Fault-Tolerant Soft Sensors for Dynamic Systems

Hongtian Chen®, Member, IEEE, and Biao Huang®, Fellow, IEEE

Abstract—Unpredicted faults occurring in automation systems deteriorate the performance of soft sensors and may even lead to incorrect results. To address the problem, this study develops three novel data-driven approaches for development of soft sensors. The three proposed soft sensors have fault-tolerant abilities. They are, respectively, called measurement space-aided scheme (MSaS), subspace-aided scheme (SSaS), and improved MSaS (IMSaS). As means to obtain more accurate results of soft sensors in the online phase: 1) MSaS constructs an optimal estimator of faults in the measurement space; 2) SSaS removes the influences caused by unknown sensor faults with the aid of a constructed subspace; and 3) IMSaS is an improved version of MSaS, eliminating the influences of the past prediction error that may accumulate and affect the current prediction result. They are the output-driven fault-tolerant soft sensors because their implementations rely on system measurements only. Furthermore, performance analysis is also conducted to investigate the estimation errors. Both the sufficient and necessary conditions for these designs are provided, and illustrations of the effectiveness and feasibility of the three proposed fault-tolerant soft sensors based on two case studies are given.

Index Terms—Data-driven methods, dynamic systems, fault-tolerant soft sensors.

I. INTRODUCTION

A MODERN automation system consists of multiple control loops where various smart sensors are embedded. In practice, a part of variables (such as the operation quality and control performance indicators) are not online measurable [1]. Therefore, soft sensors can be of use, especially for the process industries [1], [2], [3], [4], [5]. Roughly speaking, the soft sensor is a regression model that can timely predict some unmeasurable variables [5]. When the prediction results of unmeasurable variables are obtained, further fault detection or classification can be achieved. Its wide ranges of applications include performance prediction, state estimation, real-time control, performance optimization, fault estimation, etc. [6], [7], [8], [9], [10].

Thanks to a massive amount of available historical data, data-driven soft sensors have been extensively studied over the past two decades [2], [7], [9]. They are alternatives to hardware sensors and play an indispensable role in large-scale and complex systems. The significant advantage lies in their direct data-driven designs and implementations without resorting to explicit system models [3].

Regarded as the supervised learning methods, partial least squares [11], support vector machine [12], neural networks [9], etc. [13], [14], [15] have been extensively exploited in constructing soft sensors. By establishing the mapping from measured process variables to prediction results, soft sensors can be formulated as a prediction or estimation task that boils down to the regression problem [8]. In addition, latent variables, treated as knowledge or feature representations of data [16], are popular in practice because of the high dimension of data [17]. Principal component regression, partial least squares, and autoencoder are representative methods, and their latent variables are compacted representations of original signals. In [18], a probabilistic representation was used to construct soft sensors where the latent variables can help improving the prediction performance. Transfer learning was used in [9] to design soft sensors for handling unlabeled data in the target domain, taking the unchanged knowledge from the source-domain data. With the aid of a memory network having an attention mechanism, Yuan et al. [15] proposed a soft-sensor model that takes the spatiotemporal quality-relevant interactions into account. In addition, a semisupervised soft sensor was used in [19] for predicting online operation performance, where just-in-time learning helps select the performance-related variables. As summarized in [20], some Student-t distribution-based approaches were also developed to address heavy-tail outliers. However, these methods are only suitable for static systems.

Plant data are frequently characterized by considerable dynamic behaviors in industrial settings. Typically, the sequential measurements and operational performance indices are autocorrelated. Therefore, dynamic soft sensors have also been investigated and gained attention [21], [22], [23]. By modeling the dynamic relationship between plant variables and performance indices, Ding et al. [22] designed a soft sensor for dynamic systems for performance prediction along with fault diagnosis. Based on multivariate measurements, a novel soft sensor was proposed in [24] using maximum likelihoods, showing the robustness to unpredictable outliers. By incorporating the infrequently sampled performance index, Shardt et al. [23] developed a performance predictor for dynamic systems with consideration of time delays. Online designs that consider soft-sensor biases [6], reduced-order performance predictors [25], adaptive soft sensors [26], etc. [21] have also been investigated.

In practical application of soft sensors, one of the challenges is the fault caused by aging component and unexpected
malfunctions. These unpredicted faults lead to considerable disruptions to the soft-sensor performance. There have been few studies involving soft-sensor designs with consideration of the faults, especially for dynamic systems. It is desirable to develop soft sensors to restore the operational systems affected by a fault or disturbance through a redundant design. In the study, a fault-tolerant soft sensor is defined by the soft sensors of tolerant abilities to unexpected faults. These considerations motivate fault-tolerant soft sensors in this article with threefold contributions.

1) Three novel data-driven fault-tolerant soft sensors are proposed for dynamic systems. They possess abilities to tolerate the unknown sensor faults.

2) Through rigorous theoretical analysis, both sufficient and necessary conditions for the existence of the three proposed algorithms are developed. They lay the foundations for designs and implementations of fault-tolerant soft sensors for dynamic systems.

3) Comparative studies, discussions of pros and cons, and feasibility studies of the three proposed schemes are made from the algorithm designs to practical implementations.

The rest of this study starts with modeling soft sensors for both static and dynamic systems in Section II, followed by presenting the objectives of this work. Then the three proposed fault-tolerant soft sensors are, respectively, developed in Sections III–V, wherein theoretical analysis and implementation procedures are presented. The sufficient and necessary conditions are provided in Section VI, based on which comparisons among these fault-tolerant soft sensors are made. Section VII illustrates the effectiveness of the three proposed fault-tolerant soft sensors via a numerical simulation and an application to a practical debutanizer column. Section VIII concludes this study.

II. SYSTEM MODELS AND PROBLEM FORMULATION

A. System Descriptions

Consider a system $S$ given in Fig. 1. Depending on the operating conditions, $S$ may be described by a static or dynamic model. To be specific, given an operating condition, $S$ has the following form:

$$ S: \quad z(k) = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} n(k) \\ \end{bmatrix}, \quad u \in R^{k_u}, \quad y \in R^{k_y} \quad (1) $$

in which $u$ is the system input, $y$ is the system output, $n \in R^{k_u+k_y}$ represents the unknown noises, $k$ is the time instant, and there is a (static linear) relationship between $u(k)$ and $y(k)$. It is the well-known Box–Jenkins model [27]. An alternative description of $S$ given in (1) is based on the “latent variables” [28].

When there are dynamics in $S$, (1) becomes

$$ S: \quad x(k+1) = Ax(k) + Bu(k) + w(k) $$

$$ y(k) = Cx(k) + Du(k) + v(k) \quad (2) $$

where $w \in R^{k_w}$ and $v \in R^{k_v}$ are stochastic noises (different from disturbances); $A \in R^{k_x \times k_x}$, $B \in R^{k_x \times k_u}$, $C \in R^{k_y \times k_x}$, $D \in R^{k_y \times k_v}$ are the deterministic system matrices; and $x \in R^{k_x}$ is the system state.

Fig. 1. Automation systems with actuator and sensor faults.

B. Soft Sensors for Static and Dynamic Systems

For the static model given in (1), an intuitive solution for soft sensors has been proposed in [29]

$$ z(k) = P_{z}z(k) + P_{z}z(k), \quad I_{z} = P_{z} + P_{z} \quad (3a) $$

$$ \hat{\xi}(k) = \Theta P_{z}z(k), \quad \xi(k) = \hat{\xi}(k) + e(k) \in R^{k} \quad (3b) $$

where $\xi$ is the unmeasured variable in the online phase (or may be measured but with significant delays), the subscript “$\hat{\xi}$” signifies the variable $\xi$-related terms, $\hat{\xi}(k)$ is the prediction of $\xi$ based on the latent variable $P_{z}z(k)$ and the mapping $\Theta$, and the associated prediction error is $e(k)$. In (3a), $P_{z}$ is orthogonal to $P_{z}$. It is worth mentioning that in most of the publications, $z$ in (3a) is usually replaced by $y$. In fact, two kinds of descriptions are equivalent because it is not necessary to distinguish $u$ from $y$ for a static system.

Remark 1: The static model (1), together with the corresponding soft sensor given in (3a) and (3b), plays an essential role in designing soft sensors for dynamic systems. It benefits from the stacked system variables that can transform time-series features into static ones.

For dynamic systems, we introduce the state-driven and output-driven solutions, respectively. With the aid of a gain matrix $K$, the following full-dimensional observer can serve as a soft sensor to predict the unknown variable $\xi$:

$$ \hat{y}(k) = C\hat{x}(k) + Du(k) \quad (4a) $$

$$ \hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K(y(k) - \hat{y}(k)) $$

$$ = (A - KC)\hat{x}(k) + [B - KD]\begin{bmatrix} u(k) \\ y(k) \end{bmatrix} \quad (4b) $$

$$ \hat{\xi}(k) = C_{\hat{\xi}}\hat{x}(k) + D_{\hat{\xi}}\begin{bmatrix} u(k) \\ y(k) \end{bmatrix} \quad (4c) $$

In (4a)–(4c), both the innovation and prediction are obtained based on $\hat{x}(k)$. Therefore, it is called the state-driven (or dynamic feature-based [30]) soft sensor. Alternatively, $\hat{\xi}(k)$ can be predicted in real-time based on an output-driven soft sensor (see [22])

$$ x_{f}(k+1) = Hx_{f}(k) + Jy(k) $$

$$ \hat{\xi}(k) = Lx_{f}(k) + My(k) $$

which is the output-driven solution.

