

Article

Particle-Filter-Based Fault Diagnosis for the Startup Process of an Open-Cycle Liquid-Propellant Rocket Engine

Jiyoung Cha ¹, Sangho Ko ^{2,*} and Soon-Young Park ³¹ Centre for Aeronautics, Cranfield University, Cranfield MK43 0AL, UK; jiyoung.cha@cranfield.ac.uk² Department of Smart Air Mobility, Korea Aerospace University, 76 Hanggongdaehang-ro, Deogyang-gu, Goyang-si 10540, Republic of Korea³ Rocket Engine Department, Korea Aerospace Research Institute, 169-84 Gwahak-ro, Daejeon 34133, Republic of Korea; psy@kari.re.kr

* Correspondence: sanghoko@kau.ac.kr

Abstract: This study introduces a fault diagnosis algorithm based on particle filtering for open-cycle liquid-propellant rocket engines (LPREs). The algorithm serves as a model-based method for the startup process, accounting for more than 30% of engine failures. Similar to the previous fault detection and diagnosis (FDD) algorithm for the startup process, the algorithm in this study is composed of a nonlinear filter to generate residuals, a residual analysis, and a multiple-model (MM) approach to detect and diagnose faults from the residuals. In contrast to the previous study, this study makes use of the modified cumulative sum (CUSUM) algorithm, widely used in change-detection monitoring, and a particle filter (PF), which is theoretically the most accurate nonlinear filter. The algorithm is confirmed numerically using the CUSUM and MM methods. Subsequently, the FDD algorithm is compared with an algorithm from a previous study using a Monte Carlo simulation. Through a comparative analysis of algorithmic performance, this study demonstrates that the current PF-based FDD algorithm outperforms the algorithm based on other nonlinear filters.

Keywords: fault detection and diagnosis; particle filter; CUSUM algorithm; multiple-model method; liquid-propellant rocket engine; startup process



Citation: Cha, J.; Ko, S.; Park, S.-Y. Particle-Filter-Based Fault Diagnosis for the Startup Process of an Open-Cycle Liquid-Propellant Rocket Engine. *Sensors* **2024**, *24*, 2798. <https://doi.org/10.3390/s24092798>

Academic Editor: Hossam A. Gabbar

Received: 30 March 2024

Revised: 24 April 2024

Accepted: 25 April 2024

Published: 27 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Liquid-propellant rocket engines (LPREs) are mostly utilized for space launch vehicles (SLVs) or reusable launch vehicles (RLVs) for New Space because of their higher specific impulse and the accurate control capability required for precise injection into orbit [1–5]. However, given the complexity of LPREs and their multitude of components, achieving high reliability necessitates employing condition-based maintenance techniques. These include non-destructive inspection (NDI), fault detection and diagnosis (FDD), and strategies for prognosis and health management (PHM) based on sensor measurements obtained from the engines [6–26]. In terms of the FDD of LPREs, researchers have actively studied two distinguishing methods: data-driven and model-based methods. Due to the insufficiency of LPRE fault scenarios during actual firing tests, model-based methods are widely studied. Typically, these methods are based on the fault characteristics that are derived from a mathematical model with artificially inserted faults [14,15]. In the same context, this study focuses on developing a new FDD methodology using measurement data with these motivations and requirements.

In the case of LPRE operational failures, Refs. [27,28] show that a significant proportion (exceeding 30%) of engine failures occur in the startup process due to various factors: leakage and blockage in the propellant pipeline, combustion instabilities caused by the water hammer in the priming process, inappropriate initial setting and opening times of each valve, etc. Most of these failures were immediately catastrophic, so the failures caused not only mission failure but also damage to both materials and individuals. To resolve

these problems, in a previous study, Cha et al. [21] applied the extended Kalman filter (EKF) and unscented Kalman filter (UKF) as a model-based method to develop an FDD algorithm that could physically analyze fault conditions in the startup process of an LPRE. However, the usage of the two nonlinear Kalman filters has some limitations, namely, false alarms due to linearization, global approximation, and the assumption errors of the PDF of the fault condition [29,30] in the residual generation and fault detection processes with the Neyman–Pearson (NP) theory [31]. Therefore, in order to overcome these limitations of the previous study, this study proposes a new FDD algorithm employing a particle filter (PF) and a modified cumulative sum (CUSUM) algorithm, as shown in Figure 1. In the figure, the process of the FDD algorithm is delineated: the current study (depicted within solid boxes) and the previous study [21] (represented by dashed boxes).

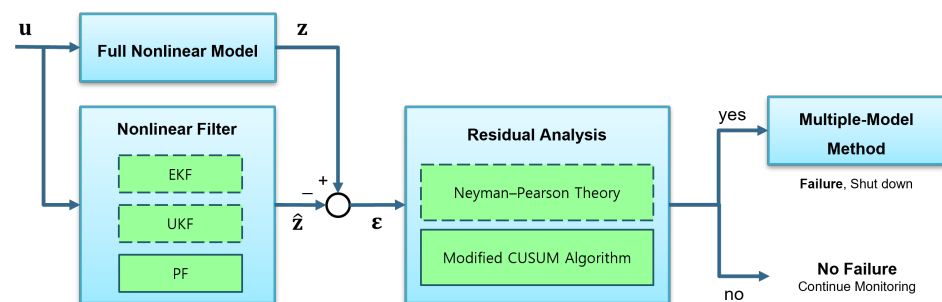


Figure 1. The structure of the FDD algorithm.

To validate the performance of the proposed FDD algorithm, we employed two fault scenarios, a turbopump efficiency deficiency and pipeline blockage, which are common sources of LPRE failures [27]. The dynamic response data were acquired by applying fault scenarios to the mathematical model of the LPRE described in Ref. [15]. Then, we designed the filters, generated residuals, and detected and diagnosed each fault with the CUSUM algorithm and multiple-model (MM) method using the residuals. For each process of the FDD algorithm, a Monte Carlo simulation was conducted to compare the performance of each FDD algorithm. This work analyzed and compared the FDD algorithms quantitatively and qualitatively under random noise conditions. Through this work, we demonstrate that the FDD algorithm with the PF and CUSUM algorithm has superior performance compared to those in the previous study.

In this study, we selected an open-cycle LPRE as a base model to study, and a brief introduction is presented in Section 2. Section 3 explains the PF to generate residuals, which are the basis of decision-making in the FDD algorithm, and Section 4 describes the FDD algorithm, which is composed of a modified CUSUM algorithm to detect a fault and the MM method to diagnose the fault using the residuals. Then, Section 5 compares the results of the newly proposed FDD algorithm with those of the previous study using the EKF/UKF. Finally, in Section 6, the concluding remarks and limitations of this study are discussed.

2. Dynamics of LPREs

2.1. Dynamic Simulation of LPREs

Using the MATLAB/Simulink environment, we constructed an open-cycle LPRE simulation program for dynamic simulation [32] for the open-cycle LPRE employed in this study, as shown in Figure 2. The mathematical model is described in more detail in Appendix A.

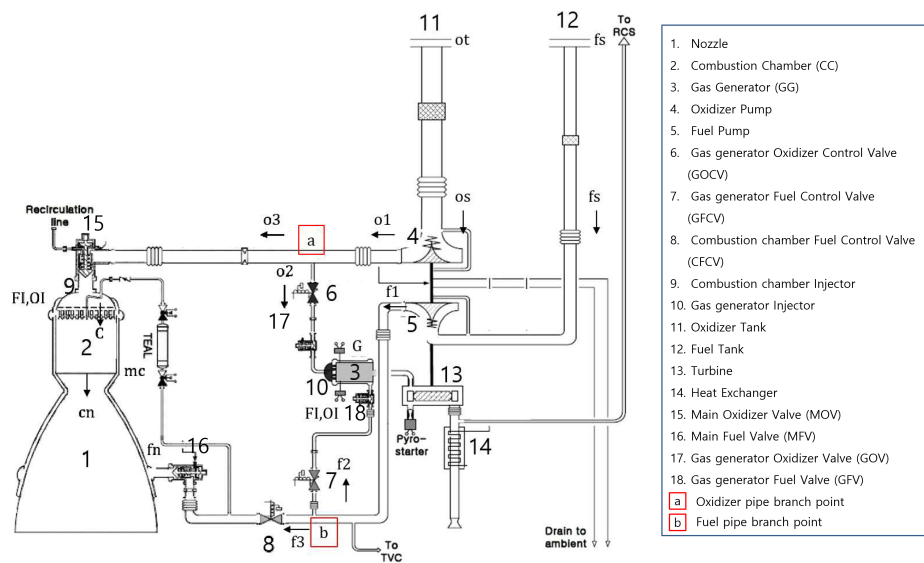


Figure 2. A schematic of a liquid-propellant rocket engine [32].

