Fault Detection in Liquid Rocket Engines via Segmented Vector Autoregressive Modeling

Tianxiang Teng[1], Zhirong Zhong[2], Jimin Liu[1], Zhongyi Zhang[2], Meng Ma[1, *]

[1]School of Mechanical Engineering, Xi'an Jiaotong University.

[2]School of Future Technology, Xi'an Jiaotong University.

*Corresponding author: Meng_Ma@mail.xjtu.edu,cn

Received Month X, XXXX | Accepted Month X, XXXX | Posted Online Month X, XXXX

Ensuring the safe operation of liquid rocket engine (LRE) systems requires reliable fault diagnosis, yet the scarcity of real fault data limits deep learning applications despite their modeling strengths. We address this by developing an offline detection method based on piecewise stationary Vector Autoregressive (VAR) modeling, employing a two-phase approach that first identifies candidate change points through block fused LASSO regularization and subsequently refines them using Smoothly Clipped Absolute Deviation (SCAD) regularization to leverage its asymptotic unbiasedness. Validated on a high-fidelity LRE simulation dataset (26 sensors, 2000 time points) with injected faults including turbopump efficiency degradation, hydrogen turbine leakage, and valve failures across 48 scenarios, our method achieves 100% precision (±50-sample tolerance) in fault timing detection without requiring training data, demonstrating superior performance to conventional ARMA models while overcoming the data dependency of neural networks.

Keywords: LRE health monitoring, VAR model, SCAD regularization

I INTRODUCTION

The LRE, often referred to as the "heart" of a spacecraft, is a critical component that ensures the flight and safe launch of the rocket. However, LREs are required to operate for extended periods under harsh conditions such as high temperatures, high pressures, strong corrosion, and high energy release, making them prone to failures within the spacecraft system. If a LRE fails during ignition or flight, it can lead to severe accidents such as explosions in a very short time frame (50ms), causing not only significant economic losses but also endangering the lives of astronauts. Therefore, research on fault diagnosis technology for LREs is particularly important. The detection of abrupt changes in multiple time series is a typical problem in the field of fault diagnosis for LREs.

Recent years, LRE fault detection approaches are divided into three types by some scholars[1]: signal processing-based approaches, model-driven approaches, and data-driven approaches. Data-driven fault detection methods involve the direct

processing of sensor-measured operational data from LRE. Alternatively, data driven techniques—such as correlation functions or autoregressive moving average (ARMA[2]) models—may extract signal features (e.g., variance, frequency) to diagnose potential engine failures. Deng et al.[3] developed an ARMA model-based fault detection method for the main-stage fault diagnosis of high-thrust hydrogen-oxygen staged-combustion cycle engines, and successfully validated the reliability of the approach through hardware-in-the-loop simulation. Xue et al.[4] developed a real-time fault simulation system based on ARMA models for reusable liquid rocket engines, conducting simulation tests on typical fault scenarios and validating the algorithm's applicability through hardware-in-the-loop testing. However, since ARMA processes each signal independently, it is prone to overlooking system-level fault characteristics.

The VAR model was proposed by Christopher Sims in 1980 as an innovation to traditional simultaneous equation models[5]. VAR model is widely applied in fields such as economics[6][7][8]and engineering[9]. The VAR model can be seen as a multivariate extension of the univariate AR model. By capturing the dynamic relationships between multiple time series variables, the VAR model can jointly predict the future values of these variables.

However, high-dimensional VAR models suffer from the issues of having a large number of parameters and being computationally challenging to solve. Richard et al.[10] proposed a two-stage method to fit sparse VAR (sVAR) models where many AR coefficients are zero. In the first stage, non-zero AR coefficients are selected based on estimates of partial spectral coherence (PSC) and the use of BIC. PSC can be used to quantify the conditional relationships between marginal series in multivariate processes. A second refinement stage is then applied to further reduce the number of parameters. To address the issue of high-dimensional VAR models requiring the estimation of a large number of parameters and potential inference problems, Monica Billio et al.[11] proposed a new Bayesian nonparametric (BNP) Lasso prior (BNP-Lasso) for high-dimensional VAR models, which can improve estimation efficiency and prediction accuracy. In 2022, Aramayis et al.[12] improved the sparse VAR model (msVAR) by using time series graphical lasso (TSGlasso) for sparse inverse spectral density matrix estimation, avoiding direct high-dimensional matrix inversion, and introducing false discovery rate (FDR) controlled multiple hypothesis testing methods in the model refinement stage, thereby improving the two-stage sparse VAR framework proposed by Davis et al.

Despite these advancements, existing methods still face several limitations in the context of LRE fault diagnosis. Signal processing and ARMA-based approaches often overlook system-level dependencies. Traditional VAR methods, while able to capture multivariate dynamics, encounter computational difficulties in high dimensions and strictly require global stationarity—a condition rarely satisfied in real LRE operations, where signals often exhibit piecewise stationarity across different stages (e.g., start-up ignition, steady combustion, shutdown depressurization). Moreover, most existing VAR-based fault detection methods rely on LASSO regularization, which enforces sparsity but introduces bias by excessively shrinking large coefficients, thereby limiting detection accuracy.

To address these issues, this study introduces a segmented VAR framework that incorporates the piecewise stationarity

assumption and improves detection performance by replacing LASSO with SCAD regularization. The SCAD penalty alleviates the bias problem of LASSO, ensuring asymptotic unbiasedness and more accurate identification of significant variables while maintaining sparsity. In this way, our method specifically addresses two major shortcomings of existing approaches: (1) the inability of ARMA or traditional VAR methods to handle non-stationary LRE fault data, and (2) the limitations of LASSO-regularized VAR in precisely identifying fault points under sparse conditions.

This paper focuses on an offline fault diagnosis method for LRE operational data based on the data generation mechanism of the VAR model. First, we construct a numerical simulation dataset containing various potential fault types during the LRE startup process. Then, we improve the VAR-based change point detection method proposed in [1] by replacing the LASSO regularization in the second step with the SCAD regularization method. The improved detection method, employing SCAD regularization, exhibits asymptotic unbiasedness and the Oracle property, enabling more accurate identification of significant variables while maintaining sparsity and avoiding excessive shrinkage of large coefficients. Finally, we apply this method to the fault diagnosis dataset, achieving favorable detection performance. The contributions of this study are summarized as follows:
(1) We propose a novel VAR-based fault point detection algorithm incorporating SCAD regularization.
(2) Through simulations of the Liquid Rocket Engine (LRE) system, we generated a synthetic dataset covering both normal and faulty states.
(3) We conducted fault point detection experiments on a simulated liquid rocket engine (LRE) dataset using the proposed fault detection method, achieving 100% precision. In addition, the algorithm was validated on real-world LRE fault data, further demonstrating its effectiveness.

The remainder of this paper is organized as follows: Part II introduces the theoretical background of the research and elaborates on the proposed method in detail. Part III and IV verifies the effectiveness and superiority of the proposed model based on simulation experiments and hot-fire testing experiments. Part V summarizes the main content of this paper.

