Article

Sensor Fusion Architecture for Fault Diagnosis with a Predefined-Time Observer

Ofelia Begovich ©, Adrián Lizárraga © and Antonio Ramírez-Treviño *©

Cinvestav del IPN, Unidad Guadalajara, Av. del Bosque 1145, Zapopan Jalisco C.P. 45019, Mexico; ofelia.begovich@cinvestav.mx (O.B.); adrian.lizarraga@cinvestav.mx (A.L.)

* Correspondence: antonio.ramirezt@cinvestav.mx

Abstract: This study focuses on generating reliable signals from measured noisy signals through an enhanced sensor fusion method. The main contribution of this research is the development of a novel sensor fusion architecture that creates virtual sensors, improving the system’s redundancy. This architecture utilizes an input observer to estimate the system input, then it is introduced to the system model, the output of which is the virtual sensor. Then, this virtual sensor includes two filtering stages, both derived from the system’s dynamics—the input observer and the system model—which effectively diminish noise in the virtual sensors. Afterwards, the same architecture includes a classical sensor fusion scheme and a voter to merge the virtual sensors with the real measured signals, enhancing the signal reliability. The effectiveness of this method is shown by applying merged signals to two distinct diagnosers: one utilizes a high-order sliding mode observer, while the other employs an innovative extension of a predefined-time observer. The findings indicate that the proposed architecture improves diagnostic results. Moreover, a three-wheeled omnidirectional mobile robot equipped with noisy sensors serves as a case study, confirming the approach’s efficacy in an actual noisy setting and highlighting its principal characteristics. Importantly, the diagnostic systems can manage several simultaneous actuator faults.

Keywords: sensor fusion; fault diagnosis; predefined-time diagnoser; nonlinear systems; mobile robot

1. Introduction

Modern industrial processes and systems are becoming larger and more complex, demanding high reliability and performance from the system [1]. To meet these demands, modern systems incorporate sophisticated control and decision-making structures, including fault diagnosis and recovery stages, to exhibit intelligent behavior even under abnormal conditions.

Regarding fault diagnosis [2–4], this stage deals with abnormal behaviors (faults) by detecting, locating, and identifying them. The output of this stage can either alert human operators, or be used by a fault-tolerant system with the aim of removing hazardous situations. One of the most widely reported approaches in fault diagnosis is the fault model-based approach [5–7], where the observers, who are the core group focused on generating results [1,2], are used to determine the existence of faults. All previously mentioned results assume that the input signal is sufficiently free of noise; however, this assumption rarely hold in real situations, limiting the use of sophisticated diagnosers. In particular, the proposed diagnoser in those works are based on asymptotic observers or finite-time observers. When noise is considered, the performance of such diagnosers is reduced, producing false fault detection. In other words, the accuracy of diagnosers depends on the reliability of the measured signals; nevertheless, in many real cases the sensor noise limits the reliability of such information.

This problem is overcome by using sensor fusion algorithms [8–11]; unfortunately, in many cases there are not enough different sensors to use such algorithms. Moreover,
the system sensors may be measuring signals which are not comparable with each other, which may make merging them with a sensor fusion technique difficult. To avoid this last drawback, in the literature changing the sensor coordinates has been proposed [12,13]; regrettably this strategy is limited to very particular cases.

This work proposes a novel sensor fusion architecture capable of generating several virtual sensors comparable with each other. These are merged in a sensor fusion stage to obtain more reliable output signals for subsequent processes, such as fault diagnosis, observers, and controllers. The idea behind the proposed architecture is to include two filter stages that generate feasible system inputs and outputs based on the system model, significantly reducing signal noise. Additionally, a voter enhances the reliability of virtual sensor signals by selecting those with the minimum variance.

To show the efficiency of the proposed architecture, its output is used by two diagnosers, one based on a high-order sliding mode (HOSM) observer, and the other based on an enhanced predefined-time observer herein introduced. Both cases show that when the proposed architecture is used, the efficiency of the diagnosers is improved, i.e., the false positive/negative fault detections are reduced, and the fault diagnosis is improved.

This work is organized as follows. Section 2 introduces the sensor fusion architecture, emphasizing how from a subset of output signals and using an input observer scheme, the virtual sensors may be implemented. This section also introduces a novel diagnoser based on an enhanced predefined-time observer. Section 3 introduces a three-wheeled omnidirectional mobile robot, that is the real system on which the proposed architecture and diagnosers are tested. Section 4 shows the experiments and the derived results. All of them show that the use of the proposed architecture improves the diagnosis of faults. Finally, the conclusions are presented.

2. Sensor Fusion Architecture

Figure 1 depicts the proposed sensor fusion architecture and the diagnoser. In the proposed approach, each virtual sensor gives an estimate of the entire set of measured system outputs, and every virtual sensor is merged, using a classical fusion stage, with the measured system output generating a cleaner estimated signal. Afterwards, a voter is used to select the most reliable estimated signal. This single output of the architecture is a reliable estimate of the system output that may be used by a diagnoser or any other device demanding reliable measures of the system. It is worth remarking that every virtual sensor is generated from a subset of the system outputs using an element that we name a homogenizer.