The open-cycle LPRE is analyzed differently depending on whether the operating valves are in a steady state or transient state [33]. During the startup process, one of the transient states, four main valves (Nos. 15, 16, 17, and 18 in Figure 2) affect the performance of the engine and cause it to operate at full thrust as swiftly as possible. Upon reaching the steady state, known as full thrust, the LPRE performance can be controlled using three control valves (Nos. 6, 7, and 8 in Figure 2). Finally, during the shutdown process, another transient state, the LPRE operation is terminated by the four main valves.

To ensure the accuracy of the simulation results, they were compared with experimental data from the steady and transient states. In the steady state (full thrust) of the simulation, with all valves and parameters fixed, a maximum error of 3.7% and a minimum error of 0% were observed [32]. However, during transient states such as the startup or shutdown procedures, four main valves were fully opened and closed at the appropriate time, respectively [16]. The comparison of transient states (startup and shutdown processes) is provided in Appendix B.

2.2. Fault Modeling

A fault is an unexpected change or an unpermitted deviation in system parameters from the standard no-fault condition, so a fault in a system may lead to system malfunction or mission failure [27]. This can be performed mathematically by combining the corresponding fault types; hence, we artificially injected each fault signal following the proven approach outlined in Refs. [15,20], which is verified with actual datasets. For the fault conditions in this study, turbopump faults and pipeline blockages, which are representative faults [15,20,27], are considered sudden faults:

$$\theta(t) = \begin{cases} \theta_0 & \text{for } t < t_0 \\ \theta_0(1 - f) \triangleq \theta_1 & \text{for } t_0 \leq t \end{cases} \quad (1)$$

where θ is a parameter of the system, θ_0 is the parameter in a normal condition, and $f = \frac{\Delta\theta}{\theta_0}$ represents the fault factor.

Even though Equation (1) describes fault development as a step function, the fault mathematical model can also be modeled by applying linear, quadratic, cubic, and exponential functions and changing the fault development time depending on the fault characteristics.

2.2.1. Turbopump Fault

There are many factors for a turbopump fault, such as cavitation, rotor bearing faults, and turbine blade faults, which are difficult to implement individually by mathematical modeling. However, most faults in a pump or turbine induce a decrease in efficiency. Therefore, using Equation (1), we express a fault condition mathematically by injecting an efficiency decrease ($\theta = \eta_t$ for turbine or η_p for pump in Equation (1)) considering an abruptly occurring ablation of turbine nozzle blades [15,20].

2.2.2. Pipeline Fault

Pipeline faults are mostly leakages or hole blockages [15,20], but we only consider a pipeline blockage to be simple in this study by decreasing the pipe cross-sectional area (A). For this, using Equation (1), we express a blockage fault due to some impurities in propellants by injecting the area variation ($\theta = A$ in Equation (1)) into a part of the pipeline (CC oxidizer pipeline (o3) and CC fuel pipeline (f3) in Figure 2).

3. Particle Filter for Residual Generation

This section delineates the PF to generate the residuals for an FDD algorithm of an open-cycle LPRE. In previous research, Cha et al. [21] developed the FDD algorithm with the EKF/UKF. However, the linearization process in the EKF yields approximation errors in each prediction/update step, and it is difficult to globally approximate based on a small set of trial points in the UKF [29,30]. Consequently, these issues may lead to poor detection or high false alarm rates for the FDD method. The PF, a method based on Monte Carlo simulation principles for a nonlinear and non-Gaussian dynamic model, is appealing for its ability to adeptly handle various nonlinear distribution characteristics of measurement noise because it updates probability densities using a Bayesian approach [34,35].

The dynamic simulation program can be expressed using continuous state dynamics and a discrete measurement procedure as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(t), \quad (2)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \quad (3)$$

where $\mathbf{u}_k \in \mathbb{R}^m$, $\mathbf{x}_k \in \mathbb{R}^n$, and $\mathbf{z}_k \in \mathbb{R}^l$ are the input, state, and output variables, respectively, and $\mathbf{w}(t) \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^l$ are the state and measurement noises, respectively, assuming normal distributions with the covariances $\mathbf{Q}(t)$ and $\mathbf{R}_k$, respectively.

For the nonlinear system, the PF was proposed to represent and recursively generate an approximation for the conditional probability density function (PDF), $p(\mathbf{x}_k | \mathcal{Z}_k)$, where $\mathcal{Z}_k = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$. The fundamental concept of the PF is to obtain and portray the necessary PDF using particles in a swarm. The swarm can be regarded as the realization of random samples from the required PDF in each step. Thus, as the number of particles increases, they tend to approach the necessary PDF more closely [34,35]. Figure 3 illustrates the process of the PF algorithm. Then, we generate the residuals (ϵ_k) using the system output variables ($\mathbf{z}_k$) and estimated output variables ($\hat{\mathbf{z}}_k$) as follows:

$$\epsilon_k = \mathbf{z}_k - \hat{\mathbf{z}}_k \quad (4)$$

(Step 1) Assumption:

Set N particles randomly $\{\mathbf{x}_i(t): i = 1, 2, \dots, N\}$ from the PDF $p(\mathbf{x}_k|Z_k)$.

(Step 2) Prediction:

Use sample values from the system noise PDF to generate a new swarm of points.

$$\hat{\mathbf{x}}_i(t) = f(\mathbf{x}_i(t), \mathbf{u}(t)) + \mathbf{w}_i(t), i = 1, 2, \dots, N$$

(Step 3) Update:

Assign a weight $\omega_k(i)$ to each $\hat{\mathbf{x}}_{k|k-1}(i)$ and update the state with the weight.

$$\omega_k(i) = \frac{p(\mathbf{z}_k|\hat{\mathbf{x}}_{k|k-1}(i))}{\sum_{j=1}^N p(\mathbf{z}_k|\hat{\mathbf{x}}_{k|k-1}(j))} \text{ where } \hat{\mathbf{x}}_{k|k-1}(i) = \int_{(k-1)dt}^{kdt} \dot{\hat{\mathbf{x}}}_i(t) dt$$

$$\hat{\mathbf{z}}_k = h(\hat{\mathbf{x}}_k) \text{ where } \hat{\mathbf{x}}_k = \sum_{i=1}^N \omega_k(i) \hat{\mathbf{x}}_{k|k-1}(i)$$

Calculate the residuals $\boldsymbol{\varepsilon}_k$.

$$\boldsymbol{\varepsilon}_k = \mathbf{z}_k - \hat{\mathbf{z}}_k$$

(Step 4) Resampling:

Use resamples to avoid degeneracy by comparing threshold (N_T) and the effective particle sample size (N_{eff}) with “true weight,” which cannot be evaluated precisely. So, estimate $\hat{N}_{eff}$ obtained by

$$N_{eff} \triangleq \frac{N}{1 + \text{Var}(\omega_k^*(i))} \leq N_T \Rightarrow \hat{N}_{eff} \triangleq \frac{1}{\sum_{i=1}^N (\omega_k(i))^2} \leq N_T$$

Resample N times independently from the distribution. The obtained particles satisfy the probability based on a proper sample generated from the posterior PDF $p(\mathbf{x}_k|Z_k)$.

$$\Pr(\mathbf{x}_k^*(i) = \mathbf{x}_k(j)) = \omega_k(j)$$

Figure 3. The four-step process for the particle filter [34].

4. Fault Detection and Diagnosis Using Residuals

Now, the CUSUM and MM methods use the residuals generated by the PF for the FDD process. As illustrated in Figure 1, the algorithm progresses through two blocks after residuals are generated by the PF: (1) the residual analysis block to detect faults and (2) the MM method block to diagnose faults.