II METHODOLOGY

VAR Model

Suppose we have a piecewise stationary time series dataset. This data contains $n+1$ time points, with m_0 change points denoted as $0 = t_0 < t_1 < \cdots < t_{m_0} < t_{m_0+1} = n$. Then, for any stationary segment where $t_{j-1} < t < t_j$. the VAR model can be expressed as:

$$X_t = B_j' X_{t-1} + \dot{o}_t \quad and \quad B_j = L^* + S_j^* \quad (1)$$

Where X_t is the p dimensional vector of sensor observed time series at t, B_j is the $p \times p$ transition matrix for the $j-th$ segment that reflects time-varying effects of historical values. Further, each transition matrix is assumed to be a superposition of a stable L^* low rank component and a time varying S_j^* sparse component. Finally, we assume that the p −dimensional noise process is normally distributed; i.e. $\dot{o} \sim N(0, \sum_{\dot{o}})$. Building upon this foundation, we may postulate that the number of non-zero elements in $j-th$ sparse component S_j^* is $\|S_j^*\|_0 = s$ with $s << p^2$ and that low rank component L^*

has rank l with $l << p$ .The low-rank component L^* encodes the static cross-autocorrelation structure across all p time series, whereas S_j^* captures dynamic cross-sectional dependencies.

FISTA(Fast Iterative Shrinkage-Thresholding Algorithm)

FISTA is an accelerated optimization algorithm designed to solve large-scale linear inverse problems, particularly those involving sparse signal recovery and regularized regression. It is an enhanced version of the ISTA (Iterative Shrinkage-Thresholding Algorithm), achieving significantly faster convergence through Nesterov's momentum techniques.

FISTA solves convex optimization problems of the form:

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + g(x) \qquad (2)$$

Where $f(x)$ is a smooth convex function (e.g., least-squares loss $\frac{1}{2}\|Ax-b\|_2^2$), $g(x)$ is a non-smooth convex regularizer (e.g., ℓ_1-norm $\lambda\|x\|_1$ for sparsity).

Algorithm 1: FISTA.

Input: Objective function components (smooth part f , non-smooth part g), Lipschitz constant L for $f(x)$, initial solution x_0 ,maximum iterations K . Function to compute gradient of smooth part ∇f ,The tolerance value. ε .

Output: Optimized solution x^* .

1. **Initialize Parameters:** Set $y_1 = x_0, t_1 = 1$ .Choose step size $t_k = \frac{1}{L}$
2. **For each iteration** k **from** 1 **to** K **:**
   a. Gradient step:
      I. Compute gradient at y_k :
      $$z_k = y_k - \nabla f(y_k)$$
   b. Proximal Operator (Shrinkage):
      I. Apply proximal operator for g :
      $$x_k = prox_{t_k g}(z_k)$$
   c. Momentum Update:
      I. Update step size:
      $$t_{k+1} = (1 + sqrt(1 + 4t^2))/2$$
      II . Update extrapolation point:
      $$y_{k+1} = x_k + (t_k - 1)(x_k - x_{k-1})/t_{k+1}$$
3. **Termination:**
   If $\|x_k - x_{k-1}\| < \varepsilon$
   Break
4. **End For**
5. **Return** x^*

The Changing points Detection Procedure

The changing point detection procedure we propose mainly consisting of two steps: (A) solving a regularized regression problem

using the block fused lasso (BFL) penalty to identify candidate change points, and (B) filtering the obtained candidate points through SCAD regularization to compute a new information criterion.

Step 1: Identify Candidate Points

For a piecewise stationary time series dataset containing $n+1$ time points, we first partition it into blocks of size b_n, keeping all model parameters fixed within each block - meaning each block's endpoints serve as candidate change points. It is important to note that to accurately identify true change points, b_n cannot be set too large. Therefore, the selection of b_n follows these criteria:

$$k_n = \left\lfloor 2\sqrt{n} \right\rfloor \tag{3}$$

Thus, the time series length for each block is:

$$b_n = \left\lfloor \frac{n}{2\sqrt{n}} \right\rfloor \tag{4}$$

For $j-th$ block, We define the following variables:

$$\begin{cases} \mathbf{X}_{r_j} = [X_{r_{j-1}}, \ldots, X_{r_j - 1}] \\ \mathbf{Y}_{r_j} = [X_{r_{j-1}+1}, \ldots, X_{r_j}] \\ \delta_{r_j} = [\text{蝌}_{r_{j-1}+1}, \ldots, \ _{r_j}] \end{cases} \tag{5}$$

For global variables, we define:

$$\begin{cases} \mathrm{X} = [\mathbf{X}_{r_1}, \ldots, \mathbf{X}_{r_{k_n+1}}]' \in \mathbb{R}^{n \times p} \\ \mathrm{Y} = [\mathbf{Y}_{r_1}, \ldots, \mathbf{Y}_{r_{k_n+1}}]' \in \mathbb{R}^{n \times p} \\ \mathrm{E} = [\text{蝌}_1, \ldots, \ _{r_{k_n}}]' \in \mathbb{R}^{n \times p} \end{cases} \tag{6}$$

Furthermore, we define the variable Z as:

$$\mathrm{Z} = \begin{bmatrix} \mathbf{X}'_{r_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{X}'_{r_2} & \mathbf{X}'_{r_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}'_{r_{k_n+1}} & \mathbf{X}'_{r_{k_n+1}} & \cdots & \mathbf{X}'_{r_{k_n+1}} \end{bmatrix} \in \mathbb{R}^{n \times p k_n} \tag{7}$$

Based on (1), we can formulate the following linear regression problem

$$\mathrm{Y} = \mathrm{X} L^* + \mathrm{Z}\,\Theta + \mathrm{E} \tag{8}$$

Wherein $\Theta = [\theta'_1, \ldots, \theta'_{k_n}]' \in \mathbb{R}^{p k_n \times p}$, where $\theta'_1 = S^*_1$, then for the subsequent ones

$$\theta_i = \begin{cases} S^*_{j+1} - S^*_j & i = t_j \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Based on Equation(8), the model can be expressed as:

$$\begin{bmatrix} \mathbf{Y}_{r_1} \\ \ldots \\ \mathbf{Y}_{r_{k_n+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{r_1} \\ \ldots \\ \mathbf{X}_{r_{k_n+1}} \end{bmatrix} \bullet L^* + \begin{bmatrix} \mathbf{X}_{r_1} \bullet S^*_1 \\ \ldots \\ \mathbf{X}_{r_{k_n+1}} \bullet S^*_{r_{k_n+1}} \end{bmatrix} + \mathrm{E} \tag{10}$$

The model coefficients Θ and L can be estimated via a composite LASSO-regularized approach, as expressed in the linear regression formulation (8)

$$\begin{aligned} \left\langle \hat{\Theta}, \hat{L} \right\rangle = & \, arg\,min_{\Theta, L \in \Omega} \frac{1}{n} \| \mathrm{Y} - \mathrm{X}L - \mathrm{Z}\Theta \|_2^2 \\ & + \lambda_{1,n} \| L \|_* + \lambda_{2,n} \| \Theta \|_1 \\ & + \lambda_{3,n} \sum_{l=1}^{k_n} \left\| \sum_{j=1}^{l} \theta_j \right\|_1 \end{aligned} \tag{11}$$

We employ the FISTA (**Algorithm 1**)for efficient solution.

