Chen *et al*. *Complex Eng Syst* 2024;4:7
DOI: 10.20517/ces.2024.02

Complex Engineering Systems

Research Article

Open Access

Layered random fault injection method for the air braking system based on multiple Markov chains

Zhiwen Chen[1], **Jingke Fan**[1], **Lijuan Peng**[1], **Hao Luo**[2], **Chao Cheng**[3], **Zhiyong Chen**[4]

[1]School of Automation, Central South University, Changsha 410083, Hunan, China.
[2]Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150006, Heilongjiang, China.
[3]School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, Jilin, China.
[4]School of Engineering, The University of Newcastle, Callaghan, NSW 2308, Australia.

Correspondence to: Dr. Zhiwen Chen, School of Automation, Central South University, No. 932, Lushan South Road, Changsha, 410083, Hunan, China. E-mail: zhiwen.chen@csu.edu.cn

How to cite this article: Chen Z, Fan J, Peng L, Luo H, Cheng C, Chen Z. Layered random fault injection method for the air braking system based on multiple Markov chains. *Complex Eng Syst* 2024;4:7. http://dx.doi.org/10.20517/ces.2024.02

Received: 7 Jan 2024 **First Decision:** 31 Jan 2024 **Revised:** 21 Feb 2024 **Accepted:** 11 Mar 2024 **Published:** 1 Apr 2024

Academic Editor: Hamid Reza Karimi **Copy Editor:** Fanglin Lan **Production Editor:** Fanglin Lan

Abstract

The air braking system is crucial for the safe operation of high-speed trains but is susceptible to faults from harsh environments and prolonged use. However, faulty data in practice are still rare because of the "safety oriented principle". For this purpose, fault injection is regularly employed. Due to the stochastic nature of faults in the system, random fault injection can more realistically simulate faulty scenarios compared to the deterministic fault injection. The traditional method entails analyzing a significant amount of raw data to extract the fault distribution function, followed by random sampling. However, the obstacles lie in the scarcity of raw fault data and the labor-intensive nature of constructing the fault distribution function. This paper proposes a layered random fault injection method based on multiple Markov chains. First, a multi-layer structured fault model base is established for the system, followed by the implementation of layered fault injection. Then, the random fault types and degrees are realized using Markov chains, in which the fault probability function is determined by the state transition matrix. Subsequently, a low-complexity Alias sampling algorithm is proposed for discrete random sampling. The nominal model is transformed into a corresponding fault model based on the sampling outcomes, facilitating the acquisition of fault data. Finally, a graphical user interface is developed to present and visualize the validation results.

Keywords: Air braking system, fault simulation, fault injection, Markov chain, random sampling

1. INTRODUCTION

High-speed trains play an essential role in both railway passenger transportation and the economic development of regions[1,2]. However, owing to their elevated speeds, the braking system has emerged as a critical safety consideration[3,4]. The air braking system (ABS) serves as the central subsystem; however, its intricate operational conditions and prolonged usage frequently result in wear, aging, and sudden abnormalities in certain components. ABS constitutes a primary origin of train malfunctions, and the failure to promptly detect and address these faults can result in significant safety incidents, causing substantial casualties and property damage[5]. The derailment incident of the Talgo 250 Hybrid high-speed train in Spain on July 24, 2013, was attributed to a braking system malfunction, leading to a substantial number of casualties. However, simulating faults on real high-speed trains is expensive and challenging, with the majority of existing platforms primarily concentrating on modeling regular train operations[6]. Therefore, investigating fault injection in ABS of high-speed trains holds significant importance. To fulfill the criteria of minimizing risk, reducing costs, and achieving short development cycles, simulation-based fault injection is often used to simulate and inject faults into ABS. Fault injection is a fault simulation technique proposed in the 20th century[7]. Its essence is to artificially introduce faults into the target system with a certain strategy for a specified fault type so as to observe and analyze the operation behavior of the target system under the condition of injected faults. It has garnered significant attention[8,9].

In the fault injection of automotive systems: Abboush *et al.*[10] proposed a new method based on hardware in the loop (HIL) simulation and automatic real-time fault injection method for generating, analyzing, and collecting data samples in the presence of single and concurrent faults. In the fault injection of aviation control systems: Joshi *et al.*[11] first proposed the application of Microsoft baseline security analyzer to the fault modeling and verification process of aviation control systems. Linnosmaa *et al.*[12] used architecture analysis and design language or its fault extension language to verify the safety requirements of aviation control system fault models. In the fault injection of hydraulic systems: Liu[13] used AMESim to establish a simulation model of the hydraulic control module of the drilling machine loop and studied the fault tolerance and fault detection of hydraulic systems. Karpenko and Sepehri[14] built mathematical models and physical experiments of servo-hydraulic positioning systems and studied leakage between actuator cavities. Niksefat and Sepehri[15] established the mathematical model of the electro-hydraulic servo system and analyzed the influence of sensor fault and hydraulic pump fault on the system through experiments. In the fault injection of heating, ventilation, and air-conditioning (HVAC) systems, Kiamanesh Bahareh[16] presents modeling patterns of numerous faults in HVAC systems based on data from field failure rates and maintenance records. The extended fault injection framework supports the injection of multiple faults with exact control of the timing, locality, and values in fault-injection vectors. A multi-dimensional fault model is defined, including the probability of the occurrence of different sensor and actuator faults. Comprehensive experimental results provide insights into the behavior of the system for concrete example scenarios using multiple fault patterns. In the fault injection of the train system: Zhou *et al.*[17] built a 1:1 electric multiple-unit platform. The brake cylinder leakage fault is realized by adding components to the platform. This fault injection method simulates a single fault type and causes a certain degree of damage to the components when simulating the fault, which hinders the feasibility of conducting experiments repeatedly. Gou *et al.*[18] and Youssef *et al.*[19] reported fault injection of open-circuit faults of rectifiers, gain and noise faults of grid-side current sensors and direct current voltage sensors in a hardware-in-the-loop simulation platform. Chen *et al.*[20] used Simulink/AMESim co-simulation to realize the fault injection of brake cylinders, train pipes and other components in the train braking system and developed a new open-source benchmark for validating the fault diagnosis methods of the electro-pneumatic brake system (EPBS). Yang *et al.*[8] proposed a fault injection strategy based on signal conditioning that realized the simulation of fault scenarios, such as sensors, traction converters, and motors, in the traction drive system. The fault injection simulations mentioned above all involve deterministic faults. By observing the behavior of the system under specific fault scenarios and acquiring faulty data, it becomes possible to formulate targeted fault diagnosis algorithms[21,22]. Presently, the majority of validation methods for diagnostic algorithms entail the establishment of deterministic experiments to assess fault scenarios. Given the stochastic nature of fault occurrences in the system, random fault injection simulates the fault occurrences of the entire system in real-world scenarios. This enhances the objectivity of fault injection and introduces greater flexibility in the injection process. Therefore, employing random fault injection represents a more convenient and efficient approach for validating fault diagnosis algorithms[23]. Presently, the prevalent approach for injecting random faults into a system involves fitting a fault injection model density function using an extensive dataset derived from the system. Subsequently, the fault parameters are randomly sampled based on this function to accomplish the random fault injection[24]. However, the creation of the fault injection model density function involves multiple steps, including data processing, parameter estimation, and function fitting, rendering it a laborious process. Furthermore, the existing challenge for ABS lies in the absence of comprehensive and accurate raw statistical data on fault distribution. This scarcity hampers the comprehensive construction of the fault injection model density function. Therefore, finding a convenient and effective method for conducting random fault injection on ABS is a pressing issue.

The main contributions of this paper are as follows:
1. A layered random fault injection method based on multiple Markov chains (MCs) is proposed, leveraging multiple MCs to systematically represent fault scenarios. This approach constructs both convergent fault type MCs and fault degree MCs, effectively diminishing reliance on initial fault data and significantly streamlining the laborious process of establishing fault distribution.
2. The Alias Sampling (A-S) algorithm has been augmented, incorporating queue operations in lieu of arrays, which serves to decrease the computational complexity involved in constructing Alias tables. This adjustment enhances the efficiency of conducting random sampling within fault distributions.
3. A random fault injection and fault diagnosis platform for the ABS is designed and developed based on Simulink/AMESim co-simulation, including modules for the underlying simulation model, random fault injection, fault diagnosis, and visual display.

Preliminary research has been reported at[25]. The rest of this paper is organized as follows: Section 2 describes the construction of ABS fault model base. Section 3 proposes the method of constructing a random fault distribution based on steady-state MCs and uses the A-S algorithm for random sampling. In Section 4, the simulation results are given. Finally, Section 5 presents the main conclusions.

