TU/e EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Eindhoven University of Technology,

Department of Electrical Engineering,

Control System Group

Fault Detection and Diagnosis Using the Dynamic Network Framework

Master Thesis

Yibo Shi

Assessment committee

Chair:	prof. dr. ir. P. M. J. Van den Hof
Member 1:	dr. V. Breschi
Member 2:	dr. ir. K. Tiels
Advisory member 1:	prof. dr. S. Weiland
Advisory member 2:	ir. S. J. M. Fonken

Graduation

Program:	System and Control
Supervisor 1:	prof. dr. ir. P. M. J. Van den Hof
Supervisor 2:	ir. S. J. M. Fonken
Student ID:	1730738
Date of defense:	October 25, 2023

EINDHOVEN
UNIVERSITY OF
TECHNOLOGY

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

10-17-2023………………………………………………..…………..

Name

Yibo Shi…………………………………………………..…………..

ID-number

1730738…………………………………………………..…………..

Signature

Yibo Shi

……………………………………………………………..…………..

Submit the signed declaration to the student administration of your department.

[i] See: https://www.tue.nl/en/our-university/about-the-university/organization/integrity/scientific-integrity/
The Netherlands Code of Conduct for Scientific Integrity, endorsed by 6 umbrella organizations, including the VSNU, can be found here also. More information about scientific integrity is published on the websites of TU/e and VSNU

Fault Detection and Diagnosis Using the Dynamic Network Framework

Yibo Shi
Control Systems Group
Eindhoven University of Technology
y.shi3@student.tue.nl

***Abstract*—This thesis proposes a local model-based method for fault detection and diagnosis (FDD) within large-scale interconnected network systems, using the dynamic network framework. The classical auto- and cross-correlation methods are adapted to validate the model of the local system, ensuring an accurate model for the subsequent FDD procedure. Employing the developed model validation tests, an algorithm based on combinatorial optimization is developed to generate the optimal model validation test for local fault detection. The model validation tests also serve as the basis for developing a fault diagnosis algorithm that facilitates accurate fault locating. Additionally, the proposed fault diagnosis approach specifies the necessary location of external excitation signals within the local system to achieve such accurate fault locating. In summary, this research offers a robust, scalable solution that addresses FDD challenges in interconnected network systems.**

***Index Terms*—Faults, Interconnected systems, Dynamic network, Model validation, correlation test**

I. Introduction

Large-scale interconnected network systems have become ubiquitous in modern society, from power grids [1] to distributed control systems [2], communication networks [3], and more. These systems are designed to connect different subsystems and components to achieve specific goals, such as the efficient transfer of power, data, or goods. However, these systems are also prone to faults, which can have significant consequences, including reduced efficiency, increased costs, and even catastrophic failures. Faults in interconnected network systems can be caused by a wide range of factors, including equipment failures, communication errors, environmental factors, and cyberattacks [4], [5]. Detecting and diagnosing these faults is essential to maintaining the reliability and performance of these systems. However, the interconnection of multiple subsystems and the sheer complexity of these systems make fault detection and diagnosis a challenging task.

To address this challenge, this research aims to develop techniques for *fault detection and diagnosis (FDD)* in interconnected network systems. As the first phase of FDD, *Fault detection (FD)* involves determining whether the behavior of a system deviates from its normal state [4]. The detection of faults is crucial to ensure that a system performs its functions effectively and to prevent system breakdown. *Fault diagnosis*, on the other hand, follows fault detection and entails identifying the fault's location, type, and severity. It can also involve the implementation of measures to rectify the fault. The primary objective of fault diagnosis is to restore the system to its normal state as quickly and effectively as possible [4], [6].

FDD is often carried out using three main types of methods, data-driven methods, knowledge-based methods, and model-based methods [6]. Data-driven methods employ training and learning to forge a representation of system behavior and analyze historical data from the systems to detect patterns indicating the presence of faults [6]. The neural network [7], pattern recognition [8], and fuzzy logic [9] methods are illustrative of the data-driven FDD methods [10]. Knowledge-based approaches utilize statistical methods that often employ graphical models to indicate potential cause-effect relationships of faults. These methods include dependency graphs [11], Petri nets [12], Bayesian networks [13], fault trees [14], and binary decision trees [15]. Model-based methods depend on a mathematical system model for monitoring purposes. This model is employed to simulate the system behavior, and deviations from the anticipated behavior are intended to be detected and potentially diagnosed as faults. The model-based methods rely on the consistency check between the predicted and measured time series data, using tools such as parameter estimation, observers, and analytical redundancy [5], [16].

For large-scale interconnected network systems, FDD is more challenging due to complex structures and the large amount of data. At the same time, the presence of loops in these network systems amplifies the impact of faults and makes it more challenging to accurately locate and diagnose them. Despite the existence of various data-driven and model-based FDD approaches, each with its advantages and limitations, none of them currently is sufficient to detect and diagnose all types of faults in a complex interconnected system [6], [17]. Data-driven AI approaches are evidenced as a viable way to address this challenge, demonstrating the increasing sophistication of current machine learning systems [6]. However, it relies heavily on the availability and quality of training data, which can limit their effectiveness in detecting and diagnosing faults when data is scarce or imbalanced [17], and there was no discussion of explainable AI, where the decisions made by algorithms could be justified [6]. The model-based approach incorporates the physical understanding of the system's interconnections, making it possible to accurately reflect the topology of the network system. Additionally, it is also regarded as a viable approach for tackling the challenge of FDD in large-scale interconnected network systems. Nevertheless, there are also evident challenges associated with employing this approach. Calibrating models for large-scale systems to align with real systems is more challenging [7], meanwhile, the scalability of models and FDD methods has to be ensured which is crucial for their rapid and reliable application to systems of varying scales [18].

In recent years, there has been a growing interest in utilizing data-driven modeling for interconnected networks within the field of system identification, employing the *dynamic network framework*. The term 'dynamic networks' typically denotes large-scale spatially interconnected systems, composed of directed interconnections of transfer functions or modules represented as links or edges, and the interconnecting signals represented as vertices or nodes. In the case of an interconnected network system, this dynamic network model can illustrate the input–output behavior between each pair of observed signals, as well as the system's spatial structure, referred to as the system topology. Research on data-driven modeling for interconnected systems using the dynamic network framework has yielded several methods that consistently estimate a multi-input-multi-output (MIMO) model of the entire network system [19] and a multi-input-single-output (MISO) model of a local subnetwork within the system [20]–[23]. Furthermore, the study [24] employs an empirical Bayes method to reduce the computational cost of the data-driven modeling procedure for a local MISO subnetwork, ensuring the scalability of the modeling methods. These studies within the dynamic network framework offer opportunities to develop scalable and distributed model-based FDD methods based on local subnetworks, thus addressing the complexity of FDD in large-scale interconnected network systems. Consequently, the central research question in this thesis is: What procedure do we need to follow to perform local model-based fault detection and diagnosis for an interconnected network system, using its model in the dynamic network framework?

In this thesis, we focus on local model-based FDD within an interconnected network system, relying on MISO models of specific local parts. Throughout this thesis, these specific local parts are referred as *target MISO subnetworks*. Consequently, to achieve comprehensive FDD across the entire network system, one can execute FDD for each MISO channel within the system.

The MISO models of the original healthy systems can be obtained through estimation procedure using local identification methods [20]–[23], or other feasible modeling approaches such first principle modeling or *empirical transfer function estimate (ETFE)*. If the acquired MISO model is not sufficiently accurate, denoting the presence of model errors, the subsequent model-based FDD procedure could be compromised, potentially leading to inaccurate detection and diagnosis [25]. One strategy for mitigating or even nullifying the effects of model error on FDD, as described in [25], is to use a robust residual generator. This can generate data that are hardly affected by model errors, which facilitates the subsequent FDD procedure. However, this technique often requires significant computational resources and complicated threshold adjustments. An alternative approach to circumventing the effects of model error is to evaluate the accuracy of the model before it is used for subsequent application; this evaluation process is often referred to as *model validation*. Once the model is validated, the influences of model errors in the FDD process are considered negligible, meaning that any anomalies discovered later are not due to model errors. For complicated and large-scale network systems, it is essential to avoid excessive computation and complicated threshold tuning during the FDD process.

Therefore, in this study, we begin by performing model validation for the target MISO network, to facilitate subsequent FDD procedures. However, the study on model validation within the dynamic network framework is relatively limited in the current literature. In my previous internship work, I demonstrated that classical standard auto- and cross-correlation tests [26] could be adapted to the dynamic network framework. Nevertheless, the performance of those standard correlation tests is suboptimal in the dynamic network framework, and conservative assumptions on the network systems are required at the same time. Additionally, there are other validation tools for the open- and closed-loop systems in current literature [27], [28] that can be generalized to the dynamic network framework. Consequently, the first objective of our research is to develop a validation technique for local dynamic network scenarios that outperforms conventional auto- and cross-correlation tests in flexibility, stability, and efficiency.

Following the successful validation of the local MISO subnetwork model, it becomes imperative to develop a model-based FDD procedure based on the dynamic network framework. Model-based FDD operates on the principle of model invalidation. By utilizing an accurate model of a fault-free system, one can assess whether new data continues to validate this model. If the model remains validated by the updated data, it suggests the system is still in a healthy state. Conversely, invalidation may indicate potential faults, possible with information for further diagnosis. For model-based FDD, the two primary techniques for model invalidation are parameter estimation and residual analysis [25]. The parameter estimation method uses a reference model for a healthy system and re-identifies the local subnetwork based on updated data. Deviations from the reference model serve as the basis for fault detection and further diagnosis [25]. However, this approach demands heavy computational resources and conservative conditions on the input excitation signals, and is therefore not explored in this study. The residual analysis method compares the system measurements to analytically computed values from the model, the discrepancy between them is indicative of the presence of faults in the system [25]. Relative to parameter estimation, residual analysis has fewer requirements on excitation and computation sources, making it better suited for large-scale network systems. By organizing the subsequent study on the FDD into the fault detection phase and the diagnosis phase, we set the second objective of this study to develop a fault detection approach based on residual analysis.

In interconnected network systems, structural complexity significantly complicates fault diagnosis, which makes these systems vulnerable to faults that can propagate through their interconnected structure. Following the detection of a fault in the target MISO subnetwork, pinpointing the origin of the fault becomes the primary task for fault diagnosis that leads to subsequent system maintenance. Therefore, the third objective of our study is to develop a fault diagnosis procedure using residual analysis within the target MISO subnetwork, aiming to accurately determine the root cause of the detected fault.

The paper begins by laying the preliminary knowledge of dynamic networks and faults in engineering systems in Chapter II. In Chapter III, we present the developed model validation tests for local subnetworks, along with numerical illustrations of test performance. Chapter IV introduces our model-based fault detection methods, supported by the algorithms we have developed. Chapter V addresses additional algorithms tailored to model-based fault diagnosis, focusing on localizing the fault's root cause with minimal scope. The proposed model-based FDD procedure is illustrated numerically in Chapter VI. Concluding remarks and future research directions are provided in Chapter VII.

II. Preliminaries

A. Dynamic network

Following the setup as in [20], a dynamic network model is built up of L node signals $w_j(t),\ j \in \mathcal{L}$ with $\mathcal{L} = [1,\ L]$ the index set of all node signals. Then the network can be written in an MISO structure as:

$$w_j(t) = \sum_{k \in \mathcal{N}_j} G^0_{jk}(q) w_k(t) + R^0_j r_j(t) + v_j(t), \qquad (1)$$

where q^{-1} is the delay operator, i.e. $q^{-1}u(t) = u(t-1)$;

- $G^0_{jk}(z)$ is a rational module transfer function which is referred to as a module in the dynamic network, representing a direct causal connection from w_k to w_j;
- $\mathcal{N}_j$ is the set of indices of node signals $w_k,\ k \neq j$, for which $G^0_{jk} \neq 0$, representing the set of indices of measured signals with direct causal connections to w_j, as referred as the *in-neighbors* of node w_j [20];
- r_j is an external excitation signal which is quasi-stationary and can be manipulated by users [29];
- R^0_j is a binary selection variable, $R^0_{jl} = 1$ indicates the excitation signal $r_j(t)$ is present and $R^0_j = 0$ indicates $r_j(t)$ is not present;
- v_j is a process noise, where the vector process $v = [v_1 \cdots v_L]^\top$ is modeled as a stationary stochastic process with rational spectral density, such that there exists a p-dimensional independent white noise process $e := [e_1 \cdots e_p]^\top$, $p \leq L$, with covariance matrix $\Lambda^0 > 0$ and variance of each innovation source $\sigma^2_{e_i} > 0,\ i \in [1, p]$. The proper rational transfer function matrix $H^0(q)$ such that $v(t) = H^0(q)e^0(t)$.

We will assume that the standard regularity conditions on the data are satisfied that are required for the convergence of the prediction error identification method and the convergence in distribution[1].

Omitting $q,\ t$ and combining the L node signals, the full dynamic network can be written in a single matrix equation:

$$\begin{aligned}
\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} &= \begin{bmatrix} 0 & G^0_{12} & \cdots & G^0_{1L} \\ G^0_{21} & 0 & \ddots & G^0_{2L} \\ \vdots & \ddots & \ddots & \vdots \\ G^0_{L1} & G^0_{L2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} \\
&\qquad + R^0 \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_L \end{bmatrix} + H^0 \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_p \end{bmatrix} \\
&= G^0 w + R^0 r + H^0 e \\
&= \left(I - G^0\right)^{-1} \left(R^0 r + H^0 e\right).
\end{aligned} \qquad (2)$$

Additionally, the noise transfer function matrix H^0 satisfies the following:

- When $p = L$, referred to as the full-rank noise case, H^0 is square, stable, monic and minimum-phase [19], [20];
- When $p < L$, referred to as the singular or rank-reduced noise case, $H^0 \in \mathbb{R}^{L \times p}(z)$ is stable and has a stable left inverse $H^\dagger$ that satisfies $H^\dagger H = I \in \mathbb{R}^{p \times p}$ [19], [23].

The dynamic networks and their components considered in this research are further restricted by the following assumption.

Assumption 1. *We consider a dynamic network with the additional properties that,*

a. All module transfer functions in the network are strictly proper and stable;
b. The network is well-posed in the sense that all principal minors of $(I - G^0(\infty))$ are nonzero [31];
c. $(I - G^0)^{-1}$ is proper and stable;

[1] See [26] page 249 and [30] Lemma B.3. This includes the property that the noise signals $e(t)$ and applied excitation signals $r(t)$ have finite forth-order moments.

d. All node signals w are nonzero and no nodes are noise-free;

e. Each node can only be directly influenced by one innovation source.[2]

Assumption 1 guarantees the properness and stability of the mappings from the network's external signals r and v to its internal signals w. Furthermore, based on item e that all node signals w are nonzero, the assumption ensures that each node gets passed through by at least one of the existing external signals r and v. In this study, for the model validation in the dynamic network framework, the internal node signals do not necessarily require sufficient excitation by the external signals r and v, they only need the nonzero assumption of item e.

B. Network and noise topology

To streamline the discussion of network interconnection structure (topology) in subsequent chapters, we introduce the following definitions. To specify the network topology, we utilize a directed graph that captures both the locations and causal directions of module transfers within the network. This graph can be mathematically represented by a binary matrix, $\mathcal{T}_G \in \mathbb{R}^{L\times L}$, as defined in [20]:

$$\begin{aligned}\mathcal{T}_G(j,i) &= 0, \text{ if } G^0_{ji} \equiv 0;\\ \mathcal{T}_G(j,i) &= 1, \text{ elsewhere.}\end{aligned} \tag{3}$$

Because of the interconnection structure that we consider as in Eq. (1) it follows that $\mathcal{T}_G(i,i) = 0, \forall i \in \mathcal{L}$. In this study, the network topology $\mathcal{T}_G$ of the data-generating system is assumed to be known.

Similarly, we define another adjacency matrix $\mathcal{T}_H \in \mathbb{R}^{L\times p}$ to represent the noise topology of the network as follows:

$$\begin{aligned}\mathcal{T}_H(j,i) &= 0, \text{ if } H^0_{ji} \equiv 0;\\ \mathcal{T}_H(j,i) &= 1, \text{ elsewhere.}\end{aligned} \tag{4}$$

To streamline the utilization of information derived from the network and noise topology, we further introduce the subsequent set definitions.

Definition 1. *We define the following sets of node indices based on the network topology $\mathcal{T}_G$ and noise topology $\mathcal{T}_H$:*

- *Set $\mathcal{C}_i$ is defined as the set of node indices k for which either a directed path through modules exists from w_k to w_i, or from w_i to w_k with $k \neq i$, or both w_k and w_i are influenced by the same innovation source. The index of the node w_i itself will always be in $\mathcal{C}_i$. With respect to correlation, the set $\mathcal{C}_i$ encompasses all node signals that exhibit correlation with w_i;*
- *Set $\mathcal{J}_i$ is defined as the set of node indices k where a directed path through modules exists from w_i to w_k and the index of the node w_i itself will also be in $\mathcal{J}_i$. Notably, set $\mathcal{J}_i$ is a subset of $\mathcal{C}_i$ by definition, i.e. $\mathcal{J}_i \subset \mathcal{C}_i$;*
- *Set $\mathcal{V}_j$ is defined as the set of node indices k for which an innovation source e_j that directly influences w_j has a path to w_k*[3].

To clarify the defined notations, consider Fig. 1, where the MISO subnetwork centered around the output node w_3 serves as our target subnetwork. In this case, we have $j = 3$, $\mathcal{N}_j = \{2, 4\}$ and $\mathcal{V}_j = \{1, 2, 3, 4\}$. From the in-neighbours w_2 and w_4, we derive the sets $\mathcal{C}_2 = \mathcal{J}_2 = \{1, 2, 3, 4\}$ and $\mathcal{C}_4 = \mathcal{J}_4 = \{1, 2, 3, 4\}$ based on the definition.

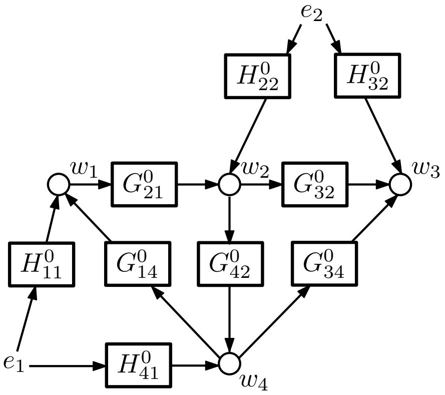

Fig. 1. Example network with target local MISO subnetwork around output node w_3.

C. One-step-ahead prediction error

The parameterized models $\{G_{jk}(q,\theta), H_j(q,\theta)\}$[4] for $k \in \mathcal{N}_j$ as described by Eq. (1) can serve as the basis for predicting the output $w_j(t)$ of the local subnetwork. For this study of model validation in the local MISO subnetwork, we utilize the MISO one-step-ahead predictor model from [20]. Consider the node signal $w_j(t)$ to be the variable subject to prediction, the predictor uses known variables, specifically $w_{\mathcal{N}_j}$, i.e. w_k for $k \in \mathcal{N}_j$, along with r_j and prior values of w_j. The one-step-ahead predictor for w_j is defined as $\hat{w}_j(t \mid t-1;\theta) := \bar{\mathbb{E}}\{w_j(t) \mid w_j^{t-1}, w_{\mathcal{N}_j}^{t-1}, r_j^t; \theta\}$[5]. Subsequently, the structure of the MISO predictor for model validation, which uses full input signals, i.e. $w_k, \forall k \in \mathcal{N}_j$, is as follows:

$$\begin{aligned}\hat{w}_j(t \mid t-1;\theta) &= (1 - H_j^{-1}(q,\theta))w_j(t)\\ &+ H_j^{-1}(q,\theta)\Big(\sum_{k\in\mathcal{N}_j} G_{jk}(q,\theta)w_k(t) + r_j(t)\Big).\end{aligned} \tag{5}$$

Remark 1. *The choice of predictor model can vary when changing to the identification scenario. Depending on different*

[2] The item e and f of **Assumption 1** allow us to construct the full MISO predictor model for the model validation purpose around any chosen output node w_j.

[3] A directed path through w_j can induce delays in the dynamic transfer due to the strict properness of G, while a directed path disjoint from w_j may result in no delays of the dynamic transfer since H can be monic.

[4] Without losing generalizability, we assume the Box-Jenkins (BJ) model structure [26] for the obtained plant and noise models.

[5] $\bar{\mathbb{E}}$ refers to $\lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N} \mathbb{E}$, and w_j^ℓ and $w_{\mathcal{N}_j}^\ell$ refer to signal samples $w_j(\tau)$ and $w_k(\tau), k \in \mathcal{N}_j$, respectively, for all $\tau \leqslant \ell$.

assumptions about network structures and noise properties, the predictor structure can change from MISO structure to MIMO structure [19], [32], and also from full input setup to partial input setup [33], [34]. However, although a dynamic network model can be obtained from different identification methods using different predictors, the validation procedure of the model can always be done with the MISO predictor as the Eq. (5) under ***Assumption 1****. Particularly in this study, we select all in-neighbors of the variable to be predicted, based on the network topology $\mathcal{T}_G$, to formulate the MISO predictor. This selection strategy could be extended to scenarios involving partial inputs in a MISO predictor for model validation.*

Following from the predictor in Eq. (5), the subsequent prediction error is defined as $\varepsilon_j(t,\theta) := w_j(t) - \hat{w}_j(t \mid t-1;\theta)$ and can also be expressed in a MISO setup as follows:

$$\varepsilon_j(t,\theta) = \\ H_j(q,\theta)^{-1}\Big(\sum_{k\in\mathcal{N}_j} \Delta G_{jk}(q,\theta)w_k(t) + H_j^0(q)e_j(t)\Big), \quad (6)$$

where $\Delta G_{jk}(q,\theta) = G_{jk}(q,\theta) - G_{jk}^0(q)$. Having obtained both the module transfer function models and the noise model $\{G_{jk}(q,\hat{\theta}_N), H_j(q,\hat{\theta}_N)\}$, we can consider the prediction error as an estimate of the innovation source $e_j(t)$ at node w_j. In this study, this error will also be referred to as the *residual* signal at node w_j.

