Fault Isolation for the Ink Deposition Process in High-End Industrial Printers

Casper van Peijpe, Farhad Ghanipoor, Youri de Loore, Pim Hacking, Nathan van de Wouw, and Peyman Mohajerin Esfahani

Abstract. This paper presents a mathematical framework for modeling the dynamic effects of three fault categories and six fault variants in the ink channels of high-end industrial printers. It also introduces a hybrid approach that combines model-based and data-based methods to detect and isolate these faults effectively. A key challenge in these systems is that the same piezo device is used for actuation (generating ink droplets) and for sensing and, as a consequence, sensing is only available when there is no actuation. The proposed Fault Detection (FD) filter, based on the healthy model, uses the piezo self-sensing signal to generate a residual, while taking the above challenge into account. The system is flagged as faulty if the residual energy exceeds a threshold. Fault Isolation (FI) is achieved through linear regression or a k-nearest neighbors approach to identify the most likely fault category and variant. The resulting hybrid Fault Detection and Isolation (FDI) method overcomes traditional limitations of model-based methods by isolating different types of faults affecting the same entries (i.e., equations) in the ink channel dynamics. Moreover, it is shown to outperform purely data-driven methods in fault isolation, especially when data is scarce. Experimental validation demonstrates superior FDI performance compared to state-of-the-art methods.

1. Introduction

Over the past decades, fault diagnosis has been a focal point of research because of its vital role in maintaining the safety and reliability of engineering systems [1, 2]. In this paper, the systems of interest are the printers of Canon Production Printing (CPP). CPP is a Canon subsidiary company located in Venlo, the Netherlands, that develops, manufactures and sells high-end industrial printers [3]. Accurate fault diagnosis for the ink channels of these printers is essential for the improving print quality and maintenance services. The ink channels are subject to different faults, such as empty ink channels, partial or complete blockage of the nozzle or drying of the local ink [4, 5]. All of these faults are examples of anomalies that would degrade image and printing quality. Preventing a low-quality print has several advantages. Firstly, improving the overall print quality results in satisfied customers and possibly attracts new businesses. Secondly, avoiding poor prints can reduce the need for reprints, saving ink, paper, and running costs associated with customer complaints and repairs. Additionally, reducing waste ink and paper is not only cost-effective, but it is also environmentally

Date: November 7, 2025.

The first and second authors contributed equally to this work. The authors are with the Delft Center for Systems and Control, Delft University of Technology ({C.D.vanPeijpe, P.MohajerinEsfahani}@tudelft.nl), the Department of Mechanical Engineering, Eindhoven University of Technology, ({F.Ghanipoor, N.v.d.Wouw}@tue.nl) and Canon Production Printing ({Youri.deLoore, Pim.Hacking}@cpp.canon).

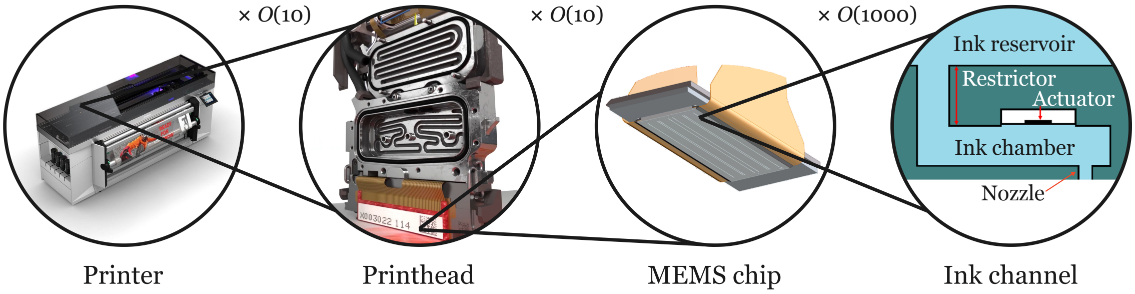

FIGURE 1. Decomposition of the printer where each component shown is smaller than and inside of the component to its left.

responsible. For these reasons, it is beneficial to detect a fault in advance and isolate faults in their early stages to compensate for a faulty ink channel before it reduces printing quality.

The printers considered in this paper consist of various components to ensure their proper functioning. In this context, the focus is on the ink channels, with up to hundreds or thousands located in the printheads, depending on the printer. In Figure 1, a decomposition is shown that indicates where each component is located and how many of each component are inside the component to its left. See [6] for more detailed information about the printers and [7] for details of the printheads.

The problem of detecting and isolating faults in the ink channel of printers is challenging. Both the time scale and the dimension of the system are microscopic, with a typical piezo self-sensing signal being tens of microseconds long [5], an actuation chamber of mere millimetres in length and drop ejection nozzles of just tens of micrometres in diameter [4]. The combination of the small time and length scales results in very small physical quantities, making it hard to observe the state of the system in real life. A so-called piezo self-sensing signal can be obtained by using the piezo actuator (used to induce ink deposition) as a sensor. In this way, it can be used to obtain information about the state of the ink channel. It is important to note that the piezo must be actuated first before the sensing signal can be obtained [8, 9, 10]. Furthermore, depending on the printer, a signal from each of more than 100,000 ink channels is collected in less than 100 ms [3]. Dealing with those large amounts of data emphasises the need for a computationally efficient method for fault diagnostics. A further challenge regarding data collection is that the piezo self-sensing signal cannot be captured during actuation, as the piezo can only serve as the actuator or the sensor at any point in time. Hence, input-output data is not available at the same time to support a solution for fault diagnostics. This also means that the state of the system cannot be assumed to be zero at the start of the sensing period since the system is being actuated before measurement acquisition starts. However, in the literature on fault diagnosis, the state of the system when measurement data is available is often assumed to be zero [11, 12, 13], which is not the case here.

The frequency content of the self-sensing signals generated by the piezo actuator can be used for Fault Detection and Isolation (FDI) through signal-based fault tree methods with tunable thresholds, which is currently being used by the manufacturer for the health monitoring of their printers. This state-of-practice approach can be seen as a data-based method [14, 15], which does not directly take into account any physics-based model of the system (ink channel dynamics). Besides, the thresholds used within this method are manual margins designed based on experience, making the method non-systematic in nature and hard to generalize to different printer types.

