


Article

Mill Feed Control System and Algorithm Based on Python

Wenkang Zhang ^{1,2,3}, Dan Liu ^{1,2,3,*}, Yu Du ^{1,2,3}, Ruitao Liu ^{1,2,3}, Daqian Wang ^{1,2,3} , Longzhou Yu ⁴
and Shuming Wen ^{1,2,3}

¹ State Key Laboratory of Clean Utilization of Complex Nonferrous Metal Resources, Kunming 650093, China; zwk0802@163.com (W.Z.); aa609028251@163.com (Y.D.); lrtao_1997@163.com (R.L.); daqianwang1997@163.com (D.W.); shumwen@126.com (S.W.)

² Yunnan Key Laboratory of Green Separation and Enrichment of Strategic Mineral Resources, Kunming 650093, China

³ Faculty of Land and Resources Engineering, Kunming University of Science and Technology, Kunming 650093, China

⁴ Yunnan Amade Electrical Engineering Company, Kunming 650033, China; godspeedtim@foxmail.com

* Correspondence: ldysyz1983@126.com

Abstract: Grinding is an important link in the process of mineral processing. It plays a vital role in mineral processing by optimizing the grinding process, improving the quality of grinding products and ensuring the follow-up operation indicators. In this paper, the Python language, intelligent theoretical control technology and mineral processing were combined to solve the problem of ore feeding control in mineral processing. Using error factor analysis, an extended control algorithm was designed. The NumPy library and data collected from the Yuan YangMou concentrator in China were used to quantitatively analyze the factors affecting the error of electronic belt scales. This paper introduces the use of Kalman filtering for electronic belt scale weight data to reduce the effect of noise and hence reduce errors. The factors affecting the process of mill feeding are also analyzed. The core ideas and methods of fuzzy control theory are summarized, and a Python-based fuzzy controller suitable for the mill feeding process that improves the overall robustness and accuracy of feeding system is implemented.

Keywords: intelligence control; grinding; Python; fuzzy control; simulation optimization



Citation: Zhang, W.; Liu, D.; Du, Y.; Liu, R.; Wang, D.; Yu, L.; Wen, S. Mill Feed Control System and Algorithm Based on Python. *Minerals* **2022**, *12*, 804. <https://doi.org/10.3390/min12070804>

Academic Editor: Saija Luukkanen

Received: 23 May 2022

Accepted: 20 June 2022

Published: 24 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mineral resources are extremely important for the development of human society. They are essential components of industry, security and economic guarantees for a country, and the embodiment of comprehensive national strength. There are few single ore deposits in China, and most of them are associated, and hence, ore tends to be lean and difficult to dress. Some enterprises are small in scale, and the use of advanced technology is not frequent, which leads to low labor productivity and the overall low utilization of mineral resources in China [1].

In mineral processing, the grinding quality (good or poor) directly affects the subsequent dressing production operation index (high or low) [2]. The grinding process, which increases the concentration of energy and material consumption, accounts for the largest portion of equipment investment in the beneficiation process [3]; therefore, optimizing the grinding process to improve the mill plant performance is extremely important [4]. It is also an important topic in mineral processing research. The feeding link of the mill is an important part of the grinding process [5], and its accuracy directly affects the production indicators of the grinding products and subsequent operations [6]. However, the grinding process itself is a time-varying multi-input/multi-output nonlinear control system [7], and there are many influencing factors, such as the field environment, nature of the ore, efficiency of the personnel, and running state of the mill [8].

In 1965, Zadeh proposed fuzzy set theory, which uses membership function to describe the differences in the transition process of controlled objects, breaking with the traditional concept of treating a non-zero value as a binary one in set theory [9].

Since the birth of fuzzy control, it has been used to obtain productive results in PID (proportion integration differentiation) control, artificial neural networks [10], expert systems, and many other fields, and has greatly promoted the research and development of intelligent control algorithms. Fuzzy control is divided into the following categories: fuzzy adaptive PID control, fuzzy neural networks, fuzzy expert control, and fuzzy adaptive predictive control [11].

Fuzzy control theory has been used for more than 50 years and has been widely used in the field of industrial control, achieving good results in different industries. In the field of mining, Mohammad evaluated the potential of fuzzy logic as an alternative method to the traditional statistical regression techniques for predicting the wear rate of high chromium alloy during phosphate grinding. The comparison of a fuzzy model and a regression model showed the superiority of the developed fuzzy model [12]. Ajaya Kumar Pani performed the modeling of a vertical roller mill used for clinker grinding using support vector regression (SVR), fuzzy inference and adaptive neuro fuzzy inference (ANFIS) techniques [13]. Khodadadi et al. [14] studied the disturbance observer of a ball mill using fuzzy PID control. The observer can observe a ball mill using fuzzy PID control. The observer can realize the control strategy of reducing or even eliminating interference in the grinding process in real time, which improves the performance of the ball mill. In addition, Zhao et al. [15] studied industrial feed electronic belt scales and addressed industrial weighing belt feeder saturation due to motor noise, friction, sensors, and highly nonlinear factors by proposing a fuzzy control algorithm for different reference input that has good robustness and validity. They compared self-tuning and fuzzy controllers and described the advantages and disadvantages of both approaches.

There are several problems in the mill feed control process: the fluctuation of mineral properties, the accuracy of electronic belt scales, the complexity of site conditions, and issues with personnel management. To address these problems, this article combines grinding control research with computer technology. First, the feeding process of the grinding mill is analyzed by adopting existing methods and mathematical analysis, the working principle and structure of each part are studied considering the practical problems in the process of mill feeding, and the factors affecting these problems are analyzed and addressed.

This paper describes the fuzzy control theory applicable to a fuzzy mill feed controller based on Python. According to an analysis of the factors, a fuzzy controller was designed and implemented for automatic mill feed regulation to optimize the feeding process. The fuzzy controller was evaluated in a simulation experiment. The proposed fuzzy controller addresses the problems of slow response and large overshoot of the traditional PID control system and makes the whole grinding and feeding system more intelligent.

2. Error Factors in the Mill Feeding Process

2.1. Main Sources of Error in Electronic Belt Scales

The operation of an electronic belt scale is a dynamic process, and the result of weighing includes randomness, so the process is regarded as a random system. The main factors affecting the accuracy are gravity, velocity, signal processing, calibration, and environmental errors [16,17].

Calibration and environmental errors belong to installation quality problems, which can be solved. Signal processing, gravity belt tension, and velocity errors also exist. Belt creep and belt running deviations have substantial influence on the accuracy of electronic belt scale measurements but can be corrected, and hence the error in electronic belt scale measurements caused by belt tension and sliding were analyzed.

2.1.1. Belt Tension Analysis

The force analysis in Figure 1 focuses on the stress of a single weight on a transport belt. Here, theta is the angle between the belt and horizontal plane, F_1 is the downward mineral aggregate force, force F_2 is the transportation force, P is the natural tension under no-load conditions, and G is the gravitation force acting on the mineral aggregate. Therefore, the stress W of a single weight on a belt can be roughly expressed as

$$W = W_p + W_1 + G - W_2 = P \cdot \cos \theta + F_1 \cdot \sin \theta + G - F_2 \cdot \sin \theta \quad (1)$$

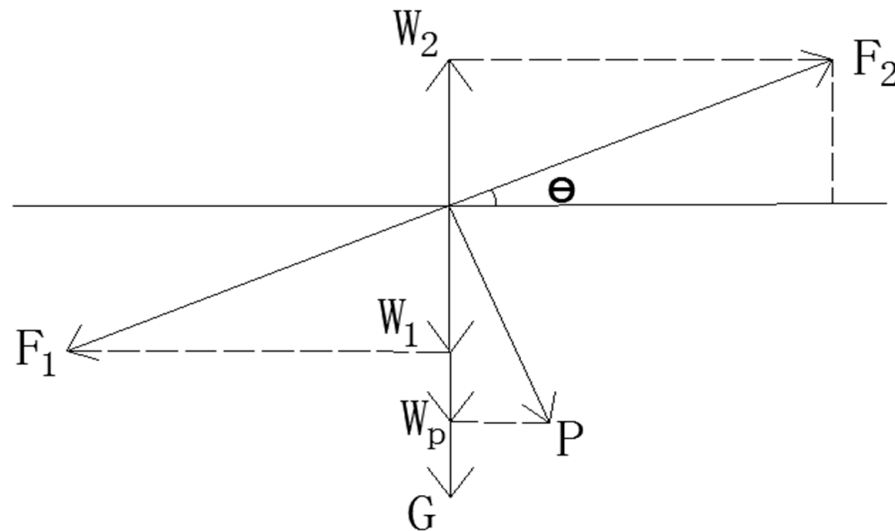


Figure 1. Stress analysis of single a weight on a transport belt.

