A Conditional Invertible Neural Network-based Fault Detection

Wenxin Sun\textsuperscript{1}, Xudong Shi\textsuperscript{1}, Weili Xiong\textsuperscript{1}, Hongtian Chen\textsuperscript{1}, and Biao Huang\textsuperscript{1}

\textsuperscript{1}Affiliation not available

May 02, 2024

Abstract

Residual generation and hypothesis test are two important components in fault detection techniques. Recent studies mainly focused on enhancing residual generation algorithms, but often overlook the Gaussian distribution assumption that necessary for hypothesis test. Based on the conditional invertible neural network (CINN), this study proposes a novel approach for mapping residual signals into near-Gaussian distributed latent variables, thereby enhancing the reliability and effectiveness of the hypothesis test approaches used for fault detection. With the specially designed architecture using CINN, the proposed mapping from residual signals to latent variables has no information loss, thus guaranteeing the accuracy of the proposed fault detection method. The main contributions of this study are two-fold: 1) To ensure that the latent variables are distributed similarly to an ideal Gaussian distribution, a novel CINN training approach is proposed; 2) historical process information is incorporated into the residual-to-latent variable mapping, dynamically refining the mapping procedures in response to the system behavior. This approach is primarily used to tackle the challenges posed by non-additive and non-Gaussian noises in industrial systems. A DC speed control system and a waste water treatment system are adopted to verify the effectiveness of the proposed fault detection approach.
Abstract—Residual generation and hypothesis test are two important components in fault detection techniques. Recent studies mainly focused on enhancing residual generation algorithms, but often overlook the Gaussian distribution assumption that necessary for hypothesis test. Based on the conditional invertible neural network (CINN), this study proposes a novel approach for mapping residual signals into near-Gaussian distributed latent variables, thereby enhancing the reliability and effectiveness of the hypothesis test approaches used for fault detection. With the specially designed architecture using CINN, the proposed mapping from residual signals to latent variables has no information loss, thus guaranteeing the accuracy of the proposed fault detection method. The main contributions of this study are two-fold: 1) To ensure that the latent variables are distributed similarly to an ideal Gaussian distribution, a novel CINN training approach is proposed; 2) historical process information is incorporated into the residual-to-latent variable mapping, dynamically refining the mapping procedures in response to the system behavior. This approach is primarily used to tackle the challenges posed by non-additive and non-Gaussian noises in industrial systems. A DC speed control system and a waste water treatment system are adopted to verify the effectiveness of the proposed fault detection approach.

Index Terms—Fault detection, conditional invertible neural networks, latent variables, residual generation.

I. INTRODUCTION

Fault detection techniques are essential for modern process control systems. In the last two decades, the area of data-driven fault detection and diagnosis has seen significant advancements [1]–[3], driven by the increasing volume of data from industrial processes. Among these efforts, generation and analysis of residual signals are of central importance. Specifically, the residual signals are required to be both sensitive to system failures and robust against normal system noises. Recent studies highlight the effectiveness of combining deep learning-based residual generation [4]–[6] with statistical hypothesis test. Such a combined approach has proven effective in producing high-quality residual signals, enhancing the accuracy of fault detection. Until now, this combined approach remains the most widely-applied framework in the design of fault detection.

Fault detection techniques typically involve two sequentially linked components: the residual generation [7] and hypothesis test [8]. The residual generation calculates the residual signals from the process data, upon which the hypothesis test algorithm is built to detect system failures. Over the past two decades, there has been an intensive investigation to residual generation, utilizing a diverse range of techniques. For instance, subspace fault detection approaches [9], [10] employ linear mappings to isolate and eliminate the normal dynamic pattern of the system from the dataset, effectively producing a residual signal highly sensitive to system faults. Besides, the auto-encoder-based residual generations [11], [12] are specifically designed to capture the image space of the system. Consequently, they produce residual signals by evaluating the discrepancy between real-time data and the image space learned. Moreover, observer-based residual generations [13], [14] focus on determining the kernel space of the systems. Following the minimum variance or least squares principles, these approaches estimate the current system state, and generate residual signals by calculating the difference between the expected system outputs and the outputs measured.

In addition to residual generation, hypothesis test is a crucial aspect of fault detection techniques. It detects system failures by comparing the variations between residual signals generated offline and online. Currently, the $T^2$ and $Q$ test [15] stand as the predominant hypothesis test approaches in fault detection applications. Typically, these test approaches rely on the Gaussian distribution assumption to ensure optimal detection performance. However, the distribution of residual signals often receive insufficient attention during the design of residual generations. This gap can result in a misalignment between the residual generation and hypothesis test algorithms. To tackle this challenge, studies have been conducted towards two main strategies. 1) Developing detection algorithms that are adept for non-Gaussian issues. For instance, numerous studies employ advanced detection algorithms (e.g. kernel density estimation [16] and one-class support vector machines [17]) for hypothesis test. This approach is capable of adaptively learning the distribution of the residual signals, thereby increasing the robustness and accuracy of the detection results. 2) Establishing non-linear mappings that can transform non-
Gaussian residual signals into near-Gaussian distributed latent variables. This mapping procedure is commonly known as Gaussian distribution transformation. Recent studies often use exponential [18] or polynomial [19] functions to establish this transformation. However, both the strategies mentioned have a common limitation: they are inadequate for tackling the non-additive noise issues within industrial systems.

