Burden Control Strategy Based on Reinforcement Learning for Gas Utilization Rate in Blast Furnace

Xiaoling Shen; Jianqi An; Min Wu; Jinhua She

Abstract: Gas utilization rate (GUR) is an important state parameter to reflect the energy consumption, the quality and production of the pig iron, and the distribution of the gas flow in a blast furnace. The GUR is mainly adjusted by burden distribution and hot-blast supply. According to the analysis of mechanism and data, burden distribution and hot-blast supply affect the GUR on a long-time scale and short-time scale, respectively. However, few of the previous researches proposed the control method for the GUR and they did not consider multi-time-scale characteristics. Thus, it is necessary to design a control strategy or system for the GUR considering the multi-time-scale characteristics, which can make the GUR have a reasonable development trend. This paper presented a burden control strategy based on a reinforcement learning algorithm for the GUR. The method improved the development trend of the GUR on a long-time scale. The experimental results demonstrated that the sequence of the parameters of the burden distribution given by the presented method ensured a reasonable development trend of the GUR on a long-time scale.

Keywords: Blast furnace, gas utilization rate, burden control strategy, reinforcement learning algorithm, long-time scale.

1. INTRODUCTION

A blast furnace (BF) is a complex reactor to convert iron ore into liquid pig iron through a series of physical changes and chemical reactions (shown in Fig. 1) (An et al., 2020, 2018b; Gomes et al., 2017). The iron ore and coke are discharged into a BF from the top to form the iron-ore layers and coke layers, which are controlled by burden distribution. The hot blast is discharged into a BF from the bottom, which is adjusted by hot-blast supply. The coke burns with the hot blast to form an upward gas flow. Then, the iron ore reacts with carbon monoxide in the upward gas flow to form liquid pig iron, slag, and BF gas flow. The BF gas flow is discharged from the top of a BF, which is called the top gas flow. The gas utilization rate (GUR) is the ratio of the carbon dioxide content to the total content of carbon monoxide and carbon dioxide in the top gas flow. The GUR, \( \rho_{CO} \), is calculated as

\[
\rho_{CO} = \frac{V_{CO_2}}{V_{CO} + V_{CO_2}}.
\]  

(1)

Improving the GUR is good for reducing consumption, improving the quality of the pig iron, and increasing the production. Thus, it is important to control the GUR.

Some researches analyzed the GUR based on mechanism analysis. For example, a definition of the GUR was given by analyzing the correlation between the GUR and the chemical reactions (Kou et al., 2016). A gas flow distribution and operation state were determined by the GUR in a BF (Xiang et al., 2013). An impact of the natural gas injection on GUR was analyzed by mathematical modeling and energy exchange (Guo et al., 2013).

In addition, some researches focused on predicting and/or optimizing GUR based on data-driven methods. For example, a relation model based on low-frequency feature extraction was established to analyze the explicit relation between the burden distribution and states (Zhang et al., 2017). Thereby, a decision-making strategy was designed to improve the GUR according to the burden distribution (Wu et al., 2018). Besides, a hybrid model was established to improve the GUR according to analyzing the position of the pile surface (Shi et al., 2016). Meanwhile, some models were built to predict the GUR, such as a model based on an online sequential extreme learning machine (Li et al., 2017) and an echo state net-
strategy based on Q-learning on a long-time scale. Section 4 analyzes experimental results to demonstrate that the presented method improves the GUR on a long-time scale. And Section 5 draws conclusions and introduces the future works.

2. REINFORCEMENT LEARNING

Reinforcement learning learns a suitable behavior through the experiences generated by the interaction between an agent and the environment (Sutton and Barto, 2011). The formalization of reinforcement learning is based on a Markov decision process, which mainly contains 5 elements, \( (S,A,P(s(t+1)|s(t),a(t)),R(t+1),\gamma) \), where

- \( \mathcal{S} \) denotes a set of states of the environment, and \( s(t) \in \mathcal{S} \) means a state at time \( t \);
- \( \mathcal{A} \) denotes a set of actions of an agent, and \( a(t) \in \mathcal{A} \) means a action selected at time \( t \);
- \( P(s(t+1)|s(t),a(t)) \) denotes the probability of selecting \( a(t) \) when \( s(t) \) is changed to \( s(t+1) \);
- \( R(t+1) \) denotes a reward given by the environment when \( s(t) \) is changed to \( s(t+1) \) by executing \( a(t) \); and
- \( \gamma \) denotes the discount rate.

