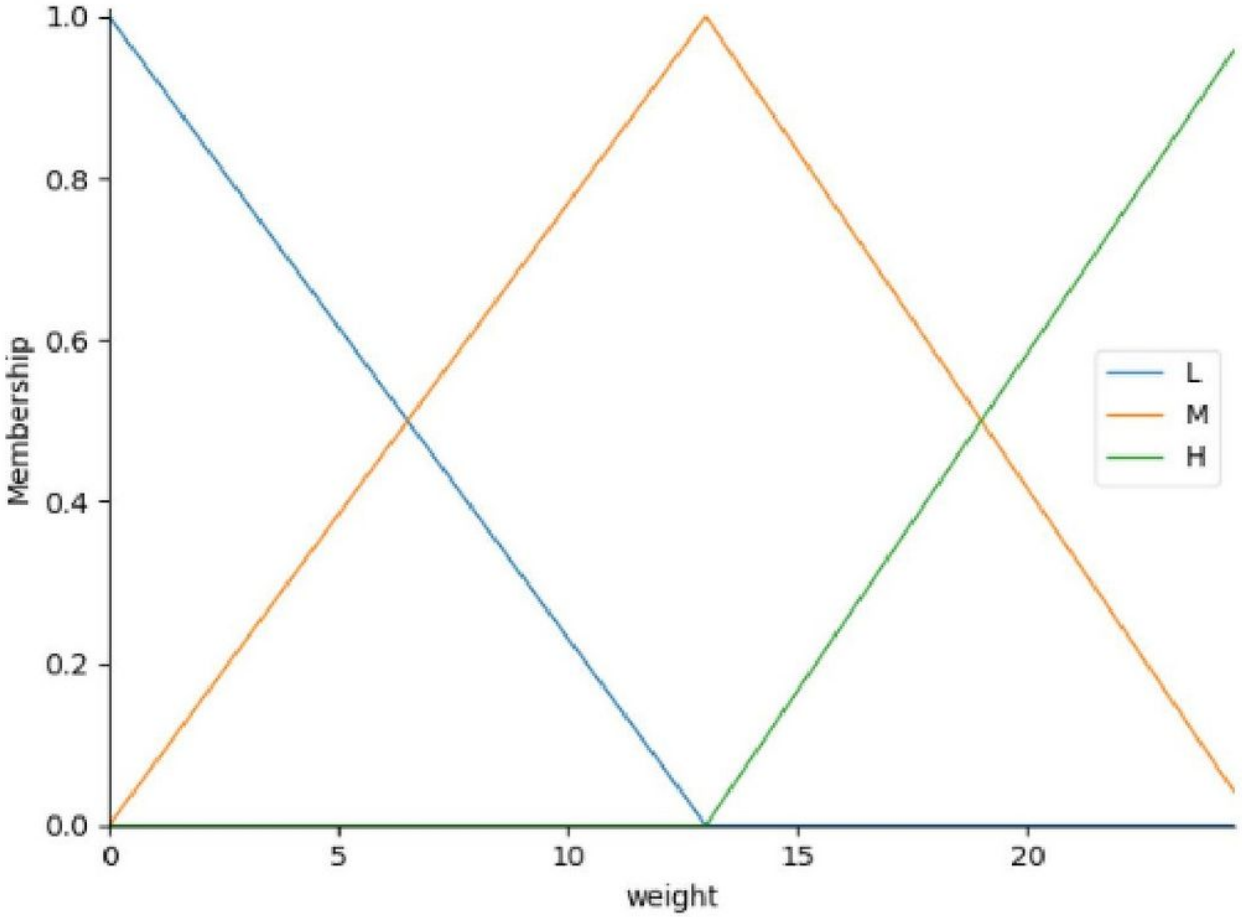

Figure 6

Membership function of ore weight

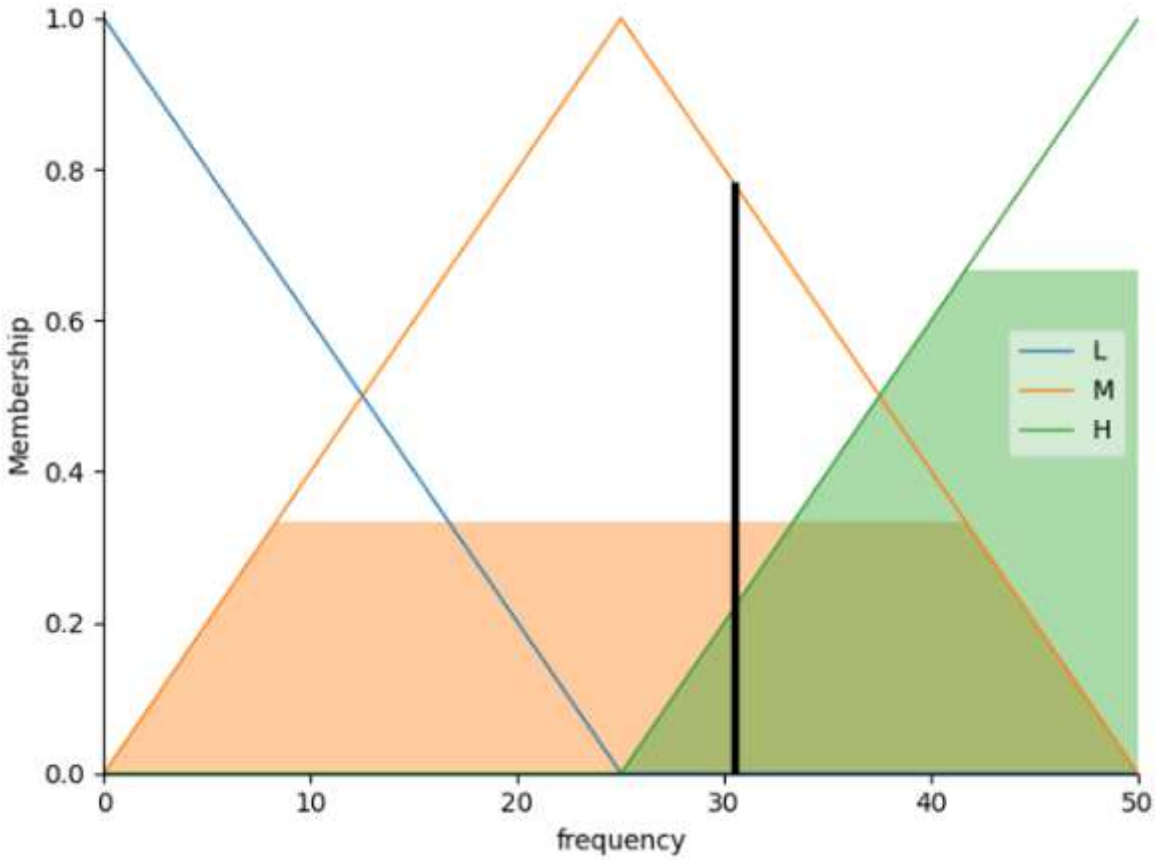

Figure 7