Currently, dealing with non-additive noise remains an open challenge in fault detection. Essentially, process data contains original signals and process noises. Original signals describe the true dynamic behavior of the system, while process noises are identically distributed and independent of the original signals. Systems affected by additive noise are generally easier to detect because the residual signals can be generated by simply subtracting the predicted original signals from the process data. Consequently, residual signals at each time instant follow a consistent distribution and are independent of the original signals. In contrast, the non-additive noise issue is challenging because the process noises affect the process data in a more complex way. The “non-additive” term indicates that basic addition or subtraction operations are not enough to effectively remove the original signals from the process data. At present, the existing residual generation techniques still face challenges in extracting residual signals that are independent of the original signals. Consequently, the state of the system can influence the distribution of related residual signals. As a result, existing test statistics remain inadequate for effective fault detection.

Based on a conditional invertible neural network (CINN) [20]–[22], this study proposes a novel Gaussian distribution transformation for addressing non-additive noise challenges. Initially, the specifically designed architecture of CINN ensures an information lossless mapping from the residual signals to latent variables, thus ensuring the reliability of the detection results. Additionally, the multi-layer structure of CINN provides enhanced flexibility in the mapping processes, improving the effectiveness of Gaussian distribution transformations. Furthermore, the delayed input-output data of the system is defined as the information vector. Recognizing that changes in the system state can affect the distribution of residual signals, the information vector is utilized to fine-tune the transformation process within CINN. This approach aims to make the distribution of latent variable in each time step closely mimics a uniform Gaussian distribution. To sum up, the main contributions of this study are as follows:

1) A neural network approach is provided for constructing the Gaussian distribution transformation. The uniquely designed network architecture offers substantial flexibility in the transformation process and guarantees a information lossless transformation from residual signals to latent variables.

2) In order to solve non-additive noise problems, lagged process information is incorporated into the Gaussian distribution transformation process. By adjusting the transformation process based on information vectors, the performance of hypothesis test is improved.

The organization of this paper is as follows: Section II details the fundamental framework of fault detection algorithms. Section III presents a CINN-based Gaussian distribution transformation, supported by numerical examples. Section IV validates the proposed CINN-based fault detection framework with simulations of DC speed control system monitoring and wastewater treatment process monitoring. Section V concludes this study.

II. FUNDAMENTAL FAULT DETECTION SCHEME

This study focuses on industrial processes characterized by the following equation

\[ y_t = \Phi(\varphi_t, w_t) + f_t, \]  \hspace{1cm} (1a)

\[ \varphi_t \triangleq \left[ y_{t-n_b:t-1}, u_{t-n_b:t} \right], \]  \hspace{1cm} (1b)

where \( \Phi \) represents the one-step transition function of the system, the column vector \( y_t \in \mathbb{R}^m \) represents the system outputs, and the vector \( f_t \in \mathbb{R}^m \) denotes the system failures. In (1b), the hyperparameters \( n_a \) and \( n_b \) are the orders of the system output and input, respectively. \( \varphi_t \) is the information vector that includes the lagged sequence of system input \( u_t \) and output \( y_t \). In many studies, dynamic models defined by (1) are also known as nonlinear autoregressive models with exogenous inputs. They are widely used to capture the dynamic behaviors of industrial systems.

A. Redundancy-based Residual Generation

The redundant measurement model \( \hat{\Phi} \), also known as a virtual sensor or analytical redundancy [23], is applied to predict the system output as \( \hat{y}_t = \hat{\Phi}(\varphi_t) \). The residual signal \( r_t \) is then obtained by comparing these predictions \( \hat{y}_t \) with the actual system outputs \( y_t \) measured by hardware devices,

\[ r_t = y_t - \hat{y}_t. \]  \hspace{1cm} (2)

Utilizing the process data \( \{\varphi_t, y_t\}_{t=1:n} \) which is sampled under normal operating conditions, the model \( \hat{\Phi} \) is typically trained by minimizing the cost function \( L \) below

\[ L = \frac{1}{n} \sum_{t=1}^{n} \| \hat{y}_t - y_t \|^2. \]  \hspace{1cm} (3)

When the data size \( n \) is sufficiently large, the cost function (3) can be approximated as follows [24]

\[ L \approx \mathbb{E}_{\varphi_t, y_t} \| \hat{y}_t - y_t \|^2 \]

\[ = \mathbb{E}_{\varphi_t} \| \hat{y}_t - \mathbb{E}(y_t | \varphi_t) \|^2 + \mathbb{E}_{\varphi_t} [\text{Var}(y_t | \varphi_t)]. \]  \hspace{1cm} (4)

The term related to \( \hat{\Phi} \) The term unrelated to \( \hat{\Phi} \)

Consequently, minimizing the cost function \( L \) effectively reduces the discrepancy between the predicted output \( \hat{y}_t \) and the expected system output \( \mathbb{E}(y_t | \varphi_t) \). In an ideal scenario, if the model \( \hat{\Phi} \) is trained to achieve optimal accuracy, such that

\[ \hat{y}_t - \mathbb{E}(y_t | \varphi_t) = 0, \]  \hspace{1cm} (5)

the residual generation (4) would serve as a kernel mapping for system (1) because of the following deductions,

\[ \{ f_t = 0 \} \implies \{ \mathbb{E}(r_t) = 0 \}, \]  \hspace{1cm} (6a)

\[ \{ f_t = 0, \ w_t = 0 \} \implies \{ r_t = 0 \}. \]  \hspace{1cm} (6b)
Here, (6a) depicts the normal operating conditions, and (6b) illustrates the noise-free normal operation conditions. As a result, the residual signal \( r_t \) becomes a key factor in detecting system failures. These residual signals are then subject to further analysis via hypothesis test, as shown in Fig. 1.