As the Q-learning algorithm is one of the most important algorithms in reinforcement learning proposed in Watkins and Dayan (1992), this paper uses a Q-learning algorithm to train the burden control sequence. The most important step of Q-learning is to train an action-value function \( Q(s(t),a(t)) \), which means the expected value when \( s(t) \) is changed to \( s(t+1) \) after executing \( a(t) \). \( Q(s(t),a(t)) \) is updated as

\[
\begin{cases}
Q(s(t),a(t)) = Q(s(t),a(t)) + \alpha \Delta R \\
\Delta R = R(t+1) + \gamma \max_a Q(s(t+1),a) - Q(s(t),a(t))
\end{cases}
\]

where \( \alpha \in (0,1] \) is the learning rate.

According to the updated method of \( Q(s(t),a(t)) \), the Q-learning considers the impact of current state and action on subsequent states. In the iron-making process, the subsequent GUR is affected by the current parameters of the operations of BF. Thus, this paper uses the Q-learning algorithm to design the burden control strategy for the GUR on a long-time scale.

3. BURDEN CONTROL STRATEGY FOR GUR

This section introduces a burden control strategy for the GUR based on Q-learning algorithm, which is designed to keep a reasonable development trend for the GUR on a long-time scale. The method contains 6 parts: the selection of the controlled state \( G(t) \), the definition of states \( \mathcal{S} \) and actions \( \mathcal{A} \), the policy of action selection \( \pi(a(t)|s(t)) \), the calculation of the reward \( R \), the update of \( Q(s(t),a(t)) \), and the strategy of burden control.

3.1 Selection of Controlled State

The main purpose of the method is to keep the GUR rising on the long-time scale. Thus, this paper uses the long-time-scale part of the GUR as the controlled state.
In order to reduce the complexity of the algorithm, this section analyzes the correlations between each parameters of burden distribution based the Pearson coefficient method. The correlation is calculated as

$$
\gamma_{jk} = \frac{\sum_{t=1}^{L} (x_k(t) - \bar{x}_k)(x_j(t) - \bar{x}_j)}{\sqrt{\sum_{t=1}^{L} (x_k(t) - \bar{x}_k)^2} \sqrt{\sum_{t=1}^{L} (x_j(t) - \bar{x}_j)^2}},
$$

where \(\gamma_{jk}\) is the correlation between \(x_k(t)\) and \(x_j(t)\); \(L\) is the length of the time series of \(x_k(t)\) and \(x_j(t)\); \(x_k(t)\) and \(x_j(t)\) are the time series of the kth and jth states of burden distribution, respectively; \(\bar{x}_k\) and \(\bar{x}_j\), the average values of \(x_k(t)\) and \(x_j(t)\), respectively. The larger \(\gamma_{jk}\) is, the stronger correlation between \(x_k(t)\) and \(x_j(t)\) is.

Table 1 shows the correlation between each state of burden distribution. It is clear that \(s_{oc1}, s_{oc2}\), and \(s_{oc3}\) are positively correlated, which means they can be adjusted by an operation. \(s_{oc1}\) and \(s_{oc2}\) are negatively correlated, which means they can also be adjusted by an operation, just by using the opposite value. \(s_{oc4}\) is not related to others.

<table>
<thead>
<tr>
<th>States</th>
<th>(s_{oc1})</th>
<th>(s_{oc2})</th>
<th>(s_{oc3})</th>
<th>(s_{oc4})</th>
<th>(s_{oc5})</th>
<th>(s_{oc6})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_{oc1})</td>
<td>1.0000</td>
<td>0.9400</td>
<td>0.8484</td>
<td>-0.024</td>
<td>-0.187</td>
<td>0.500</td>
</tr>
<tr>
<td>(s_{oc2})</td>
<td>0.9400</td>
<td>1.0000</td>
<td>0.8377</td>
<td>-0.017</td>
<td>-0.205</td>
<td>0.316</td>
</tr>
<tr>
<td>(s_{oc3})</td>
<td>0.8466</td>
<td>0.8377</td>
<td>1.0000</td>
<td>0.041</td>
<td>-0.381</td>
<td>0.348</td>
</tr>
<tr>
<td>(s_{oc4})</td>
<td>-0.024</td>
<td>-0.017</td>
<td>0.041</td>
<td>1.000</td>
<td>-0.204</td>
<td>0.231</td>
</tr>
<tr>
<td>(s_{oc5})</td>
<td>-0.187</td>
<td>-0.205</td>
<td>-0.381</td>
<td>-0.204</td>
<td>1.000</td>
<td>-0.746</td>
</tr>
<tr>
<td>(s_{oc6})</td>
<td>0.3000</td>
<td>0.317</td>
<td>0.348</td>
<td>0.231</td>
<td>-0.746</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Based on the correlation, the action, \(a(t) \in A\), is defined as

$$
a(t) = (a_T(t), a_1(t), a_2(t), a_3(t)).
$$

Thus, the next state \(s(t+1)\) after executing \(a(t)\) in \(s(t)\) is calculated as follows:

$$
\tilde{s}(t+1) = (s_{oc1}(t+1), s_{oc2}(t+1), s_{oc3}(t+1), s_{oc4}(t+1), s_{oc5}(t+1), s_{oc6}(t+1)),
$$

where

$$
s_{oc1}(t+1) = s_{oc1}(t) + a_1(t),
\noncl(t+1) = n_{oc1}(t) + a_1(t),
\noncl(t+1) = n_{oc2}(t) + a_1(t),
\noncl(t+1) = n_{oc3}(t) + a_1(t),
\noncl(t+1) = n_{oc4}(t) + a_2(t),
\noncl(t+1) = n_{oc5}(t) + a_3(t),
\noncl(t+1) = n_{oc6}(t) - a_3(t)
$$