2. PRELIMINARY

2.1. Typical faults

ABS comprises diverse components, including EP valves, train pipes, brake cylinders, and sensors. Subject to diverse and intricate operational environments and factors, each device category is inevitably prone to encountering various types and degrees of faults. This paper simulates typical fault scenarios of ABS, including sensors, EP valves, brake cylinders, and train pipes. For the sensor, four kinds of faults, including offset, gain, drift and impact, are simulated. For the EP valve, external leakage, internal leakage and spring damage are simulated. For the brake cylinder, the air leakage fault is simulated. For the train pipe, the air leakage fault is simulated. The fault parameters for each fault type are categorized into specific ranges, classifying the fault degrees into three categories based on the national standard GB/T709-2006, which divides equipment status into four levels: normal, mild, average, and severe. Train pipe fault: Compressed air is transmitted in the train pipe. Due to wear and tear over long periods of operation, the walls of the train pipe are thinned or even damaged, eventually leading to gas leaks. This means that the braking system lacks adequate air pressure, resulting in prolonged braking response time as the air pressure decreases within the braking system, thereby delaying the desired braking effect. This leads to a sluggish response of the braking system to braking commands issued by the train driver, consequently affecting the overall braking performance of the train. Sensor offset fault: This particular fault typically induces a fixed deviation in the measurement of the raw signal. The braking system

Table 1. Summary of typical faults

Fault component	Fault type	Fault degree	Changes
Sensor	Offset fault	Normal/ Mild/ Average/ Severe	The output signal increases by a fixed offset
	Gain fault	Normal/ Mild/ Average/ Severe	The output signal is multiplied proportionally
	Drifting fault	Normal/ Mild/ Average/ Severe	The output signal adds an extra signal that is proportional to time
	Shock fault	Normal/ Mild/ Average/ Severe	The output signal is added with a pulse signal at a certain time
EP valve	External air leakage	Normal/ Mild/ Average/ Severe	The pressure values of the four brake cylinders are reduced
	Internal air leakage	Normal/ Mild/ Average/ Severe	The pressure values of the four brake cylinders increase and the exhaust time becomes longer
	Spring fault	Normal/ Mild/ Average/ Severe	The speed of gas-filled changes in the charging time of the brake cylinders
Brake cylinder	Air leakage fault	Normal/ Mild/ Average/ Severe	The corresponding brake cylinder pressure value decreases
Train pipe	Air leakage fault	Normal/ Mild/ Average/ Severe	The pressure values of the four brake cylinders are reduced

relies on sensor data to ensure precise control and distribution of braking force to the wheels. Sensor offset can lead to erroneous detection of wheel speed or other parameters, causing uneven distribution of braking force across the wheels. This can result in an imbalance in the vehicle during braking, affecting both the effectiveness of braking and overall stability. Brake cylinder fault: The realization of air braking ultimately depends on the gas pressure in the brake cylinder through which the basic braking device is propelled. However, after prolonged operation, tiny gaps inevitably develop between the piston and the housing. These tiny gaps allow the gradual infiltration of gas from the gas chamber, leading to a reduction in chamber pressure. As a result, the force applied to the basic braking device is also reduced, ultimately affecting the overall braking effect. The impact of other faults is briefly summarized in Table 1.

2.2. Multi-layer fault injection model base

In order to guarantee coverage of the diverse fault scenarios outlined above, it is essential for the fault injection system to encompass a maximal number of faults. If the foundation of the fault model relies on directly enumerating fault cases, four types of fault devices are involved, with a total of eight fault types for each device, and each type has three degrees of severity. Therefore, there are a total of 24 corresponding fault scenarios. Following this enumerative method would occupy a significant amount of storage resources. Moreover, generating a significant quantity of random fault cases during subsequent fault injection and diagnosis proves to be inconvenient.

To solve the above problems, Fang *et al.*[24] have created a fault model base for the train traction system with a multi-layer structure, organized into three layers. This paper establishes a fault model base for ABS. The first layer stores the fault components; the following layer encompasses the fault types, and the subsequent layer accommodates the fault degrees, denoted as $(\boldsymbol{SC}, \boldsymbol{ST}, \boldsymbol{SD})$. The first layer stores the fault components, and there are n kinds of fault components, which constitute the fault component set $\boldsymbol{SC} = \{s_{c1}, s_{c2}, s_{c3}, \ldots, s_{cn}\}$. The second layer stores fault types, including normal operation and fault states composed of k fault types, which form a fault type set $\boldsymbol{ST} = \{s_{t1}, s_{t2}, s_{t3}, \ldots, s_{tk}\}$. The third layer stores the fault degrees, including the fault state composed of normal and l fault degrees, which forms the fault degree state set $\boldsymbol{SD} = \{s_{d1}, s_{d2}, s_{d3}, \ldots, s_{dl}\}$. A comprehensive fault injection can be denoted as $(s_{ci}, s^i_{tj}, s^j_{dk})$.

Within the three-layer fault model base, real-world scenarios allow for clear classification of both fault devices and fault types. For example, fault devices may consist of train pipes, sensors, and similar components, whereas fault types may cover leakage faults, blockage faults, and so forth. However, the differentiation of fault severity demands distinctions based on the parameter ranges influencing the fault, spanning from mild to average to

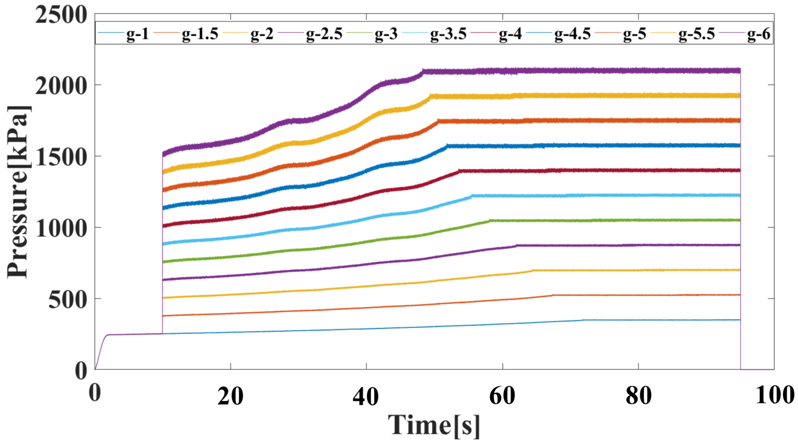


Figure 1. Sensor gain fault at different parameters.

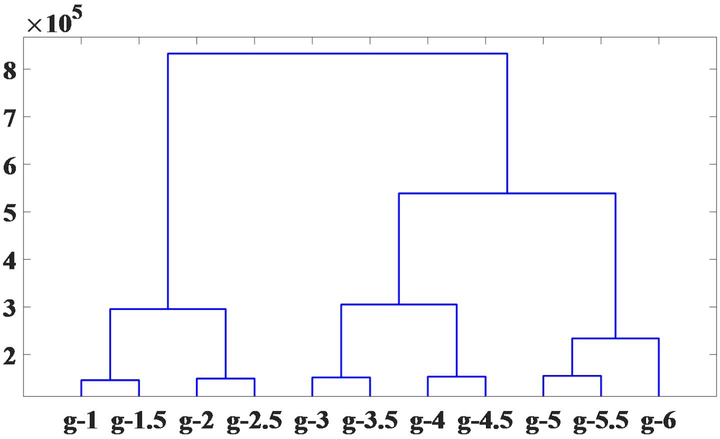


Figure 2. Cluster tree of sensor gain faults.

severe. This paper tackles this challenge by employing spectral clustering algorithms[26], which have become one of the most popular modern clustering algorithms. It is simple to implement and very often outperforms traditional clustering algorithms.

Using the gain fault of the sensor as an illustration, the fault curve is generated by selecting the gain coefficient parameter (ranging from 1 to 6 with a step size of 0.5), and the results are shown in Figure 1. Subsequently, the spectral clustering algorithm is applied for classification, and Figure 2 represents the dendrogram obtained from the classification. Due to the need to categorize the fault severity into three classes, the classification results are outlined as follows: minor (1-2.5), moderate (2.5-4.5), and severe (4.5, 6).

2.3. MC

MC[27] illustrates the interrelated transition process among different states once the system is initiated. Given the determination of the previous state of the system, the probability of the next state is independent of the previous state. In ABS, owing to the fault and repair of diverse components, there is a bidirectional transition between the normal and fault states at each level, with the probability of the next state contingent upon the previous state. As shown in Figure 3, the fault state transition in ABS can be viewed as a MC.

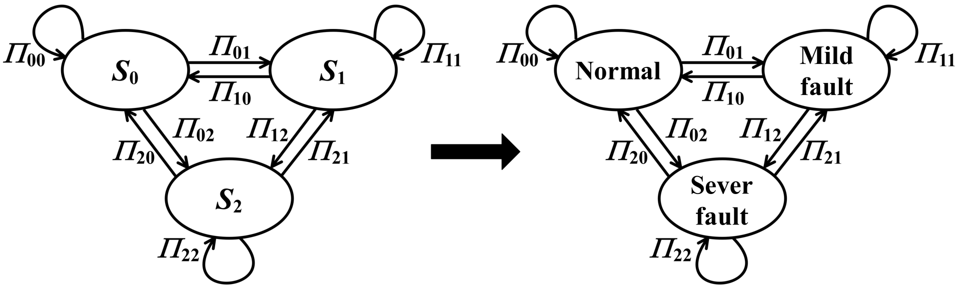


Figure 3. MC of fault. MC: Markov chain.