Motivated by the aforementioned challenges, a hybrid model- and data-based FDI approach is proposed to detect and isolate faults using piezo self-sensing signals. This approach consists of two stages: Fault Detection (FD), which is the model-based part of the proposed methodology, and Fault Isolation (FI), which is the data-based part. The proposed method for FD builds upon previous results [11, 12], where the task of FD involves generating a diagnostic signal sensitive to the occurrence of faults, known as the residual. This task is accomplished by designing a model-based filter fed by the piezo self-sensing signal. The filter formulation is modified to account for the system actuation and sensing data not being available simultaneously. For FI, inspired by [16], an isolation filter is designed using a regression operator, as well as a classification algorithm, both using the output of the FD filter (i.e., residual signal). Contrary to the previous work, (simulated) faulty data is assumed to be available. Additionally, the FI filter design is adjusted to ensure that it can appropriately deal with scenarios in which only one fault can be present at a time (as is the case for the industrial printing use case in this paper). The main contributions of this paper are the following:

(a) **Modeling printer (ink channel) dynamics and possible faults:** We model the dynamics of ink channels for a class of industrial printers considering the fault categories of (1) an empty ink channel, (2) a blocked nozzle, and (3) a dried nozzle. The modeling framework includes six different variants of these fault categories that are encountered in practice.

(b) **Methodologies for FDI:** In light of the practical limitation of actuation and sensor data being unavailable simultaneously, we develop a tailored FDI filter based on the model, in combination with available labeled experimental data. Moreover, the developed hybrid model-data fault diagnosis method addresses the limitation of conventional model-based approaches by isolating different types of faults that affect the same entries in the ink channel dynamics [17, 18, 19]. This capability is enabled by integrating data-driven classifiers, which effectively distinguish between different patterns in the residual generated by the model-based fault detector filter. Additionally, it is shown that the method outperforms purely data-driven approaches in fault isolation.

(c) **Experimental validation on an industrial printer** [3]**:** The results are validated experimentally, indicating that the proposed method provides superior performance for both fault detection and isolation compared to the current state-of-the-art method.

The content of the remainder of paper is as follows. The problem statement of the printing use case is described in Section 2. A solution approach to the problem of fault detection and isolation is outlined in Section 3. The results of this approach with simulated and experimental data are shown in Section 4. Lastly, the conclusions that can be drawn from the results are presented, together with recommendations for future work in Section 5.

2. Printer Dynamics, Faults, and the Diagnosis Problem

The ink channel dynamics of the printer are presented in Section 2.1, the possible operational faults are explained in Section 2.2 and the problem of fault detection and isolation is described in Section 2.3.

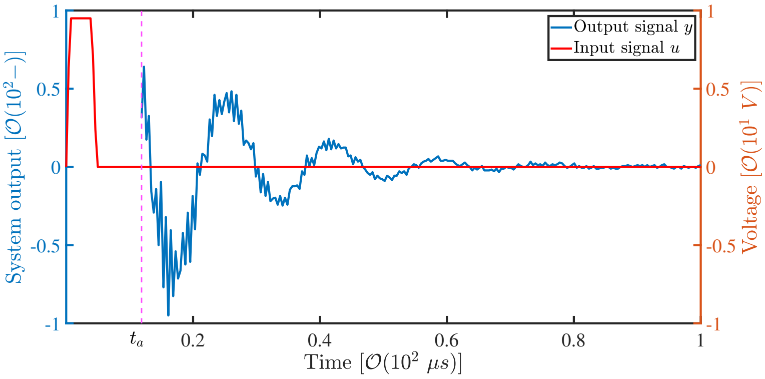

FIGURE 2. Typical input (actuation) signal and corresponding output (sensing) signal, with the output starting after the completion of the input at time t_a.

2.1. **Mathematical model for ink channel dynamics**

The smallest component in Figure 1, the ink channel, is the subject of this study. When printing, a piezo-electric actuator is (dis)charged, inducing a mechanical deformation which displaces the ink from the ink chamber to both the restrictor and nozzle; see the sub-figure on the right side of Figure 1. A strong enough actuation will lead to a droplet being jetted from the nozzle onto the paper [20]. The movement of the ink is captured in a four-dimensional state vector, denoted by x, which contains the volumes and flow rates displaced through the restrictor and the nozzle when actuating. The single variable voltage discharging the actuator is captured by the input u.

Following the fluid dynamics principles described in [5], complemented by the addition of resistance in [4], the ink channel dynamics in the healthy case can be described as follows:

$$\underbrace{\begin{bmatrix} \dot{V}_r \\ \dot{V}_n \\ \ddot{V}_r \\ \ddot{V}_n \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{1}{I_r\beta_t} & -\frac{1}{I_r\beta_t} & -\frac{R_r}{I_r} & 0 \\ -\frac{1}{I_n\beta_t} & -\frac{1}{I_n\beta_t} & 0 & -\frac{R_n}{I_n} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} V_r \\ V_n \\ \dot{V}_r \\ \dot{V}_n \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \frac{b}{I_r\beta_t} \\ \frac{b}{I_n\beta_t} \end{bmatrix}}_{B_u} u, \quad y = \underbrace{\begin{bmatrix} 0 & 0 & c & c \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} V_r \\ V_n \\ \dot{V}_r \\ \dot{V}_n \end{bmatrix}}_{x}, \tag{1}$$

where V_r and V_n are the volumes of ink displaced through the restrictor and the nozzle when actuating, respectively. The parameters I_r and I_n are the inertances in the restrictor and the nozzle, respectively, R_r and R_n are the resistances in the restrictor and the nozzle, respectively, and β_t is the total compliance of the system. The input u is the voltage applied to the actuator. The actuator constant b captures the relation between the voltage input and the influence on the channel pressure, which contributes proportionally to the time-derivative of the flow rate in the restrictor and nozzle via the factors $\frac{b}{I_r\beta_t}$ and $\frac{b}{I_n\beta_t}$, respectively. Acquisition constant c captures the relationship between the flow rates (of ink displaced through the restrictor and the nozzle) in the ink chamber and the system output y. Note that a few microseconds after each actuation, the piezoelectric actuator can be used as a sensor to measure y in (1). For a more detailed explanation of the model parameters, see [5]. A typical example of the process trajectories of both actuation and sensor acquisition is plotted in Figure 2. Since the actuator acts as a sensor after actuation, the system is treated as autonomous with the non-zero state $x_a(t_a)$ (called the the acquisition state) at starting time t_a of the acquisition. As a result, the term related to B_u can be neglected in (1) after $t = t_a$.