The research shows that the stress of a transport belt is far more complex than that analyzed in Equation (1). a mathematical model of the weighing process of an electronic belt scale was established, yielding the following equation:

$$F = nqL \pm 2K \frac{d}{L} T \quad (2)$$

Here, T is the total tension of the belt, q is the load per unit length of belt, L is the idler spacing, d is the vertical direction of the belt shape variable, n is the number of weighing rollers, and K is the belt correction factor.

As Equation (2) reveals, all quantities except for the total belt tension T can be determined. Using the design standard of a conveyor belt determined by CEMA (Consumer Electronics Manufacturers Association) for the calculation of each component, operation resistance, and efficiency, the following simplified formula can be used to calculate the effective tension of a belt:

$$T \approx K_y L (W_m + W_b) + W_m H \quad (3)$$

Here, L is the length of conveyor, K_y is the comprehensive flexural resistance coefficient, W_b is the quality of the conveyor belt per unit length, W_m is the quality of the ore material per unit length, and H is the lifting height of the ore material.

The actual conditions of a concentrator in Yuanyang County, Yunnan, China, are as follows: the weight is 10 t/h, the belt speed is 0.3 m/s, the belt length is 50 m, the rise is about 18°, the belt mass is 26 kg/m, and K_y is 0.03. According to Equation (3), the effective tension T of the belt is 197.1 kg, which was calculated by substituting T into Equation (2). Every 0.1 mm vertical change in the shape variable of the belt causes an error of approximately 0.1% in the system. Therefore, the error ϵ_1 of the tension is equal to the value of shape variable d in the vertical direction. This error is large, and it needs to be considered in the subsequent calculation of the error.

2.1.2. Belt Skid Analysis

The stress analysis of a transport belt is illustrated in Figure 2.

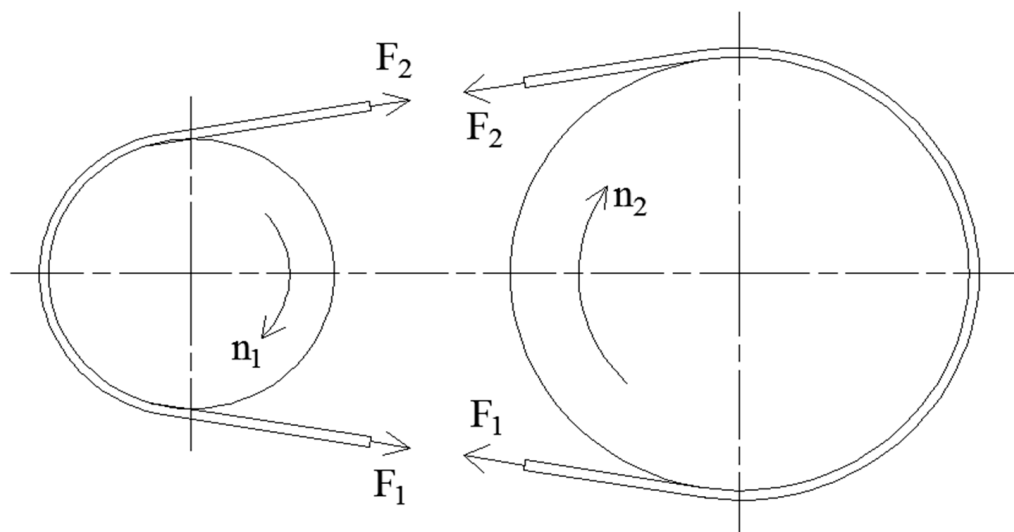


Figure 2. Stress diagram of a transport belt.

The initial tension of the belt is F_0 . The loose-edge tension F_1 and tight-edge tension F_2 are generated during operation. Assuming that the total length of the belt remains unchanged, the initial tension F_0 , loose-edge tension F_1 , tight-edge tension F_2 , and tension F_q caused by centrifugal force can be expressed as follows:

$$\begin{aligned}
 F_1 &= \frac{P \cdot e^{\mu\alpha}}{v(e^{\mu\alpha} - 1)} \\
 F_2 &= \frac{P}{v(e^{\mu\alpha} - 1)} \\
 F_q &= Q \cdot v^2 \\
 F_0 &= \frac{F_1 + F_2}{2} + F_q
 \end{aligned}
 \tag{4}$$

Here, P is the output power of the motor, μ is the friction coefficient of the belt and roller contact surface, v is the belt transport speed, and Q is the belt weight per unit length.

The belt should be tight before skidding is analyzed, so the initial tension of Equation (4) should be met during installation and calibration. After calibration, the displacement of the belt and driving roller is observed to determine if it is consistent. If the displacement of the two deviates, the transport belt slips during operation. The slip is calculated as follows:

$$\alpha = \frac{v}{r \cdot \omega}
 \tag{5}$$

Here, v is the belt transport speed, r is the radius of roller, and ω is the angular velocity of the roller during operation.

When $\alpha > 1$, the belt velocity is greater than the roller angular velocity; when $\alpha < 1$, the belt speed is less than the roller angular speed; and when $\alpha = 1$, the working condition is normal, and there is no skid. The belt skid error is expressed as $\varepsilon_2 = 1 - \alpha$.

2.2. Error Compensation of Electronic Belt Scales

Given the measurement error of electronic belt scales described in Section 2.1, it is necessary to introduce the Kalman filter algorithm to compensate for the error of the weight signal.

2.2.1. Derivation of the Model of An Electronic Belt Scale

It is assumed that the mass of the weighing platform of a weighing system is $w(t)$, the displacement of the weighing platform is $\theta(t)$, the damping coefficient is C , the spring

coefficient is K , and the mass to be weighed is m . The displacement $\theta(t)$ of entrained mass $w(t)$ changes according to the following linear relationship with the shape variable ε of the weighing sensor:

$$\theta(t) = -k_c \cdot \varepsilon(t) \quad (6)$$

Here, k_c is the elastic coefficient of the weighing cell.

The model of a scale system can be expressed as follows:

$$(w(t) + m)\ddot{\theta}(t) + c\dot{\theta}(t) + k\theta(t) = w(t)g \quad (7)$$

Dividing both sides of Equation (7) by $w(t) + m$ yields

$$\ddot{\theta}(t) + \frac{c\dot{\theta}(t)}{w(t) + m} + \frac{k\theta(t)}{w(t) + m} = \frac{w(t)g}{w(t) + m} \quad (8)$$

Define $\delta_1(t) = \theta(t)$ and $\delta_2(t) = \dot{\theta}(t)$ and substitute them into Equation (8) to obtain

$$\dot{\delta}_2(t) + \frac{c\delta_2(t)}{w(t) + m} + \frac{k\delta_1(t)}{w(t) + m} = \frac{w(t)g}{w(t) + m} \quad (9)$$

Equation (9) can be written in matrix form as follows:

$$\begin{bmatrix} \dot{\delta}_1(t) \\ \dot{\delta}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{w(t)+m} & -\frac{c}{w(t)+m} \end{bmatrix} \begin{bmatrix} \delta_1(t) \\ \delta_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{g}{w(t)+m} \end{bmatrix} w(t) \quad (10)$$

Define the following:

$$\begin{aligned} x(t) &= \begin{bmatrix} \delta_1(t) \\ \delta_2(t) \end{bmatrix} \\ \alpha(t) &= -\frac{k}{w(t) + m} \\ \beta(t) &= -\frac{c}{w(t) + m} \\ \gamma(t) &= \frac{g}{w(t) + m} \\ u(t) &= w(t) \\ y(t) &= \theta(t) \end{aligned}$$

Equation (10) can then be written as follows:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ \alpha(t) & \beta(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \gamma(t) \end{bmatrix} u(t) + z(t) \quad (11)$$

$$y(t) = [1 \quad 0]x(t) + v(t) \quad (12)$$

In Equation (11), $z(t)$ is the system noise, and in Equation (12), $v(t)$ is the observation noise. System noise and observation noise are not correlated and are independent of each other. The corresponding covariance value Q can be expressed as follows:

$$Q = \begin{bmatrix} Q_weight & 0 \\ 0 & Q_obs \end{bmatrix} \quad (13)$$

$$Q_weight = \text{cov}(weight, weight)$$

$$Q_obs = \text{cov}(Q_r, Q_r)$$

In Equation (13), Q_weight is the weight covariance of the sensing measurement, and Q_r is the belt slip covariance.