The MC possesses a distinct "memoryless" attribute, signifying that once the random variables in the nth step are known, those in the $(n+1)$th step are independent of others[28]. For the random variable set $\boldsymbol{X} = \{X_n : n > 0\}$ in the probability space with a one-dimensional countable set as the exponent set, the values of random variables are included in the countable set $\boldsymbol{S} = \{s_0, s_1, s_2, \dots, s_m\}$, which is called the state set. The "homogeneity" in a homogeneous MC is evident in the fact that the transition probability is solely connected to the state before and after the transition, rather than being dependent on the two time points. A homogeneous MC can be expressed as follows:

$$\text{Prob}\left(X_{n+1} \mid X_n, \dots, X_1\right) = \text{Prob}\left(X_{n+1} \mid X_n\right) \tag{1}$$

$\boldsymbol{\pi}_i$ is defined as its probability distribution and $\mathbf{\Pi}_{i,j}$ as its one-step transition probability. The mathematical expression is as follows:

$$\boldsymbol{\pi}_i = \text{Prob}(X_n = s_i), s_i \in \boldsymbol{S}. \tag{2}$$

$$\mathbf{\Pi}_{i,j} = \text{Prob}(X_{n+1} = s_j \mid X_n = s_i), \qquad s_i, s_j \in \boldsymbol{S}. \tag{3}$$

When all objects of a MC satisfy the following properties: recurrence, aperiodicity and pairwise connectivity, it has a unique steady-state solution[29,30], which is independent of the initial state, and can be expressed as[31]:

$$\lim_{n\to\infty} \text{Prob}\left(X_n = s_i\right) = \pi_i \tag{4}$$

The algorithm for solving the steady-state probability of MC is shown in Algorithm 1.

```
Algorithm 1: Steady-state probability of MC (SM)
Input: A state-transition matrix $M$.
Output: A steady-state probability matrix.
$Phase\_n = M$
$Phase\_n1 = Phase\_n1 * M$
while not $(Phase\_n == Phase\_n1)$.all( ) do
    $Phase\_n = Phase\_n1$
    $Phase\_n1 = Phase\_n1 * M$
Return the matrix: $Phas_n$.
```

3. LAYERED RANDOM FAULT INJECTION METHOD BASED ON MULTIPLE MCS

3.1. Fault probability distribution construction

In this paper, a multi-layer fault injection model base is established, which is divided into fault components, types and degrees. However, considering the background of ABS, the majority of components exhibit only a single type of fault. Therefore, the combination of fault components and types is called a fault type. In the

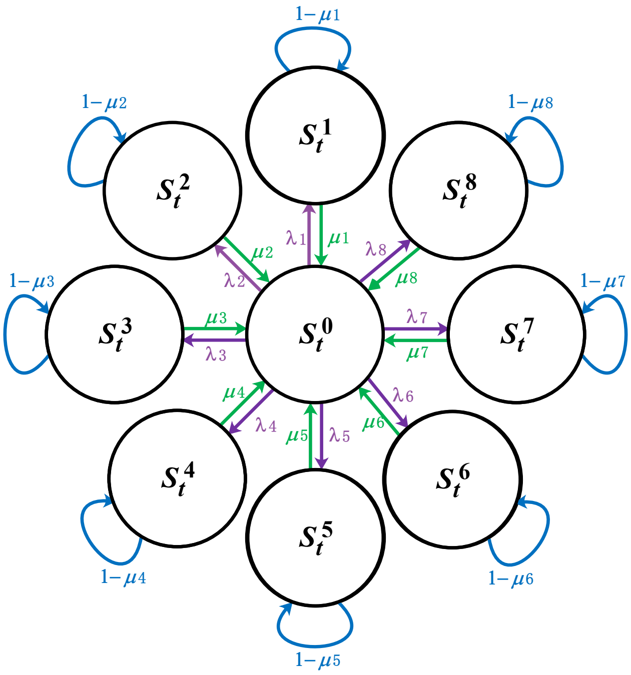


Figure 4. Fault type state transition diagram.

subsequent construction of a MC model, only the construction of fault type and fault degree MC is considered, which simplifies the model.

3.1.1 Fault type transfer MC model construction

When constructing the state transition diagram for fault types, the following conditions must be satisfied:
① When examining the state transition of an individual fault within the system, every fault state and the normal state should be mutually reachable and interconnected. However, distinct fault states should not be mutually reachable. Because the faults addressed in this manuscript are all individual faults and do not involve compound faults, that is, scenarios where two or more faults occur simultaneously. The state transition diagram must adhere to the following formula:

$$s_t^i \leftrightarrow s_t^0 \quad i = 1, 2, \ldots, k \tag{5}$$

② In the case of single fault state transfer, different fault states are unreachable, and the state transition diagram must adhere to the following formula:

$$s_t^i \nleftrightarrow s_t^j \quad i, j = 1, 2, \ldots, k, \; i \neq j \tag{6}$$

For fault type MC in Figure 4: there are k fault types except for the normal operation state. The state transition probability matrix $\boldsymbol{H_t}$ of the fault type MC can be constructed by using the steady-state probability distribution of the fault degree MC.

In Figure 4, λ_i(i=1,2,...,k) is defined as the probability of fault occurrence of the ith fault type, specifically, the probability of transition from state s_t^0 to state s_t^i. $\mu_i(i = 1, 2, ..., k)$ represents the probability of transition from

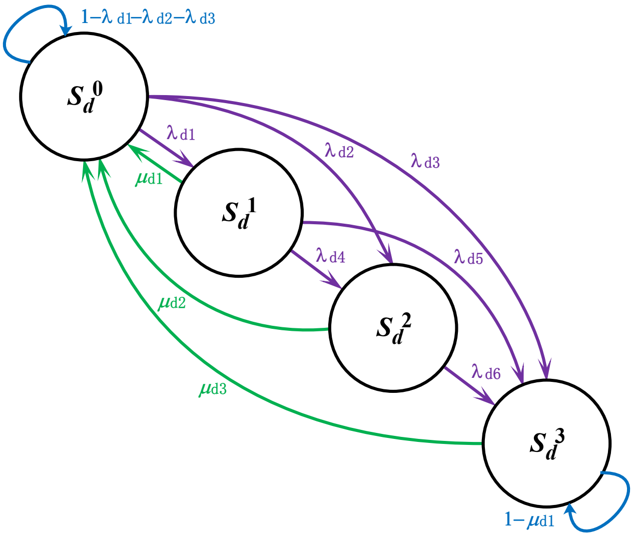


Figure 5. Fault degree state transition diagram.

state s_t^i to state s_t^0. The state transition probability matrix $\boldsymbol{H_t}$ of the fault type MC is constructed as follows:

$$\boldsymbol{H_t} = \begin{bmatrix} 1-\sum_{i=1}^{k}\lambda_i & \lambda_1 & \cdots & \lambda_k \\ \mu_1 & 1-\mu_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_k & 0 & \cdots & 1-\mu_k \end{bmatrix} \tag{7}$$

$\boldsymbol{\pi_t}$ is defined as the stationary probability distribution of MC with fault type, and $\pi_{t0}, \pi_{t1}, ..., \pi_{tk}$ are respectively the steady-state probability of normal operation state and k fault states.

$$\boldsymbol{\pi_t} = [\pi_{t0}, \pi_{t1}, ..., \pi_{tk}] \tag{8}$$

$$\boldsymbol{\pi_t}\boldsymbol{H_t} = \boldsymbol{\pi_t} \tag{9}$$

$$\sum_{i=0}^{k}\pi_{ti} = 1 \tag{10}$$

The specific value of $\pi_{ti}(i = 0, 1, ..., k)$ can be determined by simultaneously solving Eqs. (8)-(10), representing the steady-state solution. At this point, the steady-state probabilities corresponding to the aforementioned $k+1$ states have been acquired.

3.1.2 Fault degree transfer MC model construction

When constructing the state transition relationships for the fault degree state transition diagram, the following conditions must be satisfied:

① Without accounting for the repairability of the system, faults consistently progress in a more severe direction, and the state transition diagram must adhere to the following formula:

$$s_d^i \longrightarrow s_d^j \quad i, j = 1, 2, \ldots, l, \ i \leq j \tag{11}$$

② Taking into account the reparability of the system, any fault state can be directly restored to the normal state, and the state transition diagram must adhere to the following formula:

$$s_d^i \longrightarrow s_d^0 \quad i = 1, 2, \ldots, l \tag{12}$$

In Figure 5, $\boldsymbol{H_d}$ is defined as the state transition probability matrix of the fault degree MC. Using $l = 3$ as an illustration, the expression of $\boldsymbol{H_d}$ is as follows:

$$\boldsymbol{H_d} = \begin{bmatrix} 1-\sum_{i=1}^{3}\lambda_{di} & \lambda_{d1} & \lambda_{d2} & \lambda_{d3} \\ \mu_{d1} & 1-\gamma_{14}-\lambda_{d5} & \lambda_{d4} & \lambda_{d5} \\ \mu_{d2} & 0 & 1-\gamma_{26} & \lambda_{d6} \\ \mu_{d3} & 0 & 0 & 1-\mu_{d3} \end{bmatrix} \tag{13}$$

where $\gamma_{ij} = \mu_{di} + \lambda_{dj}$. $\lambda_{dj}(j = 1, 2, ..., 2l)$ represents the probability of transition from state s_d^{j-1} to state s_d^j, which is recorded as the fault rate. $\mu_{di}(i = 1, 2, ..., 2l)$ represents the probability of transition from state s_d^i to state s_d^0, which is recorded as the repair rate.