Then,

$$
s(t+1) = (s_{oc1}(t+1), s_{oc2}(t+1), s_{oc3}(t+1), s_{oc4}(t+1), s_{oc5}(t+1), s_{oc6}(t+1)),
$$

where \(s_{oc1}(t+1)\) is the median value of the interval corresponding to \(n_{oc}(t+1)\).
3.3 Policy of Action Selection

This paper takes a $\epsilon$-greedy algorithm as the policy of action selection, which is shown as

$$\pi(a(t) \mid s(t)) = \begin{cases} \epsilon/m + 1 - \epsilon & \tilde{a} = \arg \max_{a \in A} Q(s(t), a) \\ \epsilon/m & \text{else} \end{cases},$$

where $\epsilon$ is exploratory rate; $\tilde{a}$, the action that maximize $Q(s(t), a)$; $m$, the number of actions.

Thus, this paper uses the probability of $\epsilon/m + 1 - \epsilon$ to select the action that maximize $Q(s(t), a)$ and the probability of $\epsilon/m$ to randomly select the other actions.

3.4 Calculation of Reward

In order to ensure a reasonable development trend of the GUR on a long-time scale, the reward $R(t + 1)$ is designed as

$$R(t + 1) = \tilde{G}(t + 1) - \hat{G}(t),$$

where $\tilde{G}(t + 1)$ and $\hat{G}(t)$ are the prediction value of the long-time part of the GUR at time $t + 1$ and $t$, respectively, which are obtained by the long-time-scale prediction model of the GUR.

This paper uses the BP neural network algorithm to establish the long-time-scale prediction model. The parameters when training the model are as follows: hidden layers is 9; epochs, 3000; goal, $10^{-3}$; lr, 0.1; transfer function, tansig; training function, trainlm; bias learning function, learnngdm. The inputs of the prediction model contain two parts: 6 states of burden distribution and 6 historical information of the GUR. The number of history information is calculated by the partial autocorrelation function in Box and Jenkins (1971). The output of the prediction model is the long-time-scale part of the GUR. The long-time-scale part of the GUR is calculated by the methods shown in An et al. (2019).

This paper uses four-month continuous samples of real-world industrial data that were selected from the database of a 2800 m$^3$ BF. 450 samples are used for training; the rest 40 samples, for testing. The interval time between samples is 6 hours. Figure 2 shows that the prediction model accurately predicts the development trend of GUR on a long-time scale, which can be used to calculate the reward.

3.5 Update of Action-Value Function

The aim of the update is to train a table of $Q(s(t), a(t))$ by multiple iterations. For each iteration, $Q(s(t), a(t))$ is calculated based on $\pi(a(t) \mid s(t))$ and $R(t + 1)$ by (2). The update will not stop until the maximum number of iterations or $Q(s(t), a(t))$ converges.

3.6 Strategy of Burden Control

The presented method designs a strategy of burden control according to the table of $Q(s(t), a(t))$. The strategy is obtained as follows:

**Step 1** Select the action $a(t)$ as

$$a(t) = \arg \max_{a \in A} Q(s(t), a).$$

**Step 2** Calculate $\tilde{s}(t + 1)$ based on $s(t)$ and $a(t)$ by (7) and (8).

**Step 3** Calculated the state sequence of burden distribution in $s(t + 1)$ by using the interval number obtained by $\tilde{s}(t + 1)$.

**Step 4** Get the burden control strategy based on the state sequence of burden distribution.

4. EXPERIMENT AND DISCUSS

This section proves the effectiveness of the presented method by comparing the experimental result and the real-world industrial data. The data used in this experiment were selected from the database of a 2800 m$^3$ BF.

In the experiment, the maximum value of $s_T$ is 12. As the interval time between samples is 6 hours, the total period of time is 72 hours (3 days). Besides, the current prediction value of the long-time-scale part of the GUR is taken as history information in the next time, which is used as one of the inputs in the long-time-scale prediction model.