Remark 2. *The notations for the noise model $H_j(q,\theta)$ and the innovation source e_j are adopted from [20], which assumes full-rank uncorrelated noise for the dynamic network framework. In this study, which considers the reduced-rank correlated noise framework, we can continue to use the same notations $H_j(q,\theta)$ and e_j for simplicity. This is justified under* ***Assumption 1****, stipulating that each node is influenced by a single disturbance. This assumption implies that each node is directly influenced by only one single innovation source and one noise model, even for the scenario with reduced-rank correlated noise, where the innovation source e_j and noise model $H_j(q,\theta)$ may not be unique for each node w_j. However, when our focus is on predicting the output of a specific node w_j within the target MISO subnetwork, the noise model and innovation source are always present solely. Therefore, we can unambiguously use e_j and $H_j(q,\theta)$ to refer to the specific innovation source and noise model for the chosen target MISO subnetwork. And to guarantee a unique representation of Eq. (6), the noise model $H_j(q,\theta)$ of the target MISO subnetwork is restricted to be monic for the subsequent residual analysis.*

Building upon existing work in local dynamic network identification [20], [21], [23], it is possible to have accurate models of both the module and noise transfer functions $\{G_{jk}(q,\theta), H_j(q,\theta)\}$. The accurate models are obtained from consistent estimations achieved by minimizing the quadratic prediction error criterion $V_N(\theta) = \frac{1}{N}\sum_{t=0}^{N-1}\varepsilon_j^2(t,\theta) \geq \sigma_{e_j}^2$ [6]. The criterion reaches its unique minimum when the estimated parameter:

$$\hat{\theta}_N = \text{arg } \underset{\theta}{min}\ V_N(\theta). \quad (7)$$

Under standard -weak- assumptions[1], the estimated parameter converges in the number of data N, to satisfy [35]:

$$\hat{\theta}_N \rightarrow \theta^* \quad \text{w.p.1 as } N \rightarrow \infty, \quad (8)$$

with $\theta^* = \text{arg } min_\theta\ \bar{\mathbb{E}}\ \varepsilon_j^2(t,\theta)$, where $\bar{\mathbb{E}} := \frac{1}{N}\sum_{t=0}^{N-1}\mathbb{E}$ and $\mathbb{E}$ is the expectation operator. A consistent estimate is obtained if $\{G_{jk}(q,\theta^*), H_j(q,\theta^*)\} = \{G_{jk}^0(q), H_j^0(q)\}$ [20], [36]. In such cases, the residual signal $\varepsilon_j(t,\theta^*)$ is identical to the innovation source $e_j(t)$, as dictated by Eq. (6). We refer to the situation that $\varepsilon_j(t,\theta^*) = e_j(t)$ by saying that the innovation source $e_j(t)$ is also consistently estimated.

Assuming that only the models $G_{jk}(q,\theta)$ are obtained for the target local subnetwork, the prediction error in Eq. (6) turns into an estimation of the disturbance v_j on node w_j which is expressed as:

$$v_j(t,\theta) = \sum_{k\in\mathcal{N}_j} \Delta G_{jk}(q,\theta)w_k(t) + H_j^0(q)e_j(t). \quad (9)$$

In this research, $v_j(t,\theta)$ is also referred to as the *predicted disturbance*. There are certain identification algorithms, such as the *Two-Stage method* [20] and the *Instrumental Variable method* [37] which can give accurate models of module transfer functions $G_{jk}(q,\theta)$ without consistently estimating the noise model $H_j(q,\theta)$. Therefore, under the scenario that only the accurate model of $\{G_{jk}(q,\theta)\}$ is obtained, i.e. $G_{jk}(q,\theta^*) = G_{jk}^0(q)$, the equation $v_j(t,\theta^*) = v_j(t)$ holds based on Eq. (9). Therefore, the disturbance $v_j(t)$ on node w_j is said to be also consistently estimated.

Remark 3. *For model-based FDD, models consistently estimated using dynamic network identification can serve as the accurate models for the ensuing FDD procedure. However, within the target MISO subnetwork, some transfer function models can be obtained through other modeling techniques such as first principle modeling or ETFE, and some transfers can be known already, e.g. controllers with known dynamics. These models can also be used for the FDD procedure in this research, provided their accuracy. Meanwhile, the predictor model given by Eq. (5), alongside the construction of the residual $\varepsilon(t,\theta)$ and the predicted disturbance $v(q,\theta)$ as described in Eq. (6) and Eq. (9) respectively, is adaptable to models derived from the above methods.*

For simplicity of notation, we use the notation $\hat{G}_{jk}(q)$ and $\hat{H}_j(q)$ to denote the module transfer models and the noise

[6] The local transfer functions $\{G_{jk}(q,\theta), H_j(q,\theta)\}$ can also be obtained by doing a MIMO network identification with the criterion $\bar{V}(\theta) = \frac{1}{N}\sum_{t=0}^{N-1}\varepsilon^\top(t,\theta)Q\varepsilon(t,\theta)$, with $\varepsilon = [\varepsilon_1, ..., \varepsilon_L]^\top$ and the weighted matrix $Q \geq 0$ [19].

model used for FDD, which can be obtained from identification, first principle modeling, ETFE, known dynamics, etc. Additionally, we will denote the residual from Eq. (6) as $\hat{\varepsilon}_j(t)$ and the predicted disturbance from Eq. (9) as $\hat{v}_j(t)$.

D. Fault in dynamic network framework

Real-world system faults exhibit varying temporal behaviors and can be categorized into three primary types: abrupt, intermittent, and incipient faults, as delineated in the literature [4], [5]. These faults $f(t)$ can be represented as deviations of the internal system behavior or measured signals from their nominal conditions. Within the context of a dynamic network framework, an abrupt fault manifests as a sudden and lasting deviation in the network, often modeled as a step change, as illustrated in Fig. 2 (a). Intermittent faults arise at irregular intervals and are typically modeled as a composite of impulses, as depicted in Fig. 2 (c). Incipient faults gradually evolve within the system and are commonly modeled as a ramp change, as shown in Fig. 2 (b).

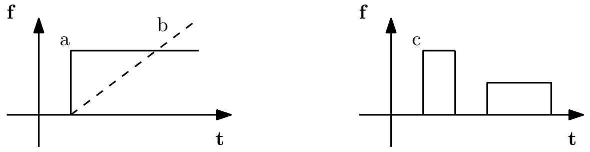

Fig. 2. Time dependency of faults: (a) abrupt; (b) incipient; (c) intermittent [4].

Beyond their temporal behaviors, faults in real-world systems can also be classified as either additive or multiplicative, as established in previous studies [4], [5]. Additive faults influence process variables by introducing an external bias, often correlating with *sensor faults* in real-world systems. Multiplicative faults impact the system's transfer behavior, typically manifesting as parameter changes. Within the dynamic network framework, we postulate that additive faults first affect the node signals w in the data-generating system and may propagate through the network's paths and loops depending on whether the faulty node is connected to a feedback loop. The additive fault is as shown in Fig. 3 (a). Conversely, we propose that multiplicative faults influence the module transfer function G^0, consequently propagating through the network to impact other node signals. The multiplicative fault is as shown in Fig. 3 (b).

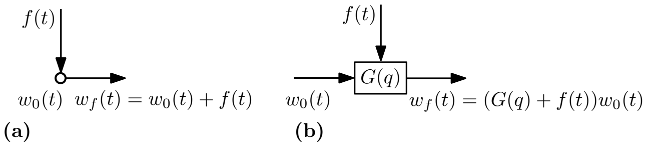

Fig. 3. Basic fault models: (a) additive fault for an output signal; (b) multiplicative fault [4].

To streamline the problem scope without undermining its generality, we focus solely on abrupt multiplicative faults within the dynamic network framework. In this study, we classify these abrupt multiplicative faults into a generalized category, formulated as the following assumptions:

Assumption 2. *Consider a nonzero module $G^0_{jk}(q)$ in the data-generating system, a fault $f(t)$ leads to module $G^0_{jk}(q)$ turning into a faulty module $G^f_{jk}(q)$, which is denoted as $G^0_{jk}(q) \leftarrow G^f_{jk}(q)$. The faulty module $G^f_{jk}(q) \neq G^0_{jk}(q)$, and it is assumed to be strictly proper and stable.*

When the faulty module $G^f_{jk}(q) = 0$, the fault $f(t)$ can be interpreted as a topology change, indicating the removal of an edge in the network. In the general case where $G^f_{jk}(q) \neq 0$, the faulty module $G^f_{jk}(q)$ is assumed to be strictly proper and stable, in accordance with **Assumption 1**. Given that an improper and unstable $G^f_{jk}(q)$ could result in system instability and potential failure, such cases are deliberately excluded from the fault detection and diagnosis scenarios considered in this study.

In response to faults under **Assumption 2**, the statistics of the residual $\hat{\varepsilon}_j(t)$ and the predicted disturbance $\hat{v}_j(t)$ diverge from those of $e_j(t)$ and $v_j(t)$ as in Eq. (1). Consequently, we opt for utilizing the residuals and predicted disturbances in constructing tests, which help to invalidate the dynamic network model when the system is no longer identical to the model. Ideally with a validated model of the target MISO subnetwork for FDD, the divergence of the residual $\hat{\varepsilon}_j(t)$ and the predicted disturbance $\hat{v}_j(t)$ from $e_j(t)$ and $v_j(t)$ can only be obtained when a faulty module $G^f_{jk}(q)$ occurs.

Thus, we begin by introducing a model validation approach for local dynamic network models which ensures the accuracy of the models used in the FDD process. When using a validated, accurate model in the fault detection phase, any data deviations that invalidate the accurate model directly indicate a detected fault. At the same time, the residuals generated between the faulty system and the accurate model provide better insight into the fault, unaffected by model errors, enriching the subsequent fault diagnosis protocol.

III. Local subnetwork model validation

A. Problem definition

The model validation procedure assesses whether the model of a dynamic system aligns with its intended application. Regarding local MISO subnetwork model validation for the application of FDD, the primary aim is to ascertain the accuracy of all transfer function models within the MISO subnetwork.

Consider a data-generating system denoted by $\{G^0(q), H^0(q)\}$ and specific models $\{\hat{G}_{jk}(q), \hat{H}_j(q)\}$ for $k \in \mathcal{N}_j$ corresponding to all modules with the target

MISO subnetwork. The local subnetwork model validation test aims to determine if the measured data set $\{w_{\mathcal{N}_j}(t), w_j(t)\}$ - potentially together with external input data $r_i(t)$ - provides evidence to reject at least one of the null hypotheses:

- ***Hypothesis*** $\mathcal{H}_a$***:*** *In the target MISO subnetwork, the obtained models* $\{\hat{G}_{jk}(q),\ \hat{H}_j(q)\}$ *of all transfers are identical to the corresponding transfers from the data-generating system* $\{G^0(q),\ H^0(q)\}$ *such that* $\hat{G}_{jk}(q) = G^0_{jk}(q)$ *and* $\hat{H}_j = H^0_j(q)$*, thus* $\hat{\varepsilon}_j(t) \triangleq e_j(t)$ *is white noise of zero mean and variance denoted by* $\sigma^2_{e_j}$*. Consequently, it is uncorrelated with itself and independent of the historical values of the input signals.*

- ***Hypothesis*** $\mathcal{H}_b$***:*** *In the target MISO subnetwork, the obtained models* $\hat{G}_{jk}(q)$ *of all modules are identical to the corresponding modules from the data-generating system* $G^0(q)$ *such that* $\hat{G}_{jk}(q) = G^0_{jk}(q)$*, thus* $\hat{v}_j(t) \triangleq v_j(t)$*.*

If the models of the local subnetwork pass the model validation test, the obtained models of the target MISO subnetwork $\{\hat{G}_{jk}(q),\ \hat{H}_j(q)\}$ (or $\hat{G}_{jk}(q)$) are considered validated by the data $\{w_{\mathcal{N}_j}(t), w_j(t)\}$ [38]. Users can select their test objective based on either of the above two hypotheses, depending on their requirement for an accurate local noise model, $\hat{H}_j(q)$.

Remark 4. *In the context of this thesis, it is important to note that strictly speaking, a certain hypothesis or model can only be invalidated based on the available data and tests. However, for the sake of simplicity, we will use the term 'validated' to describe a hypothesis or model that could not be invalidated through our testing procedures.*

The validity of $\mathcal{H}_a$ and $\mathcal{H}_b$ can be checked using tests on the sample autocorrelation function of the residual $\hat{\varepsilon}_j(t)$ and the sample cross-correlation function between either the residual $\hat{\varepsilon}_j(t)$ or the predicted disturbance $\hat{v}_j(t,)$, and the historical values of the predictor inputs $w_k(t)$ or $r_i(t)$.

Consequently, this chapter is structured as follows: Section B introduces tests based on autocorrelation and cross-correlation functions, Section C discusses the relationship between these test results and the null hypotheses for subnetwork model validation, and Section D provides numerical illustrations to support the theoretical findings.

B. Hypothesis testing

1) ***White noise test (autocorrelation test)****:* Given the estimated residual signal $\hat{\varepsilon}_j(t)$, derived from Eq. (6), we employ the white noise test to evaluate whether $\hat{\varepsilon}_j(t)$ is asymptotically self-independent, thereby can be considered a realization of a white noise process. The theoretical foundation for the white noise test is as follows:

Consider a white noise sequence $\hat{\varepsilon}_j(t)$ of length N, with zero mean and variance $\sigma^2_{e_j}$. Then it follows from a variant of the central limit theorem as in [35] and [30] that the following distribution holds:

$$\frac{1}{\sqrt{N}}\sum_{t=1}^{N}\begin{bmatrix}\hat{\varepsilon}_j(t-1)\\ \vdots \\ \hat{\varepsilon}_j(t-M)\end{bmatrix}\hat{\varepsilon}_j(t) \sim As\ \mathcal{N}\left(0, \sigma^2_{e_j}\cdot I\right). \tag{10}$$

where M represents the number of time lags under consideration, which also serves as the degrees of freedom for subsequent hypothesis tests.

For the time lag $\tau \in [1, M]$, the τ:th row of this vector is $\sqrt{N}\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$, where $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$ is the estimated autocorrelation function defined as follows:

$$\hat{R}^N_{\hat{\varepsilon}_j}(\tau) := \frac{1}{N}\sum_{t=1}^{N}\hat{\varepsilon}_j(t)\hat{\varepsilon}_j(t-\tau). \tag{11}$$

The asymptotic normal distribution in Eq. (10) consequently means that the sum of the scaled and squared version of the statistic $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$ over τ will converge to a χ^2-distribution with the freedom equal to the total number of the time lags M [35]:

$$\frac{N}{\sigma^2_{e_j}}\sum_{\tau=1}^{M}\left(\hat{R}^N_{\hat{\varepsilon}_j}(\tau)\right)^2 \sim As\ \chi^2(M). \tag{12}$$

Replacing the unknown $\sigma^2_{e_j}$ with the estimate:

$$\sigma^2_{e_j} = \frac{1}{N}\sum_{t=1}^{N}\hat{\varepsilon}_j(t)^2, \tag{13}$$

does not change the distributions in Eq. (10) and Eq, (12), asymptotically [35]. Consequently, the hypothesis $\mathcal{H}_a$ can be evaluated using the asymptotic distribution of $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$. Tests established from the literature can be applied for this purpose.

Hypothesis test 1-1 (The standard white noise test) [26].

Given a residual signal $\hat{\varepsilon}_j(t)$ *of data length* N*, the standard white noise test procedure of testing* $\mathcal{H}_a$ *is:*

$$\begin{cases} \text{if } \left|\hat{R}^N_{\hat{\varepsilon}_j}(\tau)\right| \leq \sqrt{\lambda^2/N}c_{\mathcal{N}}(\alpha), \forall\ |\tau| \leq M, \tau \neq 0, \\ \text{then accept } \mathcal{H}_a; \\ \text{if } \left|\hat{R}^N_{\hat{\varepsilon}_j}(\tau)\right| > \sqrt{\lambda^2/N}c_{\mathcal{N}}(\alpha), \exists\ |\tau| \leq M, \tau \neq 0, \\ \text{then reject } \mathcal{H}_a, \end{cases} \tag{14}$$

where the residual variance λ *is estimated from Eq. (13) and* $c_{\mathcal{N}}(\alpha)$ *corresponds to the Gaussian distributed process* $x \sim \mathcal{N}(0,1)$*, such that* $Pr(x \leq c_{\mathcal{N}}(\alpha)) = \alpha$*. Commonly, the probability level* α *is chosen as 0.95 or 0.99 [28], and the number of considered lags* M *remains at the user's choice.*

Remark 5. *Theoretically, the risk of rejecting the null hypothesis when it is actually true, also known as the false alarm rate, is equal to* $(1-\alpha)$*.*

The standard cross-correlation test is considered as a point-wise test [38]. As demonstrated in Eq. (14), the test evaluates each sample point of $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$ individually to determine if it remains within the confidence bounds, while it fails to consider the correlation between the terms of $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$ over lags τ [38]. A consequence of this is that the many models with an under-modeling error will not be invalidated by the test since the correlation in $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$, as a result of the under modeling, remains undetected [27]. Conversely, the subsequent test treats all samples in $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$ as a collective vector. This approach constructs a vector-valued statistic for hypothesis testing, based on the distribution presented in Eq. (12).

Hypothesis test 1-2 (The vector-valued white noise test) [35].

Given a residual signal $\hat{\varepsilon}_j(t)$ of data length N, the vector-valued white noise test is based on the statistic $Q_{BP}(N,M)$ which is defined as:

$$Q_{BP}(N,M) = \frac{N}{\left(\hat{R}^N_{\hat{\varepsilon}_j}(0)\right)^2} \sum_{\tau=1}^{M} \left(\hat{R}^N_{\hat{\varepsilon}_j}(\tau)\right)^2, \tag{15}$$

and the procedure of testing $\mathcal{H}_a$ is:

$$\begin{cases} \text{if } Q_{BP}(N,M) \le c_\chi(\alpha,M), \text{ then accept } \mathcal{H}_a; \\ \text{if } Q_{BP}(N,M) > c_\chi(\alpha,M), \text{ then reject } \mathcal{H}_a, \end{cases} \tag{16}$$

where $c_\chi(\alpha,M)$ corresponds to the $1-\alpha$ quantile of the χ^2-distribution with M degrees of freedom, i.e. for $x \sim \chi^2(M)$ it follows that $\Pr(x \le c_\chi(\alpha,M)) = \alpha$.

The difference between the sample-wise and vector-valued tests lies in the testing statistics each employs. The sample-wise test uses each $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$ to evaluate only the variance of each sample point, whereas the vector-valued test employs $Q_{BP}(N,M)$ to evaluate the entire covariance across multiple lags [38]. Consequently, the outcome of the sample-wise test is significantly influenced by the sample variance, which in turn is highly dependent on the system's *signal-to-noise ratio (SNR)*. In practical applications, it is also required to manually inspect the plot of autocorrelation $\hat{R}^N_{\hat{\varepsilon}_j}(\tau)$ to identify any latent correlations among sample points, even when they all stay within the confidence bounds. Despite the infrequent application of vector-valued tests in traditional model validation tasks, as indicated by [38], the limitations of sample-wise test us enough motivation to move to the vector-valued test for the model validation task in the dynamic network framework.

A improvement can be made for the statistics $Q_{BP}(N,M)$ following the reasoning of the Ljung-Box test [39]:

$$Q_{LB}(N,M) = \frac{N(N+2)}{\left(\hat{R}^N_{\varepsilon}(0)\right)^2} \sum_{\tau=1}^{M} \frac{\left(\hat{R}^N_{\varepsilon}(\tau)\right)^2}{N-\tau}, \tag{17}$$

and it is shown that the distribution for the Ljung–Box statistic is closer to a $\chi^2(M)$ distribution than is the distribution for $Q_{BP}(N,M)$ statistic for all sample sizes including small ones.

Hypothesis test 1-3 (The Ljung-Box white noise test).

Given a residual signal $\hat{\varepsilon}_j(t)$ of data length N, the Ljung-Box white noise test the procedure of testing $\mathcal{H}_a$ is:

$$\begin{cases} \text{if } Q_{LB}(N,M) \le c_\chi(\alpha,M), \text{ then accept } \mathcal{H}_a; \\ \text{if } Q_{LB}(N,M) > c_\chi(\alpha,M), \text{ then reject } \mathcal{H}_a, \end{cases} \tag{18}$$

where $Q_{LB}(N,M)$ is calculated as in Eq. (17).

The Ljung-Box white noise test is also a vector-valued test, but it has not traditionally been employed for classical model validation tasks. We introduce it in this context as a supplementary tool for model validation within the dynamic network framework. For either statistic between $Q_{BP}(N,M)$ and $Q_{LB}(N,M)$, there is no clear guide to the choice of M. If M is chosen too small, there is a danger of missing the existence of higher-order autocorrelations, but if M is chosen too large relative to the sample size, its finite-sample distribution is likely to deteriorate, diverging greatly from the χ^2 distribution [40]. According to [30], it is recommended that the selected value for M should at least exceed the number of estimated parameters in the model.

For clarity, the three white noise test methods will be referred to below as **Test 1-1**, **Test 1-2**, and **Test 1-3**.