4.1. CUSUM Algorithm for Fault Detection

The residual analysis process determines whether a fault occurred through the residual-change-checking method, for example, by identifying whether a signal exceeds a threshold [36]. However, because the sensor measurements may contain anomalies due to various factors, such as noise, impulsive noise can be associated with the threshold being exceeded during the data acquisition process or by the sensor itself. Therefore, to overcome this problem, this study proposes a residual analysis method based on a modified CUSUM algorithm, widely used for fault detection [37]. As the mean value of the residuals generated by a filter shifts when a fault occurs, the algorithm can detect whether and when a fault occurs by checking a change in the mean values between the normal condition (θ_0) and a fault condition (θ_1) (see Equation (1)) [16]. Figure 4 presents the process of the CUSUM algorithm, through which we can detect a fault considering two cases by checking whether the CUSUM algorithm results (S_k^u and S_k^l) are zero or not. Since the results are zero until the absolute value of each residual ($|\varepsilon_k|$) exceeds $\delta/2$, the variation magnitude (δ) in the process is a tuning parameter. Hence, the magnitude (δ) should be determined by considering the factor of safety (FS) of the engine. Generally, the FS depends on the components and engine

types, but it is usually at least 1.1 (10% FS), especially for the combustion chamber because of combustion instability [38,39]. Therefore, here, we set the variation magnitude (δ) to 15% ($\delta/2 = 7.5\%$) of each output variable under the normal condition to identify a fault occurring before exceeding the 10% FS.

(Step 1) Assume the probability density function:

Assume the probability density function in the log-likelihood ratio is a Gaussian distribution with constant variance (σ^2) and the mean values (μ) between the normal condition (θ_0) and a fault condition (θ_1) change.

$$s_k = \ln \frac{p_{\theta_1}(\varepsilon_k)}{p_{\theta_0}(\varepsilon_k)} \Rightarrow s_k = \frac{\mu_{\theta_1} - \mu_{\theta_0}}{\sigma_{\theta_0}^2} \left(x_k - \frac{\mu_{\theta_1} + \mu_{\theta_0}}{2} \right)$$

(Step 2) Determine the variation magnitude:

As μ_{θ_1} is unknown, determine the variation magnitude (δ) of the mean value. Then, the log-likelihood ratio becomes

$$s_k^u = \frac{\delta}{\sigma_{\theta_0}^2} \left(\varepsilon_k - \frac{\delta}{2} \right) \quad \text{with} \quad \mu_{\theta_1} = \mu_{\theta_0} + \delta.$$

Since the change can be either positive or negative, the negative case is expressed as

$$s_k^l = -\frac{\delta}{\sigma_{\theta_0}^2} \left(\varepsilon_k + \frac{\delta}{2} \right) \quad \text{with} \quad \mu_{\theta_1} = \mu_{\theta_0} - \delta.$$

(Step 3) Find the change from zero:

Find the non-zero moment using the max or min function.

$$S_k^u \text{ and } S_k^l = 0 \Rightarrow \text{normal condition } (\theta_0)$$

$$S_k^u \text{ or } S_k^l \neq 0 \Rightarrow \text{fault condition } (\theta_1)$$

with

$$S_k^u = \max\{0, S_{k-1}^u + s_k^u\} \text{ for positive change case}$$

$$S_k^l = \min\{0, S_{k-1}^l - s_k^l\} \text{ for negative change case}$$

Figure 4. The three-step process of the CUSUM algorithm [37].

To see the algorithm performance, we applied the algorithm to two fault cases: a decreased turbine efficiency fault for internal faults and the CC pressure sensor positive bias fault for sensor/valve faults. For this, we artificially injected each fault at 0.9 s during the startup process (see Figure A2) and utilized the CUSUM algorithm using the residuals generated by the PF. Figure 5 shows the results of the CUSUM algorithm for a turbine efficiency fault (Figure 5a) and the CC pressure sensor bias fault (Figure 5b) using seven normalized state variables: P_c , P_g , ω , $\dot{m}_{03}$, $\dot{m}_{f3}$, $\dot{m}_{02}$, and $\dot{m}_{f2}$ (see Table 1). In the turbine efficiency fault case, the fault affects all engine components, so all the variables change. On the other hand, a sensor bias fault causes only the corresponding sensor output variables to change. Therefore, all results of the CUSUM algorithm in Figure 5a change in the negative case (blue line, S_k^l), while in Figure 5b, only the CC pressure (P_c) changes in the positive case (red line, S_k^u) and others do not. Through this process, we can see that the CUSUM algorithm can detect a fault by checking the change in the residuals. Furthermore, we can deduce from additional information that the CUSUM algorithm can distinguish between the internal component fault and the sensor fault from the number of changes in the results.

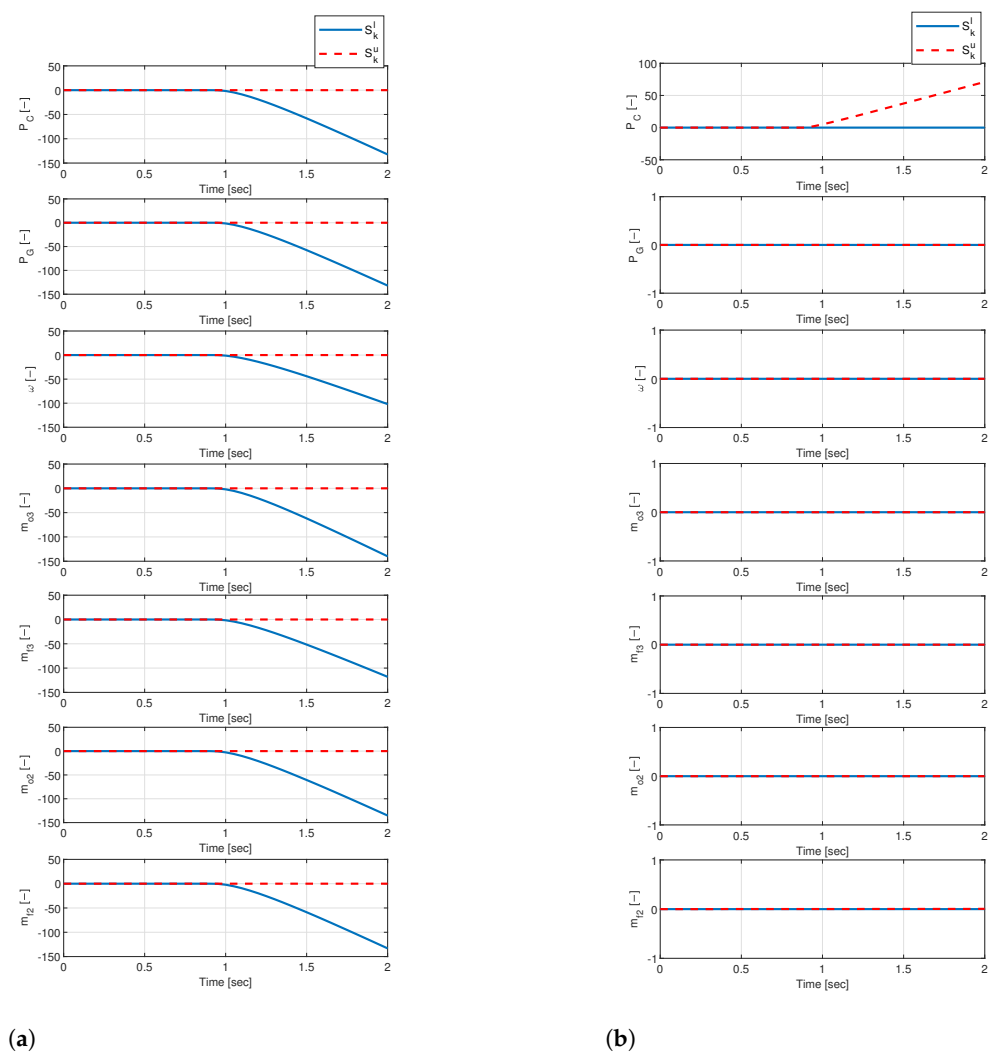


Figure 5. The results of the CUSUM algorithm (a) internal component fault results; (b) sensor fault results.

Table 1. Inverse of signal-to-noise ratio.