Figure 3 shows the state sequences of burden distribution, which is the burden control strategy. Each data is a median value of an interval, which represented an interval. Table 2 shows the intervals corresponding to the data shown in Fig. 3. The data in time 1 is the initial value of each parameter. From the results, $s_{oc1}$, $s_{oc2}$, and $s_{oc3}$ are basically the same as the initial values on the whole, $s_{oc4}$ is higher than its initial value, $s_{a1}$ is lower than its initial value and $s_{oc2}$ is higher than its initial value.

Figure 4 shows the comparison of the experimental results and real-world data of the GUR on a long-time scale. The experiment results (the red line) are obtained by the sequences of the parameters of the burden distribution (shown in Fig. 3) and the prediction model of the GUR on the long-time scale. The results are higher than real-world data. Besides, the results show that the burden control strategy changes the downward trend of the GUR on the long-time scale. Figure 5 shows that the burden control strategy increases the long-time-scale part of the GUR by 1.5% at most. The experiment demonstrates the effectiveness of the presented method.

Figure 6 shows the comparison of the fusion result and the real-world data of the GUR. The fusion result is calculated by the long-time-scale part and the short-time-scale part of the GUR. The long-time part of the GUR is calculated by...
Table 2. Intervals corresponding to burden control strategy

<table>
<thead>
<tr>
<th>Time</th>
<th>soc1</th>
<th>soc2</th>
<th>soc3</th>
<th>soc4</th>
<th>sc1</th>
<th>sc2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.451, 4.859]</td>
<td>(0.072, 0.075]</td>
<td>(0.265, 0.276]</td>
</tr>
<tr>
<td>2</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.451, 4.859]</td>
<td>(0.070, 0.072]</td>
<td>(0.276, 0.290]</td>
</tr>
<tr>
<td>3</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.042, 4.451]</td>
<td>(0.067, 0.070]</td>
<td>(0.290, 0.304]</td>
</tr>
<tr>
<td>4</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.042, 4.451]</td>
<td>(0.065, 0.067]</td>
<td>(0.304, 0.318]</td>
</tr>
<tr>
<td>5</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.451, 4.859]</td>
<td>(0.059, 0.062]</td>
<td>(0.318, 0.331]</td>
</tr>
<tr>
<td>6</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.859, 5.268]</td>
<td>(0.059, 0.062]</td>
<td>(0.331, 0.345]</td>
</tr>
<tr>
<td>7</td>
<td>(6.879, 7.104]</td>
<td>(6.691, 6.918]</td>
<td>(6.940, 7.195]</td>
<td>(4.451, 4.859]</td>
<td>(0.059, 0.062]</td>
<td>(0.331, 0.345]</td>
</tr>
<tr>
<td>8</td>
<td>(6.879, 7.104]</td>
<td>(6.691, 6.918]</td>
<td>(6.940, 7.195]</td>
<td>(5.268, 5.676]</td>
<td>(0.059, 0.062]</td>
<td>(0.331, 0.345]</td>
</tr>
<tr>
<td>9</td>
<td>(6.879, 7.104]</td>
<td>(6.691, 6.918]</td>
<td>(6.940, 7.195]</td>
<td>(4.859, 5.268]</td>
<td>(0.062, 0.065]</td>
<td>(0.318, 0.331]</td>
</tr>
<tr>
<td>10</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.859, 5.268]</td>
<td>(0.062, 0.065]</td>
<td>(0.318, 0.331]</td>
</tr>
<tr>
<td>11</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.859, 5.268]</td>
<td>(0.062, 0.065]</td>
<td>(0.318, 0.331]</td>
</tr>
<tr>
<td>12</td>
<td>(7.104, 7.328]</td>
<td>(6.918, 7.145]</td>
<td>(7.195, 7.450]</td>
<td>(4.859, 5.268]</td>
<td>(0.062, 0.065]</td>
<td>(0.318, 0.331]</td>
</tr>
</tbody>
</table>

The method presented in this paper. The short-time part is the real-world data of the GUR controlled by hot-blast supply. The result shows that the burden control strategy improves the value of the GUR as a whole by increasing the long-time-scale part of the GUR, further illustrating the effectiveness of the presented method.

5. CONCLUSION

The main contribution of this paper is to present a burden control strategy based on Q-learning algorithm for the GUR. The goal of the method is to make a reasonable development trend of the GUR on a long-time scale. The experiment demonstrates that the presented method yields a state sequence of burden distribution, which increases the GUR on the long-time scale.

The method presented in this paper improves the GUR and changes its development trend. Meanwhile, this method uses the correlations between each state to design the actions, which reduces the complexity of the method. However, this method only considers a fixed step size when training the burden control strategy. Besides, the method only gives the sequences of the ore-to-coke ratios and central-coke ratios that reflect the changes in the operating parameters of the burden distribution. Thus, we will use a...
random step size when finding the burden control strategy. And we will study the relationship between the ore-to-coke and central-coke ratios and the operation parameters.

REFERENCES