*2) **Cross-correlation test**:* Given a residual signal $\hat{\varepsilon}_j(t)$ generated from Eq. (6) or a predicted disturbance signal $\hat{v}_j(t)$ generated from Eq. (9), the cross-correlation test is used to determine whether the residual signal $\hat{\varepsilon}_j(t)$ (or $\hat{v}_j(t)$) can be considered independent with the prior values of a chosen input signal u_k. The input signal u_k is usually chosen within w_k or r_k for $k \in \mathcal{N}_j$. The theoretical foundation for the cross-correlation test can be outlined as follows:

Consider a residual signal $\{\hat{\varepsilon}_j(t)\}^N$ independent of the past input signal $u_k(t)$, based on [35] and [30], the properties of the statistic $\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k}$ are given by:

$$\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k} = \underbrace{\frac{1}{N}\begin{bmatrix} u_k(1) & u_k(2) & \cdots & \cdots & u_k(N) \\ & \ddots & \cdots & \cdots & \vdots \\ & & u_k(1) & \cdots & u_k(N-M+1) \end{bmatrix}}_{P_{u_k}} \times \underbrace{\begin{bmatrix} \hat{\varepsilon}_j(1) \\ \vdots \\ \hat{\varepsilon}_j(N) \end{bmatrix}}_{\hat{\boldsymbol{\varepsilon}}_j}, \tag{19}$$

are expected to asymptotically converge to a zero-mean Gaussian distribution as N increases:

$$\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k} \sim As\,\mathcal{N}(0,P), \tag{20}$$

the associated asymptotic covariance matrix is given by:

$$P = \frac{1}{N^2} P_{u_k} \Lambda_{\hat{\varepsilon}_j} P_{u_k}^\top, \tag{21}$$

with the residual auto-covariance matrix $\Lambda_{\hat{\varepsilon}_j} = \mathbb{E}\left[\hat{\boldsymbol{\varepsilon}}_j \hat{\boldsymbol{\varepsilon}}_j^\top\right]$. Eq. (19) implies that, when scaled and squared, the statistic $\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k}$ is expected to asymptotically converge to a χ^2 distribution with M degrees of freedom [27]:

$$\left[\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k}\right]^\top P^{-1} \left[\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k}\right] \sim As\ \chi^2(M). \tag{22}$$

The distributions in Eq. (19) and Eq. (22) remain consistent for $\hat{\mathbf{R}}^N_{\hat{v}_j u_k}$ when substituting the residual $\hat{\varepsilon}_j(t)$ with the predicted disturbance $\hat{v}_j(t)$, as per Eq. (19) and Eq. (22). Subsequently, the hypotheses $\mathcal{H}_a$ and $\mathcal{H}_b$ can be assessed through the asymptotic distributions of $\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k}$ and $\hat{\mathbf{R}}^N_{\hat{v}_j u_k}$.

Unlike the white noise test, the cross-correlation test within the dynamic network framework typically cannot directly test the very basic null hypotheses for the local subnetwork model validation, $\mathcal{H}_a$ and $\mathcal{H}_b$. This is because $\mathcal{H}_a$ and $\mathcal{H}_b$ assume that all module transfer functions in the target subnetwork are accurate. This consequently leads to $\hat{\varepsilon}_j(t)$ and $\hat{v}_j(t)$ being uncorrelated with past values of certain input signals within the network. Nevertheless, the converse is not guaranteed; demonstrating that $\hat{\varepsilon}_j(t)$ and $\hat{v}_j(t)$ are uncorrelated with past values of certain input signals can only validate certain modules within the target MISO subnetwork, which is not sufficient to validate $\mathcal{H}_a$ and $\mathcal{H}_b$. Thus, we need another hypothesis that focuses only on the signal dependency but not the validity of all modules in the MISO subnetwork, so that it can be directly tested by the cross-correlation test. Therefore, we introduce the following hypothesis, $\mathcal{H}_c$, for subsequent hypothesis testing involving residual $\hat{\varepsilon}_j(t)$ (or $\hat{v}_j(t)$) and a specific input u_k:

- **Hypothesis $\mathcal{H}_c$:** *The residual $\hat{\varepsilon}_j(t)$ (or $\hat{v}_j(t)$) is independent of the past values of the input signal u_k.*

The hypothesis $\mathcal{H}_c$ can be directly validated or rejected using the following established tests from the literature. After validating $\mathcal{H}_c$, $\mathcal{H}_a$ and $\mathcal{H}_b$ are possible to be further validated, which will be illustrated in the following section.

Hypothesis test 2-1 (The standard cross-correlation test). [35]

Given a residual signal $\hat{\varepsilon}_j(t)$ and an input signal $u_k(t)$ of data length N, the standard cross-correlation test procedure of testing $\mathcal{H}_c$ is:

$$\begin{cases} \textit{if } \left|\hat{R}^N_{\hat{\varepsilon}_j u_k}(\tau)\right| \leq \sqrt{p_\tau/N}\, c_{\mathcal{N}}(\alpha), \forall \tau \neq 0, \ \textit{then reject } \mathcal{H}_c; \\ \textit{if } \left|\hat{R}^N_{\hat{\varepsilon}_j u_k}(\tau)\right| > \sqrt{p_\tau/N}\, c_{\mathcal{N}}(\alpha), \forall \tau \neq 0, \ \textit{then accept } \mathcal{H}_c, \end{cases} \tag{23}$$

where p_τ is the (τ, τ) entry of the asymptotic covariance matrix P in Eq. (21) and $c_{\mathcal{N}}(\alpha)$ corresponds to the Gaussian distributed process $x \sim \mathcal{N}(0,1)$, such that $Pr(x \leq c_{\mathcal{N}}(\alpha)) = \alpha$.

The standard cross-correlation test is a sample-wise test, similar to the standard white noise test, for the same underlying reasons. The vector-valued cross-correlation test is as follows:

Hypothesis test 2-2 (The vector-valued cross-correlation test). [27]

Given a residual signal $\hat{\varepsilon}_j(t)$ and an input signal $u_k(t)$ of data length N, the vector-valued cross-correlation test is based on the statistic $Q_c(N, M)$ which is defined as:

$$Q_c(N, M) = \left[\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k}\right]^\top P^{-1} \left[\hat{\mathbf{R}}^N_{\hat{\varepsilon}_j u_k}\right], \tag{24}$$

and the procedure of testing $\mathcal{H}_c$ is:

$$\begin{cases} \textit{if } Q_c(N, M) \leq c_\chi(\alpha, M), \ \textit{then accept } \mathcal{H}_c; \\ \textit{if } Q_c(N, M) > c_\chi(\alpha, M), \ \textit{then reject } \mathcal{H}_c, \end{cases} \tag{25}$$

where $c_\chi(\alpha, M)$ corresponds to the $1-\alpha$ quantile of the χ^2-distribution with M degrees of freedom, i.e. for $x \sim \chi^2(M)$ it follows that $\Pr(x \leq c_\chi(\alpha, M)) = \alpha$.

Similar to the sample-wise white noise test, the sample-wise cross-correlation test shares the same limitations when validating the hypothesis $\mathcal{H}_c$, when compared to its vector-valued counterpart. However, both of the cross-correlation tests can only validate specific modules within the target MISO subnetwork can be validated. Consequently, the following subsection will discuss the specific module transfer functions within the target MISO subnetwork that each of these correlation tests can validate.

For clarity, the two cross-correlation test methods will be referred to below as **Test 2-1**, and **Test 2-2**.

C. Model validation in the MISO subnetwork

*1) **Target module set of correlation tests:*** In the context of dynamic network frameworks, the correlation test result usually depends on the accuracy of multiple estimated transfer functions, primarily because the residual signal $\hat{\varepsilon}_j(t)$ and the predicted noise $\hat{v}_j(t)$ originate from all the $\hat{G}_{jk}(q)$ functions within a chosen MISO subnetwork. To streamline the notation for model validation in MISO subnetworks, we introduce the following concept:

Definition 2. *We define the target module set of a specific correlation test as the set of all module transfer function models that can be validated by this test. Based on the signals used by the correlation test, a signal $m(t)$ or two signals $m(t)$ and $n(t)$, we denote the corresponding target module set as $\mathcal{S}_m$ or $\mathcal{S}_{mn}$.*

Therefore, the target module set of the autocorrelation test using residual $\hat{\varepsilon}_j(t)$ is denoted by $\mathcal{S}_{\hat{\varepsilon}_j}$, the target module set of the cross-correlation test using $\hat{\varepsilon}_j(t)$ and $w_k(t)$ is denoted by $\mathcal{S}_{\hat{\varepsilon}_j w_k}$. The target module set serves as a crucial attribute for each correlation test and has strong associations with both the network topology $\mathcal{T}_G$ and the noise topology $\mathcal{T}_H$.

We examine two distinct scenarios: one in which a correlation test is passed, and another where it is failed. The following assumption is applied when a correlation test is passed:

Assumption 3. *When a correlation test with the target module set $\mathcal{S}_{mn}$ is successfully passed, we assume that all associated transfer function models within $\mathcal{S}_{mn}$ are validated with a probability of α. Specifically, while errors from multiple inaccurate models within the target module set could theoretically cancel each other out and still yield a passed test, we operate under the assumption that such a scenario is practically unlikely. Therefore, we assume that the probability of successfully validating all modules in the target module set coincides with the theoretical probability α of validating the null hypothesis when it actually holds true.*

Consequently, a failed test implies the presence of at least one invalidated transfer function model within the target module set. Therefore, efforts to enhance model quality should be focused on the modules in the target module set of the failed test.

Remark 6. *In addition to the accuracy of the estimated module transfer functions, the accuracy of the noise model will also influence certain tests. To illustrate, consider the white noise test using residual $\hat{\varepsilon}_j(t)$; even if the transfer functions of all the modules within the target module set are modeled correctly, an incorrect noise model $\hat{H}_j(q)$ would lead to the residual failing to pass the white noise test. In the context of this study, it is important to note that noise modeling is not always mandatory and that the accuracy of the noise model is not always required. To develop model validation methods that are more universally applicable across different modeling scenarios, we only include the module transfer functions $\hat{G}_{jk}(q)$ in the target module set for validation. Instead of treating the noise model $\hat{H}_j(q)$ also as a transfer function to be validated, we consider the accuracy of $\hat{H}_j(q)$ as the extra prior information that can be used to assist model validation within local subnetworks.*

To augment our decision-making process, we incorporate noise information, the accuracy of the noise model $\hat{H}_j(q)$, and network noise topology $\mathcal{T}_H$, to appropriately guide the selection of different tests. In the following subsections, we will sequentially introduce the target module sets associated with various types of correlation tests and the required prior noise information for each test.

*2) **White noise test**:* The white noise test can be executed using three methods: **Test 1-1**, **Test 1-2**, and **Test 1-3**, and it directly tests the hypothesis $\mathcal{H}_a$ for the MISO subnetwork validation purpose. Therefore, an accurate noise model $\hat{H}_j(q)$ is necessitated for a white noise test to pass, in addition to the accurate module models $\hat{G}_{jk}(q)$. Subsequently, we define the prerequisites and target module set for the white noise test in the following corollary:

Corollary 1. *The white noise test, utilizing $\hat{\varepsilon}_j(t)$, is applicable only when the accurate estimated noise model $\hat{H}_j(q)$ can be obtained. Additionally, the target module set for the white noise test is denoted by*

$$\mathcal{S}_{\hat{\varepsilon}_j} = \{G_{jk} \mid k \in \mathcal{N}_j\}.$$

Proof: *See Appendix A.*

The target module set of the white noise test includes all module transfer functions in the target MISO subnetwork. However, employing the white noise test necessitates the accurate noise model $\hat{H}_j(q)$ of the target MISO subnetwork, thereby implicitly demanding knowledge of the noise topology $\mathcal{T}_H$. Hence, the prerequisites to use the white noise test include the accurate noise model $\hat{H}_j(q)$ and the noise topology $\mathcal{T}_H$.

*3) **Cross-correlation test using node signals**:* The cross-correlation test performed using $\hat{\varepsilon}_j(t)$ (or $\hat{v}_j(t)$) with node signals $w_i(t)$ from the target subnetwork MISO, can apply the methods **test 2-1** or **test 2-2** to evaluate the hypothesis $\mathcal{H}_c$. While it can not directly validate the null hypotheses $\mathcal{H}_a$ or $\mathcal{H}_b$ for the full MISO subnetwork validation purpose, the cross-correlation test validates modules within its target module set. The following corollary defines both the target module set and the noise prerequisites for the cross-correlation test using node signals:

Corollary 2. *The cross-correlation test using $\hat{\varepsilon}_j(t)$ (or $\hat{v}_j(t)$ and w_i is applied differently according to different levels of the noise information, each with its own set of available node signals for conducting the test:*

1) *When the accurate noise model $\hat{H}_j(q)$ and the noise topology $\mathcal{T}_H$ of the network can be obtained, all node signals w_i where $i \in \mathcal{N}_j$ can be used;*
2) *When the noise topology $\mathcal{T}_H$ of the network is available but the accurate noise model $\hat{H}_j(q)$ is not, node signals w_i, where $i \in \mathcal{N}_j \setminus \mathcal{V}_j$, can be used.*

While the use of $w_i(t)$ might be limited with less noise information, the target module set of the cross-correlation test using $\hat{\varepsilon}_j(t)$ (or $\hat{v}(t)$) and w_i stays the same, which is defined by:

$$\mathcal{S}_{\hat{\varepsilon}_j w_i} = \mathcal{S}_{\hat{v}_j w_i} = \{G_{jk} \mid k \in \mathcal{N}_j \cap \mathcal{C}_i\}.$$

Proof: *See Appendix B.*

*4) **Cross-correlation test using external excitation signals**:* Analogous to the cross-correlation test using node signals, the test using external excitation signals utilizes with $\hat{\varepsilon}_j(t)$ (or $\hat{v}_j(t)$) and $r_i(t)$ from the target subnetwork MISO. Again, the methods **Test 2-1** or **Test 2-2** can be used to evaluate the hypothesis $\mathcal{H}_c$, which can only validate the modules from its target module set. The following corollary defines both the target module set and the noise prerequisites for the cross-correlation test using external excitation signals:

TABLE I
THE NOISE PREREQUISITES AND TARGET SETS OF DIFFERENT MODEL VALIDATION TESTS

Target sets / Noise information / Tests	$\hat{H}_j(q)$ & $\mathcal{T}_H$	$\mathcal{T}_H$	None
White noise test with $\hat{\varepsilon}_j$	$\mathcal{S}_{\hat{\varepsilon}_j}, \forall j$	$\emptyset$	$\emptyset$
Cross-correlation test with $\hat{\varepsilon}_j$ (or $\hat{v}_j$) and w_i	$\mathcal{S}_{\hat{\varepsilon}_j w_i}, i \in \mathcal{N}_j$	$\mathcal{S}_{\hat{v}_j w_i}, i \in \mathcal{N}_j/\mathcal{V}_j$	$\emptyset$
Cross-correlation test with $\hat{\varepsilon}_j$ (or $\hat{v}_j$) and r_i	$\mathcal{S}_{\hat{\varepsilon}_j r_i}, i \in \mathcal{N}_j$	$\mathcal{S}_{\hat{v}_j r_i}, i \in \mathcal{N}_j$	$\mathcal{S}_{\hat{v}_j r_i}, i \in \mathcal{N}_j$

Corollary 3. *The cross-correlation test using $\hat{\varepsilon}_j(t)$ (or $\hat{v}_j(t)$ and r_i can be used independent of the accurate noise model $\hat{H}_j(q)$ or the noise topology $\mathcal{T}_H$. The target module set of the cross-correlation test using $\hat{\varepsilon}_j(t)$ (or $\hat{v}_j(t)$) and r_i is defined by:*

$$\mathcal{S}_{\hat{\varepsilon}_j r_i} = \mathcal{S}_{\hat{v}_j r_i} = \{G_{jk} \mid k \in \mathcal{N}_j \cap \mathcal{J}_i\}.$$

Proof: *See Appendix C.*

We consolidate the prerequisites and target module sets of various tests in Table I. Following the above analyses, we present the relationships among the target module sets for the various tests:

Proposition 1. *For dynamic network models with different network and noise topology, it always holds for the target module sets of different tests that:*

$$\mathcal{S}_{\hat{\varepsilon}_j r_i} \subseteq \mathcal{S}_{\hat{\varepsilon}_j w_i} \subseteq \mathcal{S}_{\hat{\varepsilon}_j}$$

Proof: *For a given node w_i, it naturally follows that $\mathcal{J}_i \subseteq \mathcal{C}_i$. Therefore, $(\mathcal{N}_j \cap \mathcal{J}_i) \subseteq (\mathcal{N}_j \cap \mathcal{C}_i) \subseteq \mathcal{N}_j$.* Consequently, $\{G_{jk} \mid k \in \mathcal{N}_j \cap \mathcal{J}_i\} \subseteq \{G_{jk} \mid k \in \mathcal{N}_j \cap \mathcal{C}_i\} \subseteq \{G_{jk} \mid k \in \mathcal{N}_j\}$, so the proposition is proved. ■

Based on the preceding analysis, it becomes evident that among the three tests, the white noise test imposes the most strict noise prerequisites, and simultaneously gives the largest target module set which contains all modules in the target MISO subnetwork; the cross-correlation test using r_i signals does not require additional noise information and always gives the smallest target module set; the cross-correlation test using w_i signals falls between these two extremes.

To realize the purpose of full MISO subnetwork model validation, if we can obtain the accurate noise model $\hat{H}_j(q)$ along with the noise topology $\mathcal{T}_H$, the white noise test can be directly used to evaluate the null hypothesis $\mathcal{H}_a$. In the absence of sufficient noise information, we can only resort to the cross-correlation tests to evaluate the hypothesis $\mathcal{H}_c$. The validation of $\mathcal{H}_c$ allows validation of the models within the target module set of the corresponding cross-correlation test, which includes either a subset or the entire subnetwork, as specified in **Proposition 1**. Thus, if a single test is insufficient, multiple cross-correlation tests can be performed to validate all modules within the target MISO subnetwork, effectively testing the null hypothesis $\mathcal{H}_b$. This ensures that even with limited noise information, full MISO subnetwork model validation remains achievable via cross-correlation tests.

It is important to note that the cross-correlation test using r_i signals requires the intentional introduction of the excitation signals r_i on the real-world system for the validation experiments. This stands in contrast to the other two tests, which do not require active incorporation of r_i signals. Therefore, the cross-correlation test using r_i signals is classified as an *active test*, while the remaining two are considered *passive tests*[7]. Compared to the passive test, the active test carries a higher experimental cost due to the introduction of additional r_i signals. However, it compensates by allowing for more lenient noise prerequisites. As seen in Table I, active tests are always available, irrespective of the noise information, whereas the other two passive tests are constrained by specific noise information. This observation points to a trade-off between noise information and the cost of validation experiments. More noise information can make the validation procedure more passive, while limited noise information will make the validation more reliant on r signals, thus necessitating a more active approach. Additionally, **Proposition 1** indicates that the active test involves the smallest target module set for the target module, which implies that when the test is not passed, the invalidated module can be confined within a narrower scope. This scope containing the invalidated module is defined as the *resolution* for model invalidation. Consequently, a trade-off exists between the resolution for model invalidation and the cost of validation experiments. Active tests consistently provide higher resolution for model invalidation, whereas passive tests offer lower resolution.

D. *Numerical illustration*

*1) **Experiment setup**:* To evaluate the performance of all the proposed hypothesis tests in Section B, three sets of simulation experiments are conducted: the first experiment aims to compare the effectiveness of various tests in validating the local subnetwork; the second experiment shows the robustness of the test performance against different hyperparameters; the third seeks to evaluate the sensitivity of the hypothesis tests to discrepancies between the real system and its models. The experiments utilize a data-generating network as depicted in

[7] When the cross-correlation test utilizes an existing r signal from the data-generating system, the test can also be classified as a passive test, as it eliminates the need for additional excitation signals.

Fig. 4. The target MISO subnetwork is selected with w_3 as the output, encompassing in-neighbors w_2 and w_4, and modules G^0_{12} and G^0_{13}. The white noise sources e_1 and e_2 are designed with zero mean and variance $Var(e_1) = 0.1$ and $Var(e_2) = 0.2$, respectively. The specifications for the module transfer functions and noise models are provided in Appendix D. The external excitation signal $r_2(t)$ is designed as a white noise signal with zero mean and variance $Var(r_2) = 5$ during the simulation. For simplicity of the simulations, we assume the models are provided and take the accurate module transfer function equal to the corresponding real transfer function. All modules outside of the target MISO subnetwork are considered to be accurate in the following experiments to eliminate additional interference.

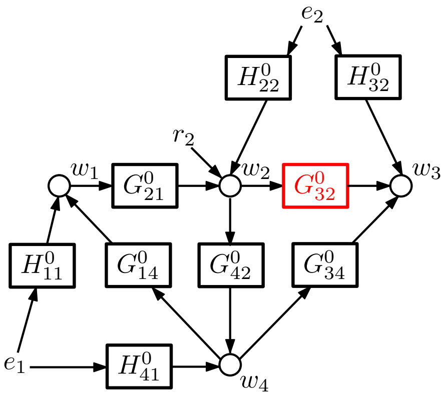

Fig. 4. Data-generating network with target local MISO subnetwork around output node w_3. The potential invalidated model G^0_{32} is marked in red.

2) ***Simulation experiment 1:*** In the first simulation experiment, we aim to compare the effectiveness of different hypothesis test methods. The following experiment scenarios are adopted:

- **Validated Scenario:** In this scenario, all modules within the target MISO subnetwork are accurate, such that $\hat{G}_{32}(q) = G^0_{32}(q)$ and $\hat{G}_{34}(q) = G^0_{34}(q)$.
- **Invalidated Scenario:** In this scenario, only the module G_{32} is inaccurate among all the modules within the target MISO subnetwork. Specifically, the inaccurate module is $\hat{G}_{32}(q) = \frac{0.312q^{-1}}{1-0.8q^{-1}+0.2q^{-2}}$, while the true module is $G^0_{32}(q) = \frac{0.39q^{-1}}{1-0.8q^{-1}+0.2q^{-2}}$.

When using a hypothesis test to validate or invalidate the modules in the target subnetwork. A failed test is labeled as a positive alarm for the invalidation, while a passed test is labeled as a negative alarm. Let P_{num} represent the total count of validated scenarios, and let N_{num} represent the total count of invalidated scenarios during the entire experiment. Furthermore, in the case that a test fails under an invalidated scenario, the result is classified as a true positive alarm (TP); in the case that a test fails under a validated scenario, the result is classified as a false positive (FP). The performance of various hypothesis tests is evaluated using the receiver operating characteristic (ROC) curve [41]. It is used to compare the true positive rate (TPR) against the false positive rate (FPR) over tuning hyperparameters, where

$$TPR = \frac{TP}{P_{num}}, \quad FPR = \frac{FP}{N_{num}}.$$

The hyperparameters for the different hypothesis tests are the confidence level α and the degrees of freedom M. As mentioned in Section B, no definitive guidelines exist for selecting the value of M. For our experiments, we set $M = 100$ and allow the confidence level α to vary within the range $[0.001, 1]$, serving as the tuning parameter for all tests to construct full ROC curves. Additionally, to make the sampling points on every ROC curve uniformly distributed, we sample the α value densely within the range $[0.9, 1]$ and sparsely within $[0.001, 0.9]$, culminating in a total of 30 samples for α.