2.2. Operational faults

The operational faults studied in this article are caused by changes in some parameters of our mathematical model in (1). These faults are naturally of a multiplicative nature with the internal states of the system x. With this in mind, we can model each fault effect on the printer dynamics as follows:

$$\begin{cases} \dot{x}(t) &= (A + B_{f_i} f_i)x(t), \quad i \in \{1, 2, \ldots, n_f\} \\ y(t) &= Cx(t), \end{cases} \tag{ODE(i)}$$

where the matrices A and C are as defined in (1) and the signal $f_i \in \mathbb{R}_+$ is the i-th fault signal taking values "0" whenever the system is healthy (i.e., no faults) or nonzero values when the fault occurs. Herein, n_f denotes the number of fault variants, which are listed below and equal six. It is practically reasonable to assume that at most one i-th fault occurs at a time. Consequently, (ODE(i)) captures the dynamics representing this sole i-th fault scenario. For every fault, the term $B_{f_i} f_i$ is added to A, where B_{f_i} captures the changes in the parameters of A. The matrix B_{f_i} may be different for every fault scenario and contains parameter changes related to that fault. Therefore, matrix B_{f_i} is different for every fault i and generally can be described as

$$B_{f_i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \delta(\frac{1}{I_r \beta_t})_i & \delta(\frac{1}{I_r \beta_t})_i & \delta(\frac{R_r}{I_r})_i & 0 \\ \delta(\frac{1}{I_n \beta_t})_i & \delta(\frac{1}{I_n \beta_t})_i & 0 & \delta(\frac{R_n}{I_n})_i \end{bmatrix}.$$

Note that faults that cause a possible change in model structure, potentially leading to a nonlinear model, are not taken into account, and that the studied fault scenarios can be described by a change in B_{f_i}. Furthermore, note that in practice, since we do not know the actual entries in the matrix B_{f_i}, we model it as matrices with zero and nonzero elements to indicate which equations are affected. These are provided below, along with the fault scenarios:

1) Empty ink channel (EC):: Refers to a fault in which an ink channel is not filled with ink. Every ink channel starts empty and subsequently it is filled with ink. An ink channel can also be fully emptied when printing. An empty channel can be modelled as a decrease in inertance of both the restrictor and nozzle, I_r and I_n, causing the sensing signal y to oscillate at a higher frequency. The corresponding fault matrix is denoted as B_{f_1} and all non-zero elements in the general fault matrix B_{f_i} are non-zero in B_{f_1} as well.

2) Blocked nozzle (BN):: Refers to a fault in which a dirt particle is present at the outlet of the nozzle. Such a particle blocks a certain percentage of the nozzle's cross-sectional area. This causes the nozzle inertance, I_n, and resistance, R_n, to increase, resulting in a faster decay of the sensing signal y. Two different types of blocked nozzle can be distinguished: (i) Fully blocked nozzle (FBN): Refers to a fault in which a nozzle is fully blocked, implying no printing can be done; (ii) Partially blocked nozzle (PBN): Refers to a fault in which a part of the nozzle is blocked, implying ink exits the nozzle incorrectly. The corresponding fault matrices are denoted as B_{f_2} and B_{f_3}, for the FBN and PBN, respectively. Both matrices have the same non-zero elements, which are $B_{f_2}(4,1)$, $B_{f_2}(4,2)$, $B_{f_2}(4,4)$, and similarly $B_{f_3}(4,1)$, $B_{f_3}(4,2)$, $B_{f_3}(4,4)$.

3) Dried nozzle (DN):: Refers to a fault in which the ink at the nozzle's outlet has evaporated, losing some of its most volatile components, such as water. This causes the viscosity to increase and therefore the resistance R_n to locally increase, leading to a faster decay of the sensing signal y. Three different drying levels are distinguished: (i) Slightly dried nozzle (SDN), (ii) Intermediately dried nozzle (IDN), (iii) Deeply dried nozzle (DDN). The corresponding fault matrices are denoted as B_{f_4}, B_{f_5}, and B_{f_6}, for the SDN, IDN, and DDN, respectively. All three matrices have the same non-zero element, which is $B_{f_4}(4,4)$ and similarly $B_{f_5}(4,4)$ and $B_{f_6}(4,4)$.

Simultaneous faults are excluded from the scope of this work, given that their probability of occurrence is about 10^6 times lower than that of single faults, and thus their impact is considered negligible for practical purposes. The problem to be solved in this paper can now be stated.

2.3. Fault detection and isolation problem

We aim to design an FD filter that can distinguish between healthy and faulty systems using the output data $y(t)$ and an approximated acquisition state $\hat{x}_a$. The FD filter generates a residual signal r that indicates whether the measurements are healthy or are affected by the fault signal f_i. Formally, the residual can be represented as a function $r(x_a, f_i)$, and the FD design is ideally defined by the following mapping requirements:

$$x_a \mapsto r(x_a, 0) \equiv 0, \tag{2a}$$

$$f_i \mapsto r(x_a, f_i) \neq 0, \quad \forall i \in \{1, 2, \dots, n_f\}. \tag{2b}$$

For fault isolation, conventional model-based methods, such as [11], use filter banks, with each filter sensitive to one fault and insensitive to others. However, this approach fails when faults share the same system dynamics entries. Isolation relies on amplifying one fault's signal while suppressing another's in the same residual [17, 18], which is impossible when both faults affect the same entry. To address this challenge, we aim to design a fault isolation (FI) filter capable of isolating the occurring fault using the output of the fault detection (FD) filter, r, building on the results in [16]. The FI filter is trained offline using available labeled experimental data for each fault scenario. In the online implementation, the filter generates a vector $\varphi^\star$ (see the output of right block in Figure 3), where each row represents the percentage of occurrence for each fault scenario, and the fault with the highest percentage is considered to be the occurring fault.

3. Fault detection and isolation approach

In this section, the proposed solution is presented. Figure 3 illustrates an overview of the proposed FDI scheme, where different types of faults f_i entering the system are detected and isolated. The methods for fault detection and fault isolation are provided in Sections 3.1 and 3.2, respectively.