Step 2: Screen Candidate Points

In the previous step, we have obtained the candidate change point set A_n, and now we need a new step to filter the existing change

point set. Specifically, we already have m change points, namely:

$$1 = s_0 < s_1 < \cdots < s_m < s_{m_0+1} = n$$

, Based on this, similar definitions as in Step 1 can be made:

$$\begin{cases} \mathbf{X}_{s_j} = [X_{s_{j-1}}, \ldots, X_{s_j - 1}] \\ \mathbf{Y}_{s_j} = [X_{s_{j-1}+1}, \ldots, X_{s_j}] \\ \delta_{s_j} = [\varepsilon_{s_{j-1}+1}, \ldots, \varepsilon_{s_j}] \end{cases} \quad (12)$$

$$\begin{cases} \mathbb{X} = [\mathbf{X}_{s_1}, \ldots, \mathbf{X}_{s_m}]' \in \mathbb{R}^{n \times p} \\ \mathbb{Y} = [\mathbf{Y}_{s_1}, \ldots, \mathbf{Y}_{s_m}]' \in \mathbb{R}^{n \times p} \\ \mathbb{E} = [\varepsilon_{s_1}, \ldots, \varepsilon_{s_m}]' \in \mathbb{R}^{n \times p} \end{cases}$$

Furthermore, we define the variable $\mathbb{Z}_{s_1,\ldots,s_m}$ as:

$$\mathbb{Z}_{s_1,\ldots,s_m} = \begin{bmatrix} \mathbf{X}'_{s_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{X}'_{s_2} & \mathbf{X}'_{s_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}'_{s_{m+1}} & \mathbf{X}'_{s_{m+1}} & \cdots & \mathbf{X}'_{s_{m+1}} \end{bmatrix} \quad (13)$$

And the corresponding coefficient matrix is given by $\Theta_{s_1,\ldots,s_m} = [\theta'_1, \ldots, \theta'_m]'$. Then, based on the variables we have defined, the following linear regression expression can be written:

$$\mathbb{Y} = \mathbb{X} L^* + \mathbb{Z}_{s_1,\ldots,s_m} \Theta_{s_1,\ldots,s_m} + \mathbb{E} \quad (14)$$

We then obtain $\Theta_{s_1,\ldots,s_m}$ by solving the following optimization problem incorporating SCAD regularization:

$$(\hat{L}, \hat{\Theta}_{s_1,\ldots,s_m}) = \arg\min_{L,\Theta_{s_1,\ldots,s_m}} \left\{ \underbrace{\sum_{i=1}^{m+1} \frac{1}{s_i - s_{i-1}} \left\| \mathbf{Y}_{s_i} - \mathbf{X}_{s_i}(\theta_{(s_{i-1},s_i)} + L) \right\|_2^2}_{\text{Prediction Error (Weighted Least Squares)}} + \underbrace{\sum_{i=1}^{m+1} \sum_j P_{\lambda,a}(\theta_{(s_{i-1},s_i),j})}_{\text{Segmented Sparsity Penalty (SCAD)}} \right. \quad (15)$$

$$\left. + \underbrace{\sum_k P_{\lambda_L,a_L}(L_k)}_{\text{Low-Rank Matrix Penalty (SCAD)}} + \underbrace{\sum_k P_{\lambda_L,a_L}(L_k)}_{\text{Low-Rank Matrix Penalty (SCAD)}} \right\}$$

In the equation, the term $P_{\lambda,a}(\bullet)$ represents the SCAD (Smoothly Clipped Absolute Deviation) regularization component. The SCAD penalty function is defined as:

$$P_{\lambda,a}(x) = \begin{cases} \lambda |x| & \text{if } |x| \le \lambda, \\ \dfrac{(a\lambda |x| - \frac{x^2 + \lambda^2}{2})}{a-1} & \text{if } \lambda < |x| \le a\lambda, \\ \dfrac{(a+1)\lambda^2}{2} & \text{if } |x| > a\lambda. \end{cases} \quad (16)$$

The total error can be expressed as:

$$\begin{aligned} \mathbb{E}_{\text{total}}(L, \Theta_{s_1,\ldots,s_m}) = & \sum_{i=1}^{m+1} \frac{1}{s_i - s_{i-1}} \left\| \mathbf{Y}_{s_i} - \mathbf{X}_{s_i}(\theta_{(s_{i-1},s_i)} + L) \right\|_2^2 \\ & + \sum_{i=1}^{m+1} \sum_j P_{\lambda,a}(\theta_{(s_{i-1},s_i),j}) \\ & + \sum_k P_{\lambda_L,a_L}(L_k) \end{aligned} \quad (17)$$

Next, we perform iterative screening of the number of change points. Let the set of change points after the k-th iteration be denoted as $s^{(k)} = (s_1^{(k)}, \ldots, s_{m_k}^{(k)})$. Therefore, the error in the current iteration can be expressed as:

$$\mathrm{E}(s^{(k)}) = \sum_{i=1}^{m_k+1} \frac{1}{s_i^{(k)} - s_{i-1}^{(k)}} \cdot \left\| \mathbf{Y}_{[s_{i-1}^{(k)}, s_i^{(k)})} - \mathbf{X}_{[s_{i-1}^{(k)}, s_i^{(k)})} (\theta_{(s_{i-1}^{(k)}, s_i^{(k)})} + L) \right\|_2^2 + \sum_{i=1}^{m_k+1} \sum_j P_{\lambda,a}(\theta_{(s_{i-1}^{(k)}, s_i^{(k)}),j}) + \sum_\ell P_{\lambda_L, a_L}(L_\ell) \quad (18)$$

We evaluates change points by sequentially testing the impact of removing each candidate point through an information criterion (IC) framework. First, the IC is defined as:

$$IC(s^{(k)}) = \mathrm{E}(s^{(k)}) + m_k \omega_n \quad (19)$$

In the equation, m_k represents the current number of change points (which dynamically changes during iterations), and ω_n denotes the sample-size-dependent penalty weight. Based on the above equation, the change in information criterion (ΔIC_r) resulting from removing the r-th change point can be expressed as:

$$\Delta IC_r = \mathrm{E}(s_{-r}^{(k)}) - \mathrm{E}(s^{(k)}) - \omega_n \quad (20)$$

A change point can be removed when ΔIC_r. By sequentially evaluating and removing each candidate point in the current change point set while computing ΔIC, we achieve effective change point screening.

III SIMULATION EXPERIMENTS

The experimental validation utilizes three types of fault simulation data from a liquid rocket engine. The model's effectiveness is verified by evaluating the accuracy in detecting the timing of simulated fault occurrences.

This experiment utilizes a liquid rocket engine simulation dataset, where numerical simulation was achieved by establishing component-level simulation modules using Amesim software.

The simulation model outputs synthetic readings from 26 sensors, with each sample consists of 2000 data points collected at a 1 kHz sampling rate during steady-state operation, representing a 2-second duration. Faults were injected at three specific time points: the 500th time point (0.5 seconds after simulation start), the 1000th time point (1.0 seconds after simulation start), and the 1500th time point (1.5 seconds after simulation start). The fault magnitudes were set to 0.8, 0.85, 0.9, and 0.95 of nominal values respectively, generating a total of 48 distinct fault scenarios.