2.2.2. Kalman Filter Based on Python

Using a Kalman filter, a mathematical model of the weighing system was established. Assuming that the sampling value of the previous state in the operation of the electronic belt scale is the same as that of the current state, then the system parameters A and B and measurement parameter H are all 1. Because there is no control in the system, the control quantity $U(k)$ is 0. The Kalman filter was implemented according to the above parameter values. The NumPy and Pylab libraries in Python were used for the development [18,19].

The Kalman filter for this model has an initial state of $X(0|0)$ and covariance $P(0|0)$. However, a Kalman filter does not converge when the covariance is 0 in the initial conditions. Hence, the Python-based Kalman filter was assigned the initial values $X(0|0) = 4$, $P(0|0) = 0$. Test samples were input to test the feasibility of the Kalman filter, and the results are shown in Figures 3 and 4. Figure 3 presents the simulation model results of the Python-based Kalman filter, in which the crosses indicate the measured values, the blue line indicates the posterior estimate, and the green line indicates the actual values. Figure 4 shows the variation in the precision of the Kalman filter output obtained using Python.

As Figures 3 and 4 reveal, as the iterations increase, error between the observed and true values of the Kalman filter still exists. After stabilization, the error between the predicted and true values is no more than 1.5%, and the accuracy continues to increase. Hence, the simulation results are consistent with the expected results. The Python-based Kalman filter is effective in filtering the weight signal of an electronic belt scale and can hence help compensate for the error in the measurement results of an electronic belt scale.

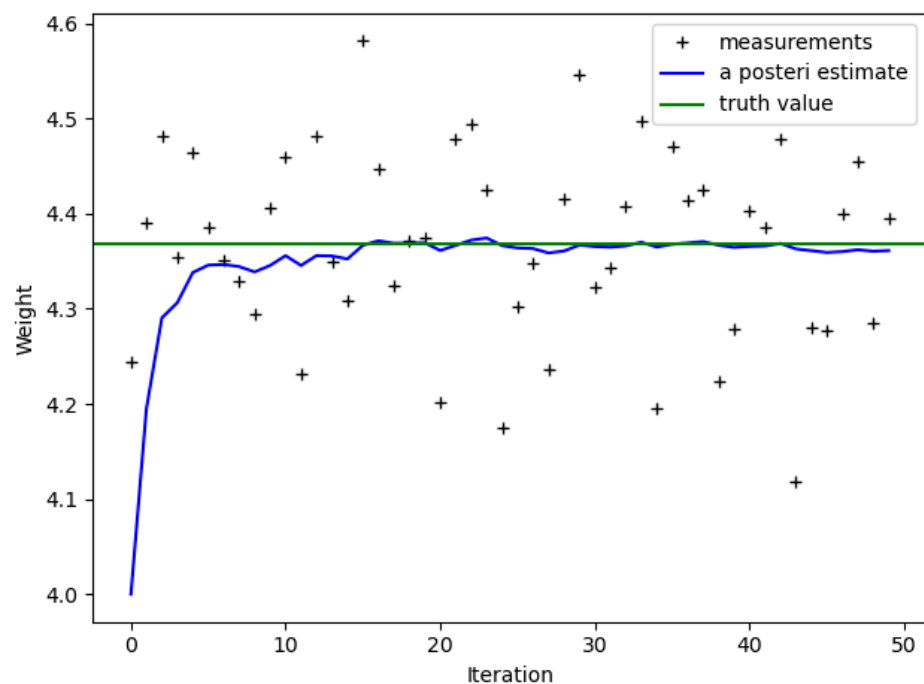


Figure 3. Simulation model of the Kalman filter.

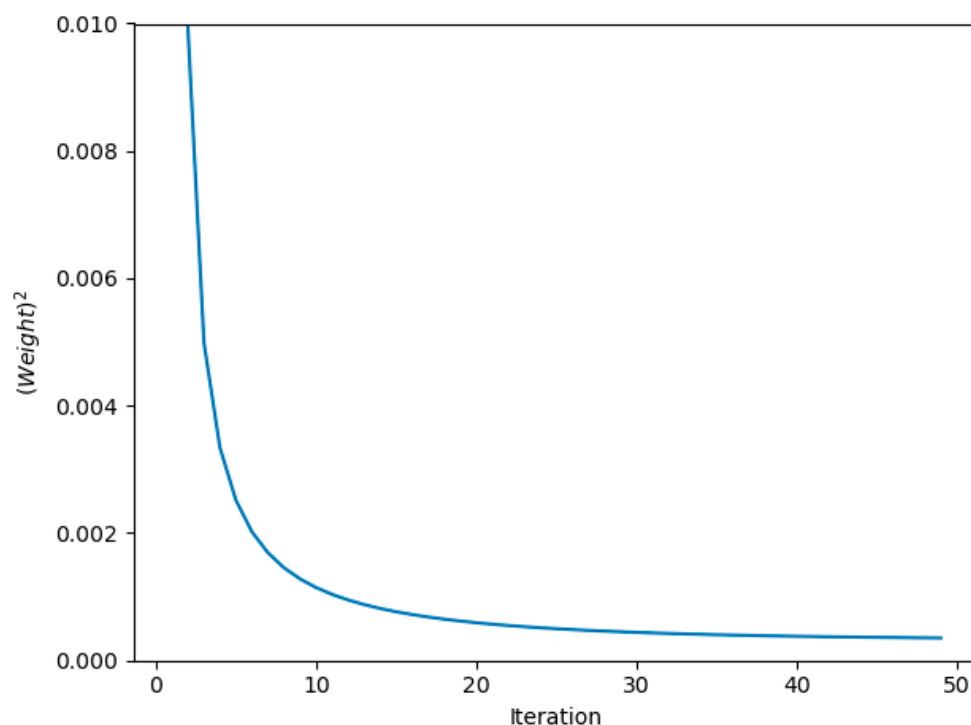


Figure 4. Variation in Kalman filter precision.

2.3. Factors Affecting the Mill Feeding Process

2.3.1. Data Preprocessing

The data used in this study were collected from the concentrator of a gold mine located in Yunnan, China. Because the data reporting, checking, and storage depends on the equipment running status and the level of staff, it is necessary to clean and filter the raw data to fill in missing data or outliers. Isolated points were estimated by interpolation to achieve the aims of preprocessing [20,21].

The ore humidity was determined using a drying method. Three 1 kg ore samples were taken, one from each ore heap, and dried in the oven at 100 °C. The percentage of lost weight is the ore humidity. The sampling period was 30 min. The motor frequency of the two existing feeders was recorded when the sample machine was operating, and a total of 20 groups of data were recorded for 5 h. Electronic belt scale weight data for a feeding machine were collected to adjust the sampling cycle, that is, to convert the actual value to the set value of a cycle. The sampling period was set to 10 min and a total of 25 groups of data were collected.

2.3.2. Linear Regression Models of the Factors Based on NumPy and Scikit-Learn

As the most widely used scientific computing tool in Python, NumPy can integrate algorithms into models and is an important component in building linear regression models. To eliminate the error caused by the randomness of the model's algorithm itself, scikit-learn was used to independently simulate, train, and evaluate the established model [20], providing guidance for its further optimization [22]. A unitary linear regression model was established for the data set, and the output of the models after 10,000 simulation experiments are shown in Figures 5–7.