**B. Hypothesis Test for Fault Detection**

Define \( r^{\text{Train}}_t \) and \( r^{\text{Test}}_t \) as the residual signals generated from training and testing data, respectively. Hypothesis test methods are designed to detect system failures by comparing the test residual signal \( r^{\text{Test}}_t \) with the sequence of training residual signal \( r^{\text{Train}}_{1:n} \). The hypothesis \( \mathcal{H} \) defined below

\[
\mathcal{H} : \{ \mathbb{E} (r^{\text{Test}}_t) = \mathbb{E} (r^{\text{Train}}_t) , \tau = 1, 2, \ldots, n \} \tag{7}
\]

characterizes the normal operating conditions of the system. If the testing algorithm confirms the hypothesis \( \mathcal{H} \), the system is then considered to be operating under normal conditions. Otherwise, it is considered that a system failure has occurred. Define \( \mu \) as the mean value of \( r^{\text{Train}}_{1:n} \) below

\[
\mu \triangleq \frac{1}{n} \sum_{\tau=1}^{n} r^{\text{Train}}_\tau . \tag{8}
\]

The hypothesis test algorithm is typically established based on the following equations

\[
J_t = D (r^{\text{Test}}_t, \mu) , \tag{9a}
\]

\[
\begin{align*}
&\{ \text{Confirm the hypothesis } \mathcal{H} \}, & \quad J_t < J^\text{th} , \\
&\{ \text{Reject the hypothesis } \mathcal{H} \}, & \quad J_t \geq J^\text{th} ,
\end{align*}
\tag{9b}
\]

where \( J_t \) is the statistic, and \( D : r^{\text{Test}}_t \times \mu \rightarrow [0, +\infty) \) quantifies the discrepancy between the test residuals \( r^{\text{Test}}_t \) and the mean \( \mu \). Both the function \( D \) and the threshold \( J^\text{th} \) are established offline, using the training residual signal \( r^{\text{Train}}_{1:n} \). The threshold \( J^\text{th} \) is set according to the following principle

\[
p(J_t \geq J^\text{th} | \mathcal{H} \text{ holds true}) = \alpha , \tag{10}
\]

where \( \alpha \) represents a pre-defined significance level, typically set between 0.001 and 0.05. When both \( r^{\text{Test}}_t \) and \( r^{\text{Train}}_t \) follow a Gaussian distribution, it would be convenient to design a hypothesis test algorithm that aligns with the principle (10). Consider the \( T^2 \) test statistic as an example, which defines \( D \) and \( J^\text{th} \) as follows

\[
D (r^{\text{Test}}_t, \mu) = \frac{n - m}{m (n^2 - 1)} \| \mu - r^{\text{Test}}_t \|^2_{\Sigma^{-1}} , \quad J^\text{th} = F_{m, n-m-1}^{-1} (1 - \alpha) . \tag{11}
\]

\[
\Sigma = \frac{1}{n - 1} \sum_{t=1}^{n} (r^{\text{Train}}_t - \mu)(r^{\text{Train}}_t - \mu)^\top.
\]

Under the Gaussian distribution assumption, the principle described in (10) is valid, ensuring the reliability of the detection results.

Currently, most hypothesis test techniques still rely on the Gaussian distribution assumption. However, this assumption is not universally applicable in fault detection applications. In the system described by (1), the process noises affect the system dynamics in a non-additive manner. Consequently, the inherent system non-linearity may cause the distribution of residual signals to diverge from a Gaussian distribution. Furthermore, as indicated by equation (1a), the distribution of residual signals can also be influenced by the information vector \( \varphi_t \). As a result, residual signal \( r_t \) under each time instant follows a distinct non-Gaussian distribution determined by \( \varphi_t \). This study uses \( p(r_t | \varphi_t) \) to represent the distribution of the residual signal at each time interval.

**C. Gaussian Distribution Transformation**

Gaussian distribution transformation is a widely used technique for solving non-Gaussian and non-additive noise issues. It employs a specifically designed non-linear mapping \( f \) to transform a non-Gaussian residual signal \( r_t \) into a near-Gaussian distributed latent variable \( z_t = f_{\varphi_t} (r_t) \), as demonstrated in Fig. 2. This transformation serves as a bridge that connects residual generation and hypothesis test, replacing the red line depicted in Fig. 1.