C. Problem Formulation and Objectives

This work intends to study fault-tolerant soft sensors for $S$ in (2) affected by the unknown faults. As shown in Fig. 1, the
sensor and actuator faults will affect system measurements and performance, respectively. Without loss of generality, \( S \) with the fault \( f \) is modeled by

\[
S^f: \quad \begin{align*}
x(k + 1) &= Ax(k) + Bu(k) + w(k) + E_a f(k) \\
y^f(k) &= Cx(k) + Du(k) + v(k) + E_s f(k)
\end{align*}
\]

which results in the following degraded prediction:

\[
x_k(k + 1) = Hx_k(k) + Jy^f_k(k) \quad (7a)
\]

\[
\xi_k(k) = Lx_k(k) + M_y^f(k) \quad (7b)
\]

\[
\xi^f(k) = \hat{\xi}^f(k) + e^f(k). \quad (7c)
\]

In (6), the two \( f \) can be different and replaced by the actuator fault \( f_a \) and sensor fault \( f_s \), which does not affect the overall procedure of the derivation presented in this work. For simplicity of the presentation, however, we will assume they are the same in the following derivations. Note that in (7a), \( y^f_k(k) \) is the term related to the actuator fault \( E_a f(k) \). It affects the variable \( \xi \) via (7b). Let \( y^f_i(k) \) be the component of \( y^f(k) \) related to sensor faults and noises. Then \( y^f(k) \) given in (6) becomes

\[
y^f(k) = y^f_k(k) + y^f_i(k), \quad y^f_i(k) = E_s f(k) + v(k) \quad (8a)
\]

\[
y^f_k(k) = Du(k) + CAx(k - 1) + CBu(k - 1) \times Cw(k - 1) + CE_a f(k - 1). \quad (8b)
\]

In practical applications, the fault \( f \) cannot be neglected because it directly affects system measurements and the variable being predicted. However, traditional soft-sensor approaches will be problematic when predicting \( \xi^f_k(k) \) if faults exist. To address this problem, the main objectives of this study, respectively, corresponding to the three proposed fault-tolerant soft sensors, are formulated as follows.

1) **Measurement-Space-Aided Scheme (MSaS):** In the measurement space, to construct an optimal estimator of \( f \) such that

\[
y^*(k) = y^f(k) - \hat{\xi}^f f(k)
\]

to design a data-driven soft sensor based on (5a) and (5b). Equation (9) implies the fault effect on the measurement is removed by the estimated fault.

2) **Subspace-Aided Scheme (SSaS):** To find a subspace of \( y \) (denoted as \( y_s \)) that is independent of sensor faults, i.e.,

\[
y_s(k) = P_s y^f(k) = P_s y^f_k(k)
\]

and to develop a soft sensor that can reveal the relationship between \( y_s \) and \( \xi^f \) such that

\[
x_k(k + 1) = Hx_k(k) + Jy_s(k) \quad (11a)
\]

\[
\xi^f(k) = Lx_k(k) + M_y^f(k). \quad (11b)
\]

3) **Improved MSaS (IMSaS):** To develop an improved version of MSaS such that

\[
\xi^f(k) \perp e^f(k - i), \quad i = 1, \ldots, N
\]

which can eliminate the cumulative error that MSaS may encounter when predicting \( \xi^f(k) \).

Before proceeding, we make three assumptions about MSaS, SSaS, and IMSaS as follows.

1) The sensor fault \( f \) is available in offline training.

2) System outputs \( y \) and \( y^f \) and the variable \( \xi \) can be measured in the offline training.

3) MSaS, SSaS, and IMSaS are data-driven soft sensors because their designs and implementations are only based on the offline datasets \{\( u, y, \xi \)\} and \{\( u, f, y^f, \xi \)\}. There is no more information about \( S, S^f, (5a), (5b) \), and (7a)–(7c) available.

In practical situations, if sensor fault data are not available in the offline phases, one may approximately simulate it by adding the sensor fault to the measurement. Then, \( f \) becomes available in the offline phase for constructing soft sensors.

### III. Measurement-Space-Aided Scheme

Based on data-based modeling, this section will develop a novel fault-tolerant soft sensor called MSaS.

To learn from data, the stacked vectors and matrices are given in (2). Following (13a) and (13b), the data-based models of both \( S \) and soft sensors

\[
\Xi = \left[ \begin{array}{c} \xi(k) \\ \vdots \\ \xi(k + N - 1) \end{array} \right] \in R^{k \times N} \quad (13a)
\]

\[
\xi_s(k) = \left[ \begin{array}{c} \xi(k) \\ \vdots \\ \xi(k + s) \end{array} \right] \in R^{k(s+1) \times N} \quad (13b)
\]

In (13a) and (13b), \( s \) represents the stack length and \( \xi(k) \) can be replaced by any variable used in (1)–(12).

#### A. Data-Based Modeling

For the soft sensor for dynamic systems given in Fig. 1, its implementation consists of two steps when using the output-based solution, i.e.,

\[
y(z) = G_y u(z) \quad (14)
\]

where \( G_{ya} \) is the transfer function from \( u(z) \) to \( a(z) \). In (14), \( G_y \) and \( G_{ya} \) describe the system dynamics and soft-sensor model, respectively.

With the help of (13a) and (13b), the data-based models of \( G_y \) and \( G_{ya} \) are detailed as follows.

1) **Data Model of \( S \):** Consider \( S \) given in (2). Following the parity-space notation, one can derive the equation:

\[
Y_{k,s} - H_{a,s} U_{k,s} = \Gamma_s X_k + H_{w,s} W_{k,s} + V_{k,s} \quad (15)
\]

where the right-hand side includes all the unknown terms, and \( H_{a,s} \in R^{k \times (s+1)}, H_{w,s} \in R^{k \times (s+1)}, \) and \( \Gamma_s \in R^{k(s+1) \times k} \) have the following forms:

\[
H_{a,s} = \left[ \begin{array}{ccc} D & & \\ \vdots & \ddots & \vdots \\ C A^{-1} B & \cdots & D \end{array} \right] \quad (16a)
\]

\[
H_{w,s} = \left[ \begin{array}{c} 0 \\ \vdots \\ C A^{-1} \end{array} \right], \quad \Gamma_s = \left[ \begin{array}{ccc} C & \vdots \\ \vdots & \vdots \end{array} \right] \quad (16b)
\]
In (4b), all the eigenvalues of \( (A - KC) \) are located inside the unit circle, which yields the following relationship [31]:
\[
\Gamma_s X_k \approx \Gamma_s L_{xp} Z_{xp} = \Gamma_s [L_u \ L_y] Z_{xp}
\]
(17)
for a large \( s \), where
\[
Z_{xp} = \begin{bmatrix} U_{k-s-1,s}^T & Y_{k-s-1,s}^T \end{bmatrix}^T.
\]
(18)
More details about (17), including derivations and descriptions of \( L_{xp} \), can be found in [32].

Based on (17), (15) can be rewritten as
\[
Y_{k,s} - H_{u,s} U_{k,s} - \Gamma_{s} L_{xp} Z_{xp} = H_{w,s} W_{k,s} + V_{k,s}
\]
(19)
which is the data model of \( G_{yu} \) that equivalently describes the dynamic behavior of \( S \) in the stacked form.

2) Data Model of Soft Sensors: Consider \( G_{f \gamma} \) whose state-space representation is given in (5a) and (5b). Similar to (15), the following equation holds:
\[
\Xi_{k,s} - H_{f \gamma,y,s} Y_{k,s} = \Gamma_{\xi,s} \hat{Y}_{k,s}
\]
(20)
where \( H_{f \gamma,y,s} \in R_{k_{y,s}x}^{k_{y,s}x} \) and \( \Gamma_{\xi,s} \in R_{k_{\xi,s}x}^{k_{\xi,s}x} \) are given as follows:
\[
H_{f \gamma,y,s} = \begin{bmatrix} M & \vdots & \vdots & \vdots \end{bmatrix}, \quad \Gamma_{\xi,s} = \begin{bmatrix} \lambda & \vdots & \vdots & \vdots \end{bmatrix}. \quad (21)
\]
In (20), no priori information about the dimension of \( X_{k,s} \) (i.e., \( k_{x,s} \)) is available. Using the past data \( \Psi_p \) and current data \( \Psi_f \), it follows from [33]:
\[
\text{rank} \left( \frac{1}{N} \Psi_f \Psi_p^T \right) = (s + 1)k_y + k_{\xi,s}
\]
(22)
where \( \Psi_p \) and \( \Psi_f \) are chosen as follows:
\[
\Psi_p = \begin{bmatrix} Y_{k,s-1,s}^T & \Xi_{k,s-1,s}^T \end{bmatrix}^T \\
\Psi_f = \begin{bmatrix} Y_{k,s}^T & \Xi_{k,s}^T \end{bmatrix}^T.
\]
(23a) (23b)
Combining (17), (23a), and (23b), (20) becomes
\[
\Xi_{k,s} - H_{f \gamma,y,s} Y_{k,s} - \Gamma_{\xi,s} L_{xp} \Psi_p = 0
\]
(24)
which is the data model of \( G_{f \gamma} \) where \( L_{xp} \) has a similar structure as \( L_{xp} \).