$\boldsymbol{\pi_d}$ is defined as the stationary probability distribution of MC with fault degree, and $\pi_{d0}, \pi_{d1}, ..., \pi_{dl}$ represent the steady-state probability of normal and l fault degree, respectively.

$$\boldsymbol{\pi_d} = [\pi_{d0}, \pi_{d1}, ..., \pi_{dl}] \tag{14}$$

$$\boldsymbol{\pi_d}\boldsymbol{H_d} = \boldsymbol{\pi_d} \tag{15}$$

$$\sum_{i=0}^{l}\pi_{di} = 1 \tag{16}$$

The specific value of $\pi_{di}(i = 1, 2, ..., l)$ can be obtained by solving Eqs. (14)-(16). The steady-state probabilities corresponding to the aforementioned four fault levels have been determined.

3.2. Random sampling of probability distribution

Upon acquiring the probability distribution of faults in ABS, efficiently simulating this distribution and conducting a substantial number of sampling tests have emerged as pivotal challenges.

3.2.1 Data processing

Random sampling from the calculated failure probability distribution may result in around 70% of the 1000 samples being in normal conditions, leading to significant resource wastage. Considering that faults are not introduced from the outset, the fault data inherently comprises both normal operational and fault data. Subsequently, following the acquisition of the steady-state probability distribution for fault types, the normal operational state can be eliminated, and a re-computation of the probability distribution can be conducted.

Eq. (8) and Eq. (14) are transformed into:

$$\boldsymbol{\pi_t} = [\pi_{t1}, \pi_{t2}, ..., \pi_{tk}]/(1-\pi_{t0}) \tag{17}$$

$$\boldsymbol{\pi_d} = [\pi_{d1}, \pi_{d2}, ..., \pi_{dl}]/(1-\pi_{d0}) \tag{18}$$

3.2.2 Random sampling method based on Alias algorithm

The Simple sampling (S-S) method shows a substantial discrepancy with the actual probability, particularly as the actual probability decreases. Additionally, each sampling operation in the S-S method has a complexity of O(K); even if the binary search method is used, the complexity is O(log(K)). Given the necessity for hundreds of samplings in a single test, a lower-complexity algorithm is required. The A-S algorithm[32] is a cutting-edge sampling algorithm that significantly mitigates the time complexity of sampling. Given a discrete random probability distribution $\text{Prob}(X = x_i) = p_i$, the A-S algorithm treats the entire probability distribution as N rectangles with an area of $1 * p_i$, and all events are converted into corresponding rectangular areas. Next, the area of each rectangle is normalized by $1/N$ to obtain N rectangles with an area of $N * p_i$. Fill the rectangles with an area of more than 1 into the rectangles with an area of less than 1, and record the details of the filling operation. Finally, N rectangles with a total area of $1 * 1$ are formed. Two arrays are constructed according to the original area and filling condition of the rectangle. One is the probability array, which is recorded as $prob$ and stores the original probability of the corresponding event of each rectangle. The other is an alias array. It stores the event number of the rectangle that is not filled enough, which is recorded as $alias$. If it is not filled, it is represented by $NULL$.

The algorithm constructed using the above method has a time complexity of O(K^2). A more efficient approach involves using two queues, A and B, to separately store node labels greater and less than 1. In each iteration, a node is selected from each queue, and the larger one is merged into the smaller one. The smaller node is then dequeued. Subsequently, the merged node is examined, and if its value is greater than 1, it is enqueued into queue A. If the value is equal to 1, it is dequeued. If the value is less than 1, it is enqueued into queue B. This algorithm achieves a time complexity of O(K). The algorithm for constructing the above A-S table and sampling process is shown in Algorithm 2. From the aforementioned process, it is evident that the A-S algorithm is designed to trade space for time. Despite having a space complexity of O(K), its time complexity is merely O(1). Given sufficient physical space, this results in a significant reduction in time complexity and an acceleration in sampling speed.

This section combines the model built in Section 2 with the fault model to extract specific faults hierarchically

Algorithm 2: Alias Sampling

```
Input: A list $P$ of length $n$, store the discrete distribution
Output: An integer in range $[0, n)$
Create arrays $Alias$ and $Prob$, each of size $n$
Create two queues, $Small$ and $Large$.
Multiply each probability by $n$.
if $p_i < 1$ then
    add $i$ to $Small$
else
    add $i$ to $Large$
while Small and Large are not empty do
    Remove the first element from $Small$, and call it $l$
    Remove the first element from $Large$, and call it $g$
    Set $Prob[l] = p_l$. And set $Alias[l]= g$
    Set $p_g := (p_g + p_l) - 1$
    if $pg<1$ then
        add $g$ to $Small$
    else
        add $g$ to $Large$
while Large is not empty do
    Remove the first element from $Large$, and call it $g$. And set $Prob[g] = 1$
while Small is not empty do
    Remove the first element from $Small$, and call it $l$. And set $Prob[l] = 1$
Generation a fair die roll from an n-sided die, call the side $i$
Flip a biased coin that comes up head with probability $Prob[i]$
if the coin comes up heads then
    return $i$
else
    return $Alias[i]$
```

from the fault model library according to sampling results, to achieve effective fault injection. The specific operational process is as follows:
① A layered fault injection model $(\boldsymbol{ST}, \boldsymbol{SD})$ for the ABS is established according to Section 2, storing collections of fault types and degrees separately;
② Users set the MC state transition matrix for fault types and degrees according to requirements and use it as input parameters; this matrix must meet certain conditions; for instance, the sum of elements in each row must equal 1;
③ The steady-state probability calculation module is responsible for calculating the steady-state probabilities of the MC state transition matrices for fault types and degrees and uses them as the fault probability distribution;
④ In the random fault generation module, the improved A-S algorithm is applied to randomly sample from the fault type probability distribution and the fault degree probability distribution, successively, obtaining (s_{ti}, s^{i}_{dj});
⑤ Upon receiving a model switching signal, the model switching module will switch the switch from the normal state model to the corresponding fault model, thus implementing the injection of random faults. The model switching signal refers to a signal when the system's randomly preset or user-defined fault injection time arrives.
Following these steps allows for the single instance of random fault injection into the ABS; repeating these steps can achieve multiple instances of random fault injections.

3.3. Overview

Most fault injections are deterministic, requiring users to manually set fault parameters for each injection. In contrast, random fault injection can more accurately simulate real-world fault scenarios, providing robust data support for subsequent fault detection and diagnosis research while reducing manual labor in extensive experiments. The prevalent method for implementing random fault injections involves fitting a large volume of raw sample data to obtain a distribution function that meets certain conditions. Then, based on this function, fault parameters are randomly sampled to achieve random fault injection. However, constructing a fault distribution function is a tedious process that requires data processing and parameter estimation. Additionally, a significant challenge in current research is the lack of raw fault data for ABSs, making it difficult to fit a fault distribution function comprehensively and accurately.

To address these issues, this chapter introduces a hierarchical random fault injection method based on multiple MCs. By leveraging the steady-state characteristics of MCs, users only need to provide the system's state transition matrix to construct a fault probability distribution, circumventing the need for complete raw data and tedious function fitting. Following the probability distribution acquisition, an improved A-S algorithm is proposed to sample the fault probability distribution accurately, efficiently, and with minimal repetition. This approach effectively resolves the challenges of constructing fault probability distributions and sampling fault probabilities, facilitating convenient and effective random fault injection into ABSs. The overall process is shown in Figure 6, and the algorithm is given in Algorithm 3.

4. SIMULATION AND ANALYSIS

4.1. Platform introduction

For the convenience of user verification, a fault injection and diagnosis platform for ABS is designed and implemented in this paper. The operational interface of this platform is illustrated in Figure 7, allowing the selection of position ② for deterministic and random fault injections. It incorporates the multiple MCs-based layered random fault injection method proposed in this paper. Click ② in Figure 7, select Random fault injection, and the random fault injection parameter setting interface, as shown in Figure 8, will pop up. ① Introduction describes the use process of the interface. ② Fault type state transition matrix setting on the left

Algorithm 3: Multiple MCs-based layered random fault injection method

```
Input: A fault type state-transition matrix $Mt$. A fault degree state-transition matrix $Md$. The sample
       size $n$
Output: An array of fault type sample results $Res\_t$. An array of fault degree sample results $Res\_d$
$Pt = \mathrm{SM}(Mt)$
$Pd = \mathrm{SM}(Md)$
for $i = 1$ to $n$ do
    $Res\_t(end + 1) = \mathrm{A\text{-}S}(Pt)$
for $i = 1$ to $n$ do
    $Res\_d(end + 1) = \mathrm{A\text{-}S}(Pd)$
Return the array of fault degree sample results $Res\_t$
Return the array of fault degree sample results $Res\_d$
```