Simulation System Construction

This study simulates a liquid rocket engine system to generate operational data under both normal and fault conditions.

The liquid rocket engine (LRE) system, comprising turbopump assemblies, thrust chambers, and auxiliary subsystems (Fig. 1), was simulated under fault conditions by perturbing parameters in the nominal model. Four failure modes were implemented: turbopump efficiency degradation, hydrogen turbine leakage, valve actuation faults, and cooling jacket breaches.

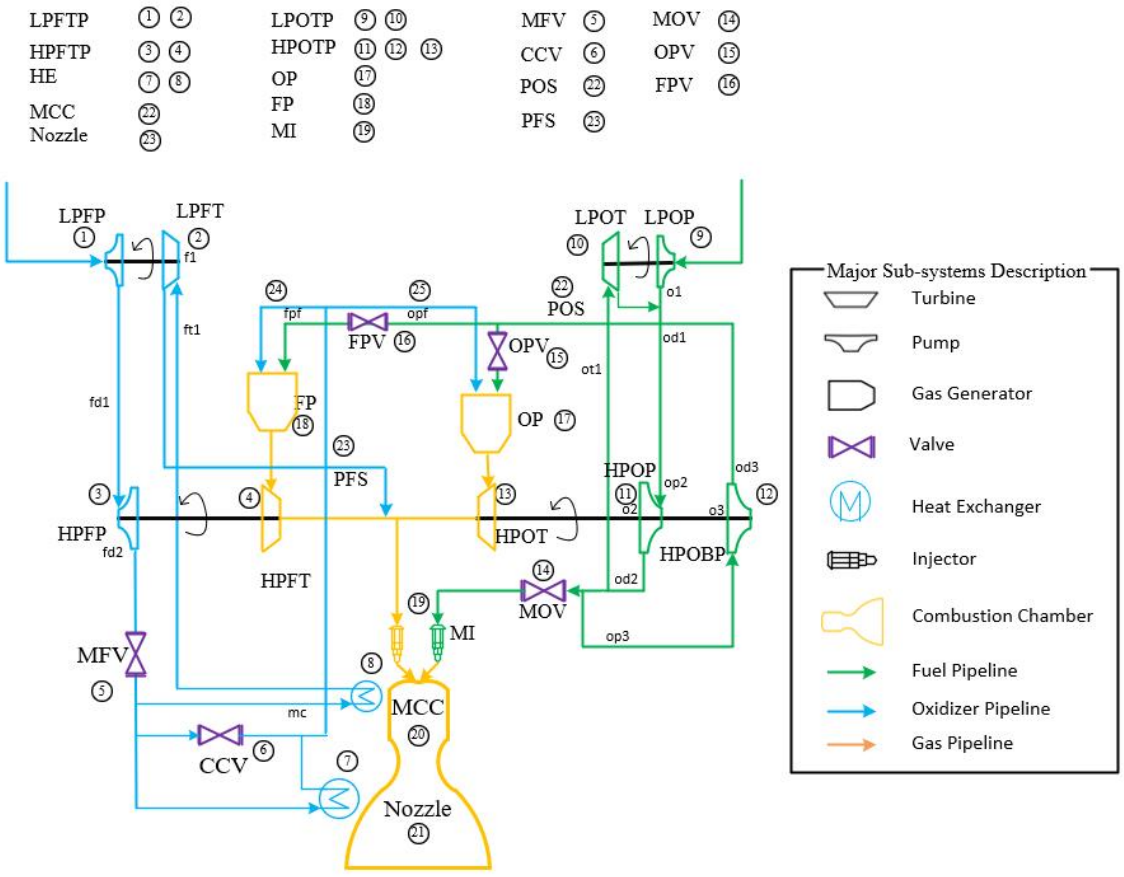


Fig1. Schematic of LRE and selected failure modes.

The fault simulation of the liquid rocket engine is based on a simulation model, employing a fault-injection approach to conduct a systematic and dynamic comprehensive analysis of various potential failure modes in a high-thrust hydrogen-oxygen engine. Table 1 summarizes common fault modes in liquid rocket engines. The investigation implements fault injection on three critical failure modes: turbopump efficiency degradation, hydrogen turbine module leakage, and valve actuation failure, representing common yet high-impact scenarios in liquid rocket engine operation.

Table 1. LRE failure modes.

Components	**Classification**	**Fault Mode**	**Fault Performance**
Turbopump	Centrifugal Pump	(1) Impeller damage (2) Bearing wear or damage (3) Pump cavitation	Pump efficiency decrease
	Pump efficiency decrease	(1) Blade detachment (2) Bearing wear or damage	Turbine efficiency decrease

		(3) Turbine blade erosion (4) Gas flow obstruction (5) Turbine inlet flow leakage	Downstream stream flow rate decrease
Pipeline	Gas pipeline	(1) Pipeline blockage (2) Pipeline leakage	Increased flow resistance
	Liquid pipeline		Downstream flow rate decrease
Thrust Chamber	Combustion chamber	Combustion deterioration	Combustion efficiency decrease
	Gas generator	Combustion deterioration	
	Cooling jacket	Cooling jacket blockage	Increased flow resistance
		Cooling jacket leakage	Downstream flow rate decrease
	Nozzle	(1) Nozzle deformation (2) Large nozzle detachment	Nozzle efficiency decrease
Others	Regulating valve	Stuck during switching	Reduced flow area
	Cavitation tube	Cavitation tube blockage	Increased flow resistance
	Sonic nozzle	Sonic nozzle blockage	

Turbine Efficiency Decrease

During operation, turbomachinery components may experience various failure modes including rotor rubbing/jamming, shaft fracture, turbine blade detachment, pump blade fracture, and turbopump cavitation, all of which can lead to varying degrees of efficiency degradation in the turbine assembly. A detailed analysis of the failure mechanisms reveals that when rotor rubbing and partial turbine blade detachment occur during operation, the former increases the torque demand on the turbopump assembly while the latter enhances the flow resistance between propellant and blades. Both effects contribute to reduced turbine actuation capability, ultimately manifesting as decreased turbine assembly efficiency[15]. To simulate these fault conditions, an efficiency correction factor f is introduced to modify the operational efficiency of both the turbine and centrifugal pump. This approach models the rotational speed reduction caused by power loss and the consequent decrease in pump work output, achieving accurate fault simulation as demonstrated in Fault Mode 1 of Figure 1. The mathematical representation of this relationship is as follow:

$$P_{turbine} = dpQ\eta_{turbine} f = n_{turbine} T \tag{21}$$

Where $P_{turbine}$ is the power, dp is the pressure difference across the turbine, Q is the volumetric flow rate, $\eta_{turbine}$ is the turbopump efficiency, f is the correction factor, $n_{turbine}$ is the common rotational speed of both the turbine and centrifugal pump, and T is the torque.

Hydrogen Turbine Module Leakage

Hydrogen, as a fuel, is relatively prone to leakage due to its small molecular weight. Furthermore, hydrogen turbopumps operate at extremely high rotational speeds reaching tens of thousands of revolutions per minute. The coaxial design of these turbopumps, combined with higher pressure at the turbine end compared to the pump end, creates conditions conducive to hydrogen leakage into the pump and surrounding environment[16].