Currently, two main challenges must be addressed when designing a Gaussian distribution transformation. Firstly, to ensure an information lossless transformation, the non-linear function \( f \) must be a bijective mapping. This requires that an inverse mapping \( f^{-1} \) exists, capable of reconstructing \( r_t \) from \( z_t \). Second, to tackle the non-additive noise problem mentioned, it is necessary to incorporate the information vector \( \varphi_t \) into the entire transformation process.

**III. CINN BASED GAUSSIAN DISTRIBUTION TRANSFORMATION**

This section presents a novel Gaussian distribution transformation utilizing CINN, which effectively addresses the two challenges outlined in Section II-C. Subsequently, a cost
A invertible layer

Invertible layers

Fig. 3. The model structure of CINN, i.e. \( z_t = f_{\varphi_t}(r_t) \).

A invertible layer

Invertible layers

Fig. 4. The inverse operation of CINN, i.e. \( r_t = f_{\varphi_t^{-1}}(z_t) \).

function is developed to quantify the discrepancy between the conditional distribution \( p(z_t | \varphi_t) \) and the ideal Gaussian distribution, thereby guiding the training procedures of CINN. Here, \( I \) denotes the identity matrix, and \( q(z_t) = N(0, I) \) signifies the target Gaussian distribution. The objective of the training process is to ensure that the conditional distribution \( p(z_t | \varphi_t) \) closely mimics the target distribution.

A. Model Structure

Typically, CINN is composed of multiple stacked invertible layers. To enhance the effectiveness of Gaussian distribution transformation, CINN in this study also incorporates a fully connected neural network \( S \) for scaling operations. This sub-section first outlines the structure of the invertible layers and then explains the logic behind integrating the network \( S \) into CINN. Each invertible layer is constructed using two single hidden layer neural networks, \( F \) and \( G \), as shown in Fig. 3. Besides, each invertible layer comprises a regular input, a conditional input (which feeds in \( \varphi_t \) in this study), and an output. Taking the first invertible layer as an example, the residual signal \( r_t = [r_{1,t}, r_{2,t}, \ldots, r_{m,t}] \) serves as the regular input, and \( h_t = [h_{1,t}, h_{2,t}, \ldots, h_{m,t}] \) denotes the output. The calculation procedure for this layer is detailed as follows

\[
\begin{align*}
\bar{h}_t &= r_t + F([r_t, \varphi_t]) \quad \text{h}_t = r_t + G([h_t, \varphi_t]),
\end{align*}
\]  

(13)

where \( r_t \) and \( h_t \) are defined below

\[
\begin{align*}
\bar{r}_t &= [r_{1,t}, r_{2,t}, \ldots, r_{m/2,t}], \\
\bar{h}_t &= [h_{1,t}, h_{2,t}, \ldots, h_{m/2,t}, \ldots, h_{m,t}],
\end{align*}
\]

(14)

and “\( \lfloor \cdot \rfloor \)” denotes the flooring function. The vectors \( \bar{h}_t \) and \( \bar{h}_t \) in (13) are defined according to the same logic as presented in (14). Consequently, the operation (13) is invertible because the residual signal \( r_t = [\bar{r}_t, \bar{r}_t] \) can be derived from \( h_t \) by using the following equations

\[
\begin{align*}
\bar{r}_t &= \bar{h}_t - F([\bar{h}_t, \varphi_t]) \quad \text{r}_t = h_t - F([h_t, \varphi_t]),
\end{align*}
\]

(15)

as is shown in Fig. 4. The inverse operation (15) ensures the information lossless property of each invertible layer.

Furthermore, based on the equation (13), the determinant of the Jacobian matrix \( \partial h_t^T / \partial r_t \) can be calculated as follows

\[
\begin{align*}
\det \left( \frac{\partial h_t^T}{\partial r_t} \right) &= \det \left( \frac{\partial [\bar{h}_t, \bar{h}_t]^T}{\partial [\bar{r}_t, \bar{r}_t]} \right) \\
&= \det \left( \frac{\partial [\bar{h}_t, \bar{h}_t]^T}{\partial [\bar{r}_t, \bar{r}_t]} \right) \times \det \left( \frac{\partial [\bar{r}_t, \bar{r}_t]^T}{\partial [\bar{r}_t, \bar{r}_t]} \right) \\
&= \det \left[ \begin{bmatrix} 1 & \frac{\partial h_t}{\partial h_t} \\ 0 & 1 \end{bmatrix} \right] \times \det \left[ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right] = 1.
\end{align*}
\]

(16)

By denoting the output of the last invertible layer as \( o_t \), the following equation holds

\[
\begin{align*}
\det \left( \frac{\partial o_t^T}{\partial r_t} \right) &= 1, \\
\det \left( \frac{\partial r_t^T}{\partial o_t} \right) &= 1.
\end{align*}
\]

(17)

The multi-layer structure of CINN offers high flexibility in achieving information lossless mapping. However, merely stacking invertible layers is insufficient to guarantee effective Gaussian distribution transformation. Specifically, the invertible layers are unable to control the entropy of \( o_t | \varphi_t \), as demonstrated below

\[
\mathbb{H}(o_t | \varphi_t) = - \int p(o_t | \varphi_t) \ln p(o_t | \varphi_t) \, do_t
\]

\[
= - \int p(r_t | \varphi_t) \ln \left( \frac{p(r_t | \varphi_t)}{\det(\partial r_t^T / \partial o_t)} \right) \, dr_t = \mathbb{H}(r_t | \varphi_t).
\]