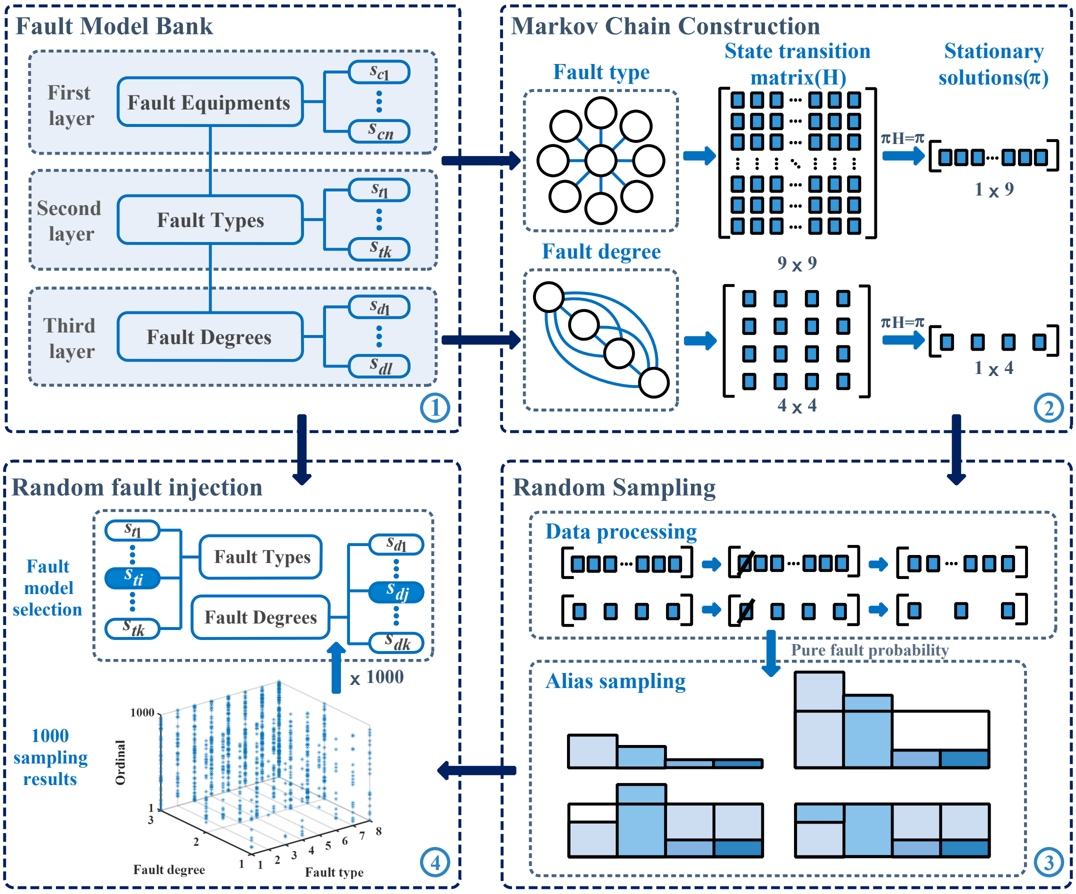
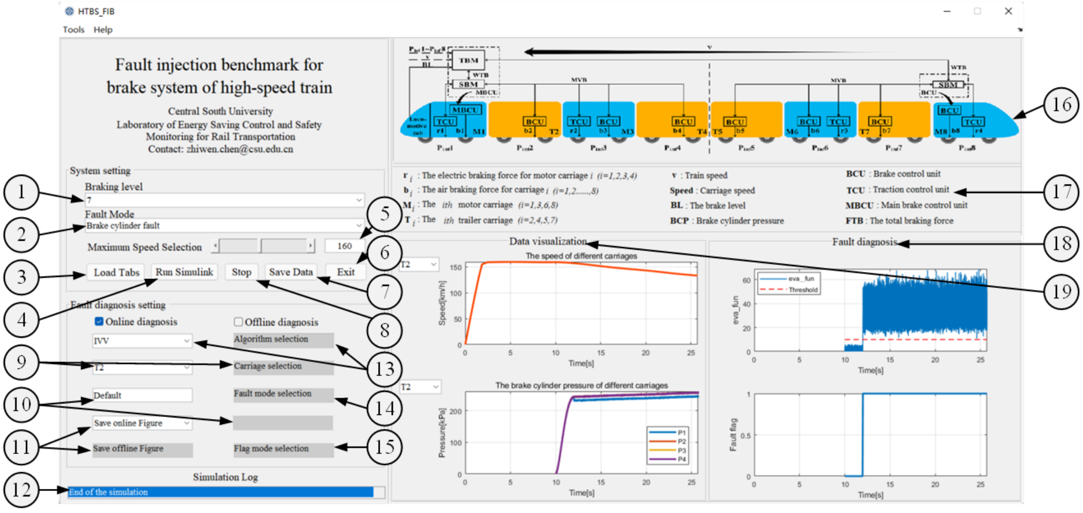


Figure 6. Multiple MCs-based layered random fault injection method. MC: Markov chain.

should be filled in. ④ Fault degree state transition matrix setting on the right should be filled in. Click OK after filling in; the platform will automatically calculate the steady-state solution of the MC according to the state transition matrix filled in by the user and display it in ③ Steady state probability of fault types and ⑤ Steady state probability of fault degrees below. Click ⑥ Start Fault Injection; the platform will automatically conduct random fault injection according to user settings, and the final simulation results will be displayed in the ⑱ data visualization module and ⑲ fault diagnosis module in Figure 7. In addition, for the specific operation process, please refer to the platform's help document for details, which is available at http://gfist.csu.



Fault injection benchmark for
brake system of high-speed train

Central South University
Laboratory of Energy Saving Control and Safety
Monitoring for Rail Transportation
Contact: zhiwen.chen@csu.edu.cn

System setting

Braking level: 7

Fault Mode: Brake cylinder fault

Maximum Speed Selection: 160

Load Tabs | Run Simulink | Stop | Save Data | Exit

Fault diagnosis setting

☑ Online diagnosis □ Offline diagnosis

IVV | Algorithm selection

T2 | Carriage selection

Default | Fault mode selection

Save online Figure

Save offline Figure | Flag mode selection

Simulation Log: End of the simulation

r_i : The electric braking force for motor carriage i $(i=1,2,3,4)$
b_i : The air braking force for carriage i $(i=1,2......,8)$
M_i : The ith motor carriage $(i=1,3,6,8)$
T_i : The ith trailer carriage $(i=2,4,5,7)$

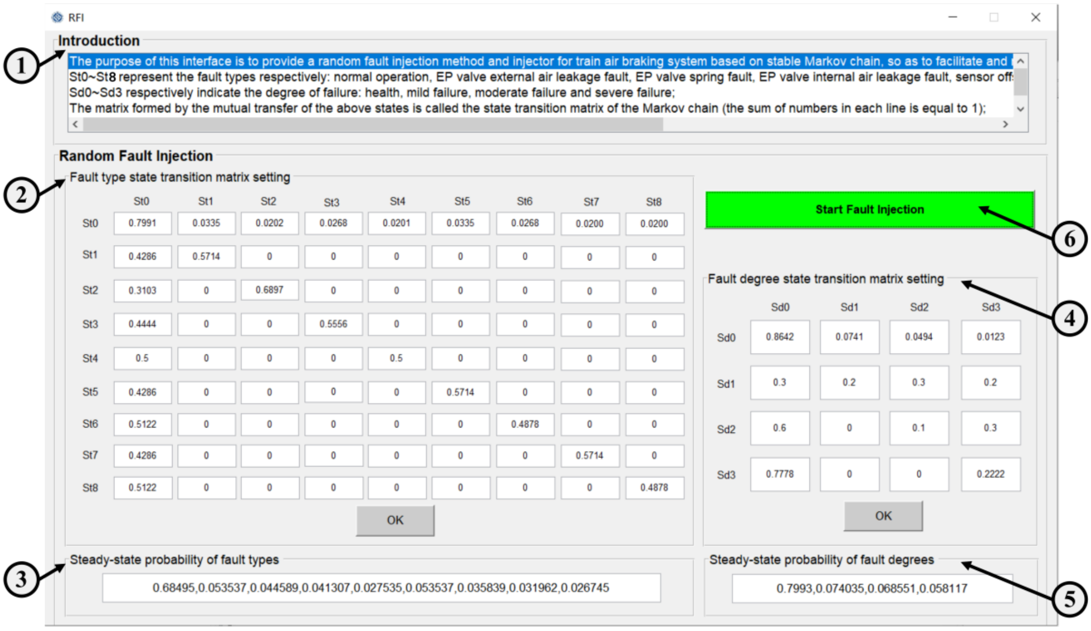
v : Train speed
Speed : Carriage speed
BL : The brake level
BCP : Brake cylinder pressure

BCU : Brake control unit
TCU : Traction control unit
MBCU : Main brake control unit
FTB : The total braking force

Data visualization

Fault diagnosis

Figure 7. EPBS_FIDMV main function model diagram. ①Set up the fault level. ②Set the fault type. ③Load form. ④Run the simulation. ⑤Set up the initial braking speed. ⑥Exit interface. ⑦Save observation point data. ⑧Stop the simulation. ⑨Select carriage. ⑩Display user-defined algorithm path. ⑪Save the online and offline fault diagnosis results. ⑫Lists simulation logs. ⑬Select fault diagnosis algorithms. ⑭Load offline fault data. ⑮Set the offline fault flag mode. ⑯Display train operation structure diagram. ⑰Introduce abbreviations. ⑱Display the online and offline fault diagnosis results. ⑲Display the data curve of the fault observation point.