In this failure scenario, liquid hydrogen leaking directly into the pump and environment essentially introduces two additional flow paths to the engine system. To model this condition, a valve assembly with a maximum flow area A is added to each flow path. The opening size of these valves is controlled by external signals to simulate varying degrees of leakage severity. The mathematical representation of this leakage model is as follows:

$$\dot{m}_3 = \dot{m}_1 + \dot{m}_2 \tag{22}$$

$$\dot{m}_2 = c_q A \tau \sqrt{2\rho\Delta p} \tag{23}$$

Where $\dot{m}_3$ and $\dot{m}_1$ are the flow rate through the primary valve, $\dot{m}_2$ is the flow rate through the leakage valve, and A is the maximum flow area of the leakage passage.

Valve Opening failure

Valve control constitutes a critical factor for normal engine startup. The simulation of valve failures - including failure to open, slow opening, and blockage - can be achieved by adjusting the timing and response speed of five main valves: the Main Oxidizer Valve (MOV), Main Fuel Valve (MFV), Fuel Preburner Oxidizer Valve (FPOV), Combustion Chamber Coolant Valve (CCV), and Oxidizer Preburner Oxidizer Valve (OPOV). The following equations primarily regulate flow through control functions to simulate these valve failure modes:

$$\dot{m} = c_q A \tau \sqrt{2\rho\Delta p} \tag{24}$$

Where $\dot{m}$ is the flow rate through the valve, c_q is the flow coefficient, A is the maximum flow area, τ is the control function, ρ is the average density of the fluid flowing through the valve, and Δp is the pressure difference between the two ports of the valve.

Data Preprocessing

The data preprocessing module consists of two main steps: data normalization and noise generation.

Data Normalization

Since the variables calculated by the liquid rocket engine's mathematical model exhibit significant differences in magnitude, normalization is essential to ensure consistency across different parameters. To address this, all simulation-derived data are uniformly scaled using the Min-Max normalization method, which transforms the values into a common range. This technique involves identifying the global minimum and maximum values of the dataset before applying the normalization formula:

$$X_{nomalized} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{25}$$

where X is the original data value, $X_{\min}$ is the minimum value in the dataset, and $X_{\max}$ is the maximum value in the dataset. This transformation scales all features to a fixed range, ensuring consistent treatment of variables regardless of their original measurement scales.

The normalized simulation data (without noise injection) for all three types of faults are illustrated in the figure.

Noise Generation

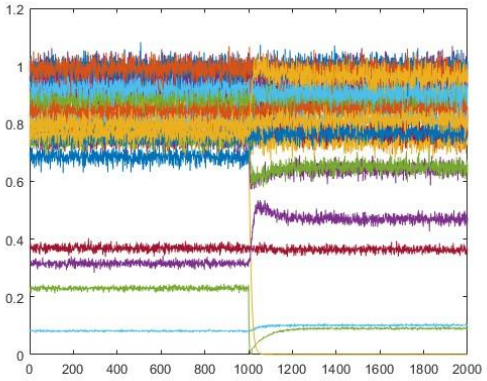
In the process of sensor measurements for liquid rocket engines, environmental interference and measurement errors inevitably introduce random noise into the actual data. To accurately simulate this phenomenon, this study employs a Gaussian noise model to enhance the normalized data.

In this study, Gaussian noise was injected into the normalized signals to realistically simulate sensor uncertainty in liquid rocket engine measurements. Specifically, three noise intensity levels were considered: 2.5%, 5%, and 7.5% of the signal amplitude. Among these, the 5% noise level closely reflects the signal-to-noise ratio (SNR) observed in actual liquid rocket engine sensor data, while the 2.5% and 7.5% levels were introduced as additional robustness tests to evaluate the stability of the proposed method under lower and higher noise conditions.

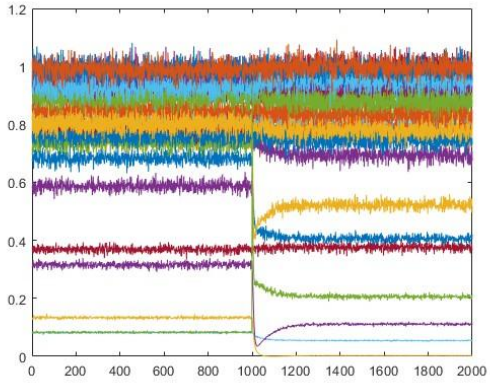
The mathematical expression of the Gaussian noise model is as follows:

$$X_{noisy} = X_{normalized} + \text{蝌} \quad \sim \mathbb{N}\ (0,\sigma^2)\ (26)$$

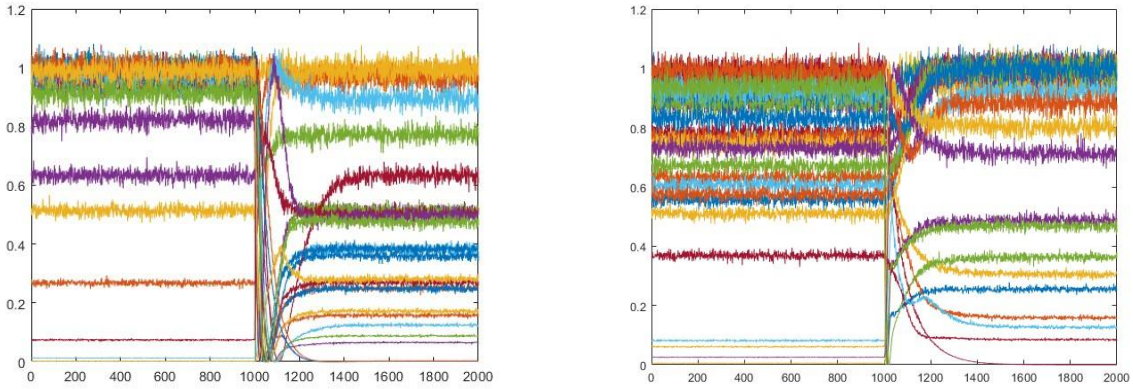
Where $X_{normalized}$ is the normalized value within $[0,1]$, ò is the Gaussian-distributed noise term, and σ determines the noise magnitude as a given percentage (2.5%, 5%, or 7.5%) of signal amplitude. This formulation allows us to evaluate robustness under different noise intensities that approximate varying levels of sensor uncertainty. By doing so, the additive noise model retains the primary signal trends while superimposing controlled random fluctuations. This design allows us to approximate sensor measurement disturbances under different noise levels without completely distorting the essential dynamics of the simulated engine data.



(a)



(b)



(c) (d)

Fig 2. The preprocessed simulation data for four types of faults. (a), low-pressure fuel turbine efficiency degradation fault. (b), abnormal oxygen main valve opening. (c), high-pressure oxygen turbine efficiency degradation Fault. (d), high-pressure fuel turbine efficiency degradation fault.