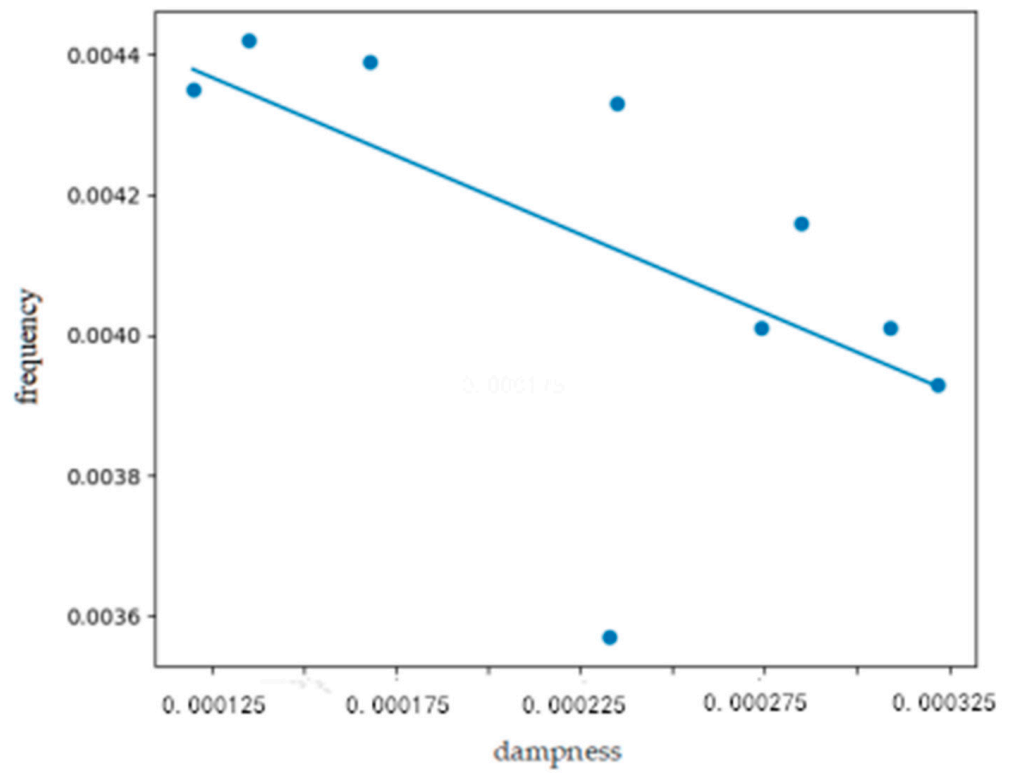


Figure 5. Linear regression model of ore humidity and motor frequency.

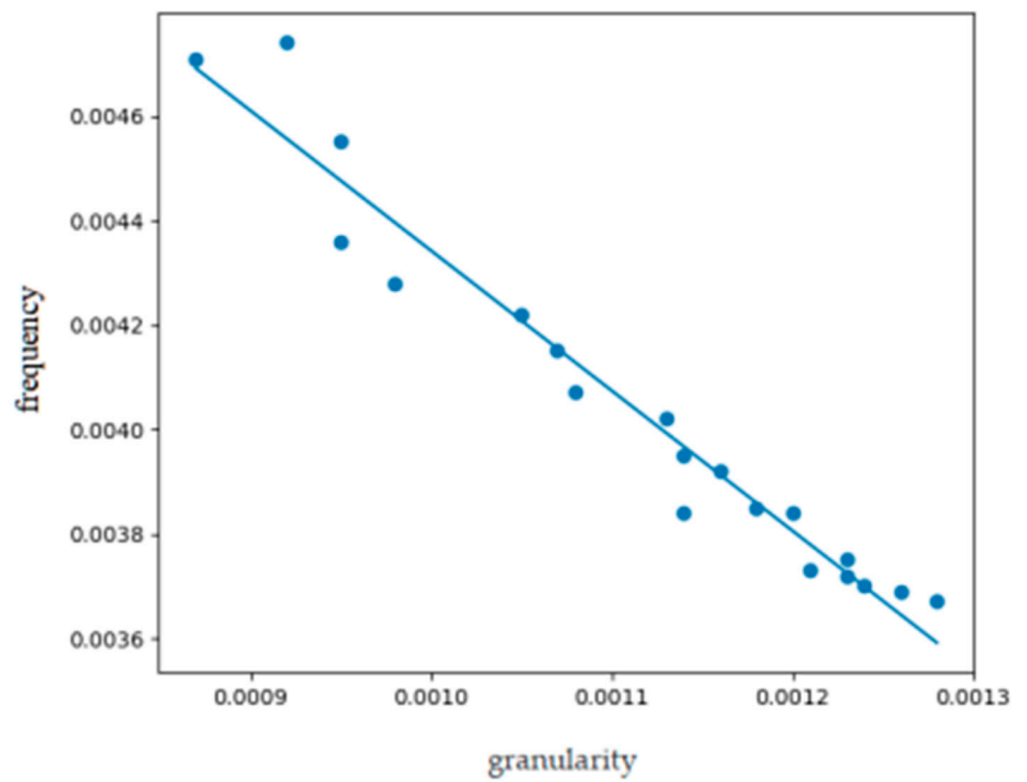


Figure 6. Linear regression model of ore granularity and motor frequency.

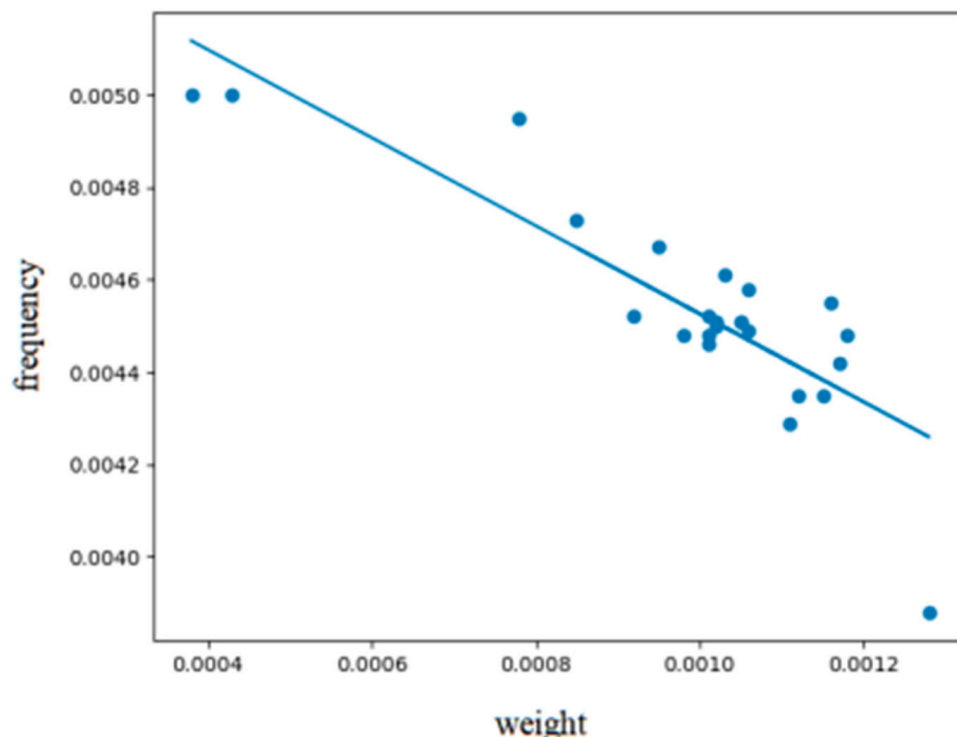


Figure 7. Linear regression model of electronic belt scale weight data and motor frequency.

The three regression model analyses show that the data and the model of the linear regression have a certain residual error. This is because the ore humidity, ore particle size, and the electronic belt scale weight data are the three main factors affecting motor frequency, but other factors exist, such as the voltage fluctuation, ore material adhesion, and feeder gate size. These other factors cause the deviation between the predicted and actual values. The degree of fit of the linear regression model should be determined before the linear correlation of the variables is studied. The degree of fit is determined by the coefficient of determination. The coefficients of determination and correlation coefficients of the three models are shown in Table 1.

Table 1. Determination and correlation coefficients of the regression model output.

Influence Factor	Coefficient of Determination	Coefficient of Association
Ore dampness	0.6189	−0.7867
Ore granularity	0.9750	−0.9874
Electronic belt weighing	0.8819	−0.9391

As Table 1 reveals, the coefficients of determination of ore particle size and electronic belt scale weight data are both above 0.8, indicating high goodness of fit, whereas the coefficient of determination of ore humidity is 0.6189, which is in the range 0.5–0.8, indicating low goodness of fit. The correlation value of coefficients of ore particle size and electronic belt scale weight data are both less than −0.9, showing a high degree of negative linear correlation. The correlation coefficient of ore humidity is −0.7867, also indicating a high degree of linear correlation. However, considering that the goodness of fit of the regression model is 61.89%, the ore humidity is not the main factor affecting the system. The ore particle size and electronic belt scale weight data are used as the input variables of the fuzzy controller, and the motor frequency is the output variable.