It should be pointed out that in (20)–(24), we do not consider the difference between \( \Xi_{k,s} \) and \( \hat{Y}_{k,s} \). In the offline training phase, \( \Xi_{k,s} \) is obtained and can be used to construct a soft sensor. In the online phase, the obtained soft-sensor model can be used to make a prediction, i.e., obtaining \( \hat{Y}_{k,s} \).

B. Parameter Identification

As described above, this study considers two separate models to describe the complete dynamic relation from \( u(k) \) to \( \xi(k) \). Certainly, an integrated single model for example, \( \hat{\xi}(z) = G_{f \gamma} G_{yu} u(z) \)
(25)
and its data-based model are also suitable for describing the dynamic behaviors. The main reasons and merits of using two separate models are summarized in the following Remark 2.

**Remark 2:** As shown in Fig. 1, the fault \( f \) affects both \( y \) and \( \xi \). Two separate models make the investigation of the fault-related components in \( y^f \) and \( \xi^f \) easier. In addition, they can simplify the design procedures for fault-tolerant soft sensors by exploiting the interaction between \( G_{f \gamma} \) and \( G_{yu} \).

Based on the two separate models, identification of all the parameters in MSaS is introduced as follows.

1) Identification of \( S \): Given the fault-free datasets \( Z_{xp}, U_{k,s}, \) and \( Y_{k,s} \), a QR factorization (or LQ factorization) is performed according to
\[
\begin{bmatrix} Z_{xp} \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{21} & R_{31} \\ R_{21} & R_{22} & R_{32} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}
\]
(26)
whose purpose is, by minimizing the effects caused by the unknown \( (H_{w,s} W_{k,s} + V_{k,s}) \), to identify \( H_{u,s} \) and \( \Gamma_s L_{xp} \). Therefore, the following two equations hold:
\[
\begin{bmatrix} Z_{xp} \\ U_{k,s} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{21} & R_{31} \\ R_{21} & R_{22} & R_{32} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}
\]
(27a) (27b)
Combining (19), (26), (27a), and (27b) yields
\[
\begin{bmatrix} \Gamma_s L_{xp} \ H_{u,s} \end{bmatrix} = \begin{bmatrix} R_{31} & R_{32} \end{bmatrix} \begin{bmatrix} R_{11} \\ R_{21} \end{bmatrix}
\]
(28)
where \( \dagger \) represents the pseudoinverse operator.

2) Identification of Soft-Sensor Models: Using the fault-free data \( \Psi_p \) and \( \Psi_f \), identification of soft-sensor models can be achieved, similar to the identification of \( S \). To be specific, its implementation can be summarized as follows:
\[
\begin{bmatrix} \Psi_p^T \\ \Psi_f^T \end{bmatrix} = \begin{bmatrix} R_{x1} & R_{x2} & R_{x3} \\ R_{x1} & R_{x2} & R_{x3} \end{bmatrix} \begin{bmatrix} Q_{\xi} \\ Q_{\xi} \end{bmatrix}
\]
(29)
which results in
\[
\begin{bmatrix} \Xi_{k,s} \\ \Psi_p \end{bmatrix} = \begin{bmatrix} R_{x1} & R_{x2} & R_{x3} \\ R_{x1} & R_{x2} & R_{x3} \end{bmatrix} \begin{bmatrix} \Psi_p^T \Psi_f^T \end{bmatrix}
\]
(30)

3) Identification of \( f \)-Related Terms: For the sake of simplicity, \( Y_{f,s}^f \), as well as \( Y^f(k) \), represents the system output related to the sensor fault. Considering \( S^f \) given in (6), (15) and (24) become
\[
Y_{f,s}^f - H_{u,s} U_{k,s} = H_{f,s} F_{k,s} + \Gamma_s X_{k,s} + H_{w,s} W_{k,s} + V_{k,s}
\]
\[
\Xi_{f,s}^f - H_{f,y,s} Y_{k,s} - \Gamma_{\xi,s} L_{xp} \Psi_p^f = 0
\]
(31a) (31b)
where \( H_{f,s} \in R_{k_{y,s}x}^{k_{y,s}x} \) is
\[
H_{f,s} = \begin{bmatrix} E_s \\ \vdots \\ 0 \end{bmatrix}
\]
(32)
Algorithm 1 Data-Driven MSaS: Identification of $S$ and Soft Sensors in Offline Phases

1: Stack the data matrices $Z_{kp}$, $U_{k,s}$, and $Y_{k,s}$;
2: Perform the first QR factorization according to (26);
3: Obtain $[\Gamma_s, L_{xp} \ H_{u,s}]$ based on (28);
4: Stack the data matrices $\Psi_p$ and $\Psi_f$;
5: Perform the second QR factorization according to (29);
6: Calculate $[\Gamma_{\xi,s}, L_{x\xi} \ P \ H_{\xi,y,s}]$ via (30).

Based on (28) and (31a), one can obtain

$$\hat{H}_{s,s} = \mathcal{F} F_{s,s} (F_{s,s}^T \hat{F}_{s,s})^{-1}. \quad (34)$$

In the online phase, $f$ can be estimated according to

$$\hat{F}_{k,s} = (\hat{H}_{s,s} F_{s,s})^{-1} H_{s,s} \mathcal{F}_{s,s} \mathcal{F}_{s,s}$$

which provides optimal prediction. Then the online prediction using MSaS can be obtained through

$$\hat{\xi}_s = \left[ \begin{array}{c} Y_{k,s} - H_{u,s} \hat{F}_{k,s} \\ \hat{\xi}_s \end{array} \right]$$

$$\hat{\hat{y}}_{k,s} = \left[ \begin{array}{c} \hat{Y}_{k,s} - H_{u,s} \hat{F}_{k,s} \\ \hat{\xi}_s \end{array} \right]$$

$$\hat{\hat{\hat{y}}}_{k,s} = \left[ \begin{array}{c} \hat{F}_{k,s} \left[ \begin{array}{c} \gamma \xi \end{array} \right] \end{array} \right]$$

C. Designs’ Procedures

On the basis of the above analysis, the schematic of the proposed MSaS approach is presented in Fig. 2, where $f$ is an optimal estimation of a sensor fault. For clarity, Algorithms 1–3 summarize the implementation procedures of MSaS step by step.

Retrospecting the implementation procedures and mathematical descriptions, further discussions on how to achieve one-step prediction and how to adjust the prediction error are given in Remarks 3 and 4, respectively.

Remark 3: In MSaS, $\hat{\hat{\hat{\xi}}}_s(k)$ contains multiple-step prediction. Using the identification results [including parameters and $f$ is an optimal estimation of a sensor fault. For clarity, Algorithms 1–3 summarize the implementation procedures of $\hat{\xi}_s$ and $F_{k,s}$, respectively.

Algorithm 2 Data-Driven MSaS: Identification of the Sensor Fault-Related Terms in the Offline Phase

**Known parameters:** $[\Gamma_s, L_{xp} \ H_{u,s}]$ and $[\Gamma_{\xi,s}, L_{x\xi} \ P \ H_{\xi,y,s}]$

**Input:** $U_{k,s}$ and $F_{k,s}$

**Output:** $H_{f,s}$

1: Load data sets $U_{k,s}$ and $F_{k,s}$;
2: Obtain $Y_{k,s}^T$, that is the output of $\mathcal{F}$;
3: Let $H_{f,s} F_{k,s}$ be $\mathcal{F}$ according to (33);
4: Define $H_{f,s}$ via (34).

Algorithm 3 Data-Driven MSaS: Online Prediction

1: Read the new data and form $u_s(k)$, $y_s(k)$;
2: Calculate $H_{f,s} f_s(k)$ according to

$$H_{f,s} f_s(k) = y_s(k) - \Gamma_s L_{xp} H_{u,s} y_s(k) - H_{f,s} f_s(k); \quad (37)$$

3: Estimate $f_s(k)$ based on

$$\hat{f}_s(k) = (H_{f,s}^T H_{f,s})^{-1} H_{f,s}^T Y_{k,s}; \quad (38)$$

4: Obtain $\hat{y}_s(k)$ via

$$\hat{y}_s(k) = y_s(k) - H_{f,s} \hat{f}_s(k); \quad (39)$$

5: Predict $\hat{\xi}_s(k)$ based on

$$\hat{\xi}_s(k) = [\Gamma_{\xi,s}, L_{x\xi} \ P \ H_{\xi,y,s}] \hat{\hat{\hat{y}}}_s(k); \quad (40)$$

Note: In the algorithm, $y_s(k)$ represents the online stacked system output that may not contain faults.
IV. SUBSPACE-ALIANTED SCHEME

This section will develop an approach called SSaS, whose purpose is to improve the accuracy of fault-tolerant soft sensors for long-term predictions by reducing the cumulative error.