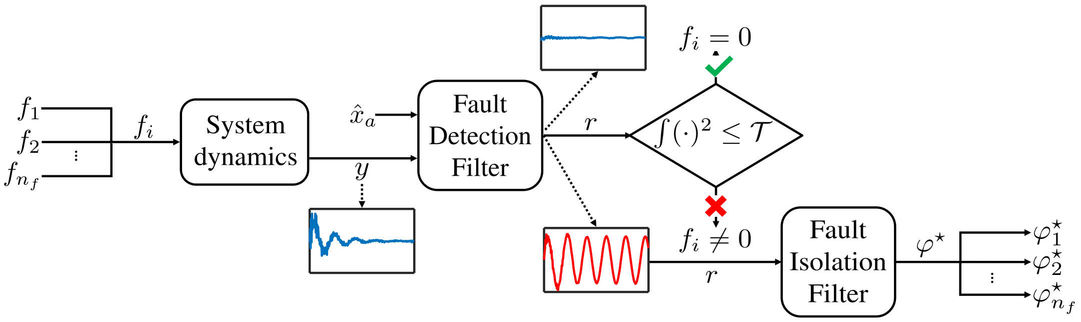

FIGURE 3. Overview of the proposed FDI scheme.

3.1. **Fault detection filter**

As a preliminary step, an accurate model is obtained using an off-the-shelf grey-box system identification method in MATLAB, namely extended least squares for parameter estimation [21]. This method uses healthy sensor data to estimate system parameters in (1) via solving the following optimization problem:

$$\{A, C, \hat{x}_a\} = \underset{A,C,\hat{x}_a}{\arg\min} \int_{t_a}^{t_f} |y(t) - Ce^{At}\hat{x}_a| dt, \tag{3}$$

where t_f denotes the final time of the signal y collected from the printer. The optimization problem is solved using the interior-point algorithm [22, Ch. 11] and yields an MSE of 2.48×10^{-18}. The resulting matrices A and C and the estimated acquisition state $\hat{x}_a$ are then used to develop a model-based fault detection filter.

To construct the FD filter, first, the state-space representation of the system in (ODE(i)) is converted from time domain to the Laplace domain as follows:

$$\begin{cases} sX(s) - x_a & = AX(s) + B_{f_i}\Psi_i(s), \\ Y(s) & = CX(s), \end{cases} \tag{4}$$

where, $s \in \mathbb{C}$ is the complex Laplace variable used in the Laplace transform. $X(s), Y(s)$ and $\Psi_i(s)$ are the Laplace transforms of $x(t), y(t)$ and $f_i x(t)$, respectively. Equation (4) can be written as

$$\underbrace{\begin{bmatrix} -sI + A \\ C \end{bmatrix}}_{H(s)=H_0+H_1 s} X(s) + \begin{bmatrix} x_a \\ 0 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ -I \end{bmatrix}}_{L(s)} Y(s) + \underbrace{\begin{bmatrix} B_{f_i} \\ 0 \end{bmatrix}}_{F_i(s)} \Psi_i(s) = 0, \tag{DAE(i)}$$

where $H(s) \in \mathbb{C}^{(n+m)\times n}$, $L(s) \in \mathbb{C}^{(n+m)\times m}$, and $F_i(s) \in \mathbb{C}^{(n+m)\times n}$, for all $i \in \{1, 2, \ldots, n_f\}$. This system representation is also known as the Differential-Algebraic Equation (DAE) (represented in the Laplace domain).

The FD filter can be stated as follows, generating the residual signal defined in the Laplace domain:

$$R(s) := \frac{N(s)L(s)}{\alpha(s)} Y(s) + \frac{N(s)}{\alpha(s)} \begin{bmatrix} \hat{x}_a \\ 0 \end{bmatrix}, \tag{5}$$

where $R \in \mathbb{C}$ is the residual in the Laplace domain, $\hat{x}_a$ denotes the approximated acquisition state from (3), $N(s)$ is designed to solve the problem stated in Section 2.3 (or more specifically to satisfy

(2)), and $\alpha(s)$ is chosen such that the transfer function of the residual generator $R(s)$ is (strictly) proper. The method used to design $\alpha(s)$ is described later. This residual can be transformed to the time domain to form the signal r. To design $N(s)$, the residual in (5) must be expressed in terms of both state-dependent and fault-dependent terms. It follows from (5) and (DAE(i)) that

$$R(s) = -\frac{N(s)H(s)}{\alpha(s)}X(s) - \frac{N(s)F_i(s)}{\alpha(s)}\Psi_i(s) + \frac{N(s)}{\alpha(s)}\begin{bmatrix} \hat{x}_a - x_a \\ 0 \end{bmatrix}. \tag{6}$$

To satisfy (2a), the first term in (6) must be zero (i.e., $N(s)H(s) = 0$). Similarly, to satisfy (2b), the second term must be nonzero (i.e., $N(s)F_i(s) \neq 0$). It is shown in [11] that these conditions on the transfer functions can be expressed in terms of matrices as follows:

$$\bar{N}\bar{H} = 0, \tag{7a}$$

$$\bar{N}\bar{F}_i \neq 0, \quad \forall i \in \{1, 2, \ldots, n_f\}, \tag{7b}$$

with $\bar{H} \in \mathbb{R}^{d_N(n+m)\times n(d_N+1)}$ as

$$\bar{H} := \left.\begin{bmatrix} H_0 & H_1 & & \\ & \ddots & \ddots & \\ & & H_0 & H_1 \end{bmatrix}\right\} d_N,$$

where, H_0 and H_1 as defined in (DAE(i)), d_N is the order of the filter, $\bar{N} \in \mathbb{R}^{1\times d_N(n+m)}$, and $\bar{F}_i \in \mathbb{R}^{d_N(n+m)\times n d_N}$ as given by

$$\bar{F}_i := \left.\begin{bmatrix} F_i & & \\ & \ddots & \\ & & F_i \end{bmatrix}\right\} d_N,$$

where F_i as defined in (DAE(i)). Generally, to create the FD filter, the linear programming problem in (7) is solved [11]. However, for this specific system (i.e., the ink channel system), the linear programming constraint (7a) has only one solution, meaning solving the linear programming problem is not needed. An approach to design $\bar{N}$ is to calculate the left null space of $\bar{H}$ as $\bar{N} = (\mathcal{N}(\bar{H}^\top))^\top = \begin{bmatrix} N_0 & N_1 & \cdots & N_{d_N} \end{bmatrix}$.

As for implementation purposes, the fault detection filter formulation should be derived in the Laplace domain, $\bar{N}$ can be converted to $N(s)$ using

$$N(s) := \sum_{i=0}^{d_N} N_i s^i.$$

Note that clearly from (6), it can been verified that if $\hat{x}_a - x_a$ is small, the residual signal will only generate the second term, which is zero in the healthy case because $\Psi_i(s)$ (i.e., the Laplace transform of $f_i x(t)$) is zero, and non-zero when a fault occurs, as the condition in (7b) imposes.