3. Python-Based Fuzzy Control Algorithm

3.1. Fuzzification and Establishment of Membership Function

The control purpose of the fuzzy controller is to make the ore feeding amount of the feeding system achieve dynamic balance, that is, when the central control room gives the expected value for the ore feeding amount, the whole feeding system controls the ore feeding amount of the feeder according to the data output motor frequency fed back by the online particle size analyzer and the electronic belt scale so as to maintain the total ore feeding amount unchanged, but can independently optimize the ore feeding according to the particle size on the transport belt. Python has a powerful standard library and open source. Among them, the scikit fuzzy module library can help us avoid the tedious and complex fuzzification and defuzzification operations as described above when designing the fuzzy controller. We can complete the construction of the fuzzy controller by clearly describing the parameters of the controlled object. The following describes how to design a fuzzy controller through the scikit fuzzy module library in Python to achieve the goal of autonomous optimization of ore feeding.

When constructing the fuzzy controller, the first step is to set the range of input and output variables according to the collected field data and actual production values of the mill plant. Hence, the ore particle size range was set to (0, 20], the step was set to 1, the electronic belt scale weight range was (0, 25], the step length was 0.5, the motor frequency range was (0, 50), and the step length was 1. The ore particle size and electronic belt scale weight were set as the preconditions, and the motor frequency was set as the post-condition. Then, using the implication operator, the membership of the pre-condition obtained by the membership function was truncated to the fuzzy set of the post-condition. This enabled the post-condition to be controlled through the preconditions, as illustrated by the code in Figure 8.

```
oresize = ctrl.Antecedent(x_oresize, 'oresize')
weight = ctrl.Antecedent(x_weight, 'weight')
frequency = ctrl.Consequent(x_frequency, 'frequency')
```

Figure 8. Definition of fuzzy controller conditions.

A fuzzy set is created after the variables have been fuzzified. This set contains all the controlled objects, that is, the ore particle size and electronic belt scale weight (the input variables) as well as the motor frequency (the output variable). The specific content of the fuzzy set is shown in Table 2, where “L,” “M,” and “H” indicate the low, medium, and high ranges of the variables, respectively.

Table 2. Fuzzy sets of the fuzzy controller.

Particle Size (Low) = L	Particle Size (Middle) = M	Particle Size (High) = H
Weight (low) = L	Weight (middle) = M	Weight (height) = H
Frequency (low) = L	Frequency (middle) = M	Frequency (high) = H

After the fuzzy sets have been established, it is necessary to describe the low, medium, and high membership ranges for ore particle size, electronic belt scale weight, and motor frequency. Triangle membership function (Fuzz.trIMF) was selected for this model. The corresponding module (Scikit-fuzzy) needs to be called from Python. After the description has been completed, the centroid method is used to resolve the “fuzz” module. The membership function of ore particle size (“oresize”) and electronic belt scale weight (“weight”) can be obtained by the “view” statement, as shown in Figures 9 and 10, respectively.

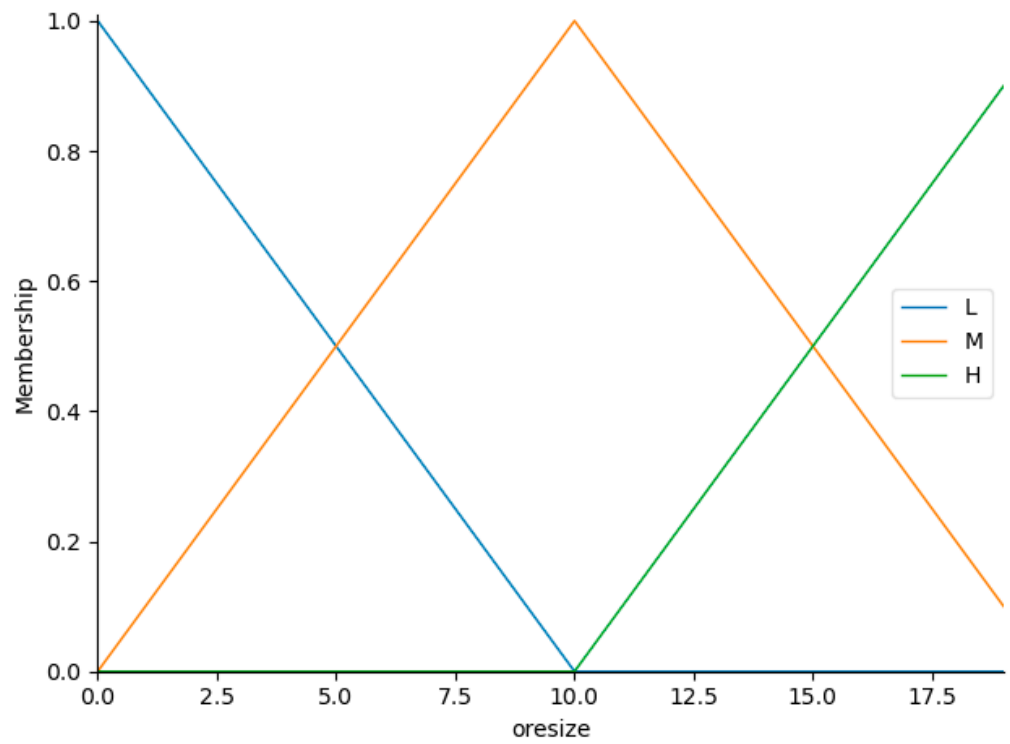


Figure 9. Membership function of ore particle size.

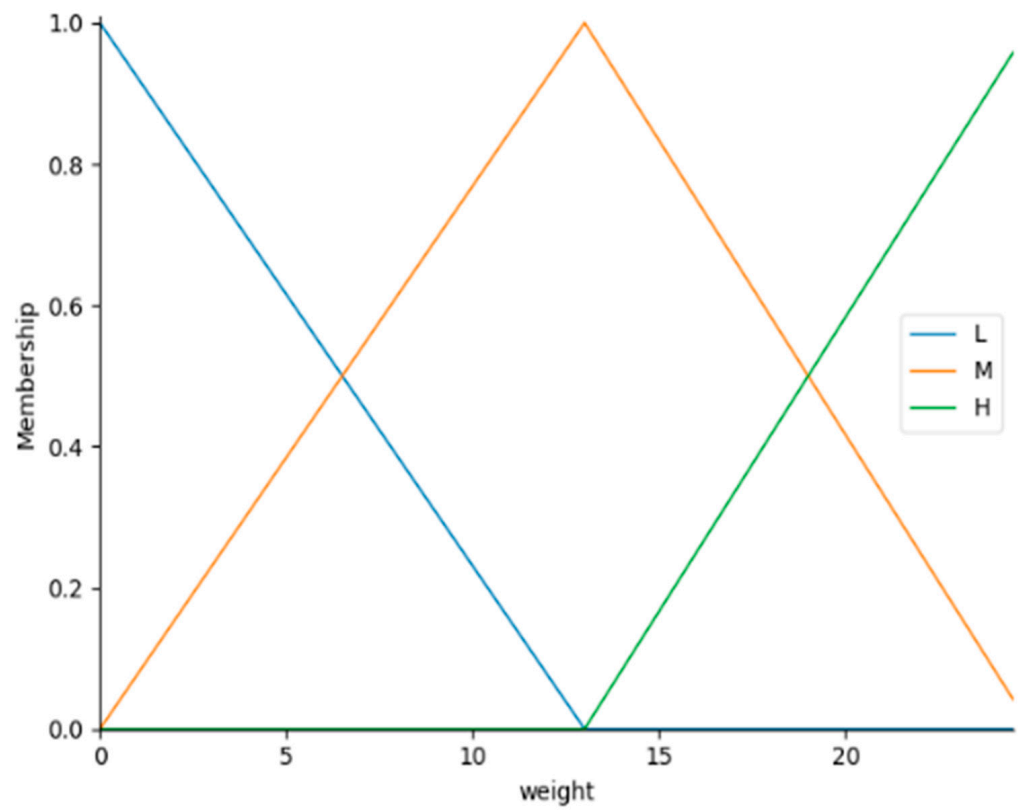


Figure 10. Membership function of electronic belt scale weight.

3.2. Construction of the Fuzzy Control Rule Base

A fuzzy control rule base is defined as a set of fuzzy rules in a fuzzy control system. The specific expression of fuzzy control rules is as follows:

$$R : \text{IF } x_1 \text{ is } F_1^l, \dots, \text{ and } x_n \text{ is } F_n^l \text{ THEN } y \text{ is } G^l$$

First, a fuzzy subset is uniformly divided, and the initial rules are generated according to the defined membership function. Then, according to the experience of field operators and related literature on beneficiation, the ore particle size, electronic belt scale weight, and motor frequency are considered individually. The fuzzy control rules can be obtained using the Cartesian product as shown in Table 3, where again, “L”, “M”, and “H” indicate the low, medium, and high ranges of the variable, respectively.