Introduction

The purpose of this interface is to provide a random fault injection method and injector for train air braking system based on stable Markov chain, so as to facilitate and
St0~St8 represent the fault types respectively: normal operation, EP valve external air leakage fault, EP valve spring fault, EP valve internal air leakage fault, sensor off
Sd0~Sd3 respectively indicate the degree of failure: health, mild failure, moderate failure and severe failure;
The matrix formed by the mutual transfer of the above states is called the state transition matrix of the Markov chain (the sum of numbers in each line is equal to 1);

Random Fault Injection

Fault type state transition matrix setting

	St0	St1	St2	St3	St4	St5	St6	St7	St8
St0	0.7991	0.0335	0.0202	0.0268	0.0201	0.0335	0.0268	0.0200	0.0200
St1	0.4286	0.5714	0	0	0	0	0	0	0
St2	0.3103	0	0.6897	0	0	0	0	0	0
St3	0.4444	0	0	0.5556	0	0	0	0	0
St4	0.5	0	0	0	0.5	0	0	0	0
St5	0.4286	0	0	0	0	0.5714	0	0	0
St6	0.5122	0	0	0	0	0	0.4878	0	0
St7	0.4286	0	0	0	0	0	0	0.5714	0
St8	0.5122	0	0	0	0	0	0	0	0.4878

Fault degree state transition matrix setting

	Sd0	Sd1	Sd2	Sd3
Sd0	0.8642	0.0741	0.0494	0.0123
Sd1	0.3	0.2	0.3	0.2
Sd2	0.6	0	0.1	0.3
Sd3	0.7778	0	0	0.2222

Steady-state probability of fault types: 0.68495,0.053537,0.044589,0.041307,0.027535,0.053537,0.035839,0.031962,0.026745

Steady-state probability of fault degrees: 0.7993,0.074035,0.068551,0.058117

Figure 8. Fault injection and diagnosis platform of ABS. ①Introduce the operation process. ②Set the fault type state transition matrix. ③ Display steady-state probability of fault type. ④Set the fault degree state transition matrix. ⑤Display steady-state probability of fault degree. ⑥Start fault injection.

edu.cn/Download.html.

4.2. Simulation results and comparative analysis

One test in the above figure is taken as an example. The fault types include normal operation state, sensor deviation fault, sensor gain fault, EP valve external leakage, train pipe air leakage fault, and so on. The above states form the fault type state set $\boldsymbol{S_t} = \{s_t^0, s_t^1, s_t^2, s_t^3, s_t^4, s_t^5, s_t^6, s_t^7, s_t^8\}$ in turn. The fault degree in-

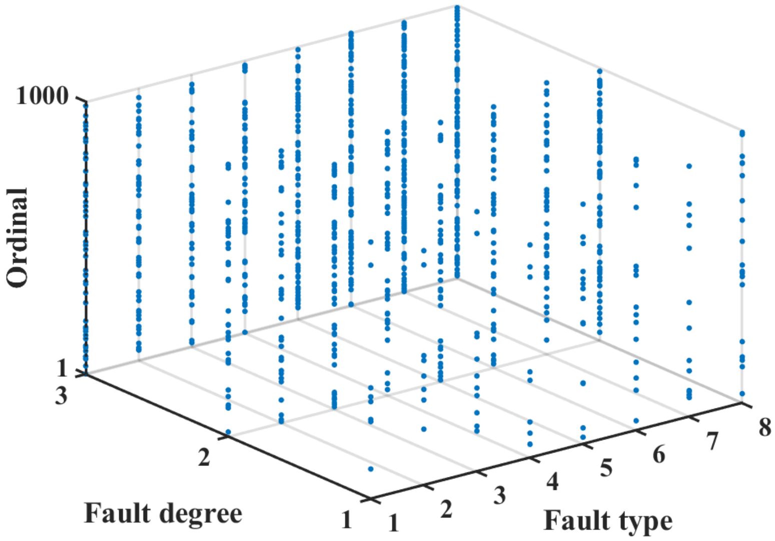


Figure 9. 1000 sampling results based on A-S.

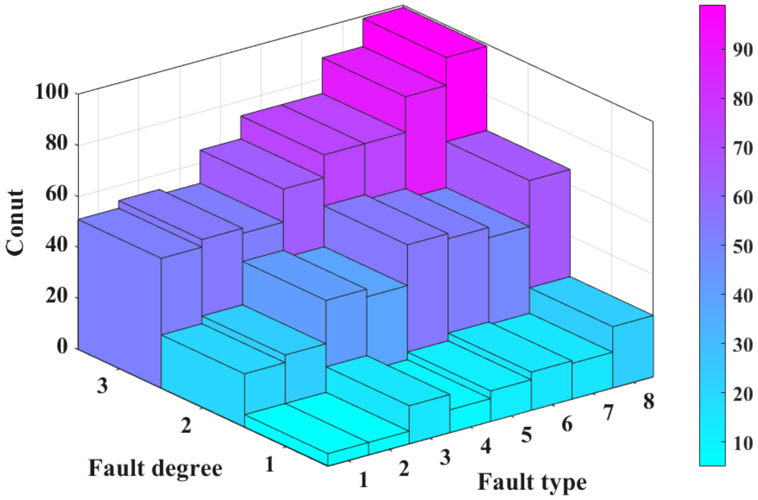


Figure 10. Frequency distribution histogram based on A-S.

cludes normal, minor, general, and severe. The above states, in turn, constitute the fault degree state set $\boldsymbol{S_d} = \{s_d^0, s_d^1, s_d^2, s_d^3, s_d^4, s_d^5, s_d^6\}$. For the fault type MC, the state transition probability matrix $\boldsymbol{H_t}$ of the fault type MC is constructed as follows:

Table 2. Frequency statistics of A-S and S-S

Fault type	Truth	Frequency statistics (1000times)		Frequency statistics (100times)	
		Alias	Simple	Alias	Simple
1	17%	17.00%	15.90%	17.00%	12.00%
2	14%	14.90%	12.70%	13.00%	16.00%
3	13%	13.50%	15.50%	15.00%	17.00%
4	9%	8.80%	9.80%	10.00%	5.00%
5	17%	16.80%	17.50%	15.00%	14.00%
6	11%	11.40%	11.20%	12.00%	9.00%
7	10%	10.00%	9.90%	8.00%	15.00%
8	9%	7.60%	7.50%	10.00%	12.00%

Table 3. Comparison of run time between A-S and S-S under different sample sizes

Sample size	100	500	1000	5000	10000	50000	100000
A-S time[ms]	0.627	1.274	1.492	2.131	2.518	6.827	12.187
S-S time[ms]	6.819	10.329	10.890	10.954	12.233	19.164	26.633

$$\boldsymbol{H_t} = \begin{bmatrix} 0.7991 & 0.0335 & 0.0202 & 0.0268 & 0.0201 & 0.0335 & 0.0269 & 0.0200 & 0.0200 \\ 0.4286 & 0.5714 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.3103 & 0 & 0.6897 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4444 & 0 & 0 & 0.5556 & 0 & 0 & 0 & 0 & 0 \\ 0.5000 & 0 & 0 & 0 & 0.5000 & 0 & 0 & 0 & 0 \\ 0.4286 & 0 & 0 & 0 & 0 & 0.5714 & 0 & 0 & 0 \\ 0.5122 & 0 & 0 & 0 & 0 & 0 & 0.4878 & 0 & 0 \\ 0.4286 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5714 & 0 \\ 0.5122 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4878 \end{bmatrix} \tag{19}$$

The specific value of π_{ti}(i=0,1,...,8) can be obtained by solving Eqs. (8)–(10) simultaneously: [0.68495, 0.053537, 0.044589, 0.041307, 0.027535, 0.053537, 0.035839, 0.031962, 0.026745], which is the steady-state solution.

For the MC of fault degree of EP valve solenoid valve leakage, the state transition probability matrix is expressed as follows:

$$\boldsymbol{H_d} = \begin{bmatrix} 0.8642 & 0.0741 & 0.0494 & 0.0123 \\ 0.3000 & 0.2000 & 0.3000 & 0.2000 \\ 0.6000 & 0 & 0.1000 & 0.3000 \\ 0.7778 & 0 & 0 & 0.2222 \end{bmatrix} \tag{20}$$

The specific value of π_{di}(i=0,1,2,3) can be obtained by solving Eqs. (14)-(16) simultaneously is [0.7993, 0.1141, 0.0685, 0.0181], which is the steady-state solution. After excluding the normal operational state, the probability distribution for pure fault types is recalculated as [0.1699, 0.1415, 0.1312, 0.0875, 0.1699, 0.1137, 0.1014, 0.0849]. Similarly, excluding the normal status, the probability distribution for pure fault degrees is recalculated as [0.5687, 0.3413, 0.0900]. These two probability distributions are subjected to 1000 tests simultaneously using the A-S algorithm. The sampling results are illustrated in Figure 9.