Due to errors caused approximation of the acquisition state (i.e., $\hat{x}_a - x_a$), modeling uncertainties, and measurement noise, a non-zero residual does not necessarily indicate that the system is faulty, which can lead to some healthy cases being misidentified as faulty. Therefore, to determine whether a residual signal is faulty, a threshold $\mathcal{T}$ on the energy of the residual signal, defined as the integral of the square of the signal, is established as

$$\mathcal{T} = \mu \max_{r_h \in \mathcal{S}_h} \left(\int_{t_a}^{t_f} r_h^2(t)\, dt \right), \tag{8}$$

where $\mathcal{S}_h$ is the healthy residual dataset which is produced by passing measurements y, including measurement noise, through the FD filter, r_h represents the healthy residuals taken from this dataset, and $\mu \in \mathbb{R}_+$ is a user-determined factor designed to make the detection method more robust against false alarms caused by approximation errors, modeling uncertainties, and measurement noise (see Section 4.2 for design examples). The selection of μ can be performed using the receiver operating characteristic (ROC) curve, which represents the trade-off between true positive and false positive rates across different threshold settings [23]. If this threshold is exceeded by the residual energy (i.e., $\int_{t_a}^{t_f} r^2(t)dt$), the system is classified as faulty (see Figure 3). Besides this threshold, other metrics, such as the maximum value of the residual, which may be computationally faster, can also be used to determine whether a signal is faulty.

The poles of the filter, represented by the denominator $\alpha(s)$ in (5), need to be selected carefully. In [11], the denominator of the FD filter is chosen as an arbitrary transfer function of sufficiently high order, with roots having negative real parts to ensure the transfer function of the filter is both proper and stable. However, it has been observed that the denominator can influence the performance of the filter for the system under consideration.

In this paper, since the energy of the residual is used for fault detection, the FD filter denominator is designed to amplify the energy of the residual, making faults easier to detect. This is achieved by placing two marginally stable poles of the FD filter at $s = \pm\omega_r i$, where ω_r is the desired residual frequency, which can be chosen arbitrarily. The frequency can be selected so that a desired number of oscillations occurs within a known time frame T. Since this pole pair does not have a negative real part, the residual does not converge to zero, and the energy continues to increase. This amplifies the presence of a fault, as a faulty residual results in a larger amplitude compared to a healthy residual. The remaining poles are chosen to be arbitrarily stable, as they do not influence the behavior of the residual significantly. Thus, the denominator can be expressed as follows:

$$\alpha(s) = (s^2 + \omega_r^2)\beta(s), \quad \omega_r = \frac{n_o}{T},$$

where $\beta(s)$ is a transfer function with roots with negative real parts of order $d_N - 2$, and n_o is the desired number of oscillations.

Once a fault has been detected, the next step is to isolate the detected fault. Towards this goal, an FI filter is designed in what follows.

3.2. **Fault isolation filter**

The FI filter is inspired by [16], where a Linear Regression (LR) model is also used for fault isolation. his approach assumes prior availability of labeled faulty data and a model that accounts for specific fault scenarios in order to train the FI filter, which limits its applicability to technical systems where such data and models are accessible, such as the special case studied here. To train the FI filter for each fault, labeled faulty data must be available. These LR models are trained offline in a discrete setting. Therefore, for offline training, continuous residuals obtained from the proposed FD filter should be discretized, and the N discretized residuals per fault class are averaged for each fault scenario as follows:

$$\rho_i[k] = \frac{1}{N}\sum_{j=1}^{N} r_i^j[k], \quad \forall i \in \{1, 2, \dots, n_f\}, k \in \{k_a, \dots, k_f\},$$

where $\rho_i[k] \in \mathbb{R}$ represents the average residual for each fault class i (see the offline training part in Figure 4, where only three faults are shown for simplicity), and k_a, k_f are the samples corresponding to t_a, t_f, respectively. Next, the matrix R, containing the averaged residuals for each fault class $\rho_i[k]$, can be constructed as follows:

$$R = \begin{bmatrix} \rho_1[k_a] & \rho_2[k_a] & \cdots & \rho_{n_f}[k_a] \\ \rho_1[k_a+1] & \rho_2[k_a+1] & \cdots & \rho_{n_f}[k_a+1] \\ \vdots & \vdots & \ddots & \vdots \\ \rho_1[k_f] & \rho_2[k_f] & \cdots & \rho_{n_f}[k_f] \end{bmatrix}.$$

During the online isolation process, LR is used to identify which expected residual $\rho_i[k]$, obtained from offline training, is closest to the measured residual signal r to be isolated (see the online isolation part in Figure 4). The LR model quantifies the contribution of each fault as a vector $\varphi^\star \in \mathbb{R}_+^{n_f}$, which is the solution to the following optimization problem:

$$\varphi^\star = \begin{cases} \arg\min_\varphi & ||R\varphi - r|| \\ \text{s.t.} & \sum_{i=1}^{n_f} \varphi_i \quad = 1, \\ & \varphi_i \quad\quad\quad \geq 0, \quad \forall i \in \{1, 2, \dots, n_f\}, \end{cases}$$

where

$$(\varphi)^\top = \begin{bmatrix} \varphi_1 & \dots & \varphi_{n_f} \end{bmatrix},$$

is used as a regressor. The fault index $i^\star$ corresponding to the highest value in $\varphi^\star$ is considered as the isolated fault, i.e., $i^\star =_i \varphi_i^\star, i \in \{1, 2, \dots, n_f\}$ (for the simplified example illustrated in Figure 4, $i^\star$ is 2). Instead of LR models, other classifiers, such as the k-nearest neighbors algorithm (KNN), can be used (see [24]). In this case, the residual signal is assigned to the class of its nearest neighbor. In the following sections, the FD and FI filters are evaluated using both simulation and experimental data concerning the ink channel dynamics in an industrial printer.

4. Simulation and Experimental Results

First, the structure of the data is analyzed in Section 4.1. Then the FD and FI filters are tested for both the simulation and experimental data in Sections 4.2 and 4.3, respectively.

4.1. Data analysis

To evaluate both filters, simulated data and experimental data are used. Simulated data is generated based on knowledge of faults and the nature of the available experimental data. 3,825 output signals, y, are generated, where 2,475 are healthy signals and every fault class contains 225 signals. Hence, all of the six fault classes mentioned in Section 2 are considered. One healthy example and three examples of faulty signals per fault category are illustrated in Figure 5.