(18)

Therefore, it is difficult to make \( p(o_t | \varphi_t) \) closely mimic a constant Gaussian distribution. To directly illustrate this limitation, a numerical case study is presented. First, the residual signals \( r_{1:5000} \) are generated as follows

\[
\begin{align*}
r_t &= (1 + w_{1,t}) (\sin(\varphi_t) + 0.1) \left[ \begin{array}{c} \cos(\varphi_t + w_{2,t}) \\ \sin(\varphi_t + w_{2,t}) \end{array} \right], \\
\varphi_t &\sim \mathcal{U}(-\pi, \pi), \quad w_t \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0.01 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0.25 \end{bmatrix}\right). \\
\end{align*}
\]

(19)

The residual signals conditioned on \( \varphi_t = 0.5 \), \( \varphi_t = 1.5 \), and \( \varphi_t = 3.0 \) are illustrated as red points in Figs. 5(a), 5(b), and 5(c), respectively. Following the operations by the trained invertible layers, the outputs \( o_t \) conditioned on various \( \varphi_t \) values are depicted by orange points in Figs. 5(d) through 5(f).

Clearly, invertible layers can control the pattern of \( p(o_t | \varphi_t) \), guiding it to closely mimic a Gaussian distribution. However, simply stacking several invertible layers is still insufficient. Conducting \( T^2 \) test based on \( o_{1:5000} \) can result in inadequate threshold boundaries (where \( J_t = J_{th} \)), as illustrated in Figs. 5(a) through 5(f).

The neural network \( S \) is constructed to overcome the limitation of the invertible layers. It is a multilayer neural network that takes \( \varphi_t \) as its input and produces a scalar output \( S(\varphi_t) \). Based on the neural network \( S(\varphi_t) \), the scaling operation is performed as follows

\[
z_t = o_t \times \exp \left[ S(\varphi_t) \right].
\]

(20)

Following this operation, the entropy \( \mathbb{H}(z_t | \varphi_t) \) can be controlled. As shown in Figs. 5(g), 5(h) and 5(i), the distribution \( p(z_t | \varphi_t) \) can mimic a constant Gaussian distribution across
various instances of the information vector $\varphi_t$, thereby enhancing the effectiveness of hypothesis test. Furthermore, the exponential function in (20) is applied prior to the production. This approach is applied to avoid the zero multiplier conditions. In summary, the CINN-based Gaussian distribution transformation $z_t = f\varphi_t(r_t)$ proceeds in two main stages: 1) The residual signal $r_t$ passes through the invertible layers to produce the output vector $o_t$; 2) the latent variable $z_t$ is derived by applying the scaling operation described in (20).

**B. Cost Function for Gaussian Distribution Transformation**

Initially, the discrepancy between $p(z_t|\varphi_t)$ and the target distribution $q(z_t)$ is quantified using the following KL divergence,

$$
\mathbb{E}_{\varphi_t}\left\{\text{KL}\left[p(z_t|\varphi_t)\|q(z_t)\right]\right\} \\
= \int p(\varphi_t) \int p(z_t|\varphi_t) \ln \frac{p(z_t|\varphi_t)}{q(z_t)} dz_t d\varphi_t \\
= -\mathbb{E}_{\varphi_t} \mathbb{H}(z_t|\varphi_t) - \mathbb{E}_{z_t} \ln q(z_t). 
$$

(21)

Based on (18) and (20), the term $\mathbb{H}(z_t|\varphi_t)$ in (21) can be derived as follows

$$
\mathbb{H}(z_t|\varphi_t) = mS(\varphi_t) + \mathbb{H}(y_t|\varphi_t), 
$$

(22)

where the term $\mathbb{H}(y_t|\varphi_t)$ represents a constant, unaffected by the residual generation model $\hat{\Phi}$ and CINN $f$. Thus, the cost function $L$ can be derived from the following deductions

$$
\begin{align*}
\mathbb{E}_{\varphi_t}\left\{\text{KL}\left[p(z_t|\varphi_t)\|q(z_t)\right]\right\} \\
\approx -\mathbb{E}_{\varphi_t} mS(\varphi_t) - \mathbb{E}_{z_t} \ln q(z_t) \\
\approx -\frac{1}{n} \sum_{t=1}^{n} \left[mS(\varphi_t) + \ln q(z_t)\right] \\
\approx \frac{1}{n} \sum_{t=1}^{n} \left[\frac{1}{2} \|f_{\varphi_t}(r_t)\|_2^2 - mS(\varphi_t)\right] \triangleq L. 
\end{align*}
$$

(23)

Therefore, minimizing the loss function $L$ effectively reduces the difference between $p(z_t|\varphi_t)$ and $q(z_t)$. Through this minimization process, the parameters of $S$ and those of $F$ and $G$ in each invertible layer are optimized simultaneously.

**IV. RESULTS**

This section provides two simulations to illustrate the benefits that CINN provides for the traditional fault detection methods. In each simulation, a variety of fault detection algorithms are evaluated. For residual generation, models such as the auto-regressive with extra inputs (ARX) model, neural network (NN), long short-term memory (LSTM), and gated
Fig. 6. Dynamic structure diagram of the DC speed control system.