A. Parameter Identification

Consider $S^f$ in (6). Its extended form is given in (31a). Then a block-Hankel matrix $Z_{zp}$ is defined as

$$Z_{zp} = \begin{bmatrix} F_{k-s-1}^T U_{k-s-1}^T (Y_f^k)^T_{k-s-1} \end{bmatrix}^T .$$

(42)

1) Identification of $S^f$: Performing a QR factorization according to

$$\begin{bmatrix} Z_{zp} \\ F_{k,s} \\ U_{k,s} \\ Y_{k,s}^f \end{bmatrix} = \begin{bmatrix} R_{11} \\ R_{21} \\ R_{31} \\ \vdots \end{bmatrix} \begin{bmatrix} Q_{11} \\ Q_{21} \\ Q_{31} \end{bmatrix}$$

(43)

results in the following equations:

$$\Gamma_s L_{xp} = R_{41} R_{41}^{-1} - R_{42} R_{42}^{-1} R_{41} R_{41}^{-1} - R_{43} R_{43}^{-1} R_{43} R_{43}^{-1}$$

(44a)

$$H_{k,s} = R_{42} R_{42}^{-1} - R_{43} R_{43}^{-1} R_{43} R_{43}^{-1}$$

(44b)

$$H_{k,s} = R_{43} R_{43}^{-1}$$

(44c)

Because $\Gamma_s L_{xp} Z_{zp} = \Gamma_s X_{k,s}^f \in R^{k_f(s+1) \times N}$ and $X_{k}^f \in R^{k_f \times N}$, there exists a matrix $\Gamma_{k,s}^f$ such that

$$\Gamma_{k,s}^f X_{k,s}^f \in R^{k_f(s+1) \times N}$$

(45a)

$$\Gamma_{k,s}^f X_{k,s}^f = 0 \implies \Gamma_{k,s}^f H_{k,s} U_{k,s} + \Gamma_{k,s}^f F_{k,s} \approx 0.$$  

(45b)

Also, there always exists $F_{rm}$, based on which one obtains

$$F_{rm} \Gamma_{k,s}^f \approx 0$$

(46)

if $\Gamma_{k,s}^f H_{k,s}$ does not have the full row rank.

Remark 5: In (46), $F_{rm}$ or (its bases) spans the left null space of $\Gamma_{k,s}^f H_{k,s}$. It is of interest, with the aid of $F_{rm}$, to remove the influences caused by the sensor fault $f$. Since the subspace spanned by $F_{rm}$ is different from the measurement space $y \in R^{k_f}$, the proposed fault-tolerant soft sensor in this section is hence called SSaS.

In fact, calculation of $F_{rm} \Gamma_{k,s}^f$ according to (44a)–(46) is troublesome. To simplify the computation, an alternative solution of $F_{rm}$ is developed as follows. According to the relationship between $U_{k,s}$ and $Y_{k,s}^f$ given in (31a), we know that

$$\begin{bmatrix} U_{k,s} \\ Y_{k,s}^f \end{bmatrix} = \begin{bmatrix} I \\ H_{k,s} \end{bmatrix} \Gamma_s L_{xp} \begin{bmatrix} Z_{zp} \\ F_{k,s} \end{bmatrix}$$

(47)

Combining (43) and (47) yields

$$\begin{bmatrix} I \\ H_{k,s} \end{bmatrix} \Gamma_s L_{xp} \begin{bmatrix} Z_{zp} \\ F_{k,s} \end{bmatrix} = \begin{bmatrix} R_{31} \\ R_{32} \end{bmatrix} \begin{bmatrix} Q_{31} \\ Q_{32} \end{bmatrix}$$

(48a)

$$H_{w,s} W_{k,s} + V_{k,s} = R_{43} Q_{43}.$$  

(48b)

Therefore, $F_{rm} \Gamma_{k,s}^f$ can be directly obtained by

$$F_{rm} \Gamma_{k,s}^f = \begin{bmatrix} R_{31} \\ R_{32} \end{bmatrix} \begin{bmatrix} Q_{31} \\ Q_{32} \end{bmatrix}$$

(49)

such that

$$F_{rm} \Gamma_{k,s}^f = \begin{bmatrix} R_{31} \\ R_{32} \end{bmatrix} \begin{bmatrix} Q_{31} \\ Q_{32} \end{bmatrix} = 0$$

(50)

where $\ker(\cdot)$ denotes the kernel of a linear mapping. Further developments and additional comments on (49) will be given in Section V.

2) Identification of Soft Sensor Models: Along with (46) and Remark 5, a new subspace, whose bases are orthogonal to $H_{k,s} X_{k,s}^f$, is defined as follows:

$$y_{s}^f(k) = P_{y} y_{s}(k) \in R^{k_f}$$

(51)

in which

$$F_{rm} \Gamma_{k,s}^f = \begin{bmatrix} P_{u} \\ P_{y} \end{bmatrix}$$

(52a)

$$\begin{bmatrix} P_{u} \\ P_{y} \end{bmatrix} \begin{bmatrix} U_{k,s} \\ Y_{k,s}^f \end{bmatrix} = 0.$$  

(52b)

$$y_{s}^f(k) = P_{y} y_{s}(k) = -P_{u} H_{k,s} u_{k}(k).$$

(53b)

It indicates that the obtained $y_{s}^f$ is independent of sensor faults.

Therefore, we can define a $y_{s}^f$-based soft sensor to obtain the online prediction via (11a) and (11b). Its extended form can be described by the following equation:

$$\Xi_{k,s} \in R^{k_f(s+1) \times k_f+1}$$

(54)

where $H_{k,s} Y_{s}^f \in R^{k_f(s+1) \times k_f+1}$ and $\Gamma_{k,s} \in R^{k_f(s+1) \times k_f+1}$ have the following structures:

$$H_{k,s} Y_{s}^f = \begin{bmatrix} M \\ \vdots \\ L H^{-1} I \end{bmatrix}, \quad \Gamma_{k,s} = \begin{bmatrix} L \\ \vdots \\ L H' \end{bmatrix}.$$  

(55)

Similar to (23a), a new notation is introduced as

$$\Psi_{p} = \begin{bmatrix} Y_{s}^f_{k-s-1,x} \\ \Xi_{k,s} \end{bmatrix}$$

(56)

using fault-free data. The following QR factorization

$$\begin{bmatrix} Y_{s}^f_{k-s-1,x} \\ \Xi_{k,s} \end{bmatrix} = \begin{bmatrix} R_{11} \\ R_{21} \\ \vdots \end{bmatrix} \begin{bmatrix} Q_{11} \\ Q_{21} \end{bmatrix}$$

(57)

implies the following three relationships:

$$Q_{11} = R_{11}^{-1} Y_{s}^f_{k,s}$$

(58a)

$$Q_{21} = R_{21}^{-1} (\Psi_{p} - R_{21} \Xi_{k,s})$$

(58b)
Algorithm 4 Data-Driven SSaS: Identification of Soft Sensors (Using the Alternative Solution) in the Offline Phase

1: Form data sets \( Z_{xpf}, F_{k,s}, U_{k,s}, \) and \( Y^f_{k,s} \);
2: Do a QR factorization according to (43);
3: Obtain \( F_{rm}\Gamma^1_{s} \) via (49);
4: Define \( P_{y} \) based on (52a) which spans \( y_{s} \).

\[
\Xi_{k,s} = R_{331}Q_{11} + R_{332}Q_{22} + R_{333}Q_{33}. \tag{58c}
\]

Combining (58c) with (58a) and (58b) yields

\[
\Xi_{k,s} = R_{331}R_{331}^{-1}Y_{k,s} + R_{332}R_{332}^{-1}R_{421}^{-1}Y_{k,s}, \tag{59}
\]

Therefore, we can define a matrix \( \Gamma^1_{s} \) based on \( R_{332} \), i.e.,

\[
\Gamma^1_{s} = \ker(R_{332}) \quad \Longrightarrow \tag{60a}
\]

\[
\Xi_{k,s} := \Gamma^1_{s}\Xi_{k,s} = \Gamma^1_{s}H_{x,y,s}Y_{k,s}, \tag{60b}
\]

which can be used for the construction of fault-tolerant soft sensors. Then, the prediction result is

\[
\Xi_{k,s} = (\Gamma^1_{s})^\dagger \Xi_{k,s}. \tag{61}
\]

B. Designs’ Procedures

Based on the above analysis, the schematic of SSaS is shown in Fig. 3. It can be observed from Fig. 3 that two subspaces are involved to complete an online prediction task. In addition, it is worth pointing out that the offline training data used in SSaS are different from that used in MSaS. The design procedures of SSaS are given in Algorithms 4–6.