Table 3. Fuzzy control rules.

F		W		
		L	M	H
O	L	H	H	M
	M	H	M	L
	H	M	L	L

3.3. Simulation Experiment

After the mathematical functions of the simulation model have been determined, the fuzzy controller can be integrated into the simulation model based on Python for the simulation experiment, and the output data can be visualized. The specific process is realized through Python as follows:

- (1) Generate the grid-point coordinate matrices for the X and Y variables. Create a matrix for the Z axis at the height of X. Initialize all variables to 0.
- (2) Construct a nested loop with a range of 0 to 30, and then assign values to input variables X and Y. The values of X, Y, and Z are the values of the “oresize,” “weight,” and “frequency” variables in the fuzzy controller.
- (3) The axes 3D matplotlib library was used to map the data.

Using the above steps, a three-dimensional visualization of the simulation experiment results was obtained, and the results are shown in Figure 11.

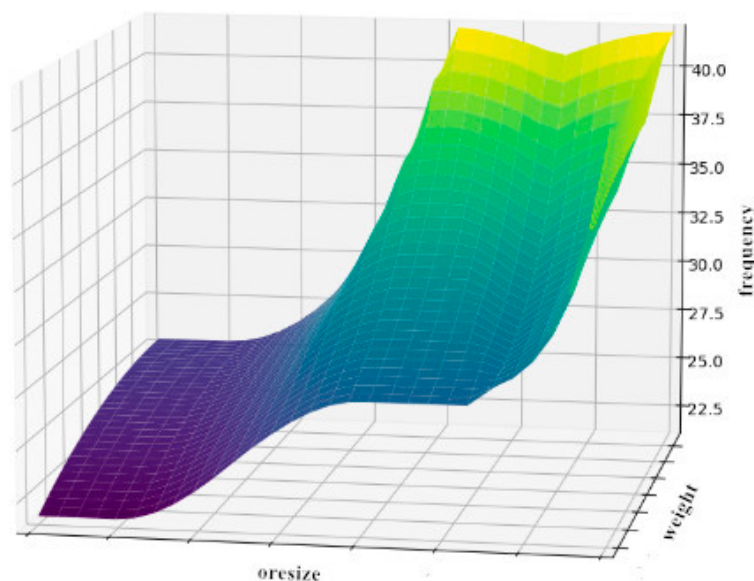


Figure 11. Three-dimensional visualization of the simulation experiment results.

Printing statements “print(‘Max’, Max(PP))” and “print(‘min’, min(PP))” output the maximum and minimum motor frequencies of the fuzzy controller of the feeding system. The simulation model obtains a maximum motor frequency of 41.67 Hz and a minimum motor frequency of 21.52 Hz. The simulation results of the maximum value agree with the actual production conditions of the concentrator.

In this study, code simulating the production conditions was added to the fuzzy controller to facilitate subsequent comparative experiments. The input values of the test were ore size = 9 and weight = 18, and the output value was frequency = 24.12. The membership function of the motor frequency under the simulated production conditions is shown in Figure 12.

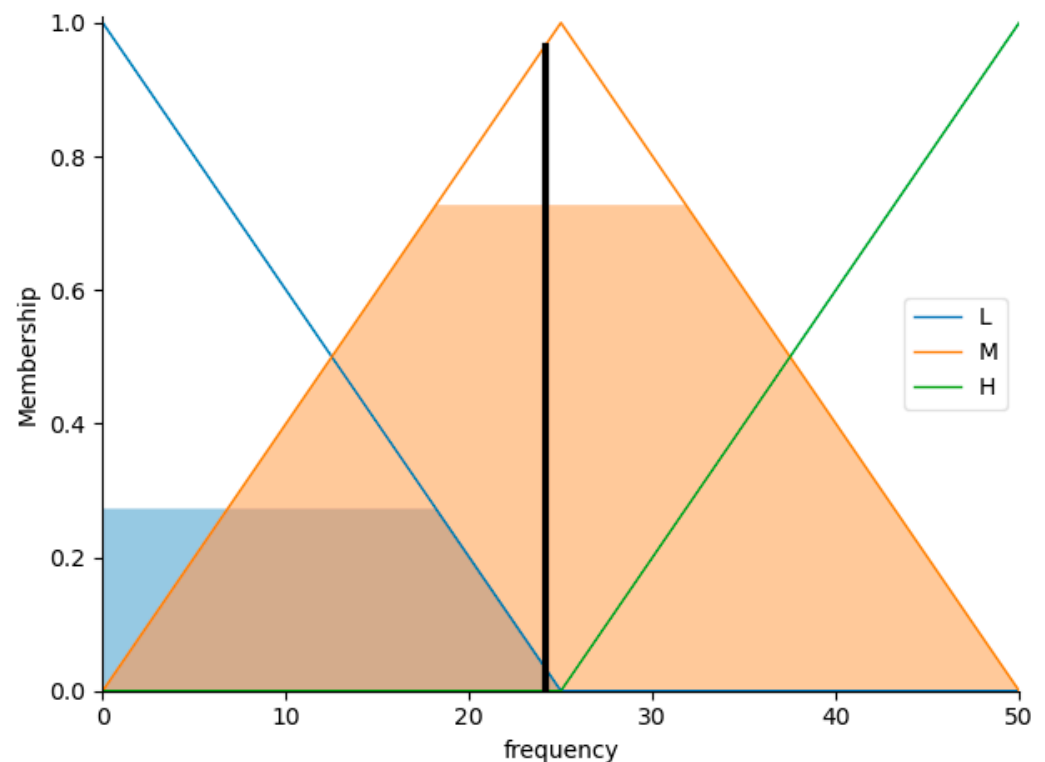


Figure 12. Membership function of motor frequency under simulated production conditions.

As Figure 12 shows, when the ore particle size is 9, the weight output by the electronic belt scale is 18, the output motor frequency is 24.12, the low membership value of motor frequency is 0.02, the medium membership value of motor frequency is 0.98, and the high membership value of motor frequency is 0. At this time, the motor frequency belongs to the range of “medium” to a very high degree, belongs to the range of “low” to a very small degree, and completely does not belong to the range of “high”.

3.4. Analysis of the Simulation Experiment Results

To better compare the advantages and disadvantages of a control system based on a traditional PID control algorithm with one based on the Python-based fuzzy mill feed control system, the following two simulation experiments were carried out:

- (1) The step responses [23] of the methods were compared.
- (2) The responses of the methods to changes in simulated field production conditions were compared to evaluate the robustness of the two control systems.

Figure 13 compares the step-response curves of the traditional PID control algorithm and the fuzzy control algorithm. At 0.01 s, a unit step signal with a given amplitude of 0.9 was input to the model. In the figure, the black curve indicates the PID control step response, and the red curve indicates the fuzzy control step response.

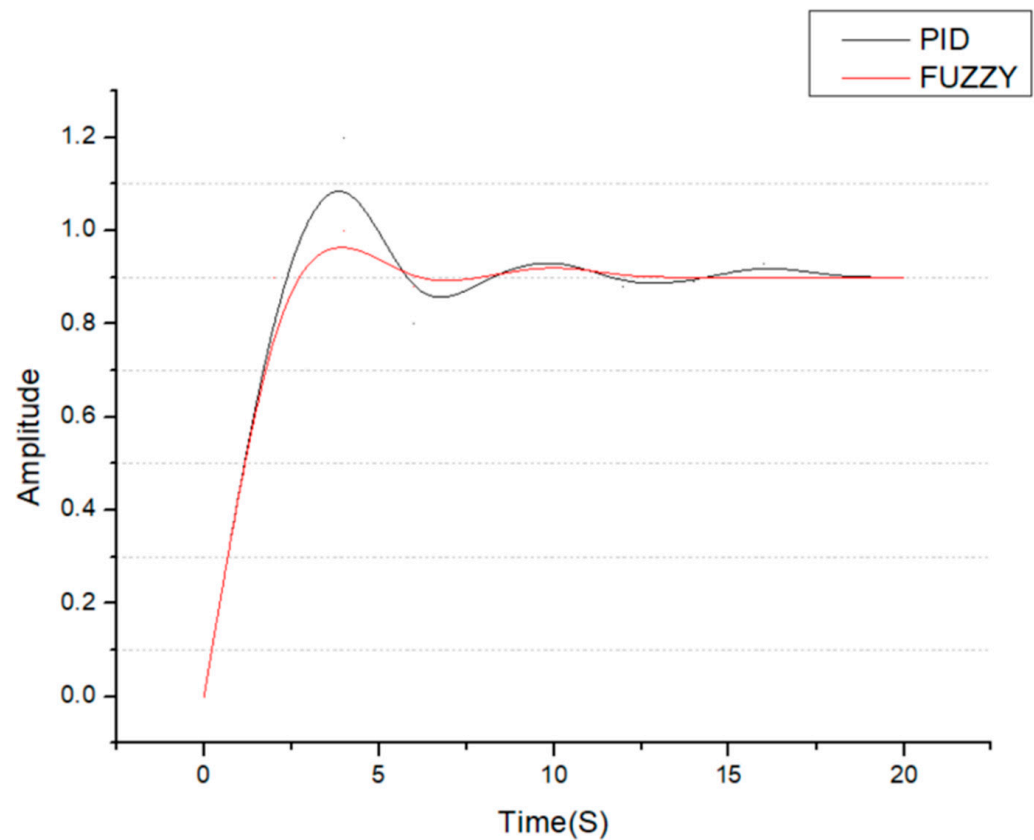


Figure 13. Comparison of traditional PID and fuzzy control step responses.