Every point on the ROC curve signifies a (FPR, TPR) pair, with the point $(0, 1)$ denoting the ideal outcome in model validation, wherein all positive alarms are indeed true positives. Accordingly, the performance of different tests can be compared by examining how close their respective points on the ROC curve are to the ideal $(0, 1)$ point, typically by computing the distance metric $dis = \sqrt{FPR^2 + (1 - TPR)^2}$. A lower value of dis indicates better test performance.

We carry out 1000 Monte Carlo simulations for both the validated and invalidated cases, setting $P_{num} = N_{num} = 1000$ and varying the data lengths N to be 500, 2500, and 5000. The resulting ROC curves for various white noise tests using the residual signal (t) and different data lengths are presented in Fig. 5. The pink curves with square markers denote **Test 1-1**, the red curves with circular markers denote **Test 1-2**, and the blue curves with triangular markers denote **Test 1-3**. Guided by the invalidated module G_{32} in the invalidated case, we select the residual signal $\hat{\varepsilon}_3(t)$ and the input signal $r_2(t)$ for the cross-correlation tests, since the target module set $\mathcal{S}_{\hat{\varepsilon}_3 r_2}$ includes the invalidated module G_{32}. The corresponding ROC curves for varying data lengths are displayed in Fig. 6. The blue curves with triangular markers denote **Test 2-1**, while the red curves with circular markers denote **Test 2-2**.

As observed in Fig. 5 and Fig. 6, the test curves tend to converge towards the $(0, 1)$ point as the data length increases. This trend towards convergence corroborates the theoretical analysis of hypothesis tests. Additionally, the curves in Fig. 6 representing the active cross-correlation tests using the excitation signal $r_2(t)$, converge more rapidly than the curves in Fig. 5, which represents the passive white noise tests relying solely on the residual signal $\hat{\varepsilon}_3(t)$. The faster convergence is attributed to the incorporation of the r signal, which improves the *signal-to-noise ratio (SNR)* and allows for more accurate testing with a smaller amount of data.

Notably, while the curves associated with the vector-valued tests (**Test 1-2** and **Test 1-3**) in Fig. 5 approach the point $(0, 1)$ as the data length grows, the curve for the sample-

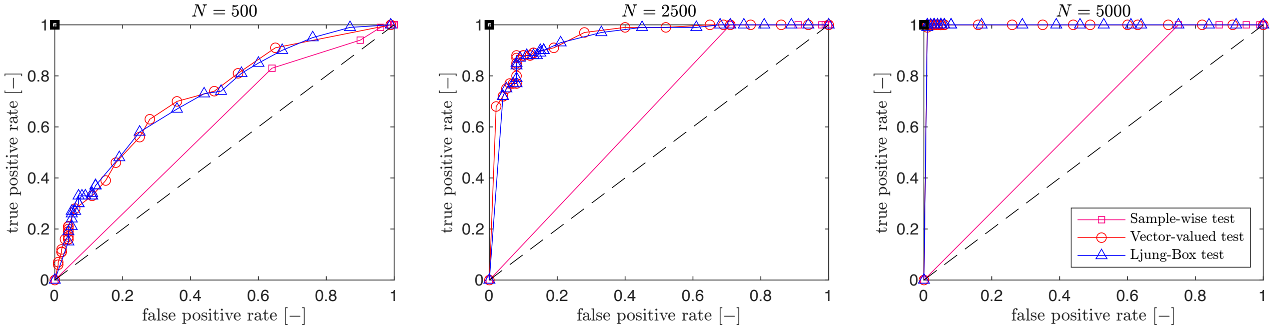

Fig. 5. TPR vs FPR for different white noise tests over tuning parameters $\alpha \in [0.001, 1]$ for different data length: $N = 500$ (left), $N = 2500$ (middle), $N = 5000$ (right). The ideal $(0, 1)$ point is marked in a black square. The dashed diagonal represents the test using a random guess, where the TPR is always equal to FPR. Thus the ROC curves of functional tests should at least be above the diagonal.

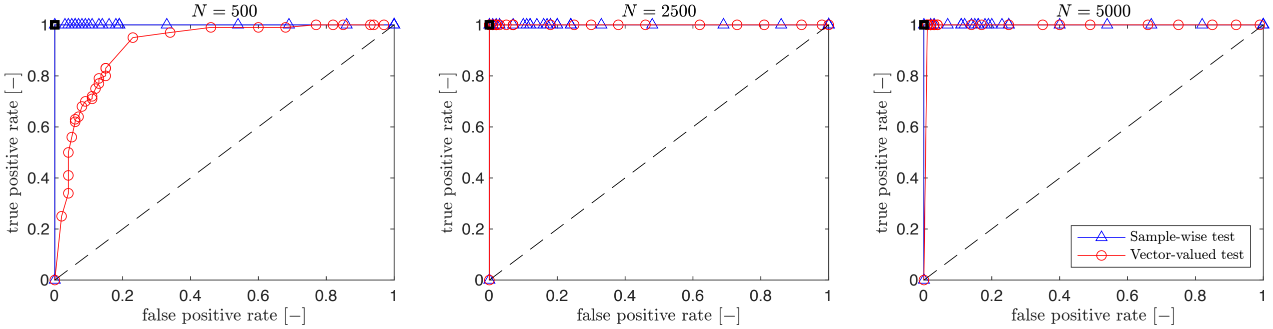

Fig. 6. TPR vs FPR for different cross-correlation tests using $r(t)$ over tuning parameters $\alpha \in [0.001, 1]$ for different data length: $N = 500$ (left), $N = 2500$ (middle), $N = 5000$ (right).

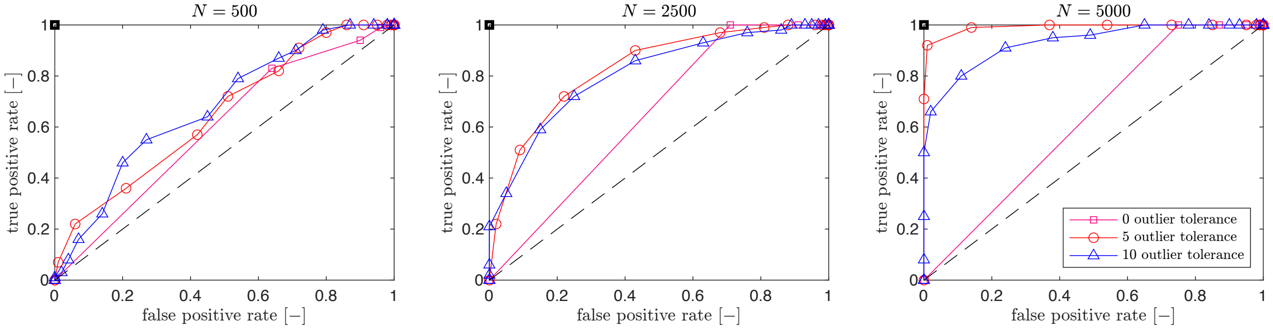

Fig. 7. TPR vs FPR for the sample-wise white noise tests with different numbers of outlier tolerance over tuning parameters $\alpha \in [0.001, 1]$ for different data length: $N = 500$ (left), $N = 2500$ (middle), $N = 5000$ (right).

wise test (**Test 1-1**) tends to stay close to the diagonal. The reason for this difference is that the sample-wise test mandates each sampling point to stay within the confidence interval to pass the test, implying a 0 tolerance for outliers. As a result, the sample-wise test curves exhibit a high FPR, indicating that the 0 tolerance for outliers is an excessively conservative criterion for dynamic network model validation. One potential explanation for this could be the low SNR in the test signals, attributable to multiple disturbance sources unique to dynamic networks, absent in classical open and closed-loop systems.

Following the above analysis, we can conclude that for the dynamic network model validation, the 0 outlier tolerance for the sample-wise test proves to be too conservative, leading to suboptimal performance. Therefore, we conducted a subsequent experiment with varied outlier tolerances for the white noise sample-wise test, maintaining the same conditions as in previous experiments. The results are shown in Fig. 7. The figure reveals that through elevating the outlier tolerance to 5 and 10, while keeping the degree of freedom M fixed at 100, the ROC curves get notably closer to $(0, 1)$, signifying a substantial improvement in test performance. To summarize, for effective dynamic network model validation using sample-wise tests, tuning the outlier tolerance as an additional hyperparameter is needed to optimize test performance. Nonetheless, this additional hyperparameter tuning requires further experimentation and still does not enable the sample-wise test to match the performance of the vector-valued tests under identical conditions. For a direct comparison, the minimum distances to the point $(0, 1)$ for each test are compiled in Table II.

TABLE II
MINIMUM DISTANCE OF DIFFERENT *ROC* CURVES TO $(0, 1)$

	$N = 500$	$N = 2500$	$N = 5000$
Test 1-1 (0 outlier tolerance)	0.66	0.71	0.75
Test 1-2	0.46	0.14	0.01
Test 1-3	0.48	0.15	0.01
Test 2-1 (using r)	0	0	0
Test 2-2 (using r)	0.22	0	0
Test 1-1 (5 outlier tolerance)	0.58	0.35	0.08
Test 1-1 (10 outlier tolerance)	0.52	0.37	0.22

The table indicates that for white noise tests (**Test 1-1**, **Test 1-2** and **Test 1-3**), vector-valued tests (**Test 1-2** and **Test 1-3**) generally outperform sample-wise tests (**Test 1-1**). However, in the case of cross-correlation tests, both types of tests exhibit similar performance. When considered alongside Fig. 6, it becomes evident that the sampling points on the curves of the vector-valued test are more densely clustered around the point $(0, 1)$ compared to the sample-wise test. This suggests greater robustness of the vector-valued tests against variations in confidence levels α. We should note that while the thesis omits the experimental results for the cross-correlation test using $w(t)$, our experiments indicate that the vector-valued test still outperforms the sample-wise test in that scenario as well. In summary, for the purpose of dynamic network model validation, vector-valued tests generally offer superior performance over sample-wise tests, making them our preferred choice.

*3) **Simulation experiment 2:*** In the second simulation experiment, we aim to show the robustness of the test performance against varying hyperparameters. The experiment only considers the invalidated scenario, where the module G_{32} is inaccurate among all the modules within the target MISO subnetwork. Specifically, the inaccurate module is $\hat{G}_{32}(q) = \frac{0.312q^{-1}}{1-0.8q^{-1}+0.2q^{-2}}$, while the true module is $G^0_{32}(q) = \frac{0.39q^{-1}}{1-0.8q^{-1}+0.2q^{-2}}$.

As evidenced by the results of experiment 1, the outcomes of each correlation test within this data-generating network stabilize when the data length reaches $N = 5000$. Thus, in this experiment, we set the data length to $N = 5000$ and varied two hyperparameters, probability level α and degrees of freedom M, to analyze their influence on the test outcomes. We selected five distinct values for the hyperparameter α: $\{0.6, 0.7, 0.8, 0.9, 0.99\}$ and ten for M: $\{10, 20, ..., 100\}$.

Our primary focus is on three distinct vector-valued tests: the white noise test using $\hat{\varepsilon}_3$ (**Test 1-2**), and the cross-correlation tests using w_2 and r_3 (**Test 2-2**). Given the similarity between **Test 1-2** and **Test 1-3** observed in experiment 1, we exclusively present the results of **Test 1-2**. The confidence threshold $c_\chi(\alpha, M)$ for each vector-valued test depends on both α and M[8], whereas the calculation of the test level Q solely hinges on the degree of freedom M. Thus, for each unique value of M, we carry out 100 Monte Carlo simulations across the three tests. Fig. 8 displays the results from the three distinct tests. Within Fig. 8, the subplots (a), (b), and (c) showcase the results of the white noise test, cross-correlation test using w, and cross-correlation test using r, respectively. In each plot, every sample point represents the test level Q for a particular M of the corresponding test. The central circle of each sample point denotes the mean value from 100 Monte Carlo simulations, while the vertical line length indicates their variance. The dotted lines depict the confidence threshold for varying values of α and M.

From Fig. 8, it is evident that, for a constant α value, there's a linear positive correlation between the confidence threshold and M within the $[10, 100]$ range. Simultaneously, for α values between $[0.6, 0.99]$, the threshold's variation is relatively minimal. Moving on to the test results: The outcomes of the three tests largely lie outside the confidence interval for varied M and α values. This suggests accurate test results, discrediting the incorrect module, except in the case of the cross-correlation test using r at $M = 10$. This underscores the robustness of our tests across diverse hyperparameter selections. For the outcomes of the cross-correlation test using r at $M = 10$, they are suboptimal because the chosen M value is too small. Furthermore, the test level Q for all three tests, analogous to the confidence threshold, shows a positive correlation with M. Notably, the growth patterns of the test level and the threshold appear to be synchronized. Based on this, we anticipate that with a continued increase in M, both the test levels and confidence thresholds for the various tests will rise in tandem, ensuring the reliability of our test results for this experiment.

This experiment underscores the robustness of vector-valued test performance against varied hyperparameter selections, obviating the need for preliminary tuning to ensure test efficacy. To utilize the test, simply select values within the recommended range of α and M; this ensures reliable test outcomes. The inherent adaptability of the vector-valued test enhances its user-friendliness, offering a significant advantage in the following automatic FDD processes.

*4) **Simulation experiment 3:*** In the third simulation experiment, we aim to evaluate the sensitivity of the hypothesis tests to discrepancies between the real system and its models. The following setup is adopted:

The transfer functions in the data-generating network adhere to those detailed in Appendix D, except for the module transfer

[8] Refer to the standard chi-square distribution for $c_\chi(\alpha, M)$.

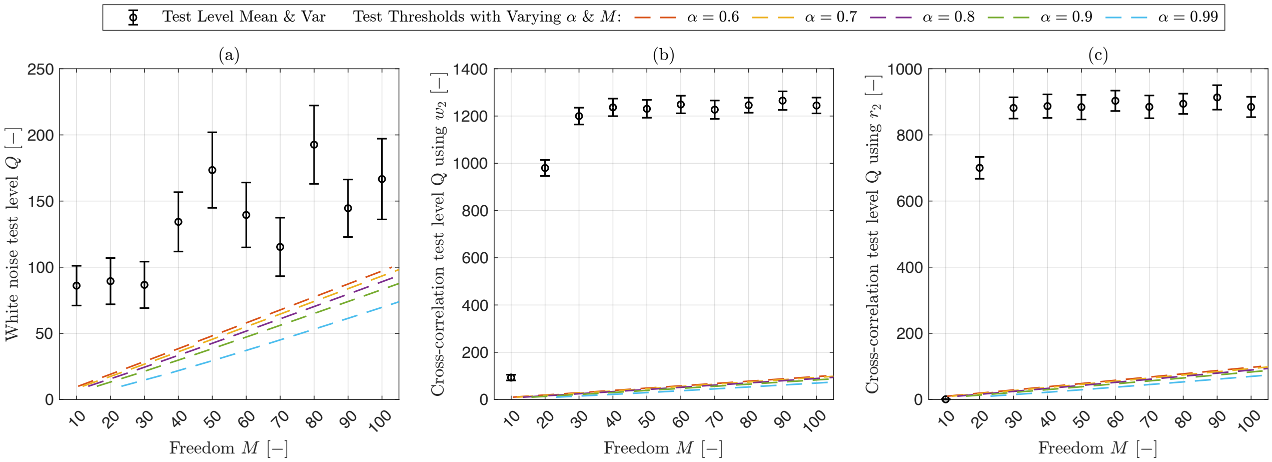

Fig. 8. Different test level Q against different confidence threshold. Each sample point represents a set of Q for a particular M; The central circle of each sample point denotes the mean value from 100 simulations, while the vertical line length indicates the variance. Dotted lines depict the threshold for varying values of α and M. (a) The white noise test using $\hat{\varepsilon}_3$; (b) The cross-correlation test using $\hat{\varepsilon}_3$ and w_2; (c) The cross-correlation test using $\hat{\varepsilon}_3$ and r_2.

function $G_{32}^0(q)$, which is defined as follows:

$$G_{32}^0(q) = \frac{1}{1 - 2 * \beta * cos(\frac{\pi}{4})q^{-1} + \beta^2 q^{-2}}.$$

The parameter β^0 is set to 0.9 for the data-generating system. For the validated local subnetwork, we set the model parameter $\hat{\beta} = \beta^0 = 0.9$, while for the invalidation case, we set the model parameter $\hat{\beta} = 0.9 - d$ where d varies within the range of $[0.001, 0.019]$. The module G_{32} is constructed in this manner to focus the deviation between the true module and its models around the resonance peak.[9]

To assess the sensitivity of the hypothesis tests, particularly for vector-valued tests, we chose 20 distinct values of d evenly distributed over the interval $[0, 0.0019]$. For every value of d, we performed 100 Monte Carlo simulations, applying four different vector-valued tests to the target subnetwork. We conducted tests with a data length of $N = 5000$, ensuring sufficient length for test result convergence. Mirroring the experiment 1, we fixed the degree of freedom at $M = 100$. The four tests encompass the vector-valued white noise test (**Test 1-2**) with the test level Q_{bp}, Ljung-Box white noise test (**Test 1-3**) with the test level Q_{lb}, vector-valued cross-correlation test (**Test 2-2**) using w_2 with the test level Q_{ew}, and vector-valued cross-correlation test using r_2 with the test level Q_{er}. The module G_{32} is always included in the target set for all selected tests, leading us to anticipate test failures when $d \neq 0$. Fig. 9 illustrates variations in the dynamic behavior of the transfer function G_{32}, as well as changes in the test levels for the various vector-valued tests for different values of d.

As depicted in the figures, the resonance peak of the transfer function G_{32} gradually escalates with an increase in d. Concurrently, consistent with our theoretical analysis,

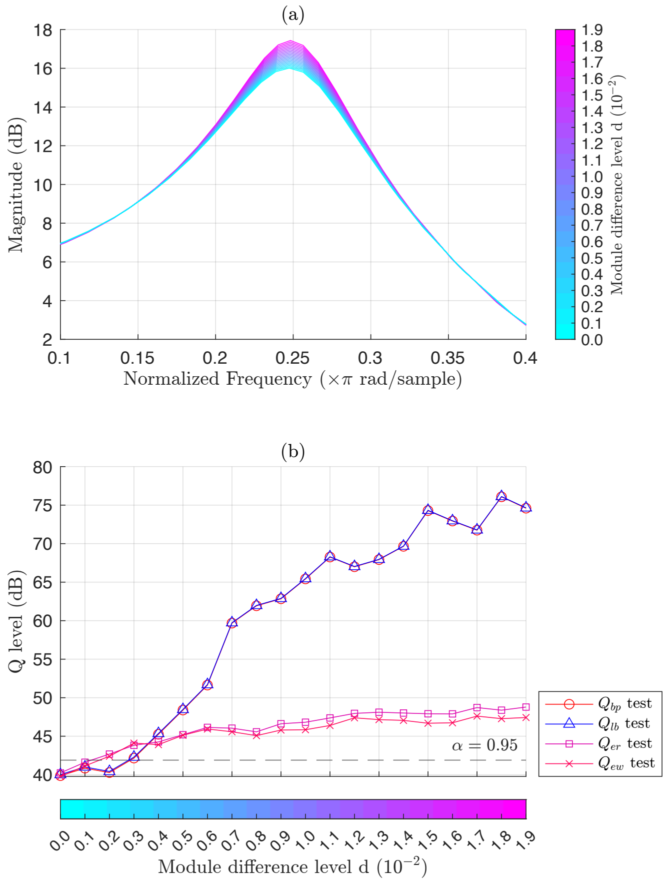

Fig. 9. (a) The bode plots of the module transfer function G_{32} with different d values; (b) The changes of the test levels Q for different vector-valued tests. The black dashed line represent the test threshold when the confidence level $\alpha = 0.95$.

[9] This simulation mimics a real-world fault in dynamic systems characterized by behavioral deviations at the resonance frequency.

there is a gradual rise in the test level Q for every vector-valued test. For the two white noise tests, the test values Q_{bp} and Q_{lb} exceed the test threshold at $d = 0.003$, and exhibited a significant increase as d ascended. The two cross-correlation test values Q_{er} and Q_{ew} exceed the test threshold already at $d = 0.002$, while the subsequent increases in d result in a smaller ascendance of test levels compared to the white noise test. The parameter β has the change ratio of $d/\beta^0 = 0.22\%$ at the point where tests initiate a failure alarm. Thus, it is evident that even minimal parameter mismatches can trigger alarms in vector-valued tests, confirming their sensitivity. Moreover, compared to the cross-correlation test, the white noise test shows a consistent sensitivity to parameter mismatches. Notably, after surpassing the test threshold, the white noise test level continues to rise significantly with increasing mismatches. This inherent property of the white noise test holds potential applications in future fault detection and diagnosis.

Generally, in the context of local network model validation, different vector-valued tests exhibit high sensitivity to invalidated modules that are included in their target module sets. Additionally, these vector-valued tests demonstrated strong convergence properties and alignment with theoretical predictions throughout the experiments. Consequently, we can consider that local subnetwork models that pass their respective vector-valued tests are deemed validated and suitable for subsequent model-based fault detection and diagnostic tasks.