The graph displays three data columns on the fault degree axis, denoted as 1 - 3 representing three fault levels. Additionally, eight data columns on the fault type axis are labeled as 1 - 8 representing eight fault types. Observing the graph, it is evident that among the 1000 sampling points, the method achieves a uniform distribution of sampling points for fault types and degrees. No substantial clustering is observed, indicating a high degree of randomness. The fault type and degree are arranged in ascending order based on their probabilities, and the frequency statistics results are presented in Figure 10. It can be seen that the probabilities of various fault degrees and types are different. The frequency distribution histogram exhibits a step-like pattern, with lower probabilities associated with higher fault degrees, aligning with the anticipated expectations for the frequency

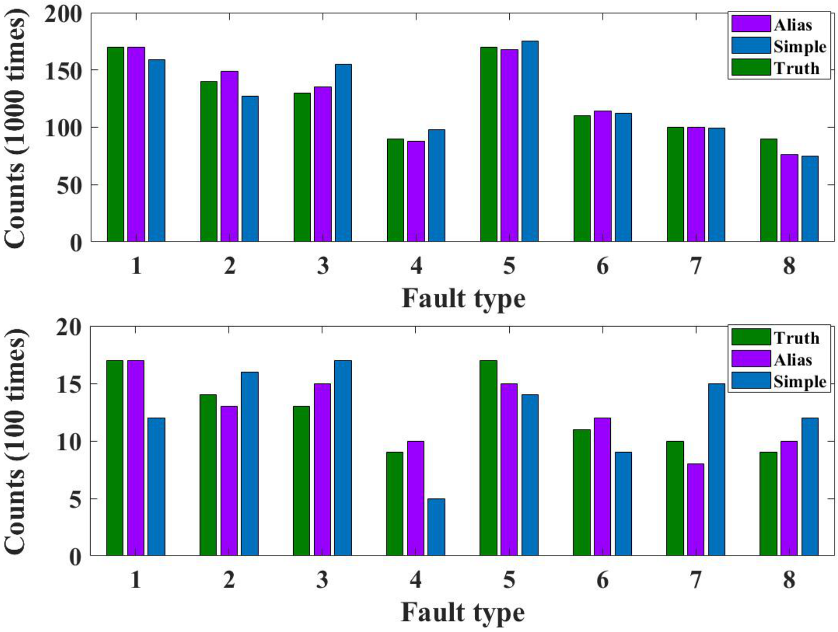


Figure 11. Frequency histogram of A-S and S-S.

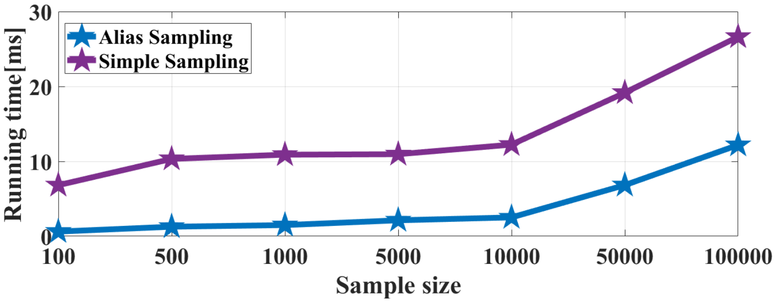


Figure 12. Run time comparison of A-S and S-S.

distribution based on fault degrees and types. The following compares A-S and S-S based on three aspects: accuracy, run time, and repetition rate.

4.2.1 Accuracy

The accuracy rate evaluation ensures that the random fault injection process closely resembles the real system situation and is more reliable. After 1000 and 100 samples, respectively, the fault type is sampled for frequency statistics and compared with the real value calculated based on MC, the frequency statistics table of A-S and S-S shown in Table 2 and the frequency histogram shown in Figure 11. The results indicate that with a large sampling volume (1000 times), the error between the frequency statistics and the actual values is small. However, with a smaller sampling amount (100 times), S-S exhibits greater errors than A-S, and the magnitude of the error amplifies with a decrease in sample probability.

4.2.2 Run time

Run time evaluation aims to minimize waiting time when injecting a large number of faults, enhancing efficiency. A-S and S-S undergo joint sampling of fault types and degrees, and the running time for sampling *n*

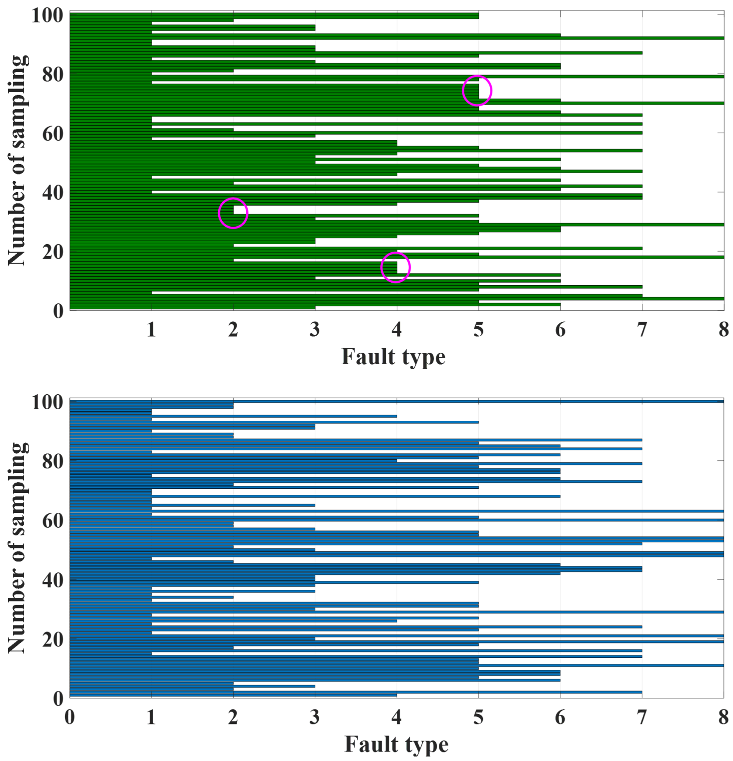


Figure 13. Repetition rate comparison of A-S and S-S.

times is computed. The results, as shown in Table 3 and Figure 12, indicate that A-S outperforms S-S in terms of sampling run time, enhancing the overall efficiency of random fault injection. This significantly contributes to reducing user waiting times and improving the overall user experience.

4.2.3 Repetition rate

The repetition rate evaluation ensures that redundant workload caused by multiple occurrences of the same fault can be avoided during small-scale random fault injection. The fault types are sampled 100 times by A-S and S-S, respectively, and the results of 100 sampling are shown in Figure 13. Any repeated sampling occurring three times or more is deemed unacceptable. The results reveal that both A-S and S-S exhibit two consecutive sampling repetitions. However, S-S has three consecutive sampling repetitions of three or more. While a high repetition rate may not influence fault injection and subsequent fault diagnosis, repeated faults can result in redundant and inconsequential fault diagnosis, leading to a waste of time and space resources.

Random fault sampling is performed using the A-S method described above, and the corresponding fault is injected into the ABS simulation model. The resulting speed curve and brake cylinder pressure curve are illustrated in Figure 14. The graph depicts that the high-speed train undergoes an acceleration phase from 0 km/h to 160 km/h within the time interval of 0 - 10 s, with the brake cylinder pressure remaining at 0 kPa. Between 10 - 105 s, the train gradually decelerates, reducing its speed from 160 km/h to 0 km/h. Throughout this process, the pressure within the brake cylinder steadily increases. In a fault-free operation of a high-speed train, the pressure across all four brake cylinders should be uniform. However, the P1 curve in the figure deviates from the overall trend, indicating a system failure during this period. To substantiate the injection of a fault into the system, Inter-variable Variance (IVV)[17,33] is employed. IVV is a statistical metric that

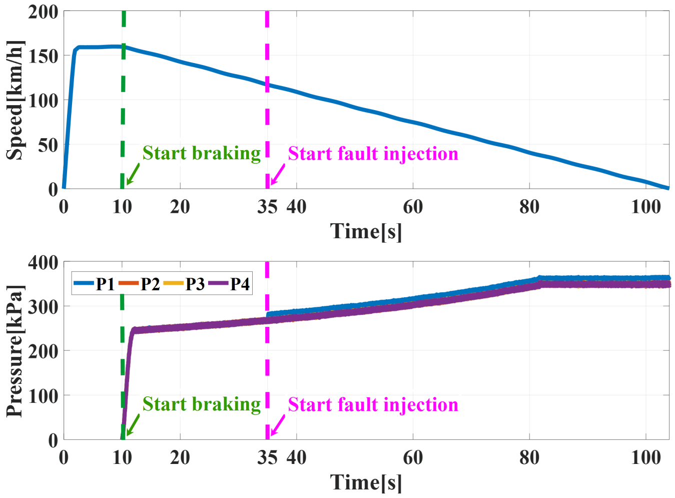


Figure 14. The speed and brake cylinder pressure of the train.