Notation	Meaning	1/SNR [%]
P_C	Combustion chamber pressure	0.47
P_G	Gas generator pressure	0.51
$\dot{m}_{o3}$	Combustion chamber oxidizer flow rate	0.48
$\dot{m}_{f3}$	Combustion chamber fuel flow rate	0.57
ω	Turbopump rotational speed	0.40
$\dot{m}_{OIC}$	Combustion chamber oxidizer injector flow rate	0.48
$\dot{m}_{FIC}$	Combustion chamber fuel injector flow rate	0.57
$\dot{m}_{OIG}$	Gas generator oxidizer injector flow rate	1.88
$\dot{m}_{FIG}$	Gas generator fuel injector flow rate	2.29
$\dot{m}_{o2}$	Gas generator oxidizer flow rate	1.88
$\dot{m}_{f2}$	Gas generator fuel flow rate	2.29

4.2. Multiple-Model Method for Fault Diagnosis

Once a fault is detected, the next step is to diagnose the fault, determining its location and severity. The MM method, consisting of multiple filters, is employed for fault diagnosis after the initial detection [16]. The decision mechanism is based on the residuals generated by a set of N filters. Each filter is designed based on different hypothesized models. If

one of the hypothesized models is true, the residuals and covariance generated by the corresponding filter are small. Therefore, the hypothesized model that produces the smallest residual and covariance can be regarded as the true model [21]. This study assumes that a fault occurs in each of the five components for each hypothesized model, including the efficiency decreases among the turbine, fuel pump, and oxidizer pump, and two blockages in the fuel and oxidizer pipelines for the hypothesized models, and the artificially injected fault signature in the turbine efficiency at 0.9 s (see Figure A2) [16].

The MM method results are depicted in Figure 6. Under the normal condition, the probability is mostly 1, and when the fault occurs, the probability changes to 0, and the turbine fault probability changes to 1. During this process, the probability of an oxidizer pump fault briefly reaches 1 due to the similarity of fault effects to those of turbine and oxidizer pump efficiency faults. Therefore, it can be confused at the beginning of the occurring fault, and the method can accurately diagnose the fault that eventually occurs.

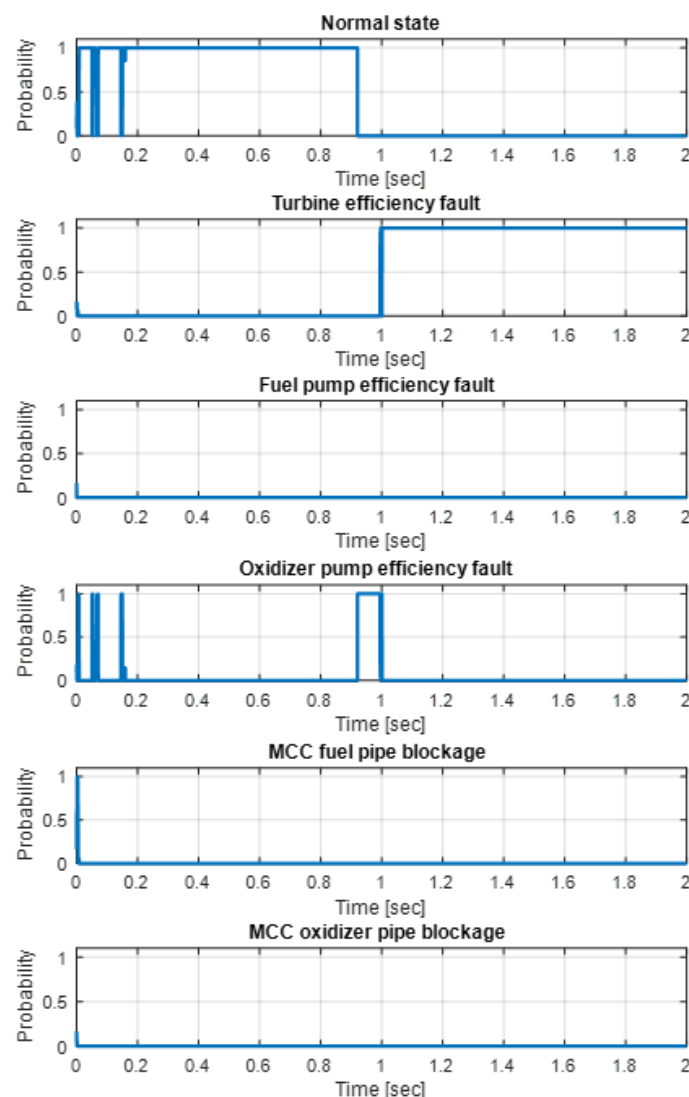


Figure 6. The results of the multiple-model method with the PF under the turbine efficiency fault [16].

5. PF-CUSUM-MM Application Results

This section validates the feasibility of the proposed FDD algorithm. For this, we artificially injected a reasonable fault into the startup process using the minimum detectable fault analysis. Then, to evaluate the FDD algorithm, we used a Monte Carlo simulation in each process of the FDD algorithm and compared the performance with the previous results using the EKF/UKF. The process of the FDD algorithm is as follows: it first detects

a fault using the residuals generated by a nonlinear filter, and then the fault diagnosis algorithm runs after the fault is detected. The overall flowchart of the proposed algorithm is shown in Figure 7.

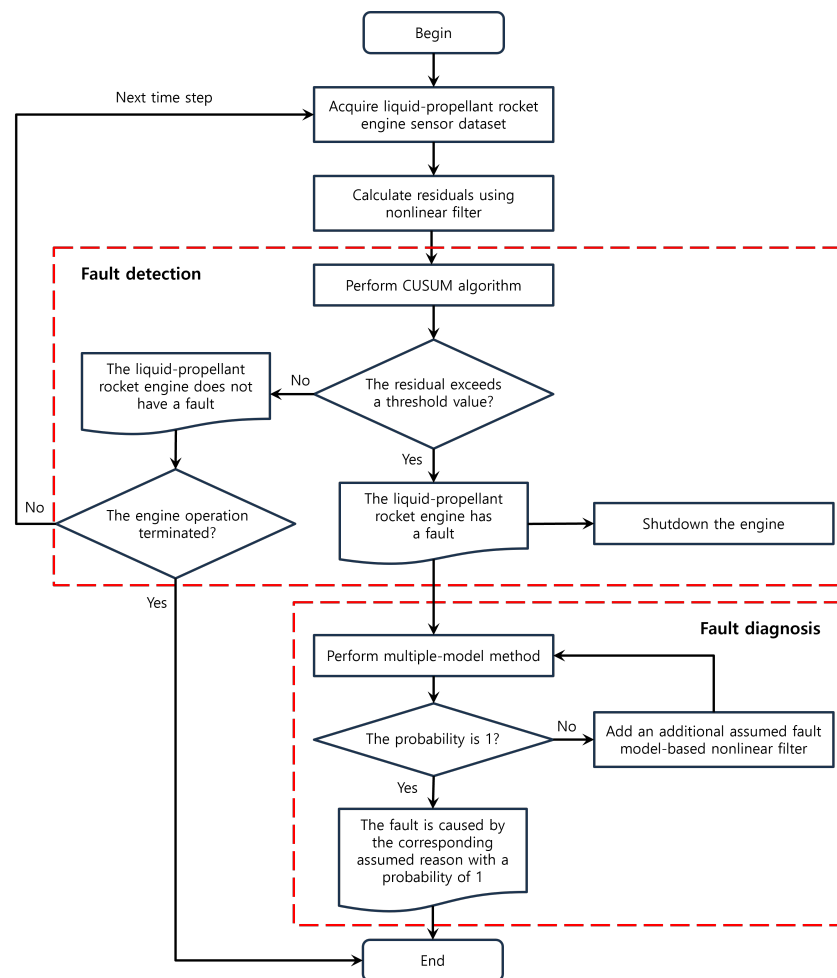


Figure 7. The flowchart of the proposed algorithm.