Membership function of motor frequency

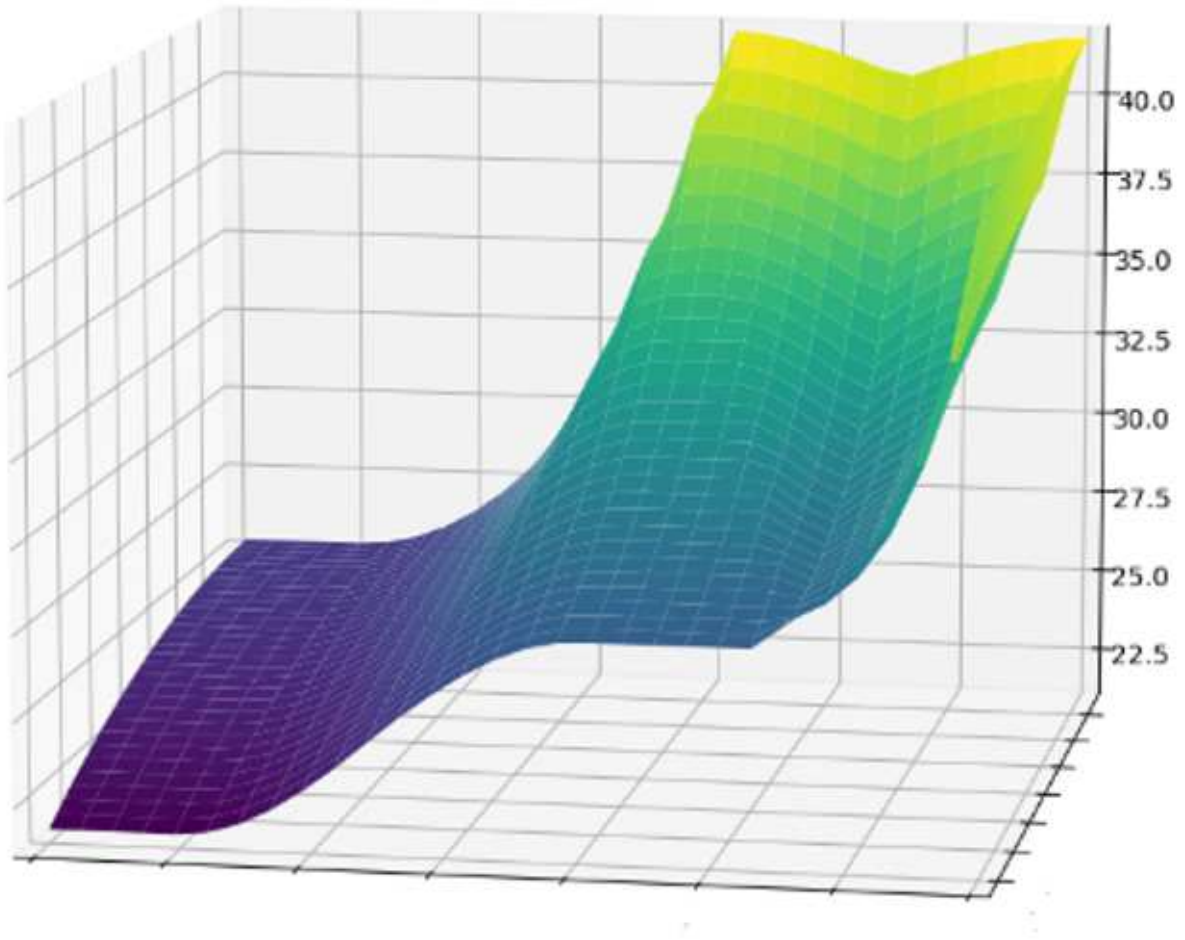

Figure 8

3D visualized simulation results of constant feeding system