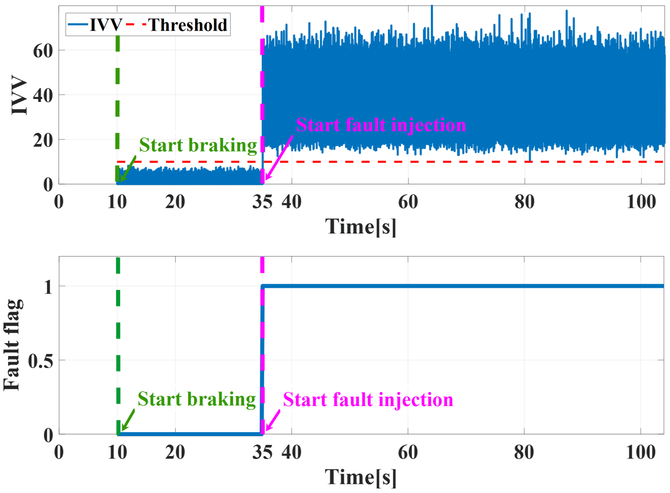


Figure 15. The result of fault diagnosis.

quantifies the variability or dispersion among different variables within a dataset. This measure is instrumental in gauging the extent of variance across the variables under examination, which is pivotal for tasks such as feature selection, elucidating the interrelations among variables, and in the realm of predictive modeling, where the independence of variables could influence the efficacy of the model. Utilizing IVV for fault detection during braking demonstrates a shift in the fault flag bit from 0 (indicating a no-fault state) to 1 (indicating a fault state), as illustrated in Figure 15.

5. CONCLUSIONS

This paper proposes a layered random fault injection method based on multiple MCs for ABS and provides a platform for fault injection and diagnosis. The random fault types and degrees are realized using MCs, in which the fault probability function is determined by the state transition matrix. The A-S algorithm is then employed for random sampling of the fault distribution, enabling efficient fault injection due to its low complexity. Additionally, a graphical user interface (GUI) is developed to present and visualize the validation results, including modules for the underlying simulation model, random fault injection, fault diagnosis, and visual display. The test results show that the proposed method can achieve more comprehensive and efficient random fault injection for ABS. The current version mainly includes the functions of random fault injection and fault diagnosis. In the future, the function of evaluating fault diagnosis algorithms will be added based on this, providing evaluation indicators suitable for ABS.

DECLARATIONS

Acknowledgments

The authors are grateful to the Guest Editors and anonymous reviewers for their constructive comments based on which the presentation of this paper has been greatly improved.

Authors' contributions

Writing-original draft and conceptualization: Chen Z, Fan J
Technical support: Luo H, Cheng C, Peng L
Validation and supervision: Chen Z

Availability of data and materials

Not applicable.

Financial support and sponsorship

This work was supported by the National Natural Science Foundation of China (Nos. U20A20186, 62173349), the Natural Science Foundation of Hunan Province (2022JJ20076), the Science and Technology Innovation Program of Hunan Province (2022RC1090), and the Key Laboratory of Energy Saving Control and Safety Monitoring for Rail Transportation under Grant (2017TP1002).

Conflicts of interest

All authors declared that there are no conflicts of interest. Zhiwen Chen is a Junior Editorial Board member in Complex Engineering Systems.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Yong Q, Hui M, Li-Min J. Development trend and active safety technology for advanced rail transit system. *China Railway* 2015;2015:77–81. DOI
2. Cheng C, Sun X, Song Y, et al. A just-in-time manifold-based fault detection method for electrical drive systems of high-speed trains. *Simu Model Prac Theor* 2023;127:102778. DOI
3. Wei W, Lin Y. Simulation of a freight train brake system with 120 valves. *Proc Inst Mech Eng Part F J Rail Rapid Transit* 2009;223:85–92. DOI
4. Wu Z, Huang NE. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Adv Adapt Data Anal* 2009;1:1–41. DOI

5. Su S, She J, Wang D, Gong S, Zhou Y. A stabilized virtual coupling scheme for a train set with heterogeneous braking dynamics capability. *Transp Res Part C: Emerg Technol* 2023;146:103947. DOI
6. Dong H, Ning B, Cai B, Hou Z. Automatic train control system development and simulation for high-speed railways. *IEEE Circuits Syst Mag* 2010;10:6–18. DOI
7. Wang T, Wang W, Zio E, Tang T, Zhou D. Analysis of configuration data errors in Communication-based Train Control systems. *Simul Model Pract Theory* 2019;96:101941. DOI
8. Yang C, Yang C, Peng T, Yang X, Gui W. A fault-injection strategy for traction drive control systems. *IEEE Trans Ind Electron* 2017;64:5719–27. DOI
9. Ding M, Chen B. Secure consensus control for multi-agent systems under communication constraints via adaptive sliding mode technique. *Complex Eng Syst* 2023;3:7. DOI
10. Abboush M, Knieke C, Rausch A. Representative real-time dataset generation based on automated fault injection and HIL simulation for ML-assisted validation of automotive software systems. *Electronics* 2024;13:437. DOI
11. Joshi A, Miller SP, Whalen M, Heimdahl MP. A proposal for model-based safety analysis. In: 24th Digital Avionics Systems Conference. vol. 2. IEEE; 2005. pp. 13–pp. DOI
12. Linnosmaa J, Pakonen A, Papakonstantinou N, Karpati P. Applicability of AADL in modelling the overall I&C architecture of a nuclear power plant. In: IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 2020, pp. 4337-4344 DOI
13. Liu T, Yu J, Sun WJ. Study on fault-tolerant technique based on knowledge modules of hydraulic fault theory. In: Advanced Materials Research. vol. 712. Trans Tech Publ; 2013. pp. 2043–50. DOI
14. Karpenko M, Sepehri N. Fault-tolerant control of a servohydraulic positioning system with crossport leakage. *IEEE Trans Control Syst Technol* 2004;13:155–61. DOI
15. Niksefat N, Sepehri N. Fault tolerant control of electrohydraulic servo positioning systems. In: Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148). vol. 6. IEEE; 2001. pp. 4472–77. DOI
16. Kiamanesh Bahareh BAOR. Fault Injection with Multiple Fault Patterns for Experimental Evaluation of Demand-Controlled Ventilation and Heating Systems. *Sensors* 2022;22:8180. DOI
17. Zhou D, Ji H, He X, Shang J. Fault Detection and Isolation of the Brake Cylinder System for Electric Multiple Units. *IEEE Trans Contr Syst Technol* 2018;26:1744-57. DOI
18. Gou B, Ge X, Wang S, et al. An open-switch fault diagnosis method for single-phase PWM rectifier using a model-based approach in high-speed railway electrical traction drive system. *IEEE Trans Power Electron* 2016;31:3816-26. DOI
19. Youssef AB, El Khil SK, Slama-Belkhodja I. State Observer-Based Sensor Fault Detection and Isolation, and Fault Tolerant Control of a Single-Phase PWM Rectifier for Electric Railway Traction. *IEEE Trans Power Electron* 2013;28:5842-53. DOI
20. Chen Z, Peng L, Fan J, et al. EPBSFIDMV: A fault injection and diagnosis methods validation benchmark for EPBS of EMU. *Contr Eng Pract* 2024;145:105873. DOI
21. Qin SJ. Survey on data-driven industrial process monitoring and diagnosis. *Annu Rev Control* 2012;36:220-34.. DOI
22. Huang M, Yin J, Yan S, Xue P. A fault diagnosis method of bearings based on deep transfer learning. *Simul Model Pract Theory* 2023;122:102659. DOI
23. He J, Lv Z, Chen X. Rolling bearing fault diagnosis method based on 2D grayscale images and Wasserstein Generative Adversarial Nets under unbalanced sample condition. *Complex Eng Syst* 2023;3:13. DOI
24. Fang D, Peng T, Yang C, Chen Z, Tao H. Random-Sampling-Based Performance Evaluation Method of Fault Detection and Diagnosis for Railway Traction System. In: 2019 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS). IEEE; 2019. pp. 570–74. DOI
25. Chen Z, Fan J, Peng L, et al. Multiple Markov chains-based layered random fault injection method for the air braking system. In: 2023 IEEE 2nd Industrial Electronics Society Annual On-Line Conference (ONCON), SC, USA, 2023, pp. 1-6. DOI
26. Von Luxburg U. A tutorial on spectral clustering. *Stat Comput* 2007;17:395-416. DOI
27. Mcginnis R, Isaacson DL, Madsen RW. Markov Chains Theory and Applications. *Contemp Sociol* 1977;6:322. DOI
28. Sericola GRB. Markov Chains and Dependability Theory. Cambridge: Cambridge University Press; 2014. DOI
29. Grassmann WK, Taksar MI, Heyman DP. Regenerative analysis and steady state distributions for Markov chains. *Oper Res* 1985;33:1107-16. DOI
30. Azimi S, Hassannayebi E, Boroun M, Tahmoures M. Probabilistic analysis of long-term climate drought using steady-state Markov chain approach. *Water Resour Manage* 2020;34:4703-24. DOI
31. Bod LWM. Introduction to probability theory and formal stochastic language theory. *Probabilistic Linguistic* 2003. DOI
32. Smith JC, Jacobson SH. An analysis of the alias method for discrete random-variate generation. *INFORMS J Comput* 2005;17:321-7. DOI
33. Ji H. Optimization-Based Incipient Fault Isolation for the High-Speed Train Air Brake System. *IEEE Trans Instrum Meas* 2022;71:1-9. DOI