Table 4 presents the parameters of the step responses. Given the same unit step signal excitation, fuzzy control has a faster response speed, smaller overshoot, and smaller error, indicating that it has better dynamic performance than traditional PID control and good long-term stability.

Table 4. Comparison of the parameters of the traditional PID control and fuzzy control step responses.

Relevant Parameters	Traditional Pid Control	Fuzzy Control
Rise time	3.5	3.4
Overshoot	22%	6%
Average error	0.25	0.03

Next, a simulation experiment to compare the response of traditional PID control and fuzzy control to changes in production conditions was performed, because the ultimate objective is to use the electronic belt scale weight to dynamically balance the belt control. Hence, the electronic belt scale weight was changed to a value corresponding to 8.5 t/h (that is, the weight when the mill is fed at a rate of 8.5 t/h), and simulation models of the two systems were built. Figures 14 and 15, respectively, show the simulation models of the two algorithms when the production conditions change.

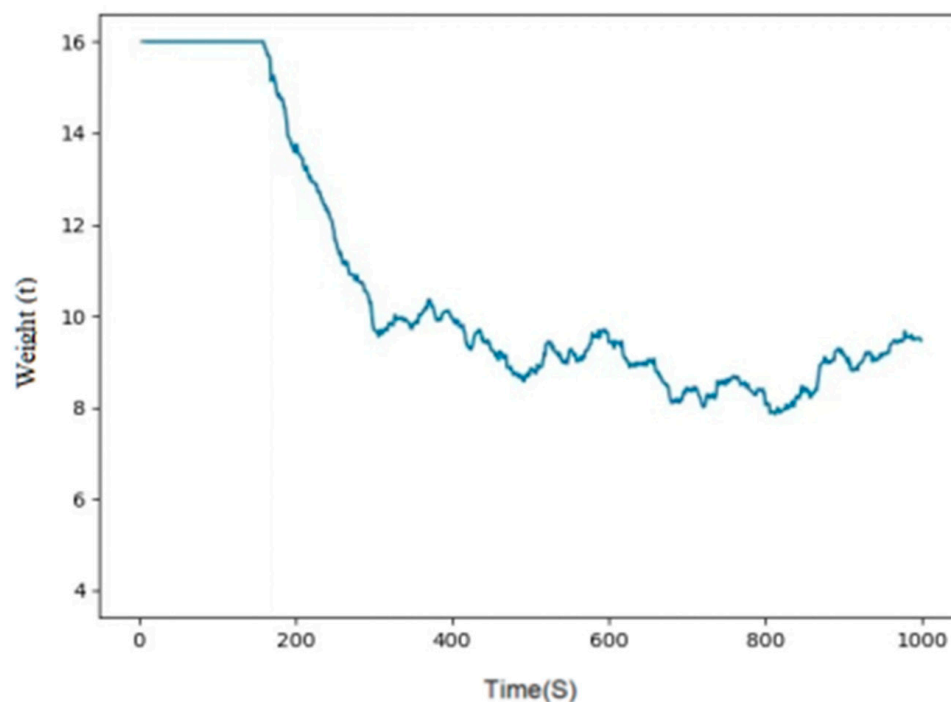


Figure 14. PID control response curve to changing production conditions.

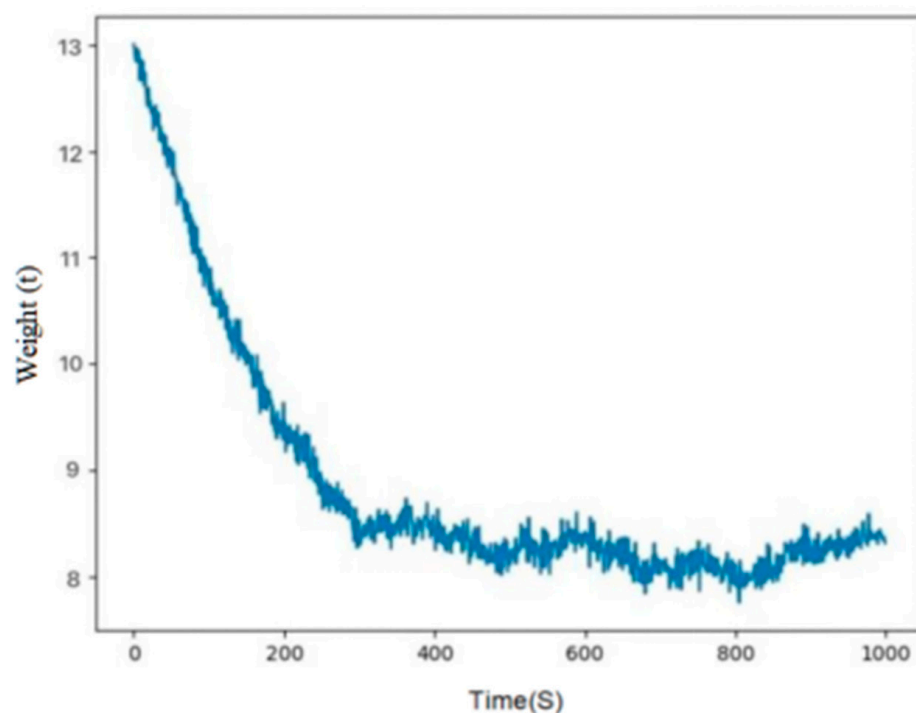


Figure 15. Fuzzy control response curve to changing production conditions.

As shown in Figure 14, the PID controller starts to adjust after a period of delay, and gradually reaches a stable value, which continues to fluctuate at 500 s, whereas the fuzzy controller responds quickly. As shown in Figure 15, when the production conditions change, the fuzzy controller starts the adjustment immediately and gradually becomes stable at about 250 s. Moreover, it begins to adjust independently according to the simulated production conditions. Although there is a certain fluctuation in the results, compared with the fluctuation of the traditional PID control algorithm, the fluctuation of the fuzzy control

algorithm has a smaller amplitude when the output reaches a stable value, which leads to higher control accuracy.

To better compare the control performance of traditional PID control and fuzzy control, print statements were used to output the values of the simulation models until they reached the expected value. Seventeen sets of data were output every 50 s from 200 s to 1000 s, and the overshoots of the two sets of data were calculated. The overshoots are compared in Figure 16.