For the two subspaces shown in Fig. 3, Remark 6 details their physical interpretations.

Remark 6: The first subspace is constructed based on the sensor fault-related data, in which \( f_{s} \) is independent of \( H_{x,y} \). As a result, the effects on \( y_{s} \) and \( e_{s} \) introduced by sensor faults will disappear. The second subspace is related to soft sensors, in which the past prediction error has no contribution to the current prediction.

### Algorithm 5 Data-Driven SSaS: Identification of Soft Sensors (Based on the Fault-Free Data) in the Offline Phase

1: Obtain \( Y_{k,s} \) based on \( S \) and (51) in the fault-free case;
2: Form data sets \( Y_{k,s}, \Psi_{p}, \) and \( \Xi_{k,s} \);
3: Perform a QR factorization according to (57);
4: Calculate \( \Gamma^1_{s} \) via (60a) and define the subspace \( \Xi_{k,s} \) via (60b);
5: Find the projection \( (\Gamma^1_{s})^\dagger \) based on (61).

### Algorithm 6 Data-Driven SSaS: Online Prediction

1: Read the new data and form \( y^f_{s} \);
2: Obtain \( Y_{k,s} \) according to (51);
3: Calculate \( e_{s} \) based on

\[
\hat{e}_{s}(k) = \Gamma^1_{s}H_{x,y,s}Y_{k,s}, \tag{62}
\]

4: Predict \( \xi_{s}(k) \) based on

\[
\hat{\xi}_{s}(k) = (\Gamma^1_{s})^\dagger \hat{e}_{s}(k). \tag{63}
\]

Note: In the algorithm, \( y^f_{s}(k) \) represents the online stacked system output that may not contain faults.

### C. Performance Analysis

Based on (7c) and (63), \( e_{s}(k) \) can be defined by

\[
e_{s}^{f}(k) = \xi_{s}(k) - \hat{\xi}_{s}(k). \tag{64}
\]

for the faulty case. In the right-hand side of (64), the superscript “f” is not used since the effects caused by the sensor faults have already been removed. Next, we will analyze the performance of SSaS from the viewpoint of the robustness to the past prediction errors.

Based on \( \Gamma^1_{s} \) defined in (60a), the orthogonality holds

\[
\hat{\xi}_{s}^{T}(k) \hat{e}_{s}^{f}(k) = 0 \tag{65}
\]

in which \( e_{s}^{f}(k) \) is induced by the past prediction error \( e_{s}^{f}(k - s - 1) \). Hence, SSaS has significant robustness to both the past estimation error and the cumulative error. The detailed theoretical analysis of (65) is presented in Appendix A.

### V. Improved Version of Measurement-Space-Aided Scheme

In this section, another approach to fault-tolerant soft sensors, called IMSaS, will be developed, which also possesses robustness to the cumulative error. After that, two alternative solutions to the QR factorization are provided.

A. Parameter Identification, Designs’ Procedures, and Performance Analysis

Consider \( S^f \) in (6). U using (34) and (35), \( \hat{Y}_{k,s} \) can be obtained via (36b). Before proceeding, \( \Gamma_{s}L_{xpf}H_{u,r} \) have to be identified [see (28)].

Based on (29), one can obtain

\[
\begin{bmatrix}
Y_{k,s} \\
\Xi_{k,s}
\end{bmatrix} = \begin{bmatrix}
R_{321} & R_{322} \\
R_{31} & R_{322}
\end{bmatrix} \begin{bmatrix}
Q_{1} \\
Q_{2}
\end{bmatrix} + R_{333}Q_{3}. \tag{66}
\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The IMSaS approach depicted in Fig. 4, the complete implementation can be divided into the following two components:

1. Data-Driven IMSaS: Identification of
2. Data-Driven IMSaS: Online Prediction

When choosing $\hat{f}(k)$, the estimation error cannot be avoided regardless of optimality of the estimator designed in (36b). Define the error as $e_{y,s}^f(k)$

$$e_{y,s}^f(k) = y_s^f(k) - H_{f,s}\hat{f}(k) - \hat{y}_s(k).$$

Combining (67) and (72) yields

$$E_y(y_s^f(k) - H_{f,s}\hat{f}(k) - e_{y,s}^f(k)) = -E_{\xi}(e_{f,s}^f(k) + \hat{z}_s(k))$$

which results in the following relationship:

$$\hat{\xi}_s(k) = (E_{\xi})^T E_y H_{f,s}\hat{f}(k) + (E_{\xi})^T E_y e_{y,s}^f(k) - (E_{\xi})^T E_y y_s^f(k) - e_{y,s}^f(k)$$

$$\implies \hat{\xi}_s(k) \perp e_{f,s}^f(k).$$

**Remark 7:** In constructing the model of fault-tolerant soft sensors, $E_{\text{rm}}$, as well as its bases, generates a subspace. In this subspace, $\hat{\xi}_s(k)$ is independent of the past prediction error $e_{f,s}^f(k-1)$ because $E_{\text{rm}}$ is approximately orthogonal to $G_{\xi,s}L_{\text{exp}}\psi_p$. The property is of practical interest and is the essential of IMSaS.

### B. Alternative Solutions to QR Factorizations

The approach taken by SSaS and IMSaS is through multiple schemes are not able to eliminate the effects caused by sensor faults present. In the presence of sensor noises, the subspace-based method is independent of the past prediction error. This approach is robust when there are only the measurement noises present. In the presence of faults, the subspace-based schemes are not able to eliminate the effects caused by sensor faults because the kernel space does not exist.

### VI. Conditions of Designs and Implementations

This section will investigate the sufficient and necessary conditions for the existence of MSaS, SSaS, and IMSaS. After that, a comparison among the three proposed fault-tolerant soft sensors will be made.

To proceed, we define a new matrix $H_{f,s}^{\text{total}}$ as

$$H_{f,s}^{\text{total}} = \begin{bmatrix} E_s & \cdots & \cdots & E_{s-n+1} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ CA^{n-1}E_s & \cdots & \cdots & E_s \end{bmatrix}$$

which considers both the actuator and sensor faults.
A. Unified Conditions

1) Unified Necessary Condition: In the parameter-identification process, a (hidden but) necessary condition for all the three proposed approaches is that the inputs of systems to identify soft sensor models satisfy a persistent excitation condition [28], [31], [32]. Taking SSaS as an example, using identification process, a (hidden but) necessary condition for SSaS, we can rewrite (47) as follows:

\[
\begin{align*}
\text{rank} \left( \begin{bmatrix} F_{k,s} & U_{k,s} \\ X_k & \end{bmatrix} Z_{yf}^T \right) &= \text{rank} \left( \begin{bmatrix} F_{k,s} & U_{k,s} \\ Y_{k,s} & \end{bmatrix} \Psi_p^T \right) \\
&= (s+1)(k_u + k_f) + k_z. \quad (77)
\end{align*}
\]

2) To identify \( \mathbf{z}_{k,s} \) for the second subspace:

\[
\begin{align*}
\text{rank} \left( \begin{bmatrix} Y_{k,s} & \mathbf{z}_{k,s} \\ X_{k,s} & \end{bmatrix} \Psi_p^T \right) &= \text{rank} \left( \begin{bmatrix} Y_{k,s} & \mathbf{z}_{k,s} \\ \mathbf{z}_{k,s} & \end{bmatrix} \Psi_p^T \right) \\
&= (s+1)k_z + k_z. \quad (78)
\end{align*}
\]

2) Unified Necessary and Sufficient Condition: \( \mathcal{G}_{yf} \) is the transfer function from \( f(z) \) to the system output \( y(z) \), as defined in (14). Essentially, \( H_{\text{total}}^{\text{SSaS}} \) given in (76) is a data-based model of \( \mathcal{G}_{yf} \). Therefore, one has:

\[
\begin{align*}
y^f(z) - y(z) &= \mathcal{G}_{yf}(z) \\
G_{yf} &= \mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{E}_u + \mathbf{E}_n \quad (79)
\end{align*}
\]

which is a mathematical description of the changes in \( y \) caused by the occurrence of \( f \) [36]. A fault can be dealt with if \( |y^f(z) - y(z)| \neq 0 \). To be specific, \( |y^f(z) - y(z)| \) should not be zero at least at some time instants. Therefore, the unified necessary and sufficient condition for the three soft sensors to be fault tolerant can be summarized as: Given the datasets that satisfy a persistent excitation condition, MSaS, SSaS, and IMSaS are feasible if and only if

\[
\mathbf{C}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{E}_u + \mathbf{E}_n \neq \mathbf{0}. \quad (80)
\]

B. Specific Conditions

Aside from the conditions mentioned above, the fundamental difference among the three proposed approaches in both the designs and implementations is reflected in the sufficient conditions. The details are given as follows.

1) Sufficient Conditions of MSaS: As presented in Section III, the design of MSaS is primarily based on least squares (used for determining parameters) and least mean squares (used for estimating the fault amplitude). As a result, there is no further sufficient condition needed. Therefore, its design has the weakest conditions among the three proposed algorithms.