The DC speed control system is utilized to validate the effectiveness of the fault detection methods mentioned. As demonstrated in Fig. 6, the entire system comprises the controller part and the motor-load part. Within the motor-load part, \( S_1 \) represents the PWM converter, which is affected by power fluctuations. Both the power fluctuations and the noise generated by the PWM conversion are considered as process noises, denoted by \( w_t \sim \mathcal{N}(0, 0.5) \), with the unit of volts (V). Besides, \( \beta = 0.0067 \) (V-min/r) and \( \gamma = 0.125 \) (V/l) represent the sensors that measure the motor speed and the rotor current, respectively. These sensors output the proportional voltage to the controller. Other components of the system are detailed in Table I. Additionally, the variables depicted in Fig. 6 are further explained in Table II. In this simulation, both recurrent unit (GRU) and the controllers (ASR and ACR) operate synchronously, executing 100 operations per second. The measurement noises \( v_{1,t} \) (A) and \( v_{2,t} \) (r/min) are both white noises following a Gaussian distribution \( \mathcal{N}(0, 1) \). The motor load within the system is characterized as follows

\[
\begin{align*}
 b_t &= 50 + 100 f_t, \\
 I_{t}^{\text{dl}} &= \begin{cases} \\
 \omega_t + b_t, & \omega_t > 0.01 b_t, \\
 100 \omega_t, & -0.01 b_t < \omega_t \leq 0.01 b_t, \\
 \omega_t - b_t, & \omega_t < -0.01 b_t.
\end{cases}
\end{align*}
\]

In this simulation, \( u_t \triangleq U_t^* \) is considered the input to the motor-load system, and \( y_t \triangleq [U_t^*, U_t^w] \) is the observed variable of the motor-load system. The model \( \Phi \) is configured with orders \( n_a = 5 \) and \( n_b = 5 \), which specifies \( \varphi_t \) as \( \{x_{-5:t-1}, u_{-5:t}\} \). The hyperparameters of the models related are detailed in Table III. During offline training, data are generated by setting

\[
\begin{align*}
 U_t^w &= 8 \sin (0.7 T t + x), \\
 f_t &= 0, \quad T = 0.01 \quad \text{specifies the sampling interval for data acquisition,} \\
 t &= 0, 1, \ldots, 20000. \quad \text{During online testing, both} \quad U_t^w \quad \text{and} \quad f_t \quad \text{are set as follows}
\end{align*}
\]

\[
\begin{align*}
 U_t^w &= 2 \sin (3 T t + x), \quad t < 1000, \\
 f_t &= 1, \quad t \geq 1000.
\end{align*}
\]

A. Monitoring DC Speed Control System

The DC speed control system is utilized to validate the effectiveness of the fault detection methods mentioned. As demonstrated in Fig. 6, the entire system comprises the controller part and the motor-load part. Within the motor-load part, \( S_1 \) represents the PWM converter, which is affected by power fluctuations. Both the power fluctuations and the noise generated by the PWM conversion are considered as process noises, denoted by \( w_t \sim \mathcal{N}(0, 0.5) \), with the unit of volts (V). Besides, \( \beta = 0.0067 \) (V-min/r) and \( \gamma = 0.125 \) (V/l) represent the sensors that measure the motor speed and the rotor current, respectively. These sensors output the proportional voltage to the controller. Other components of the system are detailed in Table I. Additionally, the variables depicted in Fig. 6 are further explained in Table II. In this simulation, both

---

TABLE I

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>( U_t^w ) (V)</td>
<td>PWM converter input</td>
<td>( U_t^w ) (V)</td>
</tr>
<tr>
<td>( E_1 ) (V)</td>
<td>Back electromotive force</td>
<td>( I_d^f ) (A)</td>
</tr>
<tr>
<td>( U_t^i ) (V)</td>
<td>Current measurement</td>
<td>( I_{d}^i ) (A)</td>
</tr>
<tr>
<td>( U_t^r ) (V)</td>
<td>Speed measurement</td>
<td>( I_{d}^{*} ) (V)</td>
</tr>
<tr>
<td>( v_{1,t} ) (V)</td>
<td>Load current</td>
<td>( v_{2,t} ) (r/min)</td>
</tr>
</tbody>
</table>

---

TABLE II

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( U_t^w ) (V)</td>
<td>Speed setpoint</td>
</tr>
<tr>
<td>( E_1 ) (V)</td>
<td>Back electromotive force</td>
</tr>
<tr>
<td>( U_t^i ) (V)</td>
<td>Current measurement</td>
</tr>
<tr>
<td>( U_t^r ) (V)</td>
<td>Speed measurement</td>
</tr>
<tr>
<td>( v_{1,t} ) (V)</td>
<td>Load current</td>
</tr>
<tr>
<td>( v_{2,t} ) (r/min)</td>
<td>System failures</td>
</tr>
</tbody>
</table>

---

FPR = \( \frac{\text{Number of normal samples misclassified}}{\text{Total number of normal samples}} \),
FNR = \( \frac{\text{Number of faulty samples misclassified}}{\text{Total number of faulty samples}} \),
Accuracy = \( \frac{\text{Number of samples correctly detected}}{\text{Total number of samples}} \).
where \( t = 0, 1, \cdots, 3200 \). To further demonstrate the effectiveness of CINN, both traditional and CINN-based fault detection approaches use the same \( \Phi \) models, avoiding separate training for each.