Implementation Details

Experiment Design

The simulation model outputs synthetic readings from 26 sensors, with each sample consists of 2000 data points collected at a 1kHz sampling rate during steady-state operation, representing a 2-second duration. Faults were injected at three specific time points: the 500th time point (0.5 seconds after simulation start), 1000th time point (1.0 seconds after simulation start), and 1500th time point (1.5 seconds after simulation start). The fault magnitudes were set to 0.8, 0.85, 0.9, and 0.95 of nominal values respectively, generating a total of 48 distinct fault scenarios. The running environment is described as follows: the CPU is a Core i5-13600KF @ 3.50GHz, the memory is 32GB, the GPU is a GTX 4070, and the programming language is R 4.4.0.

Parameter Selection

In our change-point detection method, several important tuning parameters are involved, including the regularization parameters $\lambda_{1,n}, \lambda_{2,n}, \lambda_{3,n}$ in Step 1, the penalty term ω_n in the information criterion, and the tolerance level tol used in the FISTA optimization. The following provides the approach for selecting them:

$\lambda_{1,n}$:In this study, we manually select $\lambda_{1,n}$ from the range $[\sqrt{\frac{\log p}{n}}, 10\sqrt{\frac{\log p}{n}}]$. This choice is motivated by the non-asymptotic analysis of [17]. Within this interval, we perform a grid search over a set of evenly spaced candidate values. The optimal $\lambda_{1,n}$ is then selected as the one that minimizes the validation error across the candidate grid.

$\lambda_{2,n}$:To select the regularization parameter $\lambda_{2,n}$, we adopt a block-based cross-validation approach. In the simulation study, 20% of the time series blocks are randomly selected as the validation set, with the blocks spaced evenly from a random initial point. The last time points of these selected blocks form the set T. We then

estimate Θ for a range of candidate $\lambda_{2,n}$ values using only the data excluding T. These estimates are subsequently used to predict the time series at the time points in T. The value of $\lambda_{2,n}$ that minimizes the mean squared prediction error over T is selected as the optimal parameter via cross-validation.

$\lambda_{3,n}$:The regularization parameter $\lambda_{3,n}$,which controls the inter-block sparsity structure, is theoretically required to satisfy $\lambda_{3,n} = o\left((nd_n^*)^{-1}\right)$ as stated in Assumption H3 of[14], implying that its value should vanish as the sample size n increases. For simplicity, we set $\lambda_{3,n} = 0$ throughout this study.

ω_n:The penalty weight controlling the number of change points (the larger the value, the fewer the change points). In this study, it is manually selected from the range $[\frac{1}{30}\ln(T-1)\ln(p), \frac{1}{2}\ln(T-1)\ln(p)]$,which is selected based on the theoretical range suggested by Assumption H4 in[14]. Guided by this theoretical interval, we further perform a grid search within the range to determine the optimal ω_n.

tol:The convergence tolerance ε in Algorithm 1 is manually set to 10^{-3} here.

The regularization parameter in Step 2 is selected with reference to that in Step 1.

Evaluation Index

The classification performance is evaluated based on the proportion of true positives, which focuses solely on the ratio between true positives (TP) and false positives (FP):

$$P = \frac{TP}{TP + FP} \tag{27}$$

True positive (TP) is counted when any detected break point $\hat{s}_j$ falls within $\pm p$ lags of a true change point s_j (accounting for VAR lag effects). Precision P measures the proportion of correctly identified breaks among all detections. The tolerance window is set to $p = 50$ timepoints, accommodating both the model's autoregressive structure.

Result and Discussion

For the 48 sets of fault scenario data generated by the LRE simulation model, we conducted fault point detection using three approaches: the baseline method from [14], our SCAD-regularized variant, and the LASSO-regularized variant as an additional reference. Experiments were performed under three Gaussian noise levels: 2.5%, 5%, and 7.5% of the signal amplitude. While all methods achieved 100% precision at the 2.5% noise level due to the clarity of fault signatures, under higher noise conditions (5% and 7.5%), the SCAD-based method demonstrated noticeably more robust detection performance compared to the LASSO-based approach. This agrees with the theoretical findings of [18], who showed that SCAD's non-convex penalty avoids the over-shrinkage of large coefficients that is common with LASSO, thereby preserving essential structural signals and enhancing robustness. In addition, we also conducted comparative experiments with the model in [19] to further validate the effectiveness of our method.

Figure 3 illustrates the detection results of one of the fault data sets using the improved method proposed in this paper.

Figure 3 shows the change point detection results for three different fault scenarios:

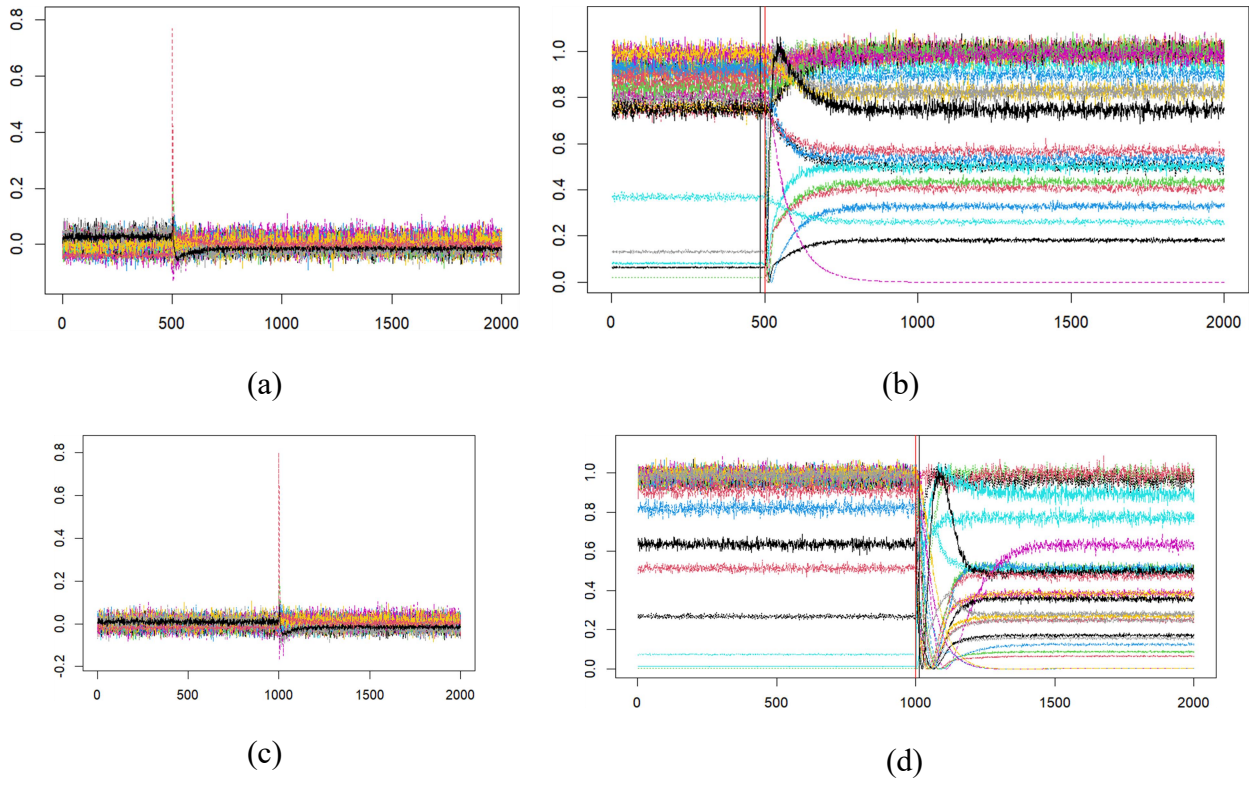
1. At the 1000th data point, an abnormal fault in the oxygen main valve opening is injected with a fault factor of 0.9.