5.1. Minimum Detectable Fault Analysis

To evaluate the performance of the FDD algorithm using an LPRE simulation with artificially injected faults, it is crucial to determine a reasonable initial fault magnitude for each fault model. Small fault sizes relative to the noise level can make detection and diagnosis challenging, potentially resulting in false alarms during the process [40]. Therefore, we analyzed the minimum detectable fault severity (MDFS) for five fault cases, each with four different fault magnitudes (5%, 10%, 15%, and 20%). This analysis compared the fault severity with the noise level, determined by the inverse of the signal-to-noise ratio (SNR) using Equation (5) (see Table 1), in scenarios involving efficiency decreases in the turbine, fuel pump, and oxidizer pump, as well as blockages in the CC oxidizer and CC fuel pipelines. However, since there are eleven output variables ($n = 11$) and a fault affects each output variable differently, we used the MDFS of each fault by calculating the mean error rate value from the normal condition value of each output variable using Equation (6) and summarize the results in Table 2.

$$\text{SNR}_i = \frac{E[s_{normal}^i]}{E[\sigma_i]}, \quad i = 1, \dots, n, \quad (5)$$

$$\text{MDFS} = \frac{1}{n} \sum_{i=1}^n \frac{E[s_{\text{fault}}^i] - E[s_{\text{normal}}^i]}{E[s_{\text{normal}}^i]}, \quad (6)$$

where σ_i represents the standard deviation of the i -th output variable noise, and s_{normal}^i and s_{fault}^i are the i -th signal of the eleven output variables under normal and fault conditions, respectively.

Table 2. The minimum detectable fault severity of each fault type and magnitude [16].

Fault Type	Fault Injection Level [%]							
	5%		10%		15%		20%	
	Peak	Final	Peak	Final	Peak	Final	Peak	Final
Turbine efficiency	5.88	5.88	11.80	11.80	17.77	17.77	23.78	23.78
Oxidizer pump efficiency	3.11	3.11	6.39	6.39	9.86	9.86	13.54	13.54
Fuel pump efficiency	2.82	2.82	5.81	5.81	8.99	8.99	12.39	12.39
CC oxidizer pipeline blockage	0.48	0.39	1.03	0.84	1.66	1.35	2.39	1.94
CC fuel pipeline blockage	1.29	1.11	2.67	2.32	4.15	3.61	5.74	5.01

All MDFSs for turbine and pump faults exceed the values listed in Table 1. This indicates that, even at the smallest fault magnitude (5%) for turbine or pump efficiency faults, the effects of the faults are greater than the noise effects. Therefore, the deviations can be identified as being caused by faults, rather than noise. In contrast, for the fuel pipeline blockage fault up to a magnitude of 5% and for the oxidizer pipeline blockage fault up to a magnitude of 15%, the MDFSs are smaller than the values listed in Table 1. Then, at those fault magnitudes, it becomes difficult to distinguish between deviations caused by faults and those caused by noise, which leads to an increase in false alarms. Therefore, this study uses 20% for each fault magnitude, where all MDFSs exceed all values listed in Table 1.

5.2. Performance Validation: A Comparison with the EKF and UKF

To confirm the algorithm performance, we compared the algorithms proposed in this study with those used in the previous study [21], which employed the other two nonlinear Kalman filters, focusing on a qualitative analysis. For this, a Monte Carlo simulation was conducted at each stage of the FDD algorithm, as illustrated in Figure 1. Subsequently, we compared the simulation results for the FDD algorithm based on each filter. In the Monte Carlo simulation process, we set two types of noise, Gaussian ($\mathcal{N}(0, \sigma^2)$) and uniform ($\mathcal{U}(-3\sigma, 3\sigma)$) distributions, and added them to the simulation data in each case.

Figure 8 shows the comparison results of the CUSUM algorithm based on the EKF, UKF, and PF with Gaussian and uniform noises. The results of each detection probability show that the CUSUM algorithm, using the residuals from the PF, is better than the other nonlinear filters. The CUSUM algorithm using the UKF is better than the EKF, except for in some fault cases (in the pipeline blockage fault) with a uniform noise distribution. This could be because the UKF generates the residuals using a set of points obtained from the Gaussian distribution, whereas the EKF uses the linearization approach. Therefore, under noise distributions other than the Gaussian distribution, the algorithm based on the UKF may yield worse results compared to those with the EKF.

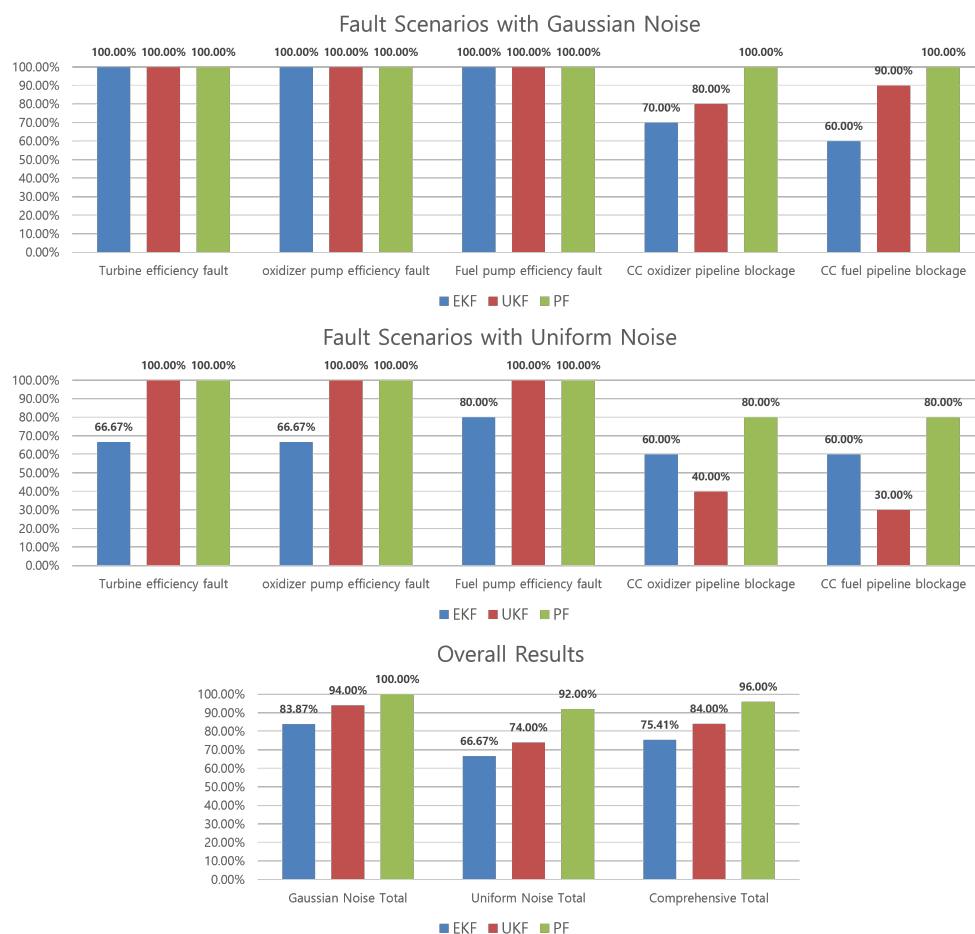


Figure 8. Monte Carlo simulation results of the CUSUM algorithm.

Subsequently, a Monte Carlo simulation was conducted using the MM method to compare the performance in the diagnostic process by applying the residuals and error covariances generated by each filter. Figure 9 shows that the MM method, employing residuals and covariance derived from the PF, outperforms the other nonlinear filters (EKF/UKF).

According to these two results, the PF is superior to the other filters in the Monte Carlo simulation. In addition, since the algorithm with the PF has a computational cost and performance that depend on the number of particles, the FDD algorithm combined with the PF is expected to demonstrate greater effectiveness in posterior analysis, independent of computational complexity. For this analysis, the algorithm utilizes secured flight status and system condition datasets transmitted via telemetry devices during flight missions as the system output variables. Through this process, the cause of the fault can be determined and analyzed. Subsequently, this posterior analysis will facilitate failure analysis and enable learning from mistakes. It will also ensure the reliability of launch vehicles and the successful performance of Post-Mission Disposal (PMD) [26].