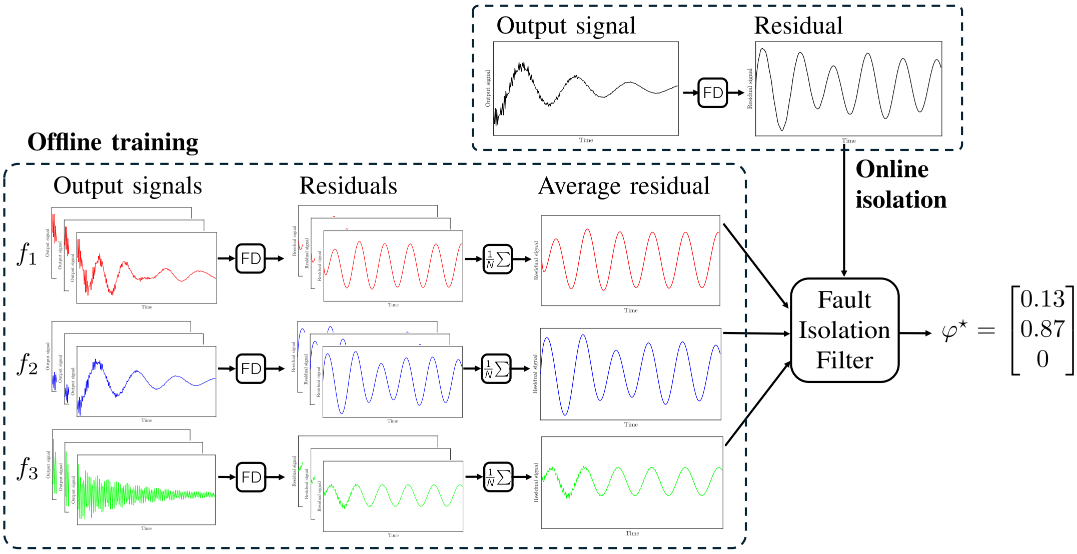

FIGURE 4. Schematic overview of the offline training (design) and online application of the FI filter.

Experimental data is collected by purposely drying the nozzle. The procedure to overcome a drying nozzle, called prefiring (mixing the ink in the nozzle and ink chamber to bring the ink in the nozzles back to a usable viscosity), is performed to collect data for fault scenarios ranging from deeply dried to healthy. Three datasets are generated: the first, with 48 signals, represents healthy nozzles, where sufficient prefires remove all dried ink. The second, with 39 signals, represents intermediately dried nozzles, where some prefires partially remove dried ink. The third, with 60 signals, represents deeply dried nozzles, with no prefires performed. Two examples of measured faulty signals per fault category are illustrated in Figure 6. Notably, separate datasets of the same length as the test data are used for the offline training of the FI filter for both the simulated and experimental data.

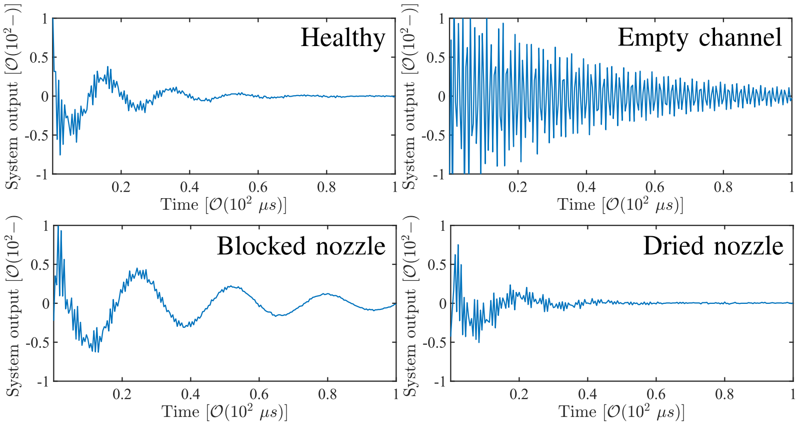

FIGURE 5. Examples of sensing signals y for simulated fault categories (here $t_a = 0$).

TABLE 1. FD Filter results for the simulated data.

		FD filter	
		Healthy	Faulty
Ground truth	Healthy	2474	1
	Faulty	92	1258

4.2. **Fault detection results**

The preliminary system identification results, obtained using the off-the-shelf grey-box method in MATLAB (explained in Section 3.1), are provided in the Appendix. The values for μ in (8) were determined by maximizing the Total Detection Rate (TDR) [25] while keeping the False Alarm Rate (FAR) [25] as close to zero as possible for different values of μ. The values are $\mu = 1$ and $\mu = 2$ for simulated and experimental data, respectively.

Some fault classes for the simulated dataset include signals that show characteristics similar to healthy signals. To prevent a large number of false positives, i.e., healthy signals being detected as faulty by the FD filter, the multiplication factor for the threshold in (8) is chosen as $\mu = 1$. For the experimental dataset, healthy and faulty signals can be distinguished more easily. Therefore, a multiplication factor of $\mu = 2$ is chosen to prevent false positives for the experimental data.

We present the FD results of the proposed method by comparing it with the State-of-Practice (SoP) approach. The SoP method relies on a signal-based fault tree approach with tunable thresholds, which are adjusted based on experience. For further details on the SoP method, refer to [14, 15].

The FD results of the simulated data are summarized in Table 1. The proposed FD filter achieves a Total Detection Rate (TDR) [25] of 97.6%, surpassing the 93.9% of the manufacturer's current method. Its False Alarm Rate (FAR) [25] is 0.04%, compared to 0% for the SoP approach. The proposed method has only one false negative, which is related to the slightly dried nozzle. Nevertheless, the substantial improvement in fault detectability halves the number of false detections, underscoring the superior reliability of the proposed approach compared to the SoP method. Furthermore, the computational complexity is lowered from the SoP's $\mathcal{O}(n \log n)$ to $\mathcal{O}(n)$, where n is the length of the signal [26].