IV. Local Fault Detection using Dynamic Network Framework

A. Problem definition

Fault detection (FD) is defined as the detection of the occurrence of faults in functional units of the process, which lead to undesired or intolerable behavior of the local system or even the entire system [5]. The FD task is to recognize that something is wrong, but this realization alone does not necessarily categorize or analyze the problem [6]. For the dynamic network system with the potential fault under **Assumption 2**, we first define the task of local fault detection using the dynamic network model as follows:

Definition 3. *For the target MISO subnetwork with the output node w_j, the universal set U is defined as the set that contains all the module transfer functions that needed to be monitored during the FD procedure, where $U \subseteq \{G_{jk} \mid k \in \mathcal{N}_j\}$. Then the local fault detection utilizes:*

- *the dataset $\{w_{\mathcal{N}_j}(t), w_j(t)\}$, potentially together with the external input data $r_i(t)$, from the data-generating system $\{G^0_{jk}(q)\}$ for $k \in \mathcal{N}_j$ and,*
- *the validated model $\{\hat{G}_{jk}(q)\}$ for the local subnetwork together with specific noise information,*

to detect whether there is a fault appearing in any module in set U, i.e. if $G^0_{jk}(q) \leftarrow G^f_{jk}(q)$ happens.

As elucidated in Section A, Chapter III, the target MISO subnetwork model validation aims to validate all models $\{\hat{G}_{jk}(q)\}$ in the MISO channel. Consequently with the validated models $\{\hat{G}_{jk}(q)\}$, the null hypothesis that $\{\hat{G}_{jk}(q)\} = \{G^0_{jk}(q)\}$ is accepted. If a fault on module $G^0_{jk}(q)$ causes it changing to $G^f_{jk}(q)$, i.e. $G^0_{jk}(q) \leftarrow G^f_{jk}(q)$, it follows that $\hat{G}_{jk}(q) \neq G^f_{jk}(q)$. With the validated model of the healthy data-generating system, we can apply the proposed model validation (invalidation) tests to detect the mismatch between the data-generating system and its healthy model, which detects faults that occur in the local subnetwork.

Experiment 1, as detailed in Section D, Chapter III, establishes that among the three categories of model validation tests, vector-valued tests consistently outperform their sample-wise counterparts. As a result, the ensuing discussions in this chapter, as well as in the subsequent chapter on fault diagnosis, will exclusively utilize vector-valued tests for model invalidation tasks. Concurrently, Experiments 2 and 3 from Section D, Chapter III confirm that these vector-valued tests can detect subtle system changes with minimal hyperparameter tuning. While the advantages of these vector-valued tests can guarantee their performance for the FD task, it should be noted that each test has its own prerequisites and target module sets. Hence, the primary challenge in using model invalidation for fault detection in local subnetworks lies in the wise selection of tests, tailored to specific network topology and the noise information. The remainder of this chapter will be devoted to addressing the issue of test selection.

B. Fault detection using model invalidation

Within the local network fault detection framework employing model invalidation, it is crucial that every module in U the user wishes to monitor is encompassed in the target module set of at least one chosen model validation test. This stipulation guarantees comprehensive fault detection coverage of the universal set U within the target MISO subnetwork. When an accurate noise model $\hat{H}_j(q)$ and noise topology $\mathcal{T}_H$ are both accessible, the passive white noise test, utilizing residual $\hat{\varepsilon}_j(t)$, is a viable choice for fault detection. This is because the target module set $\mathcal{S}\hat{\varepsilon}_j$ encompasses all module transfer functions in the target MISO subnetwork, i.e. $U \subseteq \{G_{jk} \mid k \in \mathcal{N}_j\} = \mathcal{S}_{\hat{\varepsilon}_j}$. Conversely, if the accurate noise model $\hat{H}_j(q)$ and noise topology $\mathcal{T}_H$ are not simultaneously available, the white noise test becomes infeasible for fault detection. Under such circumstances, one can only resort to cross-correlation tests, though they do not guarantee full coverage of all modules in the target MISO subnetwork. Consequently, employing multiple cross-correlation tests might be essential to ensure coverage of the universal set U. In such instances, the ensuing goal is to minimize the number of tests employed. Among those, passive tests are preferred over active tests.

To adopt an algorithmic method for generating optimal fault detection tests, our first step entails producing all potential model validation tests based on the available noise information. Subsequently, we aim to employ a different algorithm to select the most suitable test(s) from the available options for local subnetwork fault detection. For clarity in notation, we present the subsequent definition:

Definition 4. *In the specified local subnetwork with node w_j as the output, we define the aggregate set $\mathfrak{S}_j$ as the set comprises the target module sets associated with all model validation tests that can be used under the given noise conditions.*

Before generating the aggregate set $\mathfrak{S}_j$, the index sets which directly follow from the network and noise topology, namely $\mathcal{N}_j, \mathcal{V}_j, \mathcal{C}_i$ and $\mathcal{J}_i$, are necessary to construct the target module sets. Algorithms for generating these index sets, based solely on the network and noise topology $\mathcal{T}_G$ and $\mathcal{T}_H$, can be found in Appendix E. Once the required index sets are obtained, the target module sets corresponding to the applicable tests can be generated following Table I.

Subsequently, for the situation with both an accurate noise model $\hat{H}_j(q)$ and the noise topology $\mathcal{T}_H$, although all tests can be used, based on Table I, a single passive white noise test is already sufficient to detect faults in the entire local subnetwork (include set U) with minimal experiment cost. Therefore, in this case, we only need to add the white noise test into the set $\mathfrak{S}_j$; for the situation with only the noise topology $\mathcal{T}_H$ or the situation without any noise information, we need to add the target module sets of all applicable tests in the target subnetwork based on the information in Table I into the set $\mathfrak{S}_j$. At the same time, in order to facilitate the distinction between active and passive test target module sets when selecting the optimal test(s) in the following step, we will add a nonzero selection cost $C_{\hat{\varepsilon}_j w_i}$ (or $C_{\hat{v}_j w_i}$) to the target module sets of all passive tests, and a selection cost $C_{\hat{\varepsilon}_j r_i}$ to the target module sets of all active tests where $C_{\hat{\varepsilon}_j r_i} > C_{\hat{\varepsilon}_j w_i}$. We summarize the detailed procedure to generate the set $\mathfrak{S}_j$ in **Algorithm 1**.

Given the aggregate set $\mathfrak{S}_j$, encompassing the target module sets of all applicable tests for fault detection in a specific scenario, the task of choosing the optimal test(s) with respect to quantity and passivity can be reformulated as a combinatorial optimization problem, resembling a variant of the classic NP-hard weighted set cover problem [42].

Definition 5. *Within the target MISO subnetwork, for any module transfer function G that needs to be monitored and is part of the universal set U, there must exist at least one target module set of a selected test containing G, denoted by $G \in \mathcal{S}_{mn}$. Furthermore, given that a cost C_{mn} is assigned with each target module set, the second objective is to minimize the cumulative cost of the selected sets. This optimization problem can be formally defined as a Linear Integer Programming (LIP) problem, detailed below:*

Algorithm 1 Generate all the aggregate set $\mathfrak{S}_j$ for fault detection which contains the target module sets of applicable model validation tests in the target MISO subnetwork model with the output node w_j.

```
Input: Network topology $\mathcal{T}_G$, the level of noise information: with $\hat{H}_j(q)$, with $\mathcal{T}_H$ or with no noise information, the target MISO subnetwork with the output node $w_j$
Output: Set $\mathfrak{S}_j$, with each element in it assigned with a selection cost
Generate the sets associated with network topology, $\mathcal{N}_j$ and $\mathcal{C}_i$, $\mathcal{J}_i$, $\forall i \in \mathcal{N}_j$ as defined in Section B, Chapter II using algorithms in Appendix E;
Initialize an empty aggregate set $\mathfrak{S}_j = \emptyset$;
Based on the certain level of noise information, switch among the following different situations:
a. With $\hat{H}_j(q)$ and $\mathcal{T}_H$;
b. With $\mathcal{T}_H$;
c. No noise information;
if Case a then
    Add set $\mathcal{S}_{\hat{\varepsilon}_j}$ into $\mathfrak{S}_j$, with assigned cost $C_{\hat{\varepsilon}_j} = 1$. The cost is assigned to make the data structure of Case a consistent with it of Case b and Case c;
else if Case b then
    Add sets $\mathcal{S}_{\hat{\varepsilon}_j w_i}$, $\forall i \in \mathcal{N}_j / \mathcal{V}_j$ into $\mathfrak{S}_j$, with assigned cost to each set $C_{\hat{v}_j w_i}$;
    Add sets $\mathcal{S}_{\hat{v}_j r_i}$, $\forall i \in \mathcal{N}_j$ into $\mathfrak{S}_j$, with assigned cost to each set $C_{\hat{v}_j r_i}$;
else if Case c then
    Add sets $\mathcal{S}_{\hat{v}_j r_i}$, $\forall i \in \mathcal{N}_j$ into $\mathfrak{S}_j$, with assigned cost to each set $C_{\hat{v}_j r_i}$;
end if
```

$$\min_{x_{mn} \in \{0,1\}} \sum_{\mathcal{S}_{mn} \in \mathfrak{S}_j} x_{mn} C_{mn}$$

$$s.t. \sum_{G \in \mathcal{S}_{mn} \in \mathfrak{S}_j} x_{mn} \geq 1, \ \forall G \in \ U$$

The criterion that each module in U needs to be included within the target module set of at least one selected test is formalized as the constraint $\sum_{G \in \mathcal{S}_{mn} \in \mathfrak{S}_j} x_{mn} \geq 1$. Moreover, the optimization objective $\min_{x_{mn} \in \{0,1\}} \sum_{\mathcal{S}_{mn} \in \mathfrak{S}_j} x_{mn} C_{mn}$ is designed to prioritize passive tests over active tests when the first criterion is already satisfied.

Furthermore, we focus on an algorithmic approach to solve this combinatorial optimization problem. Utilizing the set $\mathfrak{S}_j$, which encompasses all target module sets of the applicable tests, we employ a greedy algorithm based on the *weighted set covering* problem which is detailed in **Definition 5** and originally proposed by [43]. We aim to construct a list denoted as $S_{selected}$ that encompasses the selected target module sets for FD, ensuring full coverage of the MISO subnetwork with the fewest sets and maximum passivity. During the algorithm's design phase, we introduced the following intermediate variables:

- UE: A set that stores all the uncovered module(s) in universal set U;
- $BestSet$: The best target module set which covers the most modules in UE with the least selection cost;
- $BestSetCov$: The number of modules that is covered by the $BestSet$ in set UE;
- $BestSetCostEf$: The cost to covering efficiency of the $BestSet$ which is defined as $C_{BestSet}/CurrentSetCov$;

To fully cover the universal set U, the set UE must be emptied, ensuring no module in U remains uncovered. When the set UE is non-empty, the algorithm iteratively searches the target module set from the aggregate set $\mathfrak{S}_j$ that can maximize module coverage with the least cost. During this search, we employ $BestSet$ to denote the optimal set, $BestSetCov$ to denote its coverage modules, and $BestSetCostEf$ for covering efficiency. Simultaneously, each target module set $\mathcal{S}_{mn}$ in the set $\mathfrak{S}_j$ is compared to $BestSet$. If $\mathcal{S}_{mn}$ surpasses $BestSet$ in module coverage and efficiency, then $BestSet$ is updated to the current $\mathcal{S}_{mn}$. Once an optimal set is determined after a round of search in $\mathfrak{S}_j$, it is added to $\mathcal{S}selected$. Subsequently, if UE becomes empty, indicating complete coverage of U, the algorithm concludes and outputs $S_{selected}$. Otherwise, the search continues in $\mathfrak{S}_j$ to select another target module set. The detailed procedure to address the problem defined in **Definition 5** can be found in **Algorithm 2**.

Algorithm 2 Optimal test(s) generating for fault detection based on weighted set covering problem [43]

```
Input: The universal set $U$, the aggregate set $\mathfrak{S}_j$
Output: The set $S_{selected}$
 1: Initialize an empty set $S_{selected}$, and initialize the set $UE$
    by populating it with all elements in set $U$;
 2: while $UE \neq \emptyset$ do
 3:     Initialize the intermediate variables $BestSet = \emptyset$,
        $BestSetCov = 0$ and $BestSetCostEf = \infty$;
 4:     for Each $\mathcal{S}_{mn} \in \mathfrak{S}_j$ with a cost $C_{mn}$ do
 5:         Calculate the current number of covered modules in
            $\mathcal{S}_{mn} \cap UE$;
 6:         Calculate the current covering efficiency as $C_{mn}$
            divided by the number of elements from step 5;
 7:         if The current covering efficiency from step 6 is
            smaller than $BestSetCostEf$ then
 8:             Update $BestSet$ to be $\mathcal{S}_{mn}$;
 9:             Update $BestSetCov$ to be the current number of covered modules from step 5, and
                $BestSetCostEf$ to be the current covering efficiency from step 6;
10:         end if
11:     end for
12:     Add $BestSet$ to $S_{selected}$;
13:     Remove all elements of $BestSet$ from $UE$;
14: end while
```

Remark 7. *In real-world scenarios, it is possible that certain w signals are not measurable or some r signals can not be allocated to the system. Under such circumstances, not all model validation tests for the local subnetwork can be performed. However, the primary objective of this study is to provide an optimal test selection strategy for fault detection, solely based on the network and noise topology, as well as available noise information. The user has the option to remove target module sets corresponding to unusable tests from the set $\mathfrak{S}_j$ generated by **Algorithm 1**, before executing **Algorithm 2**.*

Example 1. *If our objective is to detect a possible fault that occurs in the local MISO subnetwork with node w_1 being the output, with the data-generating system shown in Fig. 10. With the network and noise topology known, all the target module sets of the applicable tests can be generalized from **Algorithm 1** as follows:*

1) *The scenario with an accurate noise model $\hat{H}_1(q)$ and the noise topology $\mathcal{T}_H$: $\mathfrak{S}_1 = \{\mathcal{S}_{\hat{\varepsilon}_1} = \{G_{12}, G_{31}\}$ with $C_{\hat{\varepsilon}_1} = 1\}$;*
2) *The scenario with only the noise topology $\mathcal{T}_H$: $\mathfrak{S}_1 = \{\mathcal{S}_{\hat{v}_1 r_2} = \{G_{12}, G_{31}\}$ with $C_{\hat{v}_1 r_2} = 2$, $\mathcal{S}_{\hat{v}_1 r_3} = \{G_{12}, G_{31}\}$ with $C_{\hat{v}_1 r_3} = 2\}$;*
3) *The scenario with no noise information: $\mathfrak{S}_1 = \{\mathcal{S}_{\hat{v}_1 r_2} = \{G_{12}, G_{31}\}$ with $C_{\hat{v}_1 r_2} = 2$, $\mathcal{S}_{\hat{v}_1 r_3} = \{G_{12}, G_{31}\}$ with $C_{\hat{v}_1 r_3} = 2\}$.*

*In the first scenario, the only target module set $\mathcal{S}_{\hat{\varepsilon}_1}$ of the white noise test encompasses all modules within the target MISO subnetwork, thus it will be selected by **Algorithm 2**. In the remaining two scenarios, a minimum of one active test is included in $\mathcal{S}_{selected}$ for fault detection by **Algorithm 2**.*

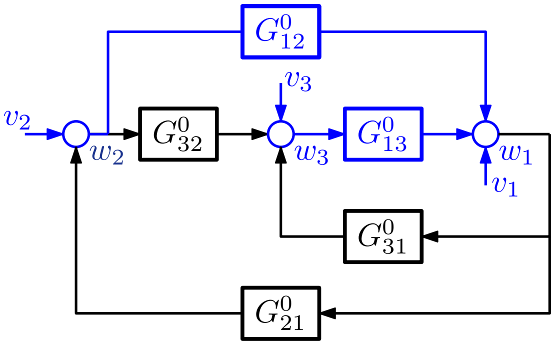

Fig. 10. Example of a 3-node data-generating network, the target MISO subnetwork contains G_{12} and G_{13} and is marked in blue. We assume the r signals can be added to the data-generating network to apply the active test.

C. Passive fault detection

The applicability of the proposed FD method is constrained by the prerequisites of the model validation tests. Specifically, in the second and third scenarios detailed in **Example 1**, if the active test is the only viable approach for model validation, it is consequently also required for fault detection. Consequently, the required r signals that are not in the data-generating system need to be allocated for the FD experiment.

However, allocating extra r signals often causes interruptions to normal system operations, thereby raising deployment challenges and risks in real-time environments. In contrast, passive methods are generally more cost-efficient with no need for extra external excitation signals. Therefore, passive methods are well suited for continuous monitoring with minimal resource burden.

The passivity of the fault detection procedure, when utilizing the model validation tests proposed in this study, hinges on the comprehensiveness of available noise information. As discussed in Section C, Chapter III, there exists a trade-off between the noise information and the experimental costs associated with fault detection. More sufficient noise information can render the detection process more passive, whereas limited noise information tends to make the procedure reliant on r signals, requiring a more active detection strategy. In the absence of both an accurate $\hat{H}_j(q)$ and the noise topology $\mathcal{T}_H$, our proposed fault detection process shifts entirely to an active approach.

To maintain a passive fault detection approach despite limited noise information, one could integrate model-based residual analysis with data-driven techniques like machine learning and pattern recognition. If an accurate noise model $\hat{H}_j(q)$ is not available, it precludes the use of passive white noise tests for monitoring the full MISO subnetwork; this limitation arises because only the predicted disturbance $\hat{v}(q)$ can be estimated from Eq. (9). Nonetheless, signal processing-based fault detection methods can also be employed. These signal-based methods first extract relevant features from $\hat{v}(q)$ and then leverage these features to monitor the full MISO subnetwork. Feature extraction can primarily occur in the time domain, frequency domain, or time-frequency domain [44], [45]. A fault can be detected in the target MISO subnetwork when the chosen feature of $\hat{v}(q)$ deviates from its nominal level and surpasses a preset confidence bound. While this enhancement in the passivity of the FD procedure will not be covered in this thesis, it is identified as a topic for future research.

Thus far, we have established a fault detection procedure based on the local dynamic network model validation. Initially, **Algorithm 1** and **Algorithm 2** autonomously select model validation tests, considering the system topology and available noise data. This process ensures the selection of the minimal tests with optimal passivity for fault monitoring and detection. Subsequently, users can engage in online fault monitoring or conduct offline fault detection experiments based on the chosen tests, after allocating the r signals that are requisite but absent in the data-generating system.

V. Fault Diagnosis using Dynamic Network Framework

A. Problem definition

Upon detecting an error in the target subnetwork, our subsequent step is to identify its root cause. In the context of the dynamic network model, this refers to the module responsible for the fault. The process to determine the location of a fault that follows fault detection is defined as *fault isolation* [4], [46]. Fault isolation is a fundamental component of the fault diagnosis process, serving as the primary objective in our model-based diagnosis within the dynamic network framework.

The task of fault isolation within a target subnetwork can be also addressed by the proposed model validation tests as the fault detection step. The model validation tests have their own target module set, and the size of these sets and the elements they contain are different as mentioned in **Proposition 1**. This allows us to locate the detected faults in the target module set of a certain test or even on a specific module by combining the results from different tests.

Example 2. *Suppose we detect a fault in the target subnetwork as shown in Fig. 11. Based on the network topology and a certain level of noise information, we have generated a set $\mathfrak{S}_5$ that contains all target module sets corresponding to applicable tests. The set $\mathfrak{S}_5$ in this case contains the following non-repeating elements:*

$$\begin{aligned}\mathcal{S}_{\hat{\varepsilon}_5} &: \{G_{51}, G_{52}, G_{53}, G_{54}\},\\ \mathcal{S}_{\hat{\varepsilon}_5 w_1} &: \{G_{51}, G_{52}, G_{53}\},\\ \mathcal{S}_{\hat{\varepsilon}_5 r_3} &: \{G_{53}, G_{54}\}.\end{aligned}$$

Assuming that there is only one fault in this target subnetwork, we can try to isolate the fault on a certain module by combining the applicable tests.

First, we consider the situation where the fault occurs on G_{54}, and we can combine tests 1 and 2 to isolate the fault. According to the relationship between the target module sets of these two tests $\mathcal{S}_{\hat{\varepsilon}_5}/\mathcal{S}_{\hat{\varepsilon}_5 r_3} = \{G_{54}\}$, when test 1 fails and test 2 passes, we can determine that the fault location is on G_{54}.

Consider another situation where the fault occurs on G_{53}, we can use test 2 and test 3 to isolate the fault. According to the relationship between the target module sets of these two tests $\mathcal{S}_{\hat{\varepsilon}_5 w_1} \cap \mathcal{S}_{\hat{\varepsilon}_5 r_3} = \{G_{53}\}$, when tests 2 and 3 fail at the same time, we can determine that the faulty module is G_{53}.

Finally, we consider the situation where the fault occurs on G_{51}. In this case, we can not isolate the fault on G_{51} by using the model validation test. Because no matter how we combine and operate the combination (taking union, intersection, or difference set) of the above three target module sets, we can not obtain a set that only contains module G_{51}. The smallest

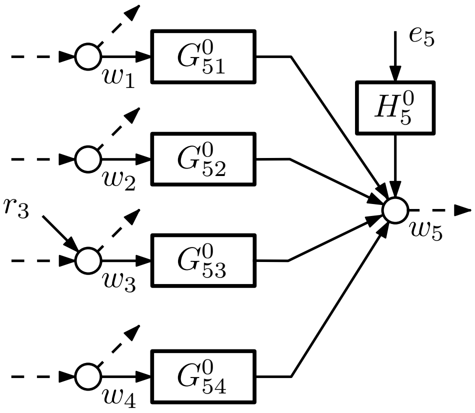

Fig. 11. Example of a target MISO subnetwork with 4 in-neighbors.

set we can get that contains module G_{51} is $\{G_{51}, G_{52}\}$, that is, we use test 1 and test 3 $\mathcal{S}_{\hat{\varepsilon}_5}/\mathcal{S}_{\hat{\varepsilon}_5 r_3} = \{G_{51}, G_{52}\}$. When test 1 fails and test 3 passes, we can isolate the fault in set $\{G_{51}, G_{52}\}$, but we can not know which module is the faulty one specifically.