The detection results of the fault detection methods are summarized in Table IV. As demonstrated, CINN effectively enhances the accuracy of the detection methods in the majority of scenarios. When incorporating CINN, each detection method mentioned achieves an accuracy exceeding 95%. To further demonstrate the effectiveness of CINN, Figs. 7 and 8 present the distributions of residual signals and latent variables, respectively. Here, the residual signals are generated using LSTM model. These residual signals are then directly fed into CINN model to produce the latent variables. The black lines in the figures represent the threshold boundaries established by \( T^2 \) test with a significance level set at \( \alpha = 0.05 \). Clearly, the distribution of residual signals significantly deviates from a Gaussian standard, thereby reducing the effectiveness of hypothesis test algorithms. In contrast, CINN effectively transforms these residual signals into latent variables that are nearly Gaussian-distributed, as shown in Fig. 8. Subsequently, the hypothesis test algorithms achieve higher accuracy.

### B. Monitoring Wastewater Treatment Process

The benchmark simulation model no. 1 (BSM1) [25], which is developed by the international water association, is also utilized in this section. This wastewater treatment system includes five biochemical reaction units and a secondary clarifier, as depicted in Fig. 9. The five reaction unites in the figure consist of two anaerobic units and three aerobic unites. The first two anaerobic units mainly perform denitrification, breaking down large organic molecules into small organic compounds. The subsequent three aerobic units primarily carry out nitrification reactions, decomposing small organic compounds into inorganic compounds. Afterwards, part of the wastewater is recirculated back to the first reaction unite, while another portion enters the secondary clarifier for solid-liquid separation.

However, the decay process of heterotrophic organisms may produce toxic substances. These toxic substances can reduce microbial activity in the reaction unites, potentially leading to a reduction in microbial activity and posing a threat to the system. In each reactor, the occurrence of toxic diffusion faults is determined by two parameters: the heterotrophic bacteria growth rate coefficient \( \mu_H \) and the heterotrophic decay coefficient \( b_H \). Under normal operating conditions, \( \mu_H \) typically equals 4.0 and \( b_H \) equals 0.3. This experiment considers four different types of system faults, as illustrated in Table V. To detect failures in the BSM1 system, this simulation tracks the concentrations of thirteen substances in both the influent and effluent. These measurements are represented as \( \mathbf{u}_t = [\mathbf{u}_{1,t}, \cdots, \mathbf{u}_{13,t}]^\top \) for the influent and \( \mathbf{y}_t = [\mathbf{y}_{1,t}, \cdots, \mathbf{y}_{13,t}]^\top \) for the effluent, as is illustrated in Fig. 9. The process variables related are detailed in Table VI.

Both the training and testing data are gathered at a frequency of one sample every 15 minutes. Each dataset contains 1343 samples. The training dataset is generated under normal operating conditions. When generating testing dataset, a system failure occurs after the process has been operating normally for

<table>
<thead>
<tr>
<th>Model</th>
<th>Model structure</th>
<th>Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARX</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>GRUs</td>
<td>2 20</td>
<td>0.1 2000</td>
</tr>
<tr>
<td>LSTM</td>
<td>2 20</td>
<td>0.1 2000</td>
</tr>
<tr>
<td>NN</td>
<td>2 20 Leaky Relu</td>
<td>0.001 5000</td>
</tr>
<tr>
<td>CINN</td>
<td>2 20 Leaky Relu</td>
<td>0.01 5000</td>
</tr>
</tbody>
</table>

The number of hidden layers in each model, and the number of invertible layers in CINN; the number of nodes in each hidden layer, and the number of hidden layer nodes contained in \( \mathcal{F} \) and \( \mathcal{G} \) within each invertible layer; the activation function.
Comparative Analysis of Detection Outcomes Across Various Models and Algorithms (DC speed control system detection simulation). The contents of the table are FPR/FNR/Accuracy, omitting the % symbol.

<table>
<thead>
<tr>
<th>Models</th>
<th>Significance level</th>
<th>Based on traditional framework</th>
<th>Based on CINN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Q test</td>
<td>T² test</td>
</tr>
<tr>
<td>ARX</td>
<td>α = 0.050</td>
<td>6.665</td>
<td>2.530</td>
</tr>
<tr>
<td>NN</td>
<td></td>
<td>7.58 /2.940</td>
<td>15.22</td>
</tr>
<tr>
<td>LSTM</td>
<td></td>
<td>5.92 /3.966</td>
<td>18.01</td>
</tr>
<tr>
<td>GRU</td>
<td></td>
<td>21.01</td>
<td>8.922</td>
</tr>
<tr>
<td>ARX</td>
<td>α = 0.010</td>
<td>2.18 /2.742</td>
<td>4.23</td>
</tr>
<tr>
<td>NN</td>
<td></td>
<td>1.79 /2.346</td>
<td>4.03</td>
</tr>
<tr>
<td>LSTM</td>
<td></td>
<td>0.93 /1.976</td>
<td>10.42</td>
</tr>
<tr>
<td>GRU</td>
<td></td>
<td>5.22 /7.965</td>
<td>32.71</td>
</tr>
<tr>
<td>ARX</td>
<td>α = 0.001</td>
<td>0.295 /7.340</td>
<td>0.04</td>
</tr>
<tr>
<td>NN</td>
<td></td>
<td>0.19 /3.328</td>
<td>0.14</td>
</tr>
<tr>
<td>LSTM</td>
<td></td>
<td>0.05 /0.966</td>
<td>4.13</td>
</tr>
<tr>
<td>GRU</td>
<td></td>
<td>0.93 /4.974</td>
<td>24.71</td>
</tr>
</tbody>
</table>