2. At the 500th data point, a fault in the high-pressure oxygen turbine efficiency is injected with a fault factor of 0.85.
3. At the 1500th data point, a fault in the low-pressure fuel turbine efficiency is injected with a fault factor of 0.8.

In (a)(c)(e), the calculation results of $(X_{t-1} - L^* X_t)$ are presented, reflecting the extraction of the low-rank component L^* of transition matrix in step 1. It is evident that the parameter estimation of L^* in step 1 is highly accurate, with the residuals containing only Gaussian noise and a distinct pulse signal at the fault injection time point

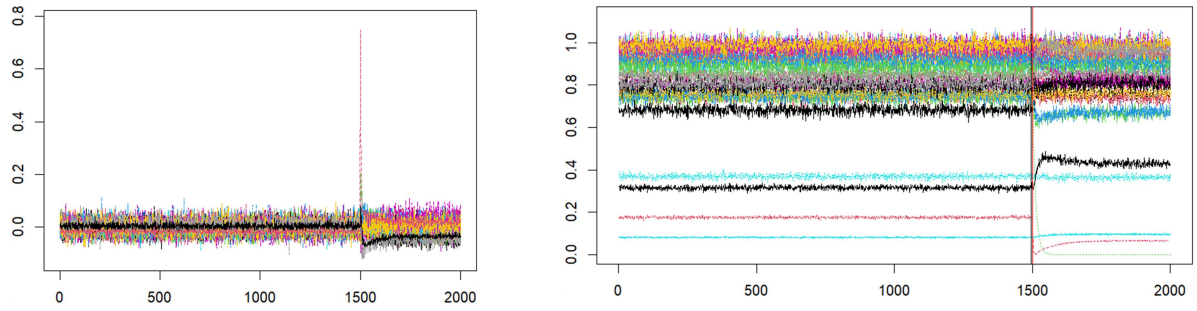
Table 2. Detection results of liquid rocket engine fault simulation data using three models under different noise levels

Noise Level	LASSO-VAR	SCAD-VAR (Proposed)	BSS -VAR
2.5%	Accuracy = 100%	Accuracy = 100%	Accuracy = 95.83%
5.0%	Accuracy = 100%	Accuracy = 100%	Accuracy = 87.5%
7.5%	Accuracy = 91.66%	Accuracy = 95.83%	Accuracy = 83.33%



(a) (b)

(c) (d)



(e) (f)

Fig 3. Fault point detection results for 3 scenarios. (a)(c)(e), the calculation results of $(X_{t-1} - L^{*} X_t)$. (b)(d)(f) Results of fault point detection. The red vertical lines indicate the actual fault injection points at 500, 1000, and 1500, respectively. The black vertical lines denote the detected fault points at 484, 1012, and 1496, respectively.

IV HOT-FIRE TESTING EXPERIMENTS

To further verify the effectiveness of the proposed change-point detection method, this paper validates it using real-world data from a hot-fire test of a certain model of liquid rocket engine. During the test, over 100 sensors were deployed on the engine to monitor its health status. A leakage fault was detected during the experiment. However, unlike in simulation scenarios, the actual fault occurrence time in the hot-fire test is unknown. The change-point detection results of the proposed method applied to the hot-fire test time series data are shown in Fig. 4. Considering that the original data contains confidential information, de-identification measures were applied. Specifically, the data was standardized and noise was injected. Only processed features and fault labels are published to ensure compliance with data security regulations.

In contrast to simulated environments, the method also proves effective when applied to real-world fault scenarios in liquid rocket engine testing.

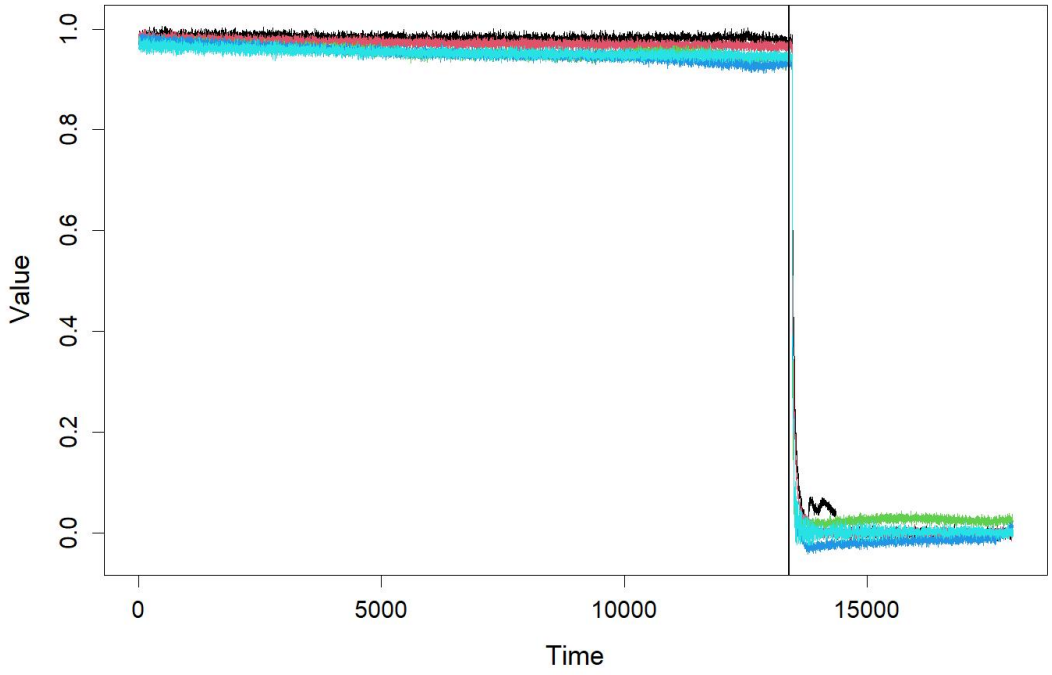


Fig 4. Fault point detection result for hot-fire testing experiments

V CONCLUSION

This paper proposes an offline change-point detection method based on the VAR model and applies it to fault diagnosis in liquid rocket engines (LREs). Fault point detection experiments were conducted on an LRE system simulation dataset, achieving a precision of 100%, which demonstrates the effectiveness of the method in the context of LRE fault diagnosis. In addition, the proposed method was further validated on real-world fault data from hot-fire testing experiments, confirming its practical applicability and robustness. Compared to the change-point detection method in [14], the proposed approach introduces the SCAD regularization technique, leveraging its non-convex penalty mechanism to reduce the shrinkage of large coefficients, thereby minimizing estimation bias and achieving a more accurate sparse solution. Furthermore, unlike artificial neural network methods, the proposed method does not require extensive training data and can localize fault points with relatively high precision.