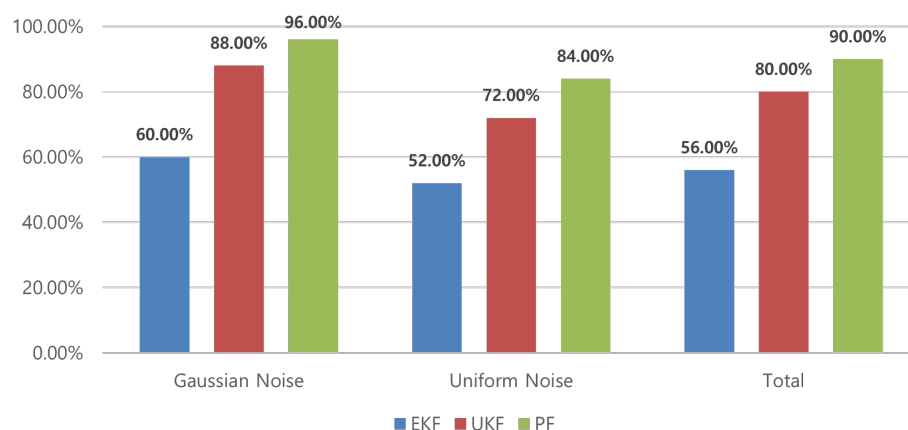


Figure 9. Monte Carlo simulation results of the multiple-model method.

6. Conclusions

This study developed an FDD algorithm consisting of a PF, CUSUM algorithm, and MM approach for an open-cycle LPRE in the startup process using a nonlinear simulation verified using experimental data from both the steady and transient states. We then designed a PF to generate residuals and detected and diagnosed faults using a modified CUSUM algorithm and MM method. To assess the FDD algorithm, we numerically evaluated the performance of the CUSUM algorithm and the MM method in each FDD step. Subsequently, we compared the performance with that obtained in a previous study [21] using the Monte Carlo simulation under various fault conditions and noise distributions. In this study, the FDD algorithm based on the PF performed better than the FDD algorithm employing the other nonlinear filters, on average. Furthermore, considering the balance between computational resources and algorithmic performance, the FDD algorithm utilizing the PF can enhance effectiveness, particularly in posterior analysis tasks. The procedures used for other fault cases can be found in Cha's study [16].

Author Contributions: Conceptualization, J.C., S.K. and S.-Y.P.; methodology, J.C. and S.K.; software, J.C.; validation, J.C., S.K. and S.-Y.P.; resources, J.C., S.K. and S.-Y.P.; preparation of the initial draft, J.C. and S.K.; modifications and revisions, J.C., S.K. and S.-Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2022-00164702).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

A	Cross-area of pipe
I_{tp}	Moment of inertia of turbopump rotor
L	Length of pipe
P_a	Oxidizer pipe branch point (a) pressure
P_b	Fuel pipe branch point (b) pressure
P_C	Combustion chamber pressure
P_{fic}	Combustion chamber fuel injector inlet pressure
P_{fig}	Gas generator fuel injector inlet pressure

P_G	Gas generator pressure
P_{oic}	Combustion chamber oxidizer injector inlet pressure
P_{oig}	Gas generator oxidizer injector inlet pressure
R_C	Combustion chamber gas constant
R_G	Gas generator gas constant
T_C	Combustion chamber temperature
T_G	Gas generator temperature
V_C	Combustion chamber volume
V_G	Gas generator volume
$\dot{m}_{cn}$	Nozzle inlet flow rate
$\dot{m}_{FIC}$	Combustion chamber fuel injector flow rate
$\dot{m}_{FIG}$	Gas generator fuel injector flow rate
$\dot{m}_{f2}$	Gas generator fuel flow rate
$\dot{m}_{f3}$	Combustion chamber fuel flow rate
$\dot{m}_G$	Gas generator outlet flow rate
$\dot{m}_{OIC}$	Combustion chamber oxidizer injector flow rate
$\dot{m}_{OIG}$	Gas generator oxidizer injector flow rate
$\dot{m}_{o2}$	Gas generator oxidizer flow rate
$\dot{m}_{o3}$	Combustion chamber oxidizer flow rate
$\dot{m}_G$	Generated mass flow rate from solid propellant
ϵ_{fc}	Amount of time delay in combustion fuel injector
ϵ_{fg}	Amount of time delay in gas generator fuel injector
ϵ_{oc}	Amount of time delay in combustion oxidizer injector
ϵ_{og}	Amount of time delay in gas generator oxidizer injector
λ	Pressure loss coefficient
ρ_f	Fuel density
ρ_o	Oxidizer density
τ_{fp}	Fuel pump torque
τ_{op}	Oxidizer pump torque
τ_{tb}	Turbine torque
ω	Angular velocity of turbopump

Appendix A. Mathematical Model of the Liquid-Propellant Rocket Engine

The mathematical modeling of the LPRE used in this paper can be expressed in the form of differential and algebraic equations (DAEs). The differential equations are derived from seven governing equations, while the algebraic equations are derived from either empirical equations or the laws of thermal-fluid dynamics [33]. To streamline the LPRE simulation model, we utilized four governing equations, assuming a constant coefficient representation for heat transfer dynamics in the mass flow rate dynamics. Using this approach, we developed an open-cycle LPRE simulation program that employs 11 first-order ordinary differential equations and 37 algebraic equations [32]. The 11 nonlinear equations of motion derived from the governing equations are listed in Table A1. Additionally, the highest level of the simulation program in the MATLAB/Simulink environment is depicted in Figure A1.

Table A1. Dynamic model equations.

Number	Governing Equation	Mathematical Equation
1	Rotational dynamics	$I_{tp} \frac{d\omega}{dt} = \tau_{tb} - \tau_{op} - \tau_{fp}$
2		$\left(\frac{L}{A}\right) \frac{d\dot{m}_{o3}}{dt} = P_a - P_{oic} - \left(\frac{\lambda}{2\rho_o A^2}\right) \dot{m}_{o3}^2$
3		$\left(\frac{L}{A}\right) \frac{d\dot{m}_{f3}}{dt} = P_b - P_{fic} - \left(\frac{\lambda}{2\rho_f A^2}\right) \dot{m}_{f3}^2$
4		$\left(\frac{L}{A}\right) \frac{d\dot{m}_{o2}}{dt} = P_a - P_{oig} - \left(\frac{\lambda}{2\rho_o A^2}\right) \dot{m}_{o2}^2$

Table A1. Cont.

Number	Governing Equation	Mathematical Equation
5	Pressure dynamics	$\left(\frac{L}{A}\right) \frac{d\dot{m}_{f2}}{dt} = P_b - P_{fig} - \left(\frac{\lambda}{2\rho_f A^2}\right) \dot{m}_{f2}^2$
6		$\left(\frac{V_c}{R_c T_c}\right) \frac{dP_c}{dt} = \dot{m}_{OIC} + \dot{m}_{FIC} - \dot{m}_{cn}$
7		$\left(\frac{V_g}{R_g T_g}\right) \frac{dP_g}{dt} = \dot{m}_{OIG} + \dot{m}_{FIG} - \dot{m}_G$
8	Time-delay equation	$\frac{d\dot{m}_{OIC}}{dt} = \frac{1}{\epsilon_{oc}} (\dot{m}_{o3} - \dot{m}_{OIC})$
9		$\frac{d\dot{m}_{FIC}}{dt} = \frac{1}{\epsilon_{fc}} (\dot{m}_{f3} - \dot{m}_{FIC})$
10		$\frac{d\dot{m}_{OIG}}{dt} = \frac{1}{\epsilon_{og}} (\dot{m}_{o2} - \dot{m}_{OIG})$
11		$\frac{d\dot{m}_{FIG}}{dt} = \frac{1}{\epsilon_{fg}} (\dot{m}_{f2} - \dot{m}_{FIG})$