10050 minutes (equivalent to 670 sampling cycles). The model orders are set as n₁₅ = 3 and nₛ = 3. The hyperparameters associated are detailed in Table VII, and the significance level α is set as 0.01.

An example is demonstrated by combining NN-based residual generation with the T² test algorithm. Using the fault detection method depicted in Fig. 1, the detection results are illustrated in Fig. 10. Additionally, Fig. 11 demonstrates the detection results of the proposed fault detection method, which uses CINN to connect NN-based residual generation with the T² test algorithm. In these figures, the blue line represents the T² statistic $J_t$, and the red line denotes the threshold $J^0$. As shown, the CINN-based fault detection method provides a more distinct separation between normal and fault conditions, resulting in higher accuracy than the traditional fault detection approach. Furthermore, the results of other fault detection methods are detailed in Table VIII. With Gaussian distribution transformation using CINN, most methods achieved higher accuracy. As illustrated in the right half of Table VIII, all CINN-based fault detection methods achieved accuracies exceeding 90%. In contrast, the traditional fault detection methods may yield poorer detection performances.

V. CONCLUSION

In traditional fault detection techniques, it is crucial for residual signals to follow a constant Gaussian distribution. However, the presence of non-Gaussian and non-additive noises can disrupt this requirement, thereby diminishing the effectiveness of fault detection methods. To tackle this issue, this study introduces a novel fault detection technique based on CINN. This technique uses CINN to transform non-Gaussian distributed residual signals into nearly Gaussian distributed latent variables, thereby enhancing fault detection performance. Moreover, the specially designed architecture of CINN ensures a transformation of residual signals to latent variables without any loss of information, thus guaranteeing the reliability of the detection results. As demonstrated by two simulation experiments in this study, the proposed CINN method effectively transforms residual signals into latent variables with near-Gaussian distributions, thereby enhancing the accuracy of traditional fault detection methods.

REFERENCES

Fig. 11. The performance of the proposed fault detection approach, which uses NN to generate residual signals and utilizes $T^2$ test for detection.

<table>
<thead>
<tr>
<th>Models</th>
<th>Fault type</th>
<th>Based on traditional framework</th>
<th>Based on CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$Q$ test</td>
<td>$T^2$ test</td>
</tr>
<tr>
<td>ARX no. 1</td>
<td>1.2/24.1/87.3</td>
<td>8.1/0.0/99.6</td>
<td>0.7/64/97.3</td>
</tr>
<tr>
<td>ARX no. 2</td>
<td>1.0/72.5/63.1</td>
<td>8.7/0.0/99.5</td>
<td>0.4/64/76.3</td>
</tr>
<tr>
<td>ARX no. 3</td>
<td>1.2/199.3</td>
<td>8.7/0.0/95.7</td>
<td>0.9/0.0/99.6</td>
</tr>
<tr>
<td>ARX no. 4</td>
<td>1.2/399.3</td>
<td>8.4/0.0/95.8</td>
<td>1.5/0.1/99.2</td>
</tr>
<tr>
<td>NARX no. 1</td>
<td>20.5/10.9</td>
<td>13.3/0.1/93.3</td>
<td>20.5/10.9</td>
</tr>
<tr>
<td>NARX no. 2</td>
<td>1.3/79.9</td>
<td>10.0/0.0/98.5</td>
<td>1.2/0.1/99.8</td>
</tr>
<tr>
<td>NARX no. 3</td>
<td>1.2/19.9</td>
<td>8.4/0.0/95.8</td>
<td>1.5/0.1/99.2</td>
</tr>
<tr>
<td>NARX no. 4</td>
<td>24.1/0.8</td>
<td>15.4/0.0/92.3</td>
<td>22.0/0.0/98.9</td>
</tr>
<tr>
<td>LSTM no. 1</td>
<td>0.6/47.5/67.0</td>
<td>3.6/0.0/98.2</td>
<td>1.2/0.1/99.8</td>
</tr>
<tr>
<td>LSTM no. 2</td>
<td>1.0/63.2/67.8</td>
<td>3.4/0.0/98.3</td>
<td>1.2/0.2/97.8</td>
</tr>
<tr>
<td>LSTM no. 3</td>
<td>1.0/97.3/50.7</td>
<td>3.1/0.0/98.1</td>
<td>1.2/0.6/97.3</td>
</tr>
</tbody>
</table>


K. Jiang, S. Hong, M. Kim, J. Na and I. Moon, “Adversarial Autoencoder...