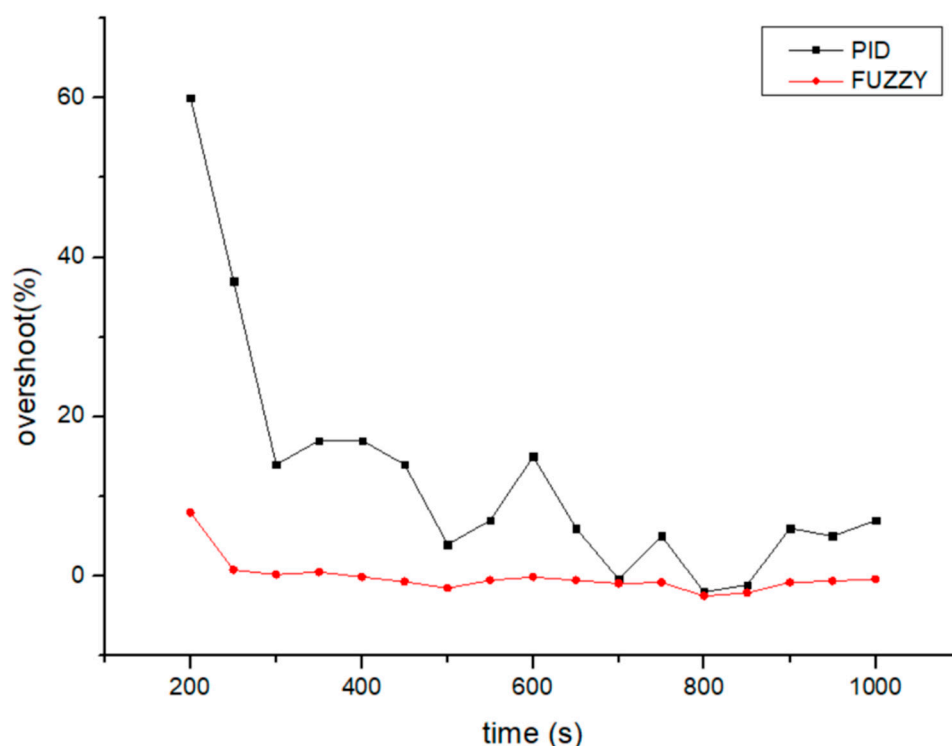


Figure 16. Comparison of the overshoots of traditional PID control and fuzzy control.

The data in Figures 14 and 15 are exported, and the overshoot is used to generate the comparison diagram of PID and fuzzy control algorithm, as shown in Figure 16. Figure 16 shows that the overshoot of the traditional PID control algorithm fluctuates when the production conditions are changed, whereas the overshoot of the fuzzy control algorithm changes substantially from 200 s to 250 s and then fluctuates very little after that.

4. Conclusions

In this paper, using a Python-based intelligent system development platform, fuzzy control was used to control to the feeding link of the grinding process so that the amount of ore fed to the mill could be adjusted according to the particle size of the ore on the conveyor belt. Moreover, the total amount of ore was dynamically balanced so that the feeding link of the mill would be more intelligent.

- (1) The Kalman filter was implemented using Python code. The results show that the error after noise filtering is less than 1.5%, indicating that Kalman filter algorithm improves the measurement accuracy of electronic belt scales.
- (2) Python was used to build a regression model to analyze the factors affecting accuracy. The results of the model indicate that the ore particle size and electronic belt scale weight could be used as the main influencing factors of the system. Therefore, the ore particle size and electronic belt scale data were used as the input, and the feeding motor frequency was used as the output.

Using the scikit fuzzy module library, a fuzzy controller suitable for the mill feeding process was developed, and a simulation experiment was carried out. The results show that the fuzzy mill feed control system based on Python can achieve the desired control purpose and performance, with good robustness. Moreover, the simulation results can be displayed in three dimensions, which is more intuitive and convenient for operators to observe and record.

Author Contributions: D.L. conceived of and designed the experiments. W.Z. and Y.D. performed the experiments and analyzed the data. R.L. and D.W. contributed materials. W.Z. wrote the paper. L.Y. and S.W. modified the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We thank the analysis laboratory of Kunming University of Science and Technology for its technical support. We thank Kimberly Moravec, from Liwen Bianji (Edanz) (www.liwenbianji.cn/ (accessed on 1 May 2022).) for editing the English text of a draft of this manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chai, T.; Ding, J.; Zhao, D.; Yan, A. Integrated Automation System of Minerals Processing and Its Applications. *IFAC Proc. Vol.* **2005**, *38*, 133–138. [[CrossRef](#)]
2. Nieto, L.; Olivares, J.; Gatica, J.; Ramos, B.; Olmos, H. Implementation of a Multivariable Controller for Grinding-Classification Process. *IFAC Proc. Vol.* **2009**, *42*, 55–60. [[CrossRef](#)]
3. Wei, D.; Craig, I.K. Grinding mill circuits—A survey of control and economic concerns. *Int. J. Miner. Process. Eng.* **2009**, *90*, 56–66. [[CrossRef](#)]
4. Ogonowski, S.; Ogonowski, Z.; Pawelczyk, M. Multi-Objective and Multi-Rate Control of the Grinding and Classification Circuit with Electromagnetic Mill. *Appl. Sci.* **2018**, *8*, 506. [[CrossRef](#)]
5. Shi, F.; Xie, W. A specific energy-based ball mill model: From batch grinding to continuous operation. *Miner. Eng.* **2016**, *86*, 66–74. [[CrossRef](#)]
6. Little, L.; Mainza, A.N.; Becker, M.; Wiese, J. Fine grinding: How mill type affects particle shape characteristics and mineral liberation. *Miner. Eng.* **2017**, *111*, 148–157. [[CrossRef](#)]
7. Jämsä-Jounela, S.L. Future Automation Systems in Context of Process Systems and Minerals Engineering. *IFAC PapersOnLine* **2019**, *52*, 403–408. [[CrossRef](#)]
8. Wang, X.; Wang, Y.; Yang, C.; Xu, D.; Gui, W. Hybrid modeling of an industrial grinding-classification process. *Powder Technol.* **2015**, *279*, 75–85. [[CrossRef](#)]
9. Zadeh, L.A. Fuzzy sets. *Inf. Control.* **1965**, *8*, 338–353. [[CrossRef](#)]
10. Li, B.; Li, Y.; Wang, H.; Ma, Y.; Hu, Q.; Ge, F. Compensation of automatic weighing error of belt weigher based on BP neural network. *Measurement* **2018**, *129*, 625–632. [[CrossRef](#)]
11. Li, S.; Xiong, F. An algorithm of characteristic expansion for fuzzy reasoning based on triple I method. In Proceedings of the 4th World Congress on Intelligent Control and Automation, Shanghai, China, 10–14 June 2002; pp. 464–468.
12. Kor, M.; Abkhoshk, E.; Tao, D.; Chen, G.; Modarres, H. Modeling and optimization of high chromium alloy wear in phosphate laboratory grinding mill with fuzzy logic and particle swarm optimization technique. *Miner. Eng.* **2010**, *23*, 713–719. [[CrossRef](#)]
13. Pani, A.K.; Mohanta, H.K. Soft sensing of particle size in a grinding process: Application of support vector regression, fuzzy inference and adaptive neuro fuzzy inference techniques for online monitoring of cement fineness. *Powder Technol.* **2014**, *264*, 484–497. [[CrossRef](#)]
14. Khodadadi, H.; Ghadiri, H. Fuzzy Logic Self-Tuning PID Controller Design for Ball Mill Grinding Circuits Using an Improved Disturbance Observer. *Min. Met. Explor.* **2019**, *36*, 1075–1090. [[CrossRef](#)]
15. Zhao, Y.; Collins, E.G.; Cartes, D.A. Self-tuning adaptive control for an industrial weigh belt feeder. *ISA Trans.* **2003**, *42*, 437–450. [[CrossRef](#)]
16. de Carvalho, R.M.; Campos, T.M.; Faria, P.M.; Tavares, L.M. Mechanistic modeling and simulation of grinding iron ore pellet feed in pilot and industrial-scale ball mills. *Powder Technol.* **2021**, *392*, 489–502. [[CrossRef](#)]
17. Behnamfard, A.; Roudi, D.N.; Veglio, F. The performance improvement of a full-scale autogenous mill by setting the feed ore properties. *J. Clean. Prod.* **2020**, *271*, 122554. [[CrossRef](#)]
18. Lafuente, D.; Cohen, B.; Fiorini, G.; García, A.; Onna, D. A Gentle Introduction to Machine Learning for Chemists: An Undergraduate Workshop Using Python Notebooks for Visualization, Data Processing, Analysis, and Modeling. *J. Chem. Educ.* **2021**, *98*, 2892–2898. [[CrossRef](#)]
19. McClarren, R.G. *Computational Nuclear Engineering and Radiological Science Using Python*; Academic Press: Cambridge, MA, USA, 2018.

20. Yang, J.; Hu, M. Filling the missing data gaps of daily MODIS AOD using spatiotemporal interpolation. *Sci. Total Environ.* **2019**, *633*, 677–683. [[CrossRef](#)]
21. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
22. Güney, Y.; Bozdogan, H.; Arslan, O. Mathematics, Robust model selection in linear regression models using information complexity. *J. Comput. Appl. Math.* **2021**, *398*, 113679. [[CrossRef](#)]
23. Kufoalor, D.; Frison, G.; Imslund, L.; Johansen, T.; Jørgensen, J. Block factorization of step response model predictive control problems. *J. Process Control* **2017**, *53*, 1–14. [[CrossRef](#)]