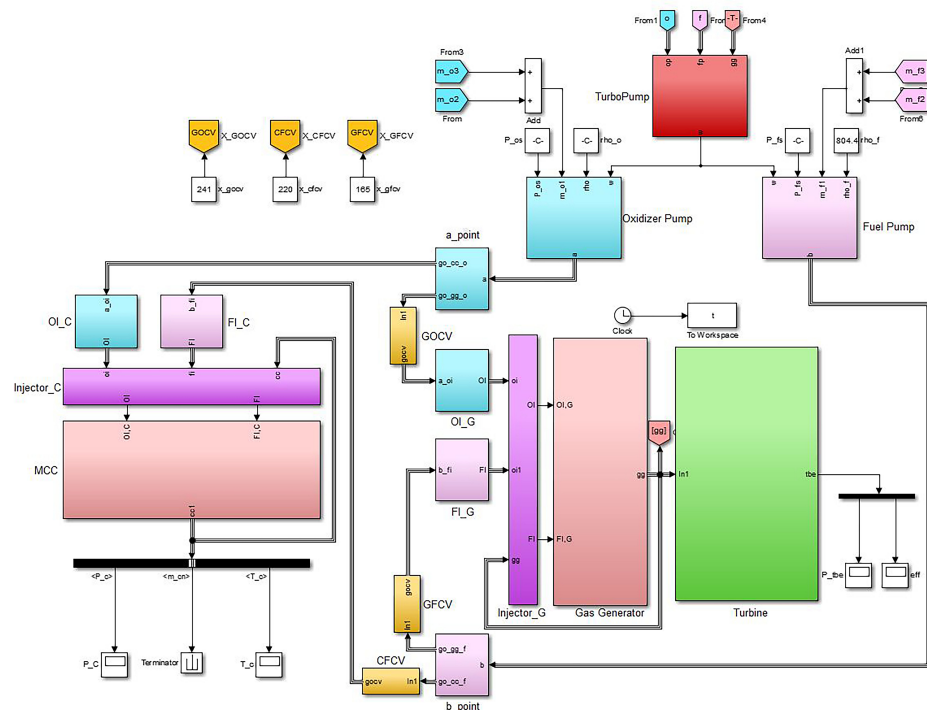


Figure A1. Simulation model of liquid-propellant rocket engine [32].

Appendix B. Verification of Dynamic Simulation

We verified the simulation results by comparing the startup and shutdown processes with the normalized simulation and experimental data of three state variables: P_c , P_g , and ω (see Table 1). In Figure A2, the startup process is delineated, progressing sequentially through the pyro-starter, CC ignition, and GG ignition stages. In the startup process, the pyro-starter first rotates the turbopump, and then ignition starts combustion in the CC. Finally, combustion begins in GG, and the turbopump accelerates until the full thrust is achieved.

In contrast, Figure A3 shows the shutdown process, delineating a sequential pattern. First, the GG oxidizer valve (GOV) and GG fuel valve (GFV) close to reduce the turbopump rotation speed and the propellant flow to the CC. Afterward, when the propellant flow to the CC is sufficiently reduced, the main oxidizer valve (MOV) closes, terminating engine operation.

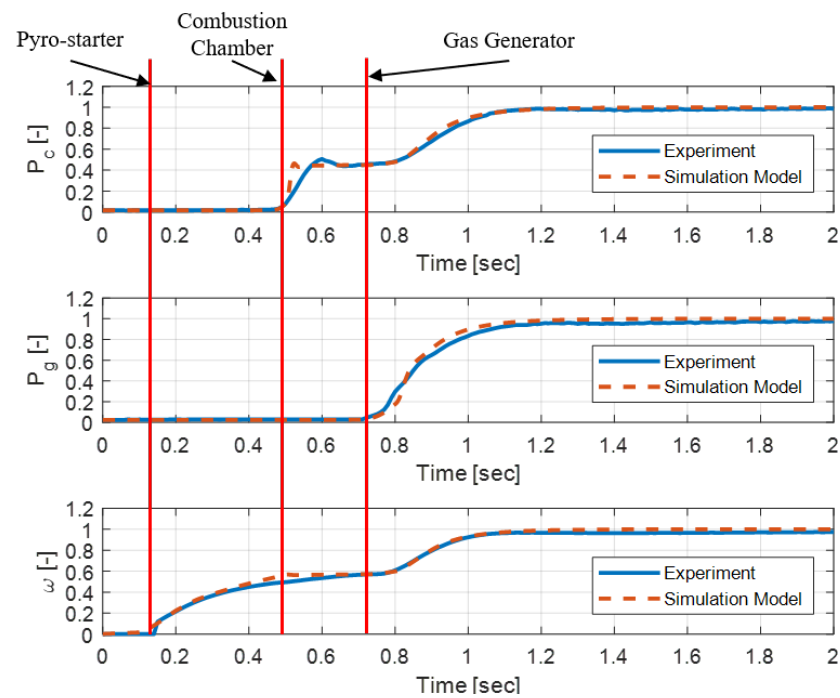


Figure A2. Comparison of the startup process between the simulation and experimental data. (**Top**) Combustion chamber pressure, (**Middle**) gas generator pressure, and (**Bottom**) turbopump rotational speed [16].

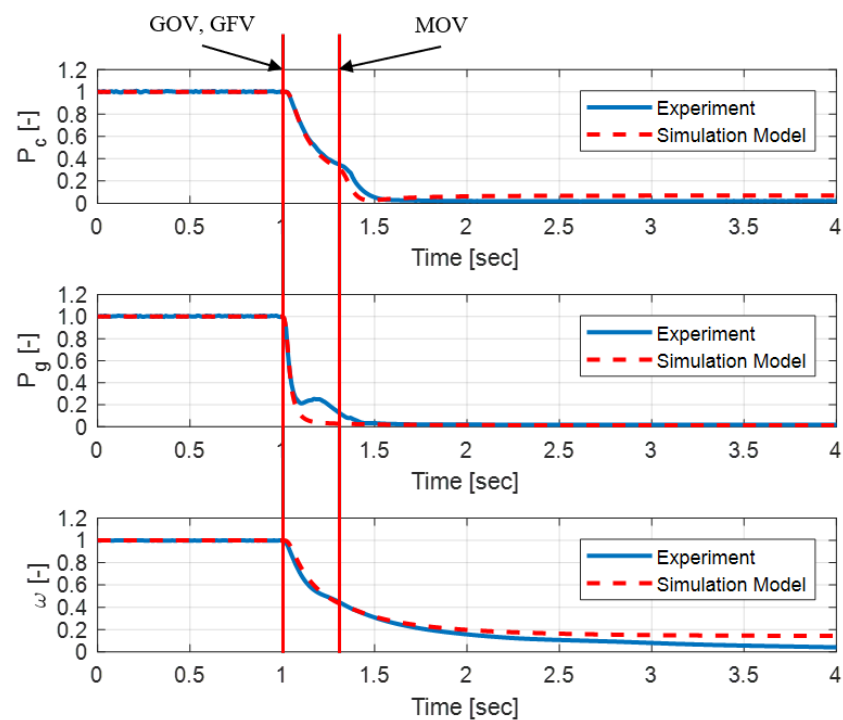


Figure A3. Comparison of the shutdown process between the simulation and experimental data. (**Top**) Combustion chamber pressure, (**Middle**) gas generator pressure, and (**Bottom**) turbopump rotational speed [16].