2) Sufficient Conditions of SSaS: To develop the sufficient condition for SSaS, we can rewrite (47) as follows:

\[
\begin{align*}
\begin{bmatrix} F_{k,s} \\ U_{k,s} \\ Y_{k,s} \end{bmatrix} &= \begin{bmatrix} \mathbf{I} & 0 & 0 \\ 0 & \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} F_{k,s} \\ U_{k,s} \\ \mathbf{H}_{\text{total}}^{\text{SSaS}} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_{k,s} \\ \mathbf{X}_k \end{bmatrix} \\
&+ \begin{bmatrix} 0 \\ 0 \\ \mathbf{H}_{w,s} \mathbf{W}_{k,s} + \mathbf{V}_{k,s} \end{bmatrix}. \quad (81)
\end{align*}
\]

Based on (45b), one obtains

\[
\begin{align*}
\Gamma_f^+ \mathbf{H}_{u,s} \mathbf{V}_{k,s} &= -\Gamma_f^+ \mathbf{H}_{\text{total}}^{\text{SSaS}} \mathbf{f}_s \implies \quad (82a) \\
\Gamma_f^+ \mathbf{Y}_{k,s} &= \mathbf{0} \quad (\text{i.e., } \Gamma_f^+ \perp \mathbf{y}, \mathbf{f}) \quad (82b)
\end{align*}
\]

even though \( \mathbf{u} \) is independent, and (82a) and (82b) indicate the first sufficient condition

\[
\Gamma_f^+ \mathbf{H}_{u,s} \mathbf{u} \neq -\Gamma_f^+ \mathbf{H}_{\text{total}}^{\text{SSaS}} \mathbf{u}. \quad (83)
\]

Following (82a), a special case, for instance, is

\[
\mathbf{E}_u = \mathbf{B}, \quad \mathbf{E}_n = \mathbf{D}, \quad \mathbb{E}(\mathbf{f}) = -\mathbb{E}(\mathbf{u}) \quad (84)
\]

where \( \mathbf{u} \) and \( \mathbf{f} \), which are independent, satisfy (77). Furthermore, according to (49) and (60a), the additional sufficient conditions for SSaS are

\[
\begin{align*}
\text{rank} \left( \begin{bmatrix} \mathbf{R}_{41} & \mathbf{R}_{42} & \mathbf{R}_{43} \\ \mathbf{R}_{41} & \mathbf{R}_{42} & \mathbf{R}_{43} \end{bmatrix} \right) &< (s+1)(k_u + k_f) \quad (85)
\end{align*}
\]

and

\[
\text{rank} \left( \mathbf{R}_{32} \right) < (s+1)k_z \quad (86)
\]

which correspond to identification of the plant model and soft sensor model, respectively. Certainly, (85) and (86) can be equivalently replaced by the datasets used in QR factorizations. As a result, (83), (85), and (86) constitute the three sufficient conditions of SSaS.

3) Sufficient Conditions of IMSaS: The design and implementation of IMSaS are done with the aid of a subspace, i.e., \( \mathbf{E}_{\text{rms}} \) in constructing soft sensors. Similar to (86), the sufficient condition can be described by

\[
\begin{align*}
\text{rank} \left( \begin{bmatrix} \hat{\mathbf{Y}}_{k,s} \\ \mathbf{z}_{k,s} \end{bmatrix} \right) &< (s+1)(k_z + k_y) \quad (87)
\end{align*}
\]

which guarantees the existence of \( \mathbf{E}_{\text{rms}} \). Note that due to the randomness of \( \hat{\mathbf{f}} \), it easily results in

\[
\begin{align*}
\text{rank} \left( \begin{bmatrix} \hat{\mathbf{Y}}_{k,s} \\ \mathbf{z}_{k,s} \end{bmatrix} \right) &\neq \text{rank} \left( \begin{bmatrix} \mathbf{Y}_{k,s} \\ \mathbf{z}_{k,s} \end{bmatrix} \right) \quad (88)
\end{align*}
\]

and

\[
\begin{align*}
\text{rank} \left( \begin{bmatrix} \hat{\mathbf{Y}}_{k,s} \\ \mathbf{z}_{k,s} \end{bmatrix} \right) &= (s+1)(k_z + k_y). \quad (89)
\end{align*}
\]

In this case, the alternative solution based on an SVD will be preferable. The readers are referred to Appendix B for an in-depth discussion.

Therefore, the relationship given in (87) is a sufficient condition of IMSaS.

Remark 9: Because of the presence of \( \hat{\mathbf{f}} \), both the sufficient and necessary conditions differ from that for the traditional system identification approaches for soft-sensor modeling. In addition, a unified necessary condition for the three proposed fault-tolerant schemes is that the dimension of \( \hat{\mathbf{f}} \) should not be more than system outputs.
C. Comparative Analysis

Based on the introduced concept, design procedures, and theoretical analysis, a comparison among the three proposed approaches is exploited. Table I presents the comparison results by considering design conditions, design complexities, robustness to cumulative errors, and computation efficiency (closely related to online time consumption).

As observed from Table I, each approach has its advantage and disadvantage. A choice should be made according to the practical requirements. Section VII will elaborate the comparisons shown in Table I through two case studies.

VII. CASE STUDIES AND APPLICATIONS

In this section, a numerical simulation and an industrial case study are adopted to carry out prediction using the three proposed fault-tolerant soft sensors.

A. Numerical Simulation

To create a suitable illustrative numerical model, slight modifications are made on a traction system given in [28].

\[
A = \begin{bmatrix}
0.0005 & -0.0009 & -0.4448 & -0.3468 \\
-0.0009 & 0.0005 & 0.3468 & -0.4448 \\
-0.0001 & 0.0001 & 0.0437 & 0.0532 \\
-0.0001 & -0.0001 & -0.0532 & 0.0437 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0.3155 \\
0.0017 \\
0.3155 \\
0.0006 \\
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
\end{bmatrix}
\]

and \(D = 0\). In addition, the variable to predict is obtained based on a dynamic model with the following parameters:

\[
H = 0.18, \quad J = [0.3 \ 0.5], \quad L = 0.3, \quad M = [1 \ 2]. \quad (91)
\]

Furthermore, \(w(k) \sim N(0, \text{diag}(0.04, 0.02, 0.04, 0.02))\) and \(v(k) \sim N(0, \text{diag}(0.01, 0.01))\), where \(N(\cdot)\) denotes Gaussian distribution.

The sensor fault \(f\) in the offline training phase is

\[
E_s = \begin{bmatrix}
1 & 0 \\
0 & 0 \\
\end{bmatrix}, \quad f = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + N\left(0, \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}\right). \quad (92)
\]

Based on the dynamic model (90) and (91), \(1 \times 10^4\) samples with the faults defined in (92) are used to identify the parameters according to Algorithms 1 and 2 of MSaS, Algorithms 4 and 5 of SSaS, and Algorithm 7 of IMSaS, where the operation point is \(u = [5 \ 10]^T\). Another \(2 \times 10^3\) samples are used for online prediction of \(\xi\).

Now we consider two kinds of faults in the online phases as follows.

1) A sensor fault occurs in \(S\) but is unrelated to the prediction variable

\[
E_s = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \quad (93)
\]

2) An actuator fault occurs in \(S\) and affects the prediction variable

\[
E_a = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^T. \quad (94)
\]

In addition, the fault magnitude is

\[
f(k) = \begin{bmatrix} [2 \ 5]^T, & k = 1, \ldots, 1000 \\
[50 \ 10]^T, & k = 1001, \ldots, 2000. \end{bmatrix}
\]

Different from the offline training, the operating point is set to be \(u = [6 \ 9]^T\). Based on Fig. 1 and (6)–(8b), \(E_s f(k)\) has no influence on \(\xi\); therefore, this term must be eliminated when making predictions.

1) Online Prediction Using the Proposed MSaS Method: By simulating both the sensor and actuator faults, Fig. 5 shows the estimation results of \(f\) and prediction result of MSaS. It can be observed that Algorithms 2 and 3 can estimate \(f\) successfully from \(\mathcal{F}_{\text{online}}\) that is affected by \(E_s f\) and \(E_a f\). As aforementioned in Section III, the estimation error shown in Fig. 5 gradually increases in terms of the negative influences on \(\xi\), resulting in unacceptable prediction results. In fact, the subgraph of Fig. 5 at the bottom illustrates the performance analysis in Sections III and IV.
2) **Online Prediction Using the Proposed SSaS Method:**

Fig. 6 depicts the prediction result using the second proposed scheme where the red solid line is the true value, and blue and black lines are the predictions using SSaS and tradition methods (see [22] without fault-tolerant abilities), respectively. When the magnitude of $f$ is small, there is no obvious difference between the two methods. However, only SSaS can obtain the accurate result for the sensor fault with large magnitudes. In comparison to MSaS, SSaS also shows the robustness to the cumulative error.