For the experimental data, the proposed FD method achieved a TDR of 100%, with no false negatives or false positives, shown in Table 2. In comparison, the SoP approach used in the industry

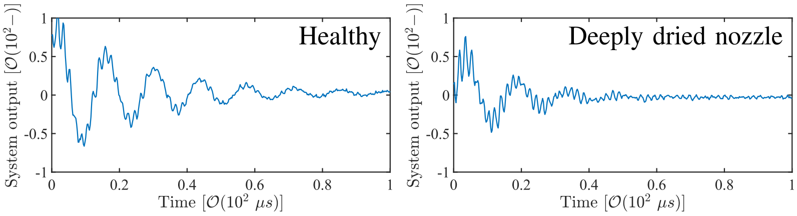

FIGURE 6. Examples of sensing signals y for experimental fault categories (here $t_a = 0$).

has TDR of 91.8%. Moreover, the FAR is 0% for the FD filter, and 4.2% for the SoP approach, highlighting the superior performance of the proposed FD filter. The improved performance of the proposed filter on experimental data is due to clearer distinctions between healthy and faulty signals. Simulated data includes slightly dried nozzles, harder to distinguish due to noise, whereas experimental faults are larger and easier to detect. Using the residual signal r instead of the output signal y leads to higher accuracy in fault isolation because r both attenuates high-frequency noise through the FD filter and amplifies small deviations in frequency, amplitude, and damping, making faults more apparent.

4.2.1. *Fault detection results using a machine learning method*

To provide additional context for the performance of the proposed FD approach, it is compared against a simple data-driven baseline. Although the primary objective of this work is to demonstrate the applicability of a theoretically established FD method on a real industrial system, benchmarking against existing alternatives offers valuable insight. The main comparison was performed with the SoP method currently deployed in the industrial process, as it represents the most relevant real-world baseline. In addition to this benchmark, we evaluate the performance of a Support Vector Machine (SVM) classifier as a representative machine learning method. The SVM model enables a comparison across several commonly used kernel functions. The results, summarized in Table 3, indicate that the linear kernel achieves slightly higher performance than the proposed FD filter, whereas the Gaussian and polynomial kernels yield inferior true detection rates.

Despite this marginal difference in performance, the proposed FD method offers a number of practical advantages. It provides a principled, model-based framework with high interpretability: the FD filter residual serves as a continuous indicator of fault severity, facilitating its direct use for FI. Moreover, the SVM approach requires significantly larger amounts of labeled data for training and incurs higher computational complexity, scaling as $\mathcal{O}(n^3)$ [27] compared to the $\mathcal{O}(n)$ complexity of the proposed filter-based scheme [26]. These factors make the proposed method particularly suitable for real-time industrial deployment where data availability and computation time are often constrained.

4.3. **Fault isolation results**

To evaluate the overall performance of the FI filter, the Harmonic Mean Average (HMA) is calculated. HMA is a standard metric used to assess the overall effectiveness of fault isolation schemes [25]. The performance of the proposed hybrid model-data fault isolation method is compared to that of a purely data-driven approach. The latter applies the FI filter directly to y, while the proposed

TABLE 2. FD Filter results for the experimental data.

		FD filter	
		Healthy	Faulty
Ground truth	Healthy	48	0
	Faulty	0	99

FD method	FAR	TDR
Proposed FD filter	0.04%	97.6%
SVM, linear	0%	100%
SVM, gaussian	0%	64.9%
SVM, polynomial	0%	70.3%

Table 3. SVM kernel function comparison

Table 4. FI results (in terms of HMA) with different training sizes for the simulated data.

Training size	Signal used	LR	KNN
2025	r	81.3%	80.0%
	y	73.1%	74.1%
1013	r	79.4%	80.1%
	y	70.5%	72.1%
203	r	80.1%	78.8%
	y	61.4%	66.4%

method applies it to r, where r is derived from the FD filter that incorporates model information. Additionally, the effect of training dataset size is examined by reducing it by 50% and 90%.

The results of the proposed FI methods using LR and KNN for the simulated data are presented in Table 4. These results demonstrate that for both methods and all training sizes, applying the FI filter to the residual signal r consistently outperforms applying it to the system output signal y, highlighting the value of model information. Among fault classes, intermediately dried nozzles have the lowest classification accuracy (see Table 5 for the confusion matrix corresponding to the method using r, LR and a training size of 2025). This is likely because their signals share similarities with those of both slightly dried nozzles and partially blocked nozzles.

Table 5. Confusion matrix of the proposed FI filter results for the simulated data (see Section 2 for the abbreviations).

		FI filter					
		EC	BN	PBN	SDN	IDN	DDN
Ground truth	EC	225	0	0	0	0	0
	BN	0	225	0	0	0	0
	PBN	0	0	190	35	0	0
	SDN	0	0	0	225	0	0
	IDN	0	0	103	113	9	0
	DDN	0	0	0	0	1	224

Table 6. FI results (in terms of HMA) with different training sizes for the experimental data.

Training size	Signal used	LR	KNN
147	r	99.0%	99.0%
	y	98.0%	98.0%
74	r	100%	100%
	y	98.0%	98.0%
15	r	99.0%	99.0%
	y	98.0%	98.0%

For comparison, the manufacturer's SoP method achieves an HMA of 45.3% on the largest training dataset. However, the SoP method has not been trained for all fault classes and would need expansion for a fair comparison.

The FI filter is also evaluated using experimental data, with results presented in Table 6. For this dataset, all methods (whether using r or y for FI and regardless of training set size) achieved high accuracy due to the smaller number of fault classes compared to the simulated data. The SoP method's HMA is 88.3%, but its primary limitation was misclassifying intermediately dried nozzles as healthy. This issue does not occur with the proposed FI filter, which significantly outperforms the SoP method.

It is worth emphasizing that the proposed method offers additional advantage. Unlike the SoP method, which performs FD and FI simultaneously, the proposed method isolates faults only after detecting them, achieved by leveraging model information in the design of the FD filter and using data for the FI filter, whereas the SoP method relies solely on data.

4.3.1. *Fault isolation results for an unknown fault*

This study assumes the prior availability of labeled faulty data and a model that accounts for specific fault scenarios in order to train the FI filter. However, there may be instances where a fault occurs that is not part of the training data. For this reason, simulations were performed using a fault scenario that was not included in the training dataset, namely the air bubble. This fault occurs when air becomes trapped inside the ink chamber. Because this fault causes major system modifications, it was not investigated further in this study. When the FI filter is subjected to an example of a faulty signal corresponding to an air bubble, its output is

$$\varphi^\star = \begin{bmatrix} 0.075 \\ 0.406 \\ 0 \\ 0 \\ 0 \\ 0.519 \end{bmatrix}.$$

This result indicates that the filter classifies the fault as a deeply dried nozzle with 51.9% certainty, which is relatively low. The FI filter was applied to 225 air bubble signals, for which the average certainty was 61.0%. This is lower than the average certainty of 82.4% for the known faults in the original dataset. In practice, the manufacturer could choose to act only on isolations with high confidence, meaning that unknown faults would initially go unaddressed. If the fault continues to be detected but not confidently isolated, the manufacturer may decide to further investigate the affected ink channel and incorporate the new fault into future training data.