As can be seen from **Example 2**, for a target MISO subnetwork, after we obtain all the target module sets $\mathfrak{S}_j$ of applicable model validation tests, we can try to find smaller sets by performing set operations such as taking union, intersection or difference set, using all elements in $\mathfrak{S}_j$. The smaller set $\mathcal{S}_{pq}$ contains fewer modules or even one module after sufficient set operations. In the process of obtaining the set $\mathcal{S}_{pq}$, the tests corresponding to the used target module sets are the tests we need to use in the process of fault isolation. At the same time, we also found that through this set operation, for some modules G_{ja}, we can find a set that only contains one module G_{ja} solely; but for other modules G_{jb}, we cannot find a set that only contains one module G_{jb} solely through these set operations. This shows that for the kind of modules G_{ja}, one fault can be completely isolated to a single module; but for kind of modules G_{jb}, the fault can only be isolated within a range containing G_{jb} and other modules. To facilitate further discussion, we give the following definitions:

Definition 6. *Within the context of model-based fault isolation using model invalidation, if a module set $\{G_{ji}\}$ that only contains G_{ji} can be derived by using every pair of target module sets $\mathcal{S}_{pq}$ and $\mathcal{S}_{mn}$ from the set $\mathfrak{S}_j$ to do the following set operations:*

- *taking union: $\mathcal{S}_{pq} \cup \mathcal{S}_{mn}$,*
- *taking intersection: $\mathcal{S}_{pq} \cap \mathcal{S}_{mn}$,*
- *taking difference: $\mathcal{S}_{pq}/\mathcal{S}_{mn}$, $\mathcal{S}_{mn}/\mathcal{S}_{pq}$,*

then we determine the fault as isolable on module G_{ji}. Conversely, if such a set $\{G_{ji}\}$ cannot be derived for module G_{ji}, the fault is deemed non-isolable on G_{ji}.

Fault isolability is differently defined in existing literature [5], [47]. Conventionally, two faults are considered isolable if their induced changes in system output can be differentiated. A pair of isolable faults can be different in terms of location, time of occurrence, severity, etc. Nonetheless, within the dynamic network framework, where structural propriety is emphasized, our primary focus is on the spatial location of the fault. Consequently, we define fault isolability solely on the basis of whether a fault is spatially distinguishable from other faults.

Remark 8. *In this study, to streamline our fault isolation analysis while maintaining generalizability, we operate under the assumption that only one fault is detected at a time in the target subnetwork. This assumption offers the potential for relaxation in subsequent research.*

As defined in **Definition 6**, fault isolability on each module can be analyzed by taking operations on all target module sets in the aggregate set $\mathfrak{S}_j$, which initially depends solely on the network and noise topology $\mathcal{T}_G$, $\mathcal{T}_H$, and the available noise information. This enables the analysis of the isolability of each module in the target MISO subnetwork prior to any FDD experiment, which means we are able to know on which modules a fault can be isolated only with topology and noise information of the system. Therefore, we initially examine each module's fault isolability within the target MISO subnetwork in the following section, subsequently proposing a fault diagnosis procedure informed by this isolability information.

B. Fault isolation synthesis

1) ***Isolability analysis****:* In this section, our objective is to formulate an algorithmic approach that uses the topology of the target system and available noise data to determine the isolability of every module within the target MISO subnetwork. As highlighted in **Definition 6**, the isolability analysis depends on set $\mathfrak{S}_j$, which includes the target module sets for all applicable model validation tests. Given that the set $\mathfrak{S}_j$ generated in **Algorithm 1** is designed for the fault detection algorithm, the selection cost allocated to each target module set within set $\mathfrak{S}_j$ is not required during fault isolation analysis. Consequently, focusing on the fault isolation phase, we construct a set comprising the target module sets for all pertinent tests from **Algorithm 3**. This algorithm mirrors **Algorithm 1**, albeit without assigning a selection cost to each target module set. The resulting set from **Algorithm 3** is labeled $\mathfrak{S}_j^{full}$, distinguishing it from set $\mathfrak{S}_j$ from the fault detection phase.

As the cross-correlation tests using r signals have the smallest target module sets compared to the other two kinds of tests as mentioned in **Proposition 1**, they are preferable for fault isolation to pinpoint more accurate fault locations. Therefore, the target module sets of all the cross-correlation tests using r signals are added in set $\mathfrak{S}_j^{full}$ even if certain r signals are not presented.

Once the aggregate set $\mathfrak{S}_j^{full}$ is obtained, a combinatorial optimization algorithm can complete the fault isolability analysis for each module transfer function in the target MISO

Algorithm 3 Generate all the aggregate set $\mathfrak{S}_j^{full}$ for fault isolation which contains the target module sets of all applicable model validation tests in the target MISO subnetwork model with the output node w_j.

```
Input: Network topology $\mathcal{T}_G$, the level of noise information: with $\hat{H}_j(q)$, with $\mathcal{T}_H$ or with no noise information, the target MISO subnetwork with the output node $w_j$
Output: Set $\mathfrak{S}_j^{full}$
Generate the sets associated with network topology, $\mathcal{N}_j$ and $\mathcal{C}_i$, $\mathcal{J}_i$, $\forall i \in \mathcal{N}_j$ as defined in Section B, Chapter II using algorithms in Appendix E;
Initialize an empty aggregate set $\mathfrak{S}_j^{full} = \emptyset$;
Based on the certain level of noise information, switch among the following different situations:
    a. With $\hat{H}_j(q)$ and $\mathcal{T}_H$;
    b. With $\mathcal{T}_H$;
    c. No noise information;
if Case a then
    Add set $\mathcal{S}_{\hat{\varepsilon}_j}$ into $\mathfrak{S}_j^{full}$;
    Add sets $\mathcal{S}_{\hat{\varepsilon}_j w_i}$, $\forall i \in \mathcal{N}_j$ into $\mathfrak{S}_j^{full}$;
    Add sets $\mathcal{S}_{\hat{\varepsilon}_j r_i}$, $\forall i \in \mathcal{N}_j$ into $\mathfrak{S}_j^{full}$;
else if Case b then
    Add sets $\mathcal{S}_{\hat{v}_j w_i}$, $\forall i \in \mathcal{N}_j / \mathcal{V}_j$ into $\mathfrak{S}_j^{full}$;
    Add sets $\mathcal{S}_{\hat{v}_j r_i}$, $\forall i \in \mathcal{N}_j$ into $\mathfrak{S}_j^{full}$;
else if Case c then
    Add sets $\mathcal{S}_{\hat{v}_j r_i}$, $\forall i \in \mathcal{N}_j$ into $\mathfrak{S}_j^{full}$;
end if
```

subnetwork $\{G_{jk}\}$ which is denoted as the universal set U. The following definition is required for further isolability analysis:

Definition 7. *The fault isolation resolution for module G_{ji} is defined as the smallest module set for that module, generated through set operations over the elements in $\mathfrak{S}_j^{full}$.*

We aim to develop an algorithm that computes the resolution for each module G_{ji} within U. If the resolution for module G_{ji} consists solely of the module itself, then, by definition, a fault is isolatable on G_{ji}; otherwise, a fault is not isolatable on G_{ji}. Subsequently, to streamline our notation, we introduce the following variables:

- The list $\mathbb{S}$ contains the resolution for every module G_{ji} in U. The function $\mathbb{S}[G_{ji}]$ retrieves the resolution associated with G_{ji}.
- The list $\mathbb{O}$ encompasses all set operations performed to obtain the resolution for every module G_{ji} in U. The function $\mathbb{O}[G_{ji}]$ retrieves all the set operations involved in determining the resolution for module G_{ji}.

Example 3. *To clarify the defined notations, consider the target MISO subnetwork in Fig. 11, where the set $\mathfrak{S}_5^{full}$ contains the following non-repeating elements:*

$$\mathcal{S}_{\hat{\varepsilon}_5 w_1} : \{G_{51}, G_{52}, G_{53}\},$$
$$\mathcal{S}_{\hat{\varepsilon}_5 r_1} : \{G_{51}\},$$
$$\mathcal{S}_{\hat{\varepsilon}_5 r_2} : \{G_{52}\}.$$

When analyzing the isolability on module G_{53}, the list $\mathbb{S}$ will give the resolution $\mathbb{S}[G_{53}] = \{G_{53}\}$ which indicates a fault is isolable on module G_{53}. The set operations to reach this isolability is shown in list $\mathbb{O}$, where $\mathbb{O}[G_{53}]$ contains the following elements:

$$\textit{Step 1: } \mathcal{S}_{\hat{\varepsilon}_5 w_1} / \mathcal{S}_{\hat{\varepsilon}_5 r_1} = \{G_{52}, G_{53}\},$$
$$\textit{Step 2: } \{G_{52}, G_{53}\} / \mathcal{S}_{\hat{\varepsilon}_5 r_2} = \{G_{53}\}.$$

The task of determining the resolution for each module with the corresponding set operations can be solved in an algorithmic approach. The approach involves a specialized search for all elements in set U, derived from set operations applied to the initial sets within $\mathfrak{S}_j^{full}$. Our objective is to identify a 'minimal' set for each module G_{ji} in the universal set U — a set which includes G_{ji} and is derived from unions, intersections, or differences of sets in $\mathfrak{S}_j^{full}$. Our proposed algorithm employs a *breadth-first search* strategy, iterating over all possible set operations for each pair of sets within the aggregate set $\mathfrak{S}_j^{full}$. Subsequently, sets resulting from these operations, if smaller, are incorporated into the aggregate set $\mathfrak{S}_j^{full}$, forming the foundation for subsequent resolution searches. Ultimately, if no further smaller sets are identified, the search concludes, yielding the resolution for each module within set U. The procedure is detailed in **Algorithm 4**, though it might not be optimized for computational efficiency.

*2) **Allocation of extra r signals for fault isolation**:* The lists $\mathbb{S}$ and $\mathbb{O}$, obtained from **Algorithm 4**, denote the fault isolability for each module and the essential tests to achieve this isolability. As the algorithms generating $\mathbb{S}$ and $\mathbb{O}$ employ all active tests to attain the smallest resolution, the selected tests for the fault isolation procedure may require some r signals absent in the data-generating system. Consequently, we must allocate those additional excitation signals r to the data-generating system for effective fault isolation.

The challenge of allocating additional excitation signals r to aid fault isolation can be framed as: Given a consistently estimated model of a data-generating local MISO subnetwork and its corresponding sets $\mathbb{S}$ and $\mathbb{O}$ derived from **Algorithm 4**, for a given target module G_{ji}, how should we allocate external excitation signals to subsequent model validation tests to precisely isolate any potential fault on G_{ji}?

To address this problem, we adopt a step-by-step methodology:

1) For the target module G_{ji}, we consult its resolution $\mathbb{S}[G_{ji}]$ to ascertain if the fault can be isolated on G_{ji};
2) If a fault is isolatable on G_{ji}, we examine the set operations documented in $\mathbb{O}[G_{ji}]$ and identify the test target

Algorithm 4 Fault isolation resolution search

```
Input: The universal set $U$, the aggregate set $\mathfrak{S}_j^{full}$
Output: The list $\mathbb{S}$ and the list $\mathbb{O}$
 1: Initialize $\mathbb{S}$ so that the isolation resolution for each module
    equal to $U$;
 2: Initialize $\mathbb{O}$ as an empty list with the proper structure;
 3: repeat
 4:   Initialize a flag variable $Updated = $ False;
 5:   for Each pair of sets $\mathcal{S}_{pq}$, $\mathcal{S}_{mn}$ in $\mathfrak{S}_j^{full}$ do
 6:     Compute the union, intersection, and difference:
        $\mathcal{S}_{pq} \cup \mathcal{S}_{mn}$, $\mathcal{S}_{pq} \cap \mathcal{S}_{mn}$, $\mathcal{S}_{pq} / \mathcal{S}_{mn}$, $\mathcal{S}_{mn} / \mathcal{S}_{pq}$. Mark
        the result as set $\mathcal{S}_{new}$;
 7:     for Each computed $\mathcal{S}_{new}$ do
 8:       if $\mathcal{S}_{new} = \emptyset$ or $\mathcal{S}_{new} \in \{\mathcal{S}\}_j^{full}$ then
 9:         Skip to the next iteration;
10:       end if
11:       Add $\mathcal{S}_{new}$ to $\mathfrak{S}_j^{full}$;
12:       for Each module $G \in \mathcal{S}_{new}$ do
13:         if $size(\mathcal{S}_{new}) < size(\mathbb{S}[G])$ then
14:           Update $\mathbb{S}[G]$ to be $\mathcal{S}_{new}$;
15:           Record the operations used to obtain $\mathcal{S}_{new}$
              from step 5 as '$\mathcal{S}_{pq}$ operation $\mathcal{S}_{mn} = \mathcal{S}_{new}$'
              and save it in $\mathbb{O}[G]$;
16:           Set $Updated = $ True to indicate a smaller set
              is found in this iteration;
17:         end if
18:       end for
19:     end for
20:   end for
21: until not $Updated$, as no smaller sets can be found.
```

sets used in the set operations with their corresponding cross-correlation tests using r signals;

3) Verify if the r signals, identified from tests in step 2, were previously present in the data-generating subnetwork. Any r signals not found within the data-generating subnetwork represent the extra r signals required for the fault isolation experiment.

Example 4. *Consider the data-generating MISO subnetwork depicted in Fig. 11, where the set $\mathfrak{S}_5^{full}$ contains the following non-repeating elements:*

$$\mathcal{S}_{\hat{\varepsilon}_5} : \{G_{51}, G_{52}, G_{53}, G_{54}\},$$
$$\mathcal{S}_{\hat{\varepsilon}_5 r_2} : \{G_{51}\},$$
$$\mathcal{S}_{\hat{\varepsilon}_5 r_3} : \{G_{53}, G_{54}\}.$$

For module G_{52} under fault isolation task, an isolation resolution of $\mathbb{S}[G_{52}] = G_{52}$ from set $\mathbb{S}$ indicates the feasibility of isolating a potential fault on module G_{52}. Concurrently, if the associated $\mathbb{O}[G_{52}]$ represents the following:

$$\textit{Step 1: } \mathcal{S}_{\hat{\varepsilon}_5} / \mathcal{S}_{\hat{\varepsilon}_5 r_3} = \{G_{51}, G_{52}\},$$
$$\textit{Step 2: } \{G_{51}, G_{52}\} / \mathcal{S}_{\hat{\varepsilon}_5 r_1} = \{G_{52}\}.$$

it suggests that excitation signals r_1 and r_3 are required to isolating a potential fault on module G_{52}. Given that solely r_3 is present in the data-generating system, there's a necessity to allocate an additional r signal on node 1 for the fault isolation procedure.

C. More accurate fault isolation based on test levels

Based on the above analysis, **Algorithm 4** aids in evaluating the isolability of the target module G_{ji} within a specified target MISO subnetwork. It becomes feasible to determine the requisite external excitation signals for the fault isolation procedure and allocate extra signals as needed for the isolation experiment. It should be noted, though, that in certain dynamic network models, isolability is not guaranteed for some module transfer functions.

Example 5. *Consider the data-generating network in Fig. 12, the focus is on the target MISO subnetwork surrounding the output node w_3. For the target MISO subnetwork, there exist validated models for the module transfers; however, there is an absence of prior knowledge regarding noise information. The set $\mathfrak{S}_3^{full}$ only contains two elements in this case:*

$$\mathcal{S}_{\hat{\varepsilon}_3 r_2} : \{G_{32}, G_{34}\},$$
$$\mathcal{S}_{\hat{\varepsilon}_3 r_4} : \{G_{32}, G_{34}\}.$$

Consequently, fault isolation is limited to the scope of $\{G_{32}, G_{34}\}$ by utilizing cross-correlation tests based on $\hat{\varepsilon}_3, r_2$ or $\hat{\varepsilon}_3, r_4$, while both tests have the same target the module set $\{G_{32}, G_{34}\}$.

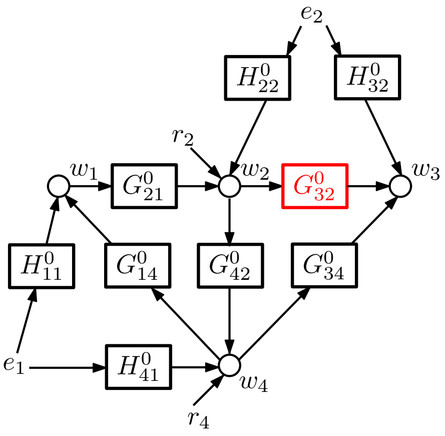

Fig. 12. Example network with target local MISO subnetwork around output node w_3. A potential fault might occur on module G_{32} (marked in red).

In **Example 5**, the presented case highlights the absence of fault isolability for the target module G_{32}. The reason for this is that within the target subnetwork, the target module sets, $\mathcal{S}_{\hat{\varepsilon}_3 r_2}$ and $\mathcal{S}_{\hat{\varepsilon}_3 r_4}$, from applicable correlation tests are identical. Consequently, **Algorithm 4** can not produce a smaller set through its set operations. If a fault occurs on module G_{32}, both the test employing $\hat{\varepsilon}_3, r_2$ and the one using $\hat{\varepsilon}_3, r_4$ would conclude a test failure. Based on this result, one can only conclude that a fault occurs within the $\{G_{32}, G_{34}\}$ scope. Despite the target module sets $\mathcal{S}_{\hat{\varepsilon}_3 r_2}$ and $\mathcal{S}_{\hat{\varepsilon}_3 r_4}$ being identical,

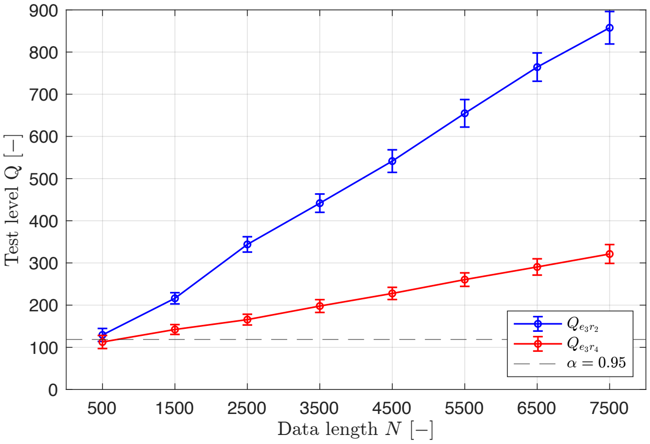

Fig. 13. Different test levels Q against the confidence threshold with $\alpha = 0.95$. The blue line shows the test levels $Q_{\hat{\varepsilon}_3 r_2}$ of the cross-correlation test using $\hat{\varepsilon}_3, r_2$, and the red line shows the test levels $Q_{\hat{\varepsilon}_3 r_4}$ of the cross-correlation test using $\hat{\varepsilon}_3, r_4$. The central circle of each sample point denotes the mean value from 100 simulations, while the vertical line length indicates the variance.

the corresponding tests exhibit differences in their test levels during experiments. As illustrated in Fig. 13, we conducted experiments with the data-generating network from Fig. 12, using tests based on $\hat{\varepsilon}_3, r_2$ and $\hat{\varepsilon}_3, r_4$. This experiment were carried out under the same conditions as Experiment 2 from Section D, Chapter III, wherein the module G_{32} is faulty and the other modules are healthy. In Fig. 13, the blue line indicates the test level $Q_{\hat{\varepsilon}_3 r_2}$ for the test using $\hat{\varepsilon}_3, r_2$, while the red line represents $Q_{\hat{\varepsilon}_3 r_4}$ for the test employing $\hat{\varepsilon}_3, r_4$. We conducted 100 Monte Carlo simulations, with the mean and variance of these displayed at each sample point in Fig. 13. Observably, both tests indicate failures, as their test levels surpass the confidence threshold. Yet, a marked distinction is evident between the test levels of the two tests, a disparity that amplifies with increasing data length N. The variability in test levels among different tests with the same target module sets can be attributed to the relative positions of the test signals and the fault in the system. r_2 is closer to the faulty module G_{32}, and the corresponding test level $Q_{\hat{\varepsilon}_3 r_2}$ stays larger than $Q_{\hat{\varepsilon}_3 r_4}$. The observed difference can aid in further fault isolation, especially when a module lacks isolability. While this study does not delve into the specific relationship between relative position and fault isolation, we consider its exploration crucial for future work.

Although the quantitative information of test levels hasn't been applied to fault isolation yet, we have developed a fault diagnosis procedure using local dynamic network model validation, with a focus on fault isolation in this chapter. Initially, **Algorithm 3** and **Algorithm 4** establish the isolation resolution for each module in the target MISO subnetwork, which uses only the information about system topology and available noise information. This stage also documents the necessary tests to achieve this resolution. Users then must allocate the additional excitation signals r to the data-generating system, in order to isolate faults in the target module G_{ji}. Ultimately, the combined results of all chosen tests determine whether the detected fault falls within the resolution scope of G_{ji}.

VI. Numerical Illustration

A. Experiment setup

In this chapter, we provide a numerical illustration of our proposed model-based fault detection and diagnosis procedure, utilizing the data-generating network depicted in Fig. 14. We have chosen the target MISO subnetwork with w_1 as the output. This includes in-neighbors w_2, w_3, and modules G^0_{12}, G^0_{13}. Disturbances v_1, v_2, v_3 on each node originate from independent innovation sources, detailed as:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} H^0_{11} & 0 & 0 \\ 0 & H^0_{22} & 0 \\ 0 & 0 & H^0_{33} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}, \tag{26}$$

where $Var(e_1) = 0.1, Var(e_2) = 0.2, Var(e_3) = 0.3$. Specifications regarding the module transfer functions and noise models can be found in Appendix F. For a streamlined simulation, we assume that the models are given and that the validated module transfer function matches the real transfer function. To prevent any additional interference, we assume that all modules outside the target MISO subnetwork are accurate in the subsequent experiments. If the target module G^0_{12} in the data-generating system is healthy, we assume its consistency with the validated model. However, when a fault arises in the target module G^0_{12}, the module changes with a damping effect on the resonance peak of the original module transfer function, as shown in Fig. 15.