References

- Muelhaupt, T.J.; Sorge, M.E.; Morin, J.; Wilson, R.S. Space traffic management in the new space era. *J. Space Saf. Eng.* **2019**, *6*, 80–87. [\[CrossRef\]](#)
- Wekerle, T.; Pessoa, J.B.; Costa, L.E.V.L.d.; Trabasso, L.G. Status and trends of smallsats and their launch vehicles—An up-to-date review. *J. Aerosp. Technol. Manag.* **2017**, *9*, 269–286. [\[CrossRef\]](#)
- Lu, Y.; Qin, X.; Chen, H. Low Cost and Reusability of Launch Vehicle. *Aerosp. China* **2019**, *17*, 37–44.
- Kim, C.T.; Yang, I.; Lee, K.; Lee, Y. Technology Development Prospects and Direction of Reusable Launch Vehicles and Future Propulsion Systems. *J. Korean Soc. Aeronaut. Space Sci.* **2016**, *44*, 686–694.
- Choo, K.; Mun, H.; Nam, S.; Cha, J.; Ko, S. A survey on recovery technology for reusable space launch vehicle. *J. Korean Soc. Propuls. Eng.* **2018**, *22*, 138–151. [\[CrossRef\]](#)
- Sun, H.; Cheng, Y.; Jiang, B.; Lu, F.; Wang, N. Anomaly Detection Method for Rocket Engines Based on Convex Optimized Information Fusion. *Sensors* **2024**, *24*, 415. [\[CrossRef\]](#) [\[PubMed\]](#)
- Kawatsu, K.; Tsutsumi, S.; Hirabayashi, M.; Sato, D. Model-based fault diagnostics in an electromechanical actuator of reusable liquid rocket engine. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 1624.
- Wu, J. Liquid-propellant rocket engines health-monitoring—A survey. *Acta Astronaut.* **2005**, *56*, 347–356. [\[CrossRef\]](#)
- Cha, J.; Ha, C.; O, S.; Ko, S. A survey on health monitoring and management technology for liquid rocket engines. *J. Korean Soc. Propuls. Eng.* **2014**, *18*, 50–58. [\[CrossRef\]](#)
- Lee, K.; Cha, J.; Ko, S. A Survey on Fault Detection and Diagnosis Method for Open-Cycle Liquid Rocket Engines through China R&D Case. *J. Aerosp. Syst. Eng.* **2017**, *11*, 22–30.
- Pan, T.; Chen, J.; Ye, Z.; Li, A. A multi-head attention network with adaptive meta-transfer learning for RUL prediction of rocket engines. *Reliab. Eng. Syst. Saf.* **2022**, *225*, 108610. [\[CrossRef\]](#)
- Kawatsu, K. PHM by using multi-physics system-level modeling and simulation for EMAs of liquid rocket engine. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019; pp. 1–10.
- Lin, R.; Yang, J.; Huang, L.; Liu, Z.; Zhou, X.; Zhou, Z. Review of Launch Vehicle Engine PHM Technology and Analysis Methods Research. *Aerospace* **2023**, *10*, 517. [\[CrossRef\]](#)
- Cha, J.; Ha, C.; Koo, J.; Ko, S. Dynamic simulation and analysis of the space shuttle main engine with artificially injected faults. *Int. J. Aeronaut. Space Sci.* **2016**, *17*, 535–550. [\[CrossRef\]](#)
- Zhang, W.; Tian, G.; Xu, Z.; Yang, Z. *Failure Characteristics Analysis and Fault Diagnosis for Liquid Rocket Engines*; Springer: Berlin/Heidelberg, Germany, 2016.
- Cha, J. Transient State Modeling, Simulation, and Fault Detection/Diagnosis of an Open-Cycle Liquid Rocket Engine. Ph.D. Thesis, Department of Aerospace and Mechanical Engineering, Korea Aerospace University, Goyang, Republic of Korea, 2019.
- Park, S.Y.; Ahn, J. Deep neural network approach for fault detection and diagnosis during startup transient of liquid-propellant rocket engine. *Acta Astronaut.* **2020**, *177*, 714–730. [\[CrossRef\]](#)
- Yu, H.; Wang, T. A method for real-time fault detection of liquid rocket engine based on adaptive genetic algorithm optimizing back propagation neural network. *Sensors* **2021**, *21*, 5026. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, T.; Ding, L.; Yu, H. Research and development of fault diagnosis methods for liquid rocket engines. *Aerospace* **2022**, *9*, 481. [\[CrossRef\]](#)
- Zhang, X.; Hua, X.; Zhu, J.; Ma, M. Intelligent Fault Diagnosis of Liquid Rocket Engine via Interpretable LSTM with Multisensory Data. *Sensors* **2023**, *23*, 5636. [\[CrossRef\]](#) [\[PubMed\]](#)
- Cha, J.; Ko, S.; Park, S.Y.; Jeong, E. Fault detection and diagnosis algorithms for transient state of an open-cycle liquid rocket engine using nonlinear Kalman filter methods. *Acta Astronaut.* **2019**, *163*, 147–156. [\[CrossRef\]](#)
- Wang, C.; Zhang, Y.; Zhao, Z.; Chen, X.; Hu, J. Dynamic model-assisted transferable network for liquid rocket engine fault diagnosis using limited fault sample. *Reliab. Eng. Syst. Saf.* **2024**, *243*, 109837. [\[CrossRef\]](#)
- Li, F.; Chen, J.; Liu, Z.; Lv, H.; Wang, J.; Yuan, J.; Xiao, W. A soft-target difference scaling network via relational knowledge distillation for fault detection of liquid rocket engine under multi-source trouble-free samples. *Reliab. Eng. Syst. Saf.* **2022**, *228*, 108759. [\[CrossRef\]](#)
- Huang, P.; Yu, H.; Wang, T. A Study Using Optimized LSSVR for Real-Time Fault Detection of Liquid Rocket Engine. *Processes* **2022**, *10*, 1643. [\[CrossRef\]](#)
- Zhang, R.; Du, H.; Wu, Y.; Qin, X.; Zhang, P. Fault Detection and Diagnosis for Thrust Drop of Launch Vehicles against Disturbances. *J. Spacecr. Rocket.* **2023**, *60*, 924–941. [\[CrossRef\]](#)
- Fernández, L.A.; Wiedemann, C.; Braun, V. Analysis of space launch vehicle failures and post-mission disposal statistics. *Aerotec. Missili Spaz.* **2022**, *101*, 243–256. [\[CrossRef\]](#)
- Harland, D.M.; Lorenz, R. *Space Systems Failures: Disasters and Rescues of Satellites, Rocket and Space Probes*; Springer Science & Business Media: New York, NY, USA, 2007.
- Kalnin, V.; Sherstiannikov, V. Hydrodynamic modelling of the starting process in liquid-propellant engines. *Acta Astronaut.* **1981**, *8*, 231–242. [\[CrossRef\]](#)
- Konatowski, S.; Kaniewski, P.; Matuszewski, J. Comparison of estimation accuracy of EKF, UKF and PF filters. *Annu. Navig* **2016**, *23*, 69–87. [\[CrossRef\]](#)

30. Gadsden, S.; Dunne, D.; Habibi, S.; Kirubarajan, T. Comparison of extended and unscented Kalman, particle, and smooth variable structure filters on a bearing-only target tracking problem. In Proceedings of the Signal and Data Processing of Small Targets 2009, San Diego, CA, USA, 2–6 August 2009; Volume 7445, pp. 113–125.
31. Steven, M.K. *Fundamentals of Statistical Processing, Volume 2: Detection Theory*; Pearson: New York, NY, USA, 1998.
32. Lee, K.; Cha, J.; Ko, S.; Park, S.Y.; Jung, E. Mathematical modeling and simulation for steady state of a 75-ton liquid propellant rocket engine. *J. Aerosp. Syst. Eng.* **2017**, *11*, 6–12.
33. Cha, J. Numerical Simulation of Chemical Propulsion Systems: Survey and Fundamental Mathematical Modeling Approach. *Aerospace* **2023**, *10*, 839. [[CrossRef](#)]
34. Djuric, P.M.; Kotecha, J.H.; Zhang, J.; Huang, Y.; Ghirmai, T.; Bugallo, M.F.; Miguez, J. Particle filtering. *IEEE Signal Process. Mag.* **2003**, *20*, 19–38. [[CrossRef](#)]
35. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [[CrossRef](#)]
36. Kim, C.W.; Park, S.Y.; Cho, W.K. Methodology of liquid rocket engine diagnosis. *Aerosp. Eng. Technol.* **2012**, *11*, 182–194.
37. Basseville, M.; Nikiforov, I.V. *Detection of Abrupt Changes: Theory and Application*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1993.
38. Bernstein, K.S.; Kujala, R.; Fogt, V.; Romine, P. *Structural Design Requirements and Factors of Safety for Spaceflight Hardware: For Human Spaceflight*; National Aeronautics and Space Administration, Lyndon B. Johnson Space Center: Washington, DC, USA, 2011.
39. Klem, M.; Fry, R. *Guidelines for Combustion Stability Specifications and Verification Procedures for Liquid Propellant Rocket Engines*; Chemical Propulsion Information Agency: Columbia, MD, USA, 1997.
40. Vanini, Z.S.; Khorasani, K.; Meskin, N. Fault detection and isolation of a dual spool gas turbine engine using dynamic neural networks and multiple model approach. *Inf. Sci.* **2014**, *259*, 234–251. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



DSPACE

<https://dspace.org/>

Particle-filter-based fault diagnosis for the startup process of an open-cycle liquid-propellant rocket engine

Cha, Jihyoung; Ko, Sangho; Park, Soon-Young

2024-05-01

MDPI

<https://doi.org/10.3390/s24092798>

Cha J, Ko S, Park SY. (2024) Particle-filter-based fault diagnosis for the startup process of an open-cycle liquid-propellant rocket engine. *Sensors*, Volume 24, Issue 9, May 2024, Article number 2798