However, the proposed method still has several limitations. First, although FISTA accelerates the optimization process, the block-wise segmentation strategy requires sequential processing of high-dimensional VAR models for each block. As the data scale increases—such as longer time series or more sensors—the number of blocks grows linearly, leading to significantly increased computational costs and challenges in meeting real-time requirements. Second, key parameters such as the regularization coefficients ($\lambda_{1,n}$, $\lambda_{2,n}$) and the convergence tolerance (ε) must be manually tuned, which relies on expert knowledge or trial-and-error. This parameter selection process is not only time-consuming

but also introduces subjectivity, potentially resulting in unstable outcomes. Finally, the method divides the time series into fixed-length blocks, implicitly assuming a uniform distribution of change points. In practice, however, failures in liquid rocket engines often concentrate in specific periods (e.g., the ignition phase). Fixed block sizes lack the flexibility to adapt to such temporal variations, potentially causing insufficient resolution in change-point-dense regions and redundant computation in sparse ones.

Moreover, the proposed framework is built on the assumption of piecewise stationarity. This assumption is supported by the operational characteristics of LREs, where the working process can be divided into several distinct phases such as ignition, steady combustion, and shutdown depressurization. Each phase typically corresponds to different control logics, fuel flow adjustments, or structural responses, which lead to clear differences between transient and steady states. While transient signals exhibit strong non-stationary behaviors (e.g., rapid thrust rise, chamber pressure oscillation, or temperature jumps), steady phases usually demonstrate small fluctuations and relatively stable inter-variable relationships, thus presenting local stationarity suitable for VAR modeling. Nevertheless, under conditions with strong high-frequency disturbances, this assumption may no longer hold, which imposes limitations on the model's generality.

We hope to further enhance the method's real-time applicability and robustness through adaptive segmentation and automated parameter selection in future work.

References

[1] Wang, T., Ding, L., & Yu, H. (2022). Research and development of fault diagnosis methods for liquid rocket engines. *Aerospace*, 9(9), 481.

[2] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

[3] Deng, C., Xue, W., Zheng, M., & Ma, H. (2020). Study on Real Time Diagnosis Method of the Main Stage Working Condition of Rocket Engine Based on Improved ARMA Model. Comput. *Meas. Control*, *28*, 33-38.

[4] Xue, W., Zhang, Q., & Wu, X. P. (2019). Based on the ARMA Model for the Liquid Rocket Propulsion Fault Detection. Comput. *Meas. Control*, *27*, 4-7.

[5] Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, 1-48.

[6] Lütkepohl, H. (2013). Vector autoregressive models. In *Handbook of research methods and applications in empirical macroeconomics* (pp. 139-164). Edward Elgar Publishing.

[7] Ahelegbey, D. F., Giudici, P., & Hashem, S. Q. (2021). Network VAR models to measure financial contagion. *The North American Journal of Economics and Finance*, *55*, 101318.

[8] Barbaglia, L., Croux, C., & Wilms, I. (2020). Volatility spillovers in commodity markets: A large t-vector autoregressive approach. *Energy Economics*, *85*, 104555.

[9] Hasanah, R. N., OMP, R. R., & Suyono, H. (2020, July). Comparison analysis of electricity load demand prediction using recurrent neural network (rnn) and vector autoregressive model (var). In *2020 12th International Conference on Electrical Engineering (ICEENG)*(pp. 23-29). IEEE.

[10] Davis, R. A., Zang, P., & Zheng, T. (2016). Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, *25*(4), 1077-1096.

[11] Billio, M., Casarin, R., & Rossini, L. (2019). Bayesian nonparametric sparse VAR models. *Journal of Econometrics*, *212*(1), 97-115.

[12] Dallakyan, A., Kim, R., & Pourahmadi, M. (2022). Time series graphical lasso and sparse VAR estimation. *Computational Statistics & Data Analysis*, *176*, 107557.

[13] Basu, S., Li, X., & Michailidis, G. (2019). Low rank and structured modeling of high-dimensional vector autoregressions. *IEEE Transactions on Signal Processing*, 67(5), 1207-1222.

[14] Bai, P., Safikhani, A., & Michailidis, G. (2020). Multiple change points detection in low rank and sparse high dimensional vector autoregressive models. *IEEE Transactions on Signal Processing,* 68, 3074-3089.

[15] Whitacker, L. H. L., Tomita, J. T., & Bringhenti, C. (2017). An evaluation of the tip clearance effects on turbine efficiency for space propulsion applications considering liquid rocket engine using turbopumps. *Aerospace Science and Technology*, 70, 55-65.

[16] Zhang, X., Hua, X., Zhu, J., & Ma, M. (2023). Intelligent fault diagnosis of liquid rocket engine via interpretable LSTM with multisensory data. *Sensors, 23*(12), 5636.

[17] Bickel, Peter J., Ya'acov Ritov, and Alexandre B. Tsybakov. "Simultaneous analysis of Lasso and Dantzig selector." (2009): 1705-1732.

[18] Fan, J., & Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, *96*(456), 1348–1360.

[19] Safikhani, Abolfazl, Yue Bai, and George Michailidis. 2021. "Fast and Scalable Algorithm for Detection of Structural Breaks in Big VAR Models." *Journal of Computational and Graphical Statistics* 31 (1): 176–89.

Appendix A

Table 1 summarizes the symbols used in this paper.

Table 1 List of symbols used in this paper.

Symbol	Meaning
X_t	Multivariate time series observation at time t
B_j	Transition matrix for the $j-th$ segment that reflects time-varying effects of historical values
L^*	Low rank component
S_j^*	Sparse component
p	Number of variables (sensor channels)
b_n	Intercept vector for the $n-th$ segment
k_n	Length of the $n-th$ segment
$\mathbf{X}_{r_j}$	Multivariate time series observations in block j
Y_{r_j}	Multivariate time series observations in block j
δ_{r_j}	Error (innovation) terms in block j
$\mathbb{X}$	Collection of all historical inputs across all blocks
$\mathbb{Y}$	Collection of all current outputs across all blocks
$\mathbb{E}$	Collection of all error terms.
$\mathbb{Z}$	Block-structured matrix built from segment predictors $\mathbf{X}_{r_j}$, describing segment-specific contributions (Eq. 7)
Θ	Parameter matrix capturing structural changes across segments(Eq. 9)
$\lambda_{1,n}, \lambda_{2,n}, \lambda_{3,n}$	Regularization parameters controlling penalties.
s_i	End index of the $i-th$ segment
$P_{\lambda,a}(x)$	SCAD (Smoothly Clipped Absolute Deviation) penalty function with parameters λ and a.
m	Number of segments (or sub-intervals) in the time series.
$P_{turbine}$	Turbine output power
Q	Mass flow rate through the turbine
$\eta_{turbine}$	Turbine efficiency
f	Rotational speed (frequency) of turbine shaft
$n_{turbine}$	Rotational speed of turbine (rpm)
T	Torque produced by turbine
$\dot{m}_3$	The total flow rate through the valve system
$\dot{m}_2$	The flow rate through the leakage valve.
$\dot{m}_1$	The flow rate through the primary valve.
c_q	Flow coefficient of the leakage passage.
A	Maximum flow area of the leakage passage.
ρ	Fluid density.
Δp	Pressure drop across the leakage valve.