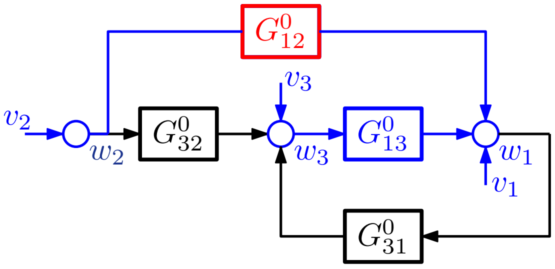

Fig. 14. Example of a 3-node data-generating network, the target MISO subnetwork contains G_{12} and G_{13}, which is marked in blue. The target module for fault isolation is G_{12} which is marked in red.

To simplify notation, we will conduct the complete fault detection and isolation procedure across three scenarios, each reflecting a different level of noise information:

- **Case 1**: The accurate noise model $\hat{H}_{11}(q)$ for the target MISO subnetwork is available, with known noise topology $\mathcal{T}_H$;
- **Case 2**: Only the noise topology $\mathcal{T}_H$ is available;
- **Case 3**: There is no available noise information.

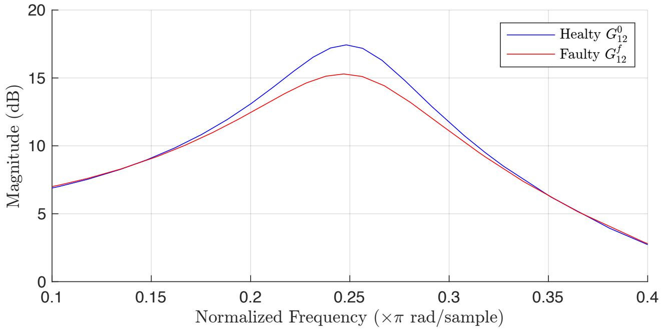

Fig. 15. The bode plots of the healthy module G^0_{12} and designed faulty module G^f_{12}.

The FDD procedure will encompass fault detection, fault isolability analysis, allocation of extra r signals, and fault isolation experiment across all three cases.

B. Numerical Illustration on Fault Detection

Prior to conducting the fault detection experiments, we can employ **Algorithm 1** and **Algorithm 2** as outlined in Section B, Chapter IV, to select the optimal test for fault detection. Utilizing the network topology $\mathcal{T}_G$ from Fig. 14, and integrating it with noise information from Case 1, Case 2, and Case 3, we can derive the aggregate sets $\mathfrak{S}_1$ for fault detection via **Algorithm 1**.

Subsequently, **Algorithm 2** processes $\mathfrak{S}_1$ for each case, producing the optimal test for fault detection. Table III summarizes the output set $\mathfrak{S}_1$ and the optimal tests across all three cases. For Case 1, the algorithm selects the white noise test when there is sufficient noise information since the test has the largest target module set to cover the entire MISO channel with optimal passivity. In Case 2, the chosen test is the cross-correlation test using $\hat{\varepsilon}_1$ and w_2. However, an alternative could be another cross-correlation test employing $\hat{\varepsilon}_1$ and w_3, given their identical target module sets that encompass the complete MISO channel. The tests based on w signals also have lower experiment costs compared to other active tests in set $\mathfrak{S}_1$. For Case 3, since set $\mathfrak{S}_1$ exclusively comprises active tests, the algorithm opts for the cross-correlation test that uses $\hat{\varepsilon}_1$ and r_2, with a target module set capable of covering the full MISO channel. Given that r_2 is absent from the data-generating network (see Fig. 14), it is necessary to add the r_2 signal during fault detection experiments for Case 3. For subsequent fault detection experiments, the introduced r_2 signal is designed as a white noise signal, characterized by $Var(r_2) = 5$.

Subsequently, we employ the selected fault detection tests, for each case, to perform a fault detection simulation experiment that monitors the entire target subnetwork. The simulation runs with a total data length of $N = 10000$. Initially, the module G^0_{12} in the data-generating system is configured to a healthy state. However, post $N = 5001$, it transitions to the faulty state G^f_{12}, as depicted in Fig. 15.

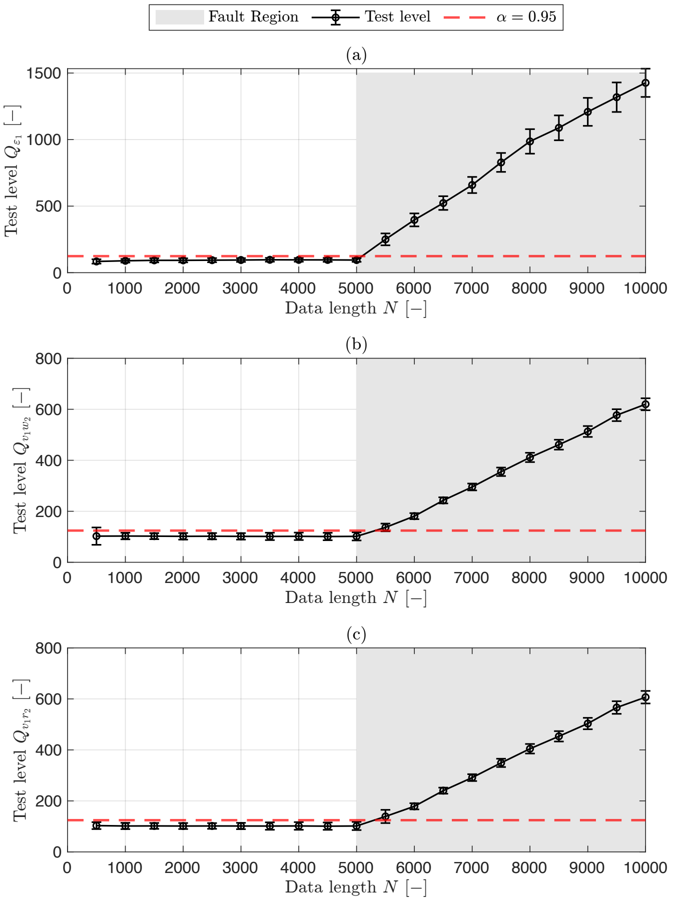

Fig. 16. Monitoring a fault with the validation tests in the target MISO subnetwork as in Fig. 14. Different test levels Q against the confidence threshold with $\alpha = 0.95$. The central circle of each sample point denotes the mean value from 100 simulations, while the vertical line length indicates the variance. (a) The test level $Q_{\hat{\varepsilon}_1}$ for Case 1; (b) The test level $Q_{\hat{v}_1 w_2}$ for Case 2; (c) The test level $Q_{\hat{v}_1 r_2}$ for Case 3.

Test levels are computed at intervals of 500 steps during the whole experiment. Each time before $N \leq 5000$ when computing the test level, all measured data is used; after $N > 5000$, the test level is computed only based on the newest 5000 data point. The experiment results for all three cases are illustrated in Fig. 16. The experiment incorporated 100 Monte Carlo simulations, the center of each sample point in Fig. 16 denotes the mean value of 100 simulations, while the vertical line indicates its variance. As shown in Fig. 16, prior to the fault onset at $N = 5001$, all three case test levels consistently stay below the confidence threshold. Post the fault event at $N = 5001$, the test levels for all cases swiftly surpass the confidence threshold, indicating that a fault within the local subnetwork is detected. The white noise test depicted in Fig. 16 (a) demonstrates greater sensitivity compared to the cross-correlation tests in Fig. 16 (b) and Fig. 16(c), given its faster

TABLE III
THE OUTPUTS OF DIFFERENT ALGORITHMS IN THE FDD PROCEDURE

	$\mathfrak{S}_1$	FD Test(s)	$\mathfrak{S}_1^{full}$	FI Test(s)
Case 1	$\mathcal{S}_{\hat{\varepsilon}_1} = \{G_{12}, G_{13}\}$	$\mathcal{S}_{\hat{\varepsilon}_1}$	$\mathcal{S}_{\hat{\varepsilon}_1} = \{G_{12}, G_{13}\}$, $\mathcal{S}_{\hat{\varepsilon}_1 w_2} = \{G_{12}, G_{13}\}, \mathcal{S}_{\hat{\varepsilon}_1 w_3} = \{G_{12}, G_{13}\}$, $\mathcal{S}_{\hat{\varepsilon}_1 r_2} = \{G_{12}, G_{13}\}, \mathcal{S}_{\hat{\varepsilon}_1 r_3} = \{G_{13}\}$	$\mathcal{S}_{\hat{\varepsilon}_1 r_2}$ & $\mathcal{S}_{\hat{\varepsilon}_1 r_3}$
Case 2	$\mathcal{S}_{\hat{v}_1 w_2} = \{G_{12}, G_{13}\}, \mathcal{S}_{\hat{v}_1 w_3} = \{G_{12}, G_{13}\}$, $\mathcal{S}_{\hat{v}_1 r_2} = \{G_{12}, G_{13}\}, \mathcal{S}_{\hat{v}_1 r_3} = \{G_{13}\}$	$\mathcal{S}_{\hat{v}_1 w_2}$	$\mathcal{S}_{\hat{v}_1 w_2} = \{G_{12}, G_{13}\}, \mathcal{S}_{\hat{v}_1 w_3} = \{G_{12}, G_{13}\}$, $\mathcal{S}_{\hat{v}_1 r_2} = \{G_{12}, G_{13}\}, \mathcal{S}_{\hat{v}_1 r_3} = \{G_{13}\}$	$\mathcal{S}_{\hat{v}_1 r_2}$ & $\mathcal{S}_{\hat{v}_1 r_3}$
Case 3	$\mathcal{S}_{\hat{v}_1 r_2} = \{G_{12}, G_{13}\}, \mathcal{S}_{\hat{v}_1 r_3} = \{G_{13}\}$	$\mathcal{S}_{\hat{v}_1 r_2}$	$\mathcal{S}_{\hat{v}_1 r_2} = \{G_{12}, G_{13}\}, \mathcal{S}_{\hat{v}_1 r_3} = \{G_{13}\}$	$\mathcal{S}_{\hat{v}_1 r_2}$ & $\mathcal{S}_{\hat{v}_1 r_3}$

growing test level. No evident performance difference in fault detection is shown between the active test in Fig. 16 (c) and the passive tests in Fig. 16 (a) and Fig. 16 (b).

Therefore, we conclude that this experiment validates the effectiveness of **Algorithm 1** and **Algorithm 2**. Furthermore, it underscores the inherent trade-off between noise information and test passivity. For Case 1, with abundant noise information, the passive white noise test is directly applicable for local fault monitoring and detection; For Case 2, given the availability of the noise topology $\mathcal{T}_H$, the passive cross-correlation test using w_2 can be applied; For Case 3, owing to the absence of noise information, it necessitates the introduction of additional r_2 signal in the data-generating system for fault detection purpose.

C. Numerical Illustration on Fault Isolation

Upon fault detection in the target MISO subnetwork, we perform fault isolation to determine where the fault originated. Initially, using **Algorithm 3** (refer to Section B, Chapter V), we generate the aggregate set $\mathfrak{S}_1^{full}$, as shown in Table III. Subsequently, using $\mathfrak{S}_1^{full}$ as input in **Algorithm 4**, the output reveals $\mathbb{S}[G_{12}] = \{G_{12}\}$ and $\mathbb{S}[G_{13}] = \{G_{13}\}$, confirming fault isolability for both modules in the target subnetwork. Consequently, with G_{12} considered as the target module, tests for fault isolation are formulated using the set $\mathbb{O}$ from **Algorithm 4**. These tests are further detailed in Table III.

For each case, based on the selected tests, we carry out the next fault isolation simulation experiments. Having identified a faulty module in the target MISO subnetwork from previous fault detection, we proceed to conduct a fault isolation experiment of length $N = 5000$, with test levels being calculated every 500 steps and each time using all measured data. Given that a fault has already been identified in the target MISO subnetwork, the ensuing fault isolation experiments can performed with a reduced horizon N.

Based on this experimental setup, we executed 100 Monte Carlo simulations; the results are illustrated in Fig. 17. Despite differences in the selected tests across the three cases, the results, as evident from Fig. 17 (a), (b), and (c), are similar. In Fig. 17, Test 1 (marked in black) corresponds to the test

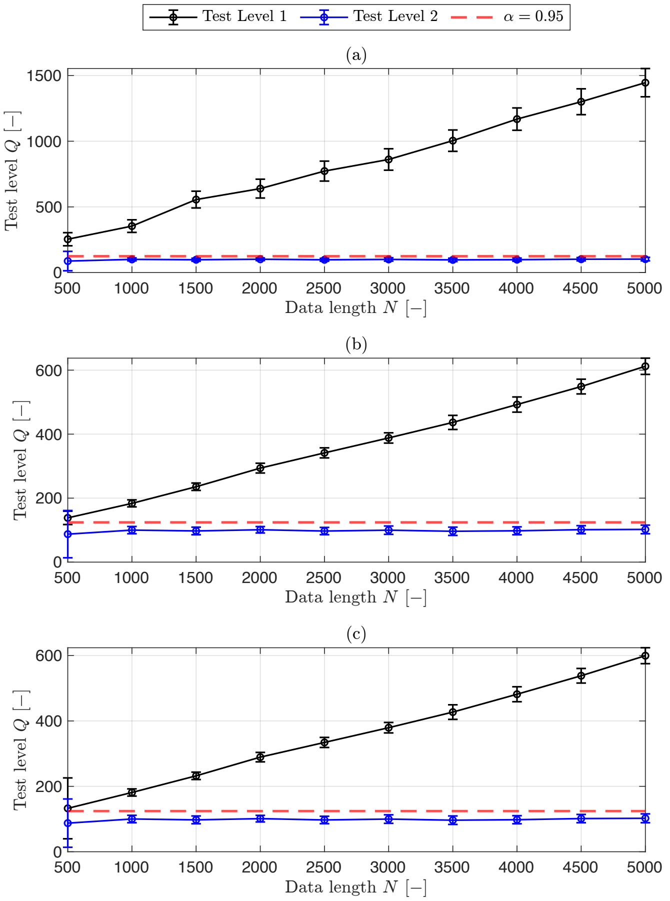

Fig. 17. Isolating a fault with the validation tests in the target MISO subnetwork as in Fig. 14. (a) The test levels $Q_{\hat{\varepsilon}_1}$ (in black) and $Q_{\hat{\varepsilon}_1 r_3}$ (in blue) for Case 1; (b) The test level $Q_{\hat{v}_1 w_2}$ (in black) and $Q_{\hat{v}_1 r_3}$ (in blue) for Case 2; (c) The test level $Q_{\hat{v}_1 r_2}$ (in black) and $Q_{\hat{v}_1 r_3}$ (in blue) for Case 3.

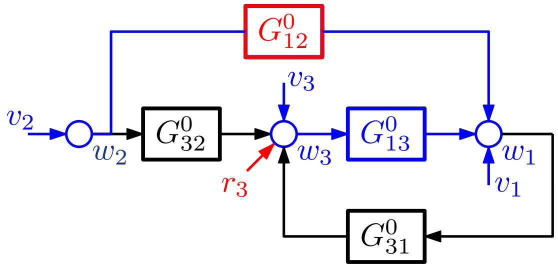

Fig. 18. The required FDD experiment setup for Case 1 and Case 2.

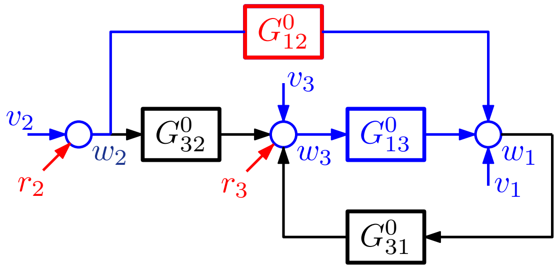

Fig. 19. The required FDD experiment setup for Case 3.

involving the target module set $\{G_{12}, G_{13}\}$ for each scenario. Conversely, Test 2 (marked in blue), the cross-correlation using $\hat{\varepsilon}_1, r_3$, solely tests G_{13}. The experimental results indicate a rapid increase of the test levels of Test 1, surpassing the confidence threshold shortly after the experiment's onset. In contrast, the test levels of Test 2 consistently stay below the threshold. This outcome suggests a failure for Test 1 and a pass for Test 2 across all cases. Hence, it is deduced that module G_{13}, tested by Test 2, is faultless. However, within the target module set $\{G_{12}, G_{13}\}$ of Test 1, a fault exists. In summary, by integrating outcomes from both tests, the fault is isolated on module G_{12} across all cases.

In the FDD procedure, both Case a and Case b require the addition of r_3 during the fault isolation experiment as depicted in Fig. 18. Conversely, for Case 2, while r_3 is essential for fault isolation, r_2 is requisite throughout the FDD process. Therefore as illustrated in Fig. 19, Case 3 necessitates the inclusion of both r_2 and r_3 in the data-generating network. Throughout the fault detection and isolation experiments, the four algorithms introduced in this study enable the automatic generation of the requisite model validation tests. Given the trade-off between noise information and test passivity, more detailed and precise noise modeling is important if one wants to utilize more passive tests in the FDD procedure.

VII. Conclusion and Future Work

This paper introduces a local model-based method for fault detection and diagnosis in large-scale interconnected network systems. Our approach relies on the models of the target MISO subnetwork with validated accuracy, and employs model invalidation tests for fault detection and diagnosis. We employ a phased approach, first adapting existing auto- and cross-correlation tests in the literature for open and closed-loop systems to suit the dynamic network framework to solve the objective of local subnetwork model validation. Simulation results demonstrate that vector-valued correlation tests outperform standard sample-wise tests within the dynamic network framework, yielding more stable and accurate validation results. Utilizing model validation tests, we convert the subsequent problem of local fault detection into a combinatorial optimization problem and devise a heuristic algorithm that automatically generates optimal tests for local fault detection given the structural properties of the network system. Furthermore, for the local fault diagnosis purpose, we demonstrate that combining results from multiple validation tests enables us to localize the detected fault more precisely, even narrowing it down to a specific module. An algorithmic approach is developed to generate the required tests for fault isolation purposes and indicate the allocation of required excitation signals. Given that large networks may be subject to correlated or reduced-rank noise, our fault detection and diagnosis method shows promise in terms of scalability and consistency.

One limitation of the present study is the assumption that in the data-generating network, each node is directly influenced by a single disturbance only. In practice, this condition may not always be feasible. In the dynamic network framework considering reduced-rank noise, a more common scenario is for each node to be influenced by several disturbance sources [48]. Consequently, a significant avenue for future research involves relaxing the constraint on the number of interference sources affecting each node.

Concurrently, as highlighted in Chapter IV, our future research direction also encompasses the development of a fully passive fault detection procedure. Our present fault detection technique, which depends on fully active tests in the absence of noise information, inherently constrains its applicable scenarios: In systems where introducing external signals at any given time is not feasible, our fault detection methods become inapplicable.

As delineated in Chapter V, our proposed fault detection and diagnosis methodology primarily uses qualitative analysis of the model validation tests, focusing solely on test outcomes — either pass or fail. In future research, we aim to leverage model validation tests for more in-depth quantitative analyses, utilizing precise test levels. Such an approach could offer a finer resolution in determining the fault's location, magnitude, or other inherent attributes.

Appendix A. Proof of Corollary 1

First, we prove that the target module set of the white noise test is $\mathcal{S}_{\hat{\varepsilon}_j} = \{G_{jk} \mid k \in \mathcal{N}_j\}$. Omitting the t, q, we can denote the residual signal $\hat{\varepsilon}_j$ as follows:

$$\begin{aligned}\hat{\varepsilon}_j &= H_j^{-1}(\sum_{k\in\mathcal{N}_j}\Delta G_{jk}w_k + H_j^0 e_j)\\ &= \underbrace{H_j^{-1}\sum_{k\in\mathcal{N}_j}\Delta G_{jk}w_k + (H_j^{-1}H_j^0-1)e_j +}_{\text{nonwhite part}} \underbrace{e_j}_{\text{white part}},\end{aligned}\tag{27}$$

where $\hat{\varepsilon}_j = e_j$ holds only when $\Delta G_{jk} = 0$ and $H_j^{-1}H_j^0 = 1$ with nonzero w_k and e_j, which is when hypothesis $\mathcal{H}_a$ holds. Therefore, it is proved that the target module set of the white noise test is $\mathcal{S}_{\hat{\varepsilon}_j} = \{G_{jk} \mid k \in \mathcal{N}_j\}$.

Then we prove that the white noise test can only be used when the noise model is obtained. If the noise model $\hat{H}_j(q)$ is not obtained, we have the estimated disturbance $\hat{v}_j$ as follows:

$$\begin{aligned}\hat{v}_j &= \sum_{k\in\mathcal{N}_j} \Delta G_{jk} w_k + H_j^0 e_j \\ &= \underbrace{\sum_{k\in\mathcal{N}_j} \Delta G_{jk} w_k + (H_j^0 - 1) e_j}_{\text{nonwhite part}} + \underbrace{e_j}_{\text{white part}} ,\end{aligned} \tag{28}$$

where the nonwhite part will always exist because of the term $(H_j^0 - 1)e_j$. Therefore, the white noise test can only be used when the noise model $\hat{H}_j(q)$ is obtained. ■

Appendix B. Proof of Corollary 2

The following lemma is essential for for the proof of **Corollary 2** and **Corollary 3**:

Lemma 2. *For the sampled quasi-stationary signals $a(t)$, $b(t)$, $c(t)$ with a data length of N and given a filter (transfer function) $G(q)$ such that $a(t) = G(q)b(t)$, the estimated cross-correlation functions $\hat{R}^N_{ac}(\tau)$ and $\hat{R}^N_{ca}(\tau)$ can be described as follows:*

1) $\hat{R}^N_{ac}(\tau) = G(q)\hat{R}^N_{bc}(\tau) \Leftrightarrow \hat{R}^N_{ac}(\tau) = g(\tau) \star \hat{R}^N_{bc}(\tau)$;

2) $\hat{R}^N_{ca}(\tau) = G^*(q)\hat{R}^N_{cb}(\tau) \Leftrightarrow \hat{R}^N_{ca}(\tau) = g(-\tau) \star \hat{R}^N_{cb}(\tau)$,

where $G^*(e^{iw}) = G(e^{-iw})$ and $\star$ is the convolution operator.