5. Conclusion

This paper has presented a method for detecting and isolating faults in the ink channels of industrial printers using model-based detection filters and linear regression- and KNN-based fault classifiers. Using residuals (generated for fault detection) in fault isolation consistently outperformed directly using system output data, highlighting the advantages of the proposed method over purely data-driven approaches. Experimental results confirmed that the proposed fault isolation (FI) filter outperformed the state-of-the practice method, particularly by reducing the misclassification of faulty signals as healthy. For future work, it is recommended to experimentally evaluate the performance of the proposed methods under additional fault scenarios, such as an air bubble in the ink channel, and to investigate potential nonlinearities arising from model changes induced by these faults. Moreover, it is recommended to investigate the application of Fault Estimation, which estimates the degree of a certain fault, to the use-case.

Acknowledgement

This publication is part of the project Digital Twin project 4.3 with project number P18-03 of the research programme Perspectief which is (mainly) financed by the Dutch Research Council (NWO). The last author also acknowledges the support of the European Research Council (ERC) under the grant TRUST-949796. The authors were enabled by Canon Production Printing Netherlands B.V. to perform research that partly forms the basis for this paper. Canon Production Printing B.V. does not accept responsibility for the accuracy of the data, opinions and conclusions mentioned in this paper, which are entirely for the account of the authors.

Appendix A. Identified system matrices

The system identification results are given as follows:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -4.33\times 10^{11} & -4.33\times 10^{11} & -1.59\times 10^{5} & 0 \\ -6.17\times 10^{11} & -6.17\times 10^{11} & 0 & -1.75\times 10^{5} \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}, \quad \hat{x}_a = \begin{bmatrix} 0.21 & 0.21 & 0.16 & 0.22 \end{bmatrix}^\top.$$

References

[1] Jie Chen and Ron J Patton. *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media, 2012.

[2] Steven X Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.

[3] Industrial Printer Manufacturer. Canon production printing webpage. https://cpp.canon, November 2023.

[4] Detlef Lohse. Fundamental fluid dynamics challenges in inkjet printing. *Annual review of fluid mechanics*, 54(1):349–382, 2022.

[5] Hans Reinten, Yogesh Jethani, Arjan Fraters, Roger Jeurissen, Detlef Lohse, Michel Versluis, and Tim Segers. Resonance behavior of a compliant piezo-driven inkjet channel with an entrained microbubble. *The Journal of the Acoustical Society of America*, 151(4):2545–2557, 2022.

[6] Canon Colorado M-series. Canon production printing webpage. https://www.canon-europe.com/business/products/wide-format-printers/colorado-m-series/, November 2023.

[7] UVgel. Canon production printing webpage. https://cpp.canon/products/uvgel/, November 2023.

[8] Johannes Mathieu Marie Simons and Mark Alexander Groninger. Printing apparatus with measuring circuit for diagnosis of condition of each electromechanical transducer, January 2004. U.S. Patent No.: US6682162B2.

[9] Byung Hun Kim, Na Rae Park, and Hyun Ho Shin. Error detection apparatus of inkjet printer head and error detection method thereof, November 2014. U.S. Patent No.: US8894178B2.

[10] Marko Mihailovic, Tjerk E. C. Hummel, Johannes M. M. Simons, Hylke Veenstra, Cornelis W. M. Venner, and Amol A. Khalate. Method for detecting an operating status of an inkjet nozzle, December 2018. U.S. Patent No.: US10144215B2.

[11] Peyman Mohajerin Esfahani and John Lygeros. A tractable fault detection and isolation approach for nonlinear systems with probabilistic performance. *IEEE Transactions on Automatic Control*, 61(3):633–647, 2016.

[12] Erik Frisk and Mattias Nyberg. Residual generation for fault diagnosis of systems described by general linear differential-algebraic equations. *IFAC Proceedings Volumes*, 35(1):137–142, 2002. 15th IFAC World Congress.

[13] Jingwei Dong, Kaikai Pan, Sergio Pequito, and Peyman Mohajerin Esfahani. Robust multivariate detection and estimation with fault frequency content information. *Available at arXiv:2310.04922*, 2023.

[14] Hassan Ghasemzadeh and Roozbeh Jafari. Physical movement monitoring using body sensor networks: A phonological approach to construct spatial decision trees. *IEEE Transactions on Industrial Informatics*, 7(1):66–77, 2011.

[15] P.J.B Hacking. Classification of jetting behavior based on self-sensing piezo actuators. Master's thesis, TU Eindhoven, 2019.

[16] Chris Van der Ploeg, Mohsen Alirezaei, Nathan van de Wouw, and Peyman Mohajerin Esfahani. Multiple faults estimation in dynamical systems: Tractable design and performance bounds. *IEEE Transactions on Automatic Control*, 67(9):4916–4923, 2022.

[17] Yao Cheng, Rixin Wang, and Minqiang Xu. A combined model-based and intelligent method for small fault detection and isolation of actuators. *IEEE Transactions on Industrial Electronics*, 63(4):2403–2413, 2015.

[18] Jingwei Dong, Kaikai Pan, Sergio Pequito, and Peyman Mohajerin Esfahani. Robust multivariate detection and estimation with fault frequency content information. *Available at arXiv:2310.04922*, 2023.

[19] Farhad Ghanipoor, Carlos Murguia, Peyman Mohajerin Esfahani, and Nathan van de Wouw. Robust fault estimators for nonlinear systems: An ultra-local model design. *Automatica*, 171:111920, 2025.

[20] Kye-Si Kwon and Wousik Kim. A waveform design method for high-speed inkjet printing based on self-sensing measurement. *Sensors and Actuators A: Physical*, 140(1):75–83, 2007.

[21] A grey-box system identification algorithm provided by MATLAB. https://nl.mathworks.com/help/ident/ref/greyest.html, 2024.

[22] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[23] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ROC Analysis in Pattern Recognition.

[24] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[25] T.A.J. van Esch. Hybrid model-based and data-based fault detection, isolation, and estimation for wafer handler robots. Master's thesis, TU Eindhoven, November 2022.

[26] C. Albert. A mechanistic dynamic emulator. *Nonlinear Analysis: Real World Applications*, 13(6):2747–2754, December 2012.

[27] Retantyo Wardoyo Abdiansah Abdiansah. Time complexity analysis of support vector machines (svm) in libsvm. *International Journal of Computer Applications*, 128(3):28–34, October 2015.