3) **Online Prediction Using the Proposed IMSaS Method:**

To rectify the incorrect predictions shown in the last subgraph of Fig. 5, a subspace of $\xi$ is adopted in the proposed IMSaS. It makes the prediction results depend only on the current error of estimating $f$. The blue curve in Fig. 7 is the prediction result of IMSaS, showing a satisfactory performance.

### B. Industrial Application

The debutanizer column is the central unit used for splitting naphtha and desulfuration [37]. It consists of six parts, i.e., the heat exchanger, bottom reboiler, feed pump, head reflux pump, reflux accumulator, and overhead condenser. Fig. 8 presents the flowchart of a debutanizer column that can remove propane and butane from the naphtha stream.

In practical operations, the concentration of butane cannot be measured in real-time, which necessitates soft sensors to obtain the desired control quality for the process. For this purpose, seven sensors are equipped in the sequential debutanizer column, as listed in Table II.

The plant data of the debutanizer column are provided by Fortuna et al. [37]. There are 2393 samples that are divided into 1393 samples for offline training and the rest for online tests. The faults in the offline and online phases are considered as follows: $E_\delta = \text{diag}(1, 0, 0)$ and the following holds.

1) **Offline Phase:**

$$f(k) = \begin{cases} 
[2 \ 0 \ 0]^T, & k = 1, \ldots, 500 \\
[5 \ 0 \ 0]^T, & k = 501, \ldots, 1393. 
\end{cases}$$

(95)

2) **Online Phase:**

$$f(k) = \begin{cases} 
[0 \ 0 \ 0]^T, & k = 1, \ldots, 250 \\
[5 \ 0 \ 0]^T, & k = 251, \ldots, 500 \\
[10 \ 0 \ 0]^T, & k = 501, \ldots, 750 \\
[20 \ 0 \ 0]^T, & k = 751, \ldots, 1000. 
\end{cases}$$

(96)

In this application, we choose $s_p = 10$ and $s_f = 5$.

1) **Online Prediction Using the Proposed MSaS Method:**

By setting $f$ via (96), Fig. 9 shows that MSaS not only can estimate the sensor fault successfully but also has a good prediction for $\xi$. There are several spikes when the fault magnitudes change.
Fig. 9. Estimation and prediction results of the proposed MSaS approach on the debutanizer column process.

The main reason is the “moving window” [see (13a) and (13b)] adopted in data-driven dynamic approaches [38].

2) Online Prediction Using the Proposed SSaS Method:
As shown in Fig. 10, the prediction result using SSaS is not influenced by the sensor fault; thus, it also has a good prediction accuracy. On the contrary, the traditional approach [21], whose result is described by the black curve, shows significant variations corresponding to the varying faults. The difference indicates that in the prediction process, SSaS can generate an effective subspace in which \( \hat{\xi} \) and \( f \) are orthogonal to each other.

3) Online Prediction Using the Proposed IMSaS Method:
As illustrated in Section V, IMSaS not only has fault-tolerant abilities but also is robust to the cumulative error. The result of IMSaS given in Fig. 11 is obtained in the presence of \( \hat{f} \) shown in Fig. 9. It can be readily observed from Fig. 11 that IMSaS shows the excellent prediction performance when the faults are present.

C. Discussion

Based on the simulation results and Table I, several additional notes are listed below.

1) IMSaS takes advantage of the orthogonality of subspace, removing the influences caused by sensor faults. It has competitive advantages among the three proposed fault-tolerant soft sensors.

2) Both the sufficient and necessary conditions are the foundations of the proposed fault-tolerant soft sensors. Because of space constraints, these conditions were only briefly investigated in this study. It deserves more in-depth investigations.

3) Extending fault-tolerant soft sensors developed in this study to closed-loop systems is possible using, for example, the orthogonal projection [39] and a technique similar to image representations [40].

4) In fact, \( f \) in Fig. 1 can be regarded as (or replaced by) the other unknown signal sources such as disturbances and outliers. From this viewpoint, robust designs will also be the natural extensions of the fault-tolerant soft sensors.

5) Among the three proposed approaches, only MSaS can detect, predict, and classify the faults, because the subspace generated by SSaS and IMSaS is orthogonal to the fault directions.

6) Using nonlinear regression through nonlinear operators with time-delay units could extend the three proposed fault-tolerant soft sensors to nonlinear dynamic systems.

VIII. CONCLUSION

This article has developed three novel fault-tolerant soft-sensor algorithms. Different from the existing methods [41], the proposed algorithms are suitable for dynamic systems with consideration of both the sensor and actuator faults. The first scheme is designed with the aid of the measurement space, in which the influence caused by sensor faults is removed via optimal estimation. The second approach is designed based on an instrumental subspace, in which the influence caused by sensor faults is eliminated owing to the orthogonality properties. The third soft sensor is an improved version of the first one, by considering the robustness to the cumulative error.

This study is the first attempt toward fault-tolerant soft sensors for dynamic systems. It is expected to open a new avenue for the development of soft sensors. In this study, sensor-fault data in the offline training phase are assumed to be known.
Future work could carry out fault-tolerant soft sensors that can address the unknown faults. Another promising research direction would be the design of fault-tolerant soft sensors for closed-loop systems.

APPENDIX A
THEORETICAL JUSTIFICATION OF (65)

In (54), $\Gamma_{k:t}X_{k:t}$ can be estimated via a non-steady-state Kalman filter. Similar to (17), one has

$$
\Gamma_{k:t}X_{k:t}(k) = \Gamma_{k:t}X_{k:t}(\hat{X}_k(k))
$$

(A.1a)

$$
\hat{X}_k(k) = \left[\psi_p^T(k) - \xi^T(k - s - 1)\right]^T
$$

(A.1b)

In the online phase, $\hat{X}_k(k - s - 1)$ will be replaced by $\hat{X}_k(k - s - 1)$, since $\hat{X}_k(k - s - 1)$ [together with $\xi^T(k - s - 1)$] is not measurable. Therefore, there exists an estimation error $e_{\xi}^T(k - s - 1)$ in (A.1b), i.e.,

$$
\xi^T(k - s - 1) = e_{\xi}^T(k - s - 1) + \hat{\xi}_k^T(k - s - 1)
$$

(A.2a)

$$
\hat{\psi}_p(k) = \psi_p(k) - \begin{bmatrix}
0 \\
\hat{e}_{\xi}^T(k - s - 1)
\end{bmatrix}
$$

(A.2b)

$$
\Gamma_{k:t}X_{k:t}(k) = \Gamma_{k:t}X_{k:t}(\hat{X}_k(k)) + \Gamma_{k:t}L_{\xi:p}^T e_{\xi}^T(k - s - 1)
$$

(A.2c)

Combining (A.2a)–(A.2c) with (54) yields

$$
\xi^T(k) = H_{\xi,y}(k) + \Gamma_{k:t}^T L_{\xi:p}^T e_{\xi}^T(k)
$$

(A.3a)

$$
\hat{\psi}_p(k) = \psi_p(k) - \begin{bmatrix}
0 \\
\hat{e}_{\xi}^T(k - s - 1)
\end{bmatrix}
$$

(A.3b)

$$
e_{\xi}^T(k) = \Gamma_{k:t}L_{\xi:p}^T e_{\xi}^T(k - s - 1)
$$

(A.3c)

As shown in (A.3c), the prediction error will be accumulated in a long-term prediction process if (A.3b) is directly used. Based on the QR factorization, one can obtain

$$
\Gamma_{k:t}L_{\xi:p}^T = R_{k:t} L_{\xi:p}^{-1}
$$

(A.4)

Furthermore, $\Gamma_{k:t}^\perp$, defined in (60a), results in

$$
\hat{\xi}_k^T(k) = \Gamma_{k:t}^\perpH_{\xi,y}(k) + \Gamma_{k:t}^\perp\hat{\psi}_p(k) - \Gamma_{k:t}^\perp e_{\xi}^T(k)
$$

(A.5)

which completes (65).

APPENDIX B
TWO ALTERNATIVE SOLUTIONS

1) Alternative Solution 1: Consider $Z_{xp}$, $U_{k:t}$, and $Y_{k:t}$. Performing an SVD according to [31]

$$
\begin{bmatrix}
U_{k:t} \\
Y_{k:t}
\end{bmatrix} = \begin{bmatrix}
U_p & U_r
\end{bmatrix}
\begin{bmatrix}
\Sigma_p & 0 \\
0 & \Sigma_r
\end{bmatrix} \approx \begin{bmatrix}
V_p^T & V_r
\end{bmatrix}
$$

(B.1)

yields

$$
U_r^T := \ker\left(\begin{bmatrix} I \\ H_{u:y} \\ \Gamma_r L_{\xi:p} \end{bmatrix}\right)
$$

(B.2)

2) Alternative Solution 2: Consider a QR factorization given in (26). Performing an SVD according to

$$
\begin{bmatrix}
R_{21} \\
R_{31}
\end{bmatrix} = \begin{bmatrix} U_p & U_r \end{bmatrix}
\begin{bmatrix}
\Sigma_p & 0 \\
0 & \Sigma_r \approx 0
\end{bmatrix} \begin{bmatrix}
V_p^T & V_r
\end{bmatrix}
$$

(B.3)

also obtains (B.2).