To prove **Corollary 2**, we first establish a proof under the conditions where the accurate noise model $\hat{H}_j(q)$ is available. Utilizing **Lemma 2**, we derive the subsequent equation by substituting Eq. (27) into the correlation function $\hat{R}^N_{\hat{\varepsilon}_j w_i} = \frac{1}{N}\sum_{t=1}^{N} \hat{\varepsilon}_j(t) w_i(t-\tau)$, while omitting the variables τ, q:

$$\hat{R}^N_{\hat{\varepsilon}_j w_i} = \underbrace{H_j^{-1}\Delta\mathbf{G}_{j\mathcal{N}_j}\mathbf{R}_{w_{\mathcal{N}_j} w_i}}_{target\ part} + \underbrace{H_j^{-1}H_j^0 R_{e_j w_i}}_{interference\ part} , \tag{29}$$

where

$$\begin{aligned}\Delta\mathbf{G}_{j\mathcal{N}_j} &= \begin{bmatrix}\Delta G_{j1}, & \Delta G_{j2}, & \ldots, & \Delta G_{jn}\end{bmatrix}, \\ \mathbf{R}_{w_{\mathcal{N}_j} w_i} &= \begin{bmatrix}R_{w_1 w_i}, & R_{w_2 w_i}, & \ldots, & R_{w_n w_i}\end{bmatrix}^\top ,\end{aligned} \tag{30}$$

with n the number of elements in set $\mathcal{N}_j$, i.e. $n = |\mathcal{N}_j|$[10]. When the cross-correlation test is passed, the left side of Eq. (29) is considered equal to 0 for $\tau > 0$[11], and the same should hold for the right side. The term $R_{e_j w_i}$ in the interference part in Eq. (29) is considered equal to 0 for $\tau > 0$, based on **Assumption 1**. If the noise model H_j is accurate, no additional nonzero terms will be introduced in the interference part for $\tau > 0$. In the target part, the term $H_j^{-1}\Delta G_{jk} R_{w_k w_i}$ for $k \in \mathcal{N}_j$ can be zero with either ΔG_{jk} or $R_{w_k w_i}$ being zero. Consequently, if $R_{w_k w_i} = 0$, the status of ΔG_{jk} becomes ambiguous. Hence, we restrict the target module set of the cross-correlation test with w_i to include only node signals in the selected MISO channel that have a non-zero correlation with w_i. Then it is proved that the target module set of the cross-correlation test using $\hat{\varepsilon}_j$ and w_i is $\mathcal{S}_{\hat{\varepsilon}_j w_i} = \{G_{jk} \mid k \in \mathcal{N}_j \cap \mathcal{C}_i\}$.

Next, we establish a proof under the condition that the noise model $\hat{H}_j(q)$ is not available, but the noise topology $\mathcal{T}_H$ is known. Here, the estimated disturbance $\hat{v}_j$ is used instead of the residual $\hat{\varepsilon}_j$. According to **Lemma 2**, we derive the subsequent equation by substituting Eq. (28) into the correlation function $\hat{R}^N_{\hat{v}_j w_i} = \frac{1}{N}\sum_{t=1}^{N} \hat{v}_j(t) w_i(t-\tau)$, while omitting the variables τ, q, θ:

$$\hat{R}^N_{\hat{v}_j w_i} = \underbrace{\Delta\mathbf{G}_{j\mathcal{N}_j}\mathbf{R}_{w_{\mathcal{N}_j} w_i}}_{target\ part} + \underbrace{H_j^0 R_{e_j w_i}}_{interference\ part} , \tag{31}$$

where the interference part can not consistently be 0 for $\tau > 0$ because the noise filter H_j^0 can propagate the nonzero values from the side of $\tau \leq 0$ to the side of $\tau > 0$. Hence, not all node signals w_i for $i \in \mathcal{N}_j$ are eligible for the cross-correlation test due to the presence of the nonzero interference part.

However, node signals w_i for $i \notin \mathcal{V}_j$ are independent of the innovation source e_j, rendering the interference term zero for all τ. Thus, it is proved that only the node signals w_i for $i \in \mathcal{N}_j \setminus \mathcal{V}_j$ can be used under the condition that the estimated noise model is not available. The target module set can be proved that $\mathcal{S}_{\hat{v}_j w_i} = \{G_{jk} \mid k \in \mathcal{N}_j \cap \mathcal{C}_i\}$, similarly to the previous case.

When the noise topology $\mathcal{T}_H$ is unknown, information regarding the set $\mathcal{V}_j$ is also unavailable. Under this condition, the selection of node signals for conducting this cross-correlation test becomes ambiguous. Hence, we conclude that performing the cross-correlation test using w_i is infeasible without knowledge of the noise topology $\mathcal{T}_H$. ■

Appendix C. Proof of Corollary 3

First, we establish a proof under the conditions where the estimated noise model $\hat{H}_j(q)$ is available. Utilizing **Lemma 2**, we derive the subsequent equation by substituting Eq. (27) into the correlation function $\hat{R}^N_{\hat{\varepsilon}_j r_i} = \frac{1}{N}\sum_{t=1}^{N} \hat{\varepsilon}_j(t) r_i(t-\tau)$, while omitting the variables τ, q:

$$\begin{aligned}\hat{R}^N_{\hat{\varepsilon}_j r_i} &= H_j^{-1}\Delta\mathbf{G}_{j\mathcal{N}_j}\mathbf{R}_{w_{\mathcal{N}_j} r_i} + H_j^{-1}H_j^0 R_{e_j r_i} \\ &= H_j^{-1}\Delta\mathbf{G}_{j\mathcal{N}_j}\mathbf{R}_{w_{\mathcal{N}_j} r_i},\end{aligned} \tag{32}$$

where

$$\begin{aligned}\Delta\mathbf{G}_{j\mathcal{N}_j} &= \begin{bmatrix}\Delta G_{j1}, & \Delta G_{j2}, & \ldots, & \Delta G_{jn}\end{bmatrix}, \\ \mathbf{R}_{w_{\mathcal{N}_j} r_i} &= \begin{bmatrix}R_{w_1 r_i}, & R_{w_2 r_i}, & \ldots, & R_{w_n r_i}\end{bmatrix}^\top .\end{aligned} \tag{33}$$

When the cross-correlation test is passed, the left side of Eq. (32) is considered equal to 0 for $\tau \geq 0$, and the same should

[10] Given a set $\mathcal{C}$, let $|\mathcal{C}|$ denote its cardinality in this thesis.

[11] For the cross-correlation test using w_i, the considered lags of the correlation function should start from $\tau = 1$ since the monic noise model might cause the node signals to be correlated at $\tau = 0$.

hold for the right side. The term $R_{e_j r_i}$ is considered 0 since the external excitation signal r_i is independent with the innovation source e_j. The term $H_j^{-1}\Delta G_{jk} R_{w_k r_i}$ for $k \in \mathcal{N}_j$ can be zero with ΔG_{jk} or $R_{w_k r_i}$ being zero. Consequently, if $R_{w_k r_i} = 0$, the value of ΔG_{jk} becomes indeterminate. Thus, the target module set of the cross-correlation test using r_i should consist of node signals in the selected MISO channel that have a nonzero correlation with r_i, i.e. $R_{w_k r_i} \neq 0$. Then it is proved that the target module set of the cross-correlation test using $\hat{\varepsilon}_j(t)$ and w_i is $\mathcal{S}_{\hat{\varepsilon}_j r_i} = \{G_{jk} \mid k \in \mathcal{N}_j \cap \mathcal{P}_i\}$.

Next, we give proof for the situation where we have no information about the noise model H_j or the noise topology $\mathcal{T}_H$. Here, the estimated disturbance $\hat{v}_j$ is used instead of the residual $\hat{\varepsilon}_j$. Based on **Lemma 2** we have the following equation by omitting the τ, q:

$$\begin{aligned}\hat{R}^N_{\hat{v}_j r_i} &= \Delta\mathbf{G}_{j\mathcal{N}_j}\mathbf{R}_{w_{\mathcal{N}_j} r_i} + H^0_j R_{e_j r_i}\\ &= \Delta\mathbf{G}_{j\mathcal{N}_j}\mathbf{R}_{w_{\mathcal{N}_j} r_i},\end{aligned} \tag{34}$$

where the left side can still be 0 when $\mathbf{G}_{jk}$ corresponding to the nonzero $R_{w_k r_i}$ is zero. Therefore, the cross-correlation test using r_i is available with any level of noise information. ■

Appendix D. Simulated Network

For the Monte Carlo simulations, we consider the following transfer functions for the data-generating network as in Fig. 4 and Fig. 12.

$$\begin{aligned}
G^0_{14} &= \frac{0.38q^{-1} - 0.24q^{-2}}{1 - 1.35q^{-1} + 0.54q^{-2}}, & H^0_{11} &= \frac{1 + 0.52q^{-1}}{1 + 0.41q^{-1}},\\
G^0_{21} &= \frac{-0.2q^{-1}}{1 - 1.3q^{-1} + 0.6q^{-2}}, & H^0_{41} &= \frac{1 - 0.2q^{-1}}{1 + 0.43q^{-1}},\\
G^0_{32} &= \frac{0.39q^{-1}}{1 - 0.8q^{-1} + 0.2q^{-2}}, & H^0_{22} &= \frac{1 + 0.44q^{-1}}{1 + 0.35q^{-1}},\\
G^0_{34} &= \frac{-0.6q^{-1}}{1 + 0.45q^{-1} + 0.12q^{-2}}, & H^0_{32} &= \frac{1 + 0.52q^{-1}}{1 + 0.45q^{-1}},\\
G^0_{42} &= \frac{-0.3q^{-1}}{1 - 0.6q^{-1} + 0.2q^{-2}}.
\end{aligned}$$

Appendix E. Algorithms for Generating Index Sets

Algorithm 5 Generating set $\mathcal{N}_j$

```
Input: The index $j$ of the output node $w_j$, the network topology $\mathcal{T}_G$
Output: The set $\mathcal{N}_j$
Initialize an empty set $\mathcal{N}_j$;
for $k = 1 : L$ do
    if $\mathcal{T}_G(j,k) \neq 0$ then
        Add $k$ into $\mathcal{N}_j$;
    end if
end for
```

Algorithm 6 Generating set $\mathcal{J}_k$ (Depth-first search (DFS))

```
Input: The index $k$ of a node $w_k$, the network topology $\mathcal{T}_G$
Output: The set $\mathcal{J}_k$
Initialize an empty set $\mathcal{J}_k$;
Initialize an empty stack $S$;
Initialize an array $visited$ of size $L$ with all values set to False;
Push $k$ into $S$;
Mark $k$ as visited: $visited(k) =$ True;
while $S$ is not empty do
    Pop the top element from $S$, name it $currentNode$;
    for Each node $w_j$ s.t. $\mathcal{T}_G(j, currentNode) = 1$ do
        if $visited(j) \neq$ True then
            Push $j$ into $S$;
            $visited(j) =$ True;
            Add $j$ into $\mathcal{J}_k$;
        end if
    end for
end while
```

Algorithm 7 Generating set $\mathcal{V}_j$

```
Input: The index $j$ of a node $w_j$, the network topology $\mathcal{T}_G$, the noise topology $\mathcal{T}_H$
Output: The set $\mathcal{V}_j$
Initialize an empty set $\mathcal{V}_j$;
for $l = 1$ : number of columns($\mathcal{T}_H$) do
    // Find the innovation source on node $w_j$.
    if $\mathcal{T}_H(j,l) \neq 0$ then
        for $m = 1$ : number of rows($\mathcal{T}_H$) do
            // Find the nodes that are influenced by the same innovation source of node $w_j$.
            if $\mathcal{T}_H(m,l) \neq 0$ then
                Generate $\mathcal{J}_m$ for node $w_m$;
                Add all elements of $\mathcal{J}_m$ in $\mathcal{V}_j$;
            end if
        end for
    end if
end for
```

Algorithm 8 Generating set $\mathcal{C}_k$

```
Input: The index $k$ of a node $w_k$, the network topology $\mathcal{T}_G$,
    the noise topology $\mathcal{T}_H$
Output: The set $\mathcal{C}_k$
Initialize an empty set $\mathcal{C}_k$;
Generate $\mathcal{J}_k$ for node $w_k$, and add all elements of set $\mathcal{J}_k$ into set $\mathcal{C}_k$;
Generate $\mathcal{V}_k$ for node $w_k$, and add all elements of set $\mathcal{V}_k$ into set $\mathcal{C}_k$;
// Do a reversed DFS.
Initialize an empty stack $S$;
Initialize an array $visited$ of size $L$ with all values set to False;
Push $k$ into $S$;
Mark $k$ as visited: $visited(k) = $ True;
while $S$ is not empty do
    Pop the top element from $S$, name it $currentNode$;
    for Each node $w_j$ s.t. $\mathcal{T}_G(currentNode, j) = 1$ do
        if $visited(j) \neq$ True then
            Push $j$ into $S$;
            $visited(j) =$ True;
            Add $j$ into $\mathcal{J}_k$;
        end if
    end for
end while
```

Appendix F. Simulated Network

For the Monte Carlo simulations, we consider the following transfer functions for the data-generating network as in Fig. 14.

$$G_{13}^0 = \frac{-0.2q^{-1}}{1-1.3q^{-1}+0.6q^{-2}}, \quad H_{11}^0 = \frac{1+0.52q^{-1}}{1+0.41q^{-1}},$$
$$G_{31}^0 = \frac{0.39q^{-1}}{1-0.8q^{-1}+0.2q^{-2}}, \quad H_{22}^0 = \frac{1+0.44q^{-1}}{1+0.35q^{-1}},$$
$$G_{32}^0 = \frac{-0.3q^{-1}}{1+0.6q^{-1}+0.2q^{-2}}, \quad H_{33}^0 = \frac{1+0.52q^{-1}}{1+0.45q^{-1}}.$$

References

[1] J. W. Pierre, D. Trudnowski, M. Donnelly, N. Zhou, F. K. Tuffner, and L. Dosiek, "Overview of System Identification for Power Systems from Measured Responses," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 989–1000, 2012.

[2] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 324–333, 2008.

[3] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff, J. Querol, L. Lei, T. X. Vu, and G. Goussetis, "Satellite Communications in the New Space Era: A Survey and Future Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 70–109, 2021.

[4] R. Isermann, *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer, 2006.

[5] S. Simani, C. Fantuzzi, and R. J. Patton, *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. Springer London, 2003.

[6] B. Dowdeswell, R. Sinha, and S. G. MacDonell, "Finding faults: A scoping study of fault diagnostics for Industrial Cyber–Physical Systems," *Journal of Systems and Software*, vol. 168, p. 110638, 2020.

[7] G. M. Milis, D. G. Eliades, C. G. Panayiotou, and M. M. Polycarpou, "A cognitive fault-detection design architecture," in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 2819–2826, 2016.

[8] H. Fang, H. Shi, Y. Dong, H. Fan, and S. Ren, "Spacecraft power system fault diagnosis based on DNN," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, pp. 1–5, 2017.

[9] M. H. Kim, S. Lee, and K. C. Lee, "A fuzzy predictive redundancy system for fault-tolerance of x-by-wire systems," *Microprocessors and Microsystems*, vol. 35, no. 5, pp. 453–461, 2011.

[10] C. Sankavaram, A. Kodali, and K. Pattipati, "An integrated health management process for automotive cyber-physical systems," in *2013 International Conference on Computing, Networking and Communications (ICNC)*, pp. 82–86, 2013.

[11] M. He and J. Zhang, "A Dependency Graph Approach for Fault Detection and Localization Towards Secure Smart Grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 342–351, 2011.

[12] X. Yang and L. Chen, "Design and fault diagnosis of Petri net controllers for Petri nets with uncontrollable and unobservable transitions," *Journal of Manufacturing Systems*, vol. 28, no. 1, pp. 17–22, 2009.

[13] Y. Chen, "Applications of Bayesian Network in Fault Diagnosis of Braking System," in *2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 1, pp. 234–237, 2011.

[14] J. Wang, L. Zhang, L. Duan, and R. X. Gao, "A new paradigm of cloud-based predictive maintenance for intelligent manufacturing," *Journal of Intelligent Manufacturing*, vol. 28, no. 5, pp. 1125–1137, 2017.

[15] P. Waszecki, M. Lukasiewycz, and S. Chakraborty, "Decentralized diagnosis of permanent faults in automotive E/E architectures," in *2015 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, pp. 189–196, 2015.

[16] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, vol. 3 of *The International Series on Asian Studies in Computer and Information Science*. Springer US, 1999.

[17] K. Tidriri, N. Chatti, S. Verron, and T. Tiplica, "Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges," *Annual Reviews in Control*, vol. 42, pp. 63–81, 2016.

[18] I. Yen, S. Zhang, F. Bastani, and Y. Zhang, "A Framework for IoT-Based Monitoring and Diagnosis of Manufacturing Systems," in *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pp. 1–8, 2017.

[19] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers, "Prediction error identification of linear dynamic networks with rank-reduced noise," *Automatica*, vol. 98, pp. 256–268, 2018.

[20] P. M. J. Van den Hof, A. G. Dankers, P. S. C. Heuberger, and X. Bombois, "Identification of dynamic models in complex networks with prediction error methods—Basic methods for consistent module estimates," *Automatica*, vol. 49, pp. 2994–3006, Oct. 2013.

[21] A. G. Dankers, P. M. J. Van den Hof, X. Bombois, and P. S. C. Heuberger, "Identification of Dynamic Models in Complex Networks With Prediction Error Methods: Predictor Input Selection," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 937–952, 2016.

[22] K. R. Ramaswamy and P. M. J. Van den Hof, "A local direct method for module identification in dynamic networks with correlated noise," *Automatica*, vol. 66, pp. 5237–5252, 2021.

[23] S. J. M. Fonken, K. R. Ramaswamy, and P. M. J. Van den Hof, "A scalable multi-step least squares method for network identification with unknown disturbance topology," *Automatica*, vol. 141, p. 110295, 2022.

[24] K. R. Ramaswamy, G. Bottegal, and P. M. J. Van den Hof, "Local Module Identification in Dynamic Networks Using Regularized Kernel-Based Methods," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4713–4718, 2018.

[25] J. Gertler, *Fault detection and diagnosis in engineering systems*. Marcel Dekker, 1998.

[26] L. Ljung, *System identification toolbox: User's guide*. Citeseer, 1995.

[27] S. G. Douma, X. Bombois, and P. M. J. Van den Hof, "Validity of the standard cross-correlation test for model structure validation," *Automatica*, vol. 44, pp. 1285–1294, 2008.

[28] T. Söderström and P. Stoica, "On covariance function tests used in system identification," *Automatica*, vol. 26, no. 1, pp. 125–133, 1990.

[29] U. Forssell and L. Ljung, "Closed-loop identification revisited," *Automatica*, vol. 35, no. 7, pp. 1215–1241, 1999.

[30] T. Söderström and P. Stoica, *System Identification*. Prentice-Hall International, 1989.

[31] M. Araki and M. Saeki, "A quantitative condition for the well-posedness of interconnected dynamical systems," *IEEE Transactions on Automatic Control*, vol. 28, no. 5, pp. 569–577, 1983.

[32] K. R. Ramaswamy, P. M. J. Van den Hof, and A. G. Dankers, "Generalized sensing and actuation schemes for local module identification in dynamic networks," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5519–5524, 2019.

[33] A. G. Dankers, P. M. J. Van den Hof, X. Bombois, and P. S. C. Heuberger, "Predictor input selection for two stage identification in dynamic networks," in *2013 European Control Conference (ECC)*, pp. 1422–1427, 2013.

[34] A. G. Dankers, P. M. J. Van den Hof, and P. S. C. Heuberger, "Predictor input selection for direct identification in dynamic networks," in *52nd IEEE Conference on Decision and Control*, pp. 4541–4546, 2013.

[35] L. Ljung, *System Identification: Theory for the User*. Prentice-Hall, 1999.

[36] H. H. M. Weerts, "Identification of Dynamic Networks with Rank-Reduced Process Noise," 2017.

[37] A. G. Dankers and P. M. J. Van den Hof, "Non-parametric identification in dynamic networks," in *2015 54th IEEE Conference on Decision and Control*, pp. 3487–3492, 2015.

[38] S. G. Douma, "From data to performance system; identification uncertainty and robust control design," 2006.

[39] A. I. McLeod, "On the Distribution of Residual Autocorrelations in Box-Jenkins Models," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 40, no. 3, pp. 296–302, 1978.

[40] F. Hayashi, *Econometrics*. Princeton University Press, 2000.

[41] S. Shi, G. Bottegal, and P. M. J. Van den Hof, "Bayesian topology identification of linear dynamic networks," in *2019 18th European Control Conference (ECC)*, pp. 2814–2819, 2019.

[42] A. Caprara, P. Toth, and M. Fischetti, "Algorithms for the Set Covering Problem," *Annals of Operations Research*, vol. 98, pp. 353–371, 2000.

[43] V. Chvatal, "A Greedy Heuristic for the Set-Covering Problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[44] W. Xu, Y. Wan, T. Zuo, and X. Sha, "Transfer Learning Based Data Feature Transfer for Fault Diagnosis," *IEEE Access*, vol. 8, pp. 76120–76129, 2020.

[45] H. Wang, Y. Tang, and R. Hong, "Multiple physical signals based residual life prediction model of slewing bearing," *Journal of Vibroengineering*, vol. 18, no. 7, pp. 4340–4353, 2016.

[46] C. Sankavaram, B. Pattipati, A. Kodali, K. Pattipati, M. Azam, S. Kumar, and M. Pecht, "Model-based and data-driven prognosis of automotive and electronic systems," 2009.

[47] C. Sankavaram, A. Kodali, D. F. M. Ayala, K. Pattipati, S. Singh, and P. Bandyopadhyay, "Event-driven Data Mining Techniques for Automotive Fault Diagnosis," *Annual Conference of the PHM Society*, vol. 2, 2010.

[48] H. H. M. Weerts, P. M. J. Van den Hof, and A. G. Dankers, "Identifiability of dynamic networks with part of the nodes noise-free," *IFAC-PapersOnLine*, vol. 49, no. 13, pp. 19–24, 2016.