REFERENCES

[1] C. Shang, X. Gao, F. Yang, and D. Huang, “Novel Bayesian framework for dynamic soft sensor based on support vector machine with finite impulse response,” IEEE Trans. Control Syst. Technol., vol. 22, no. 4, pp. 1550–1557, Jul. 2014.

[2] F. A. A. Souza, R. Araújo, and J. Mendes, “Review of soft sensor approach for process,” IEEE Trans. Soft Comput., vol. 24, no. 3, pp. 4958–4967, Apr. 2012.

[3] S. Yin, X. Zhu, and O. Kaynak, “Improved PLS focused on process,” IEEE Trans. Soft Comput., vol. 22, no. 4, pp. 1094–1103, Nov. 2013.

[4] C. Shang, X. Gao, and D. Huang, “Novel Bayesian framework for dynamic soft sensor based on support vector machine with finite impulse response,” IEEE Trans. Control Syst. Technol., vol. 22, no. 4, pp. 1550–1557, Jul. 2014.

[5] Z. Ge, “A novel approach for process,” IEEE Trans. Syst. Man Cybern., vol. 46, no. 4, pp. 4958–4967, Apr. 2012.

[6] F. A. A. Souza, R. Araújo, and J. Mendes, “Review of soft sensor approach for process,” IEEE Trans. Soft Comput., vol. 22, no. 4, pp. 1094–1103, Nov. 2013.

[7] S. Yin, X. Zhu, and O. Kaynak, “Improved PLS focused on process,” IEEE Trans. Soft Comput., vol. 22, no. 4, pp. 1094–1103, Nov. 2013.

[8] F. A. A. Souza, R. Araújo, and J. Mendes, “Review of soft sensor approach for process,” IEEE Trans. Soft Comput., vol. 22, no. 4, pp. 1094–1103, Nov. 2013.

[9] Z. Ge, “A novel approach for process,” IEEE Trans. Syst. Man Cybern., vol. 46, no. 4, pp. 4958–4967, Apr. 2012.

[10] Y. Lyu, J. Chen, and Z. Song, “Synthesizing labeled data to enhance soft sensor performance in data-scarce regions,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 12, pp. 7598–7609, Dec. 2022, doi: 10.1109/TNNLS.2021.3085869.

[11] K. Peng, K. Zhang, B. You, and J. Dong, “Quality-related prediction and monitoring of multi-mode processes using multiple PLS with application to an industrial hot strip mill,” Neurocomputing, vol. 168, pp. 1094–1103, Nov. 2015.

[12] H. Kaneko and K. Funatsu, “Applicability domain of soft sensor models based on one-class support vector machine,” AIChE J., vol. 59, no. 6, pp. 2046–2050, Jan. 2013.
[20] J. Zhu, Z. Ge, Z. Song, and F. Gao, “Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data,” *Annu. Rev. Control*, vol. 46, pp. 107–133, Jun. 2018.

[21] X. Yuan, Y. Wang, C. Yang, Z. Ge, Z. Song, and W. Gui, “Weighted linear dynamic system for feature representation and soft sensor application in nonlinear dynamic industrial processes,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1508–1517, Feb. 2018.

[22] S. X. Ding, S. Yin, K. Peng, H. Hao, and B. Shen, “A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill,” *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2239–2247, Nov. 2013.

[23] Y. A. W. Shardt, H. Hao, and S. X. Ding, “A new soft-sensor-based process monitoring scheme incorporating infrequent KPI measurements,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3834–3851, Jun. 2015.

[24] O. Wu, H. Kodamana, N. M. Jan, R. Tan, and B. Huang, “Robust soft sensor development using multi-rate measurements,” *IFAC-PapersOnLine*, vol. 51, no. 1, pp. 10190–10195, Jul. 2017.

[25] H. J. Galicia, Q. P. He, and J. Wang, “A reduced order soft sensor approach and its application to a continuous digestor,” *J. Process Control*, vol. 23, no. 3, pp. 1700–1716, Mar. 2013. doi: 10.1016/j.jprocont.2020.3029946.

[26] W. Ni, S. D. Brown, and R. Man, “A localized adaptive soft sensor for dynamic system modeling,” *Chem. Eng. Sci.*, vol. 111, pp. 350–363, May 2014.

[27] J. L. Hintze, *User’s Guide IV*. Kaysville, UT, USA: NCSS Statistical System, 2007.

[28] H. Chen, B. Jiang, S. X. Ding, and B. Huang, “Data-driven fault diagnosis for traction systems in high-speed trains: A survey, challenges, and perspectives,” *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 489–500, Apr. 2011.

[29] K. Zhang, H. Hao, Z. Chen, S. X. Ding, and K. Peng, “A comparison and evaluation of key performance indicator-based multivariate statistics process monitoring approaches,” *J. Process Control*, vol. 33, pp. 112–126, Sep. 2015.

[30] G. E. P. Box and G. M. Jenkins, *Time Series Analysis-Forecasting and Control*. Sacramento, CA, USA: Holden-Day, 1976.

[31] B. Huang and R. Kadali, *Dynamic Modelling, Predictive Control and Performance Monitoring: A Data-Driven Subspace Approach*. London, U.K.: Springer-Verlag, 2008.

[32] P. Van Overschee and B. D. Moor, “A unifying theorem for three subspace system identification algorithms,” *Automatica*, vol. 31, no. 12, pp. 1853–1864, Dec. 1995.

[33] J. Wang and S. J. Qin, “A new subspace identification approach based on principal component analysis,” *J. Process Control*, vol. 12, no. 8, pp. 84–1855, Dec. 2002.

[34] Y. Guo and B. Huang, “State estimation incorporating infrequent, delayed and integral measurements,” *Automatica*, vol. 58, pp. 32–38, Aug. 2015.

[35] X. Ding, *Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. London, U.K.: Springer-Verlag, 2014.

[36] X. Ding, *Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Berlin, Germany: Springer-Verlag, 2008.

[37] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*. Berlin, Germany: Springer, 2007.

[38] H. Chen, Z. Chai, O. Dogru, B. Jiang, and B. Huang, “Data-driven designs of fault detection systems via neural network-aided learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5694–5705, Oct. 2022. doi: 10.1109/TNNLS.2021.3071292.

[39] B. Huang, S. X. Ding, and S. J. Qin, “Closed-loop subspace identification: An orthogonal projection approach,” *J. Process Control*, vol. 15, no. 1, pp. 53–66, Feb. 2005.

[40] H. Luo, K. Li, O. Kaynak, S. Yin, M. Hoo, and H. Zhao, “A robust data-driven fault detection approach for rolling mills with unknown roll eccentricity,” *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2641–2648, Nov. 2020.

[41] C. A. C. Belchior, R. A. M. Araújo, F. A. A. Souza, and J. A. C. Landeck, “Sensor-fault tolerance in a wastewater treatment plant by means of ANFIS-based soft sensor and control reconfiguration,” *Neural Comput. Appl.*, vol. 30, no. 10, pp. 3265–3276, Nov. 2018.

---

**Hongtian Chen** (Member, IEEE) received the Ph.D. degree from the College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2019. He was a Post-Doctoral Fellow with the University of Alberta, Edmonton, AB, Canada, from 2019 to 2023. He joined Shanghai Jiao Tong University, Shanghai, China, in 2023, as an Associate Professor with the Department of Automation. His research interests include fault diagnosis and fault-tolerant control, data mining and analytics, machine learning, and cooperative control, and their applications in high-speed trains, new energy systems, industrial processes, and autonomous systems. Dr. Chen was a recipient of the Grand Prize of Innovation Award of Ministry of Industry and Information Technology of China in 2019, the Excellent Ph.D. Thesis Award of Jiangsu Province in 2020, the Excellent Doctoral Dissertation Award from the Chinese Association of Automation (CAA) in 2020, and Marie Skłodowska-Curie Actions in 2023.

**Biao Huang** (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in automatic control from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1983 and 1986, respectively, and the Ph.D. degree in process control from the University of Alberta, Edmonton, AB, Canada, in 1997. He joined the University of Alberta in 1997 as an Assistant Professor with the Department of Chemical and Materials Engineering, where he is currently a Full Professor. He was an NSERC Industrial Research Chair in control of oil sands processes and the AITF Industry Chair in process control from 2013 to 2018. His research interests include data analytics, process control, system identification, control performance assessment, Bayesian methods, and state estimation. He has applied his expertise extensively in industrial practice.

Dr. Huang is a fellow of the Canadian Academy of Engineering and the Chemical Institute of Canada. He was a recipient of the Germany’s Alexander von Humboldt Research Fellowship, the Canadian Chemical Engineering Society’s Synrude Canada Innovation and D.G. Fisher Awards, the APEGAs Summit Research Excellence Award, the University of Alberta’s McCalla and Killam Professorship Awards, the Petro-Canada Young Innovator Award, the A-Tech Outstanding Achievement Award in Science and Engineering Award, and the Best Paper Award from the *Journal of Process Control*. **