This fusion architecture generates more reliable signals, allowing the use of sophisticated diagnosers (such as those based on sliding modes) capable of diagnosing faults in a short period of time after their occurrence, and drastically reducing the number of false fault detections.
whereas the encoders measure the motors’ positions and velocities. Nevertheless, the state is estimated from the information in \( \Sigma \) of computing all the signals in \( \Sigma \), due to the assumption of observability when the system output is used to estimate the system input. For example, the GPS gives information related to the vehicle’s position and orientation, whereas the encoders measure the motors’ positions and velocities. Nevertheless, the information provided by encoders and GPS may be redundant when direct or inverse kinematics are used. Our approach leverages system characteristics to add redundancy to the system information. To formalize these concepts, the following sets and maps are defined.

**Definition 1.** Let \( \Sigma = \{y_1, \ldots, y_s\} \) be the set of system outputs, and \( y = [y_1 \ldots y_s]^T \) be the system output vector. Let \( \Sigma_i, \Sigma_j \subset \Sigma \) be two arbitrary subsets of \( \Sigma \). If there exists a function \( P_{ij} : \Sigma_i \rightarrow \Sigma_j \) such that from the information provided by the signals in \( \Sigma_i \), the signals in \( \Sigma_j \) may be estimated, then \( P_{ij} \) is the \( i \)-th homogenizer for \( \Sigma_j \). The homogenizer value \( \hat{\Sigma}_j = P_{ij}(\Sigma_i) \) is named the \( i \)-th virtual sensor of \( \Sigma_j \). If \( \Sigma_j = \Sigma \), then \( \hat{\Sigma}_j = P_{ij}(\Sigma_i) \), i.e., the whole set of output sensors is reconstructed from \( \Sigma_i \).

For example, assuming that the motor speed and position are measured in the system outputs \( y_i \) and \( y_j \), respectively, then the homogenizer \( P_{ij} \) can be implemented by an integration procedure, adding a virtual sensor for the position. Homogenizer \( P_{ij} \) may be implemented in several ways, depending on the relationship between signals in \( \Sigma_i \) and \( \Sigma_j \). This work proposes the homogenizer depicted in Figure 2, that is based on the input estimator proposed in [15]. This is for dynamical systems under the assumption of observability when the system output is \( \Sigma_i \subset \Sigma \). For this, signals in \( \Sigma_i \) are used to estimate the system input \( \hat{u} \). Afterwards, leveraging \( \hat{u} \) and the system model, the system outputs \( \hat{\Sigma}_j \), due to \( \Sigma_i \), are estimated, i.e., the proposed homogenizer is capable of computing all the signals in \( \Sigma \). Thus, the homogenizer is represented as a function \( P_i : \Sigma_i \rightarrow \Sigma \). As a notation \( \hat{\Sigma}_i = P_{ij}(\Sigma_i) \) is used to indicate that all system outputs are estimated from the information in \( \Sigma_i \). In addition to \( \hat{\Sigma}_i \), the proposed homogenizer also computes the estimated state \( \hat{x}_i \) and the estimated input \( \hat{u}_i \), obtained from \( \Sigma_i \).

In order to compute the input \( \hat{u}_i \), first, the coordinate transformation

\[
\Phi^i(x) = [(\xi^i)^T, (\eta^i)^T]^T
\]

is applied [15], where \( \xi^i = [(\xi^{ih}_i)^T \ldots (\xi^{ih}_w)^T]^T \) and \( \eta^i \) is the internal system dynamic. Notice that \( \xi^{ih} \) corresponds to the coordinate transformation obtained from the \( a \)-th output \( h_a(x) \), with \( h_a, \ldots, h_w \in \Sigma_i \). Then, by the construction of the coordinate transformation, the
states $\xi^i$ and their derivatives $\dot{\xi}^i = [(\dot{\xi}^{ih_a})^T \ldots (\dot{\xi}^{ih_w})^T]^T$ can be estimated by higher-order sliding mode (HOSM) differentiators [16] (by computing the derivatives of the outputs of the system), where $\xi^{ih_a} = [\xi^{ih_a}]_1^T$, $\xi^{ih_w} = [\xi^{ih_w}]_1^T$, and $r_h$ denotes the relative degree of the $a$-th output $h_a(x)$. Once the state $\xi^i$ and its derivative $\dot{\xi}^i$ are estimated, the input $\bar{u}^i$ is computed as reported in [15]:

$$\bar{u}^i = (E^i)^{-1} \begin{bmatrix} \dot{\xi}^{ih_a} \\ \vdots \\ \dot{\xi}^{ih_w} \end{bmatrix} - \begin{bmatrix} L^a_f h_a((\Phi^i)^{-1}) \\ \vdots \\ L^w_f h_w((\Phi^i)^{-1}) \end{bmatrix}$$

(2)

where $L_f h(x) = \sum_{i=1}^n \frac{\partial h}{\partial x_i} f_i(x)$ denotes the derivative of $h$ along $f$, see [17], and the matrix $E^i(x)$ can be constructed as [15]

$$E^i(x) = \begin{bmatrix} L_{g1} \left( L_f^{(r_a-1)} h_a \right) & \ldots & L_{g1} \left( L_f^{(r_a-1)} h_a \right) \\ \vdots & \ddots & \vdots \\ L_{g1} \left( L_f^{(r_w-1)} h_w \right) & \ldots & L_{g1} \left( L_f^{(r_w-1)} h_w \right) \end{bmatrix}$$

(3)

Notice that the proposed homogenizer depicted in Figure 2 computes the system states from $\bar{u}^i$. Then, using the system model, the system output is estimated, generating virtual system outputs.

**Signal homogenizer $i$**

![Figure 2. Implementation of the signal homogenizer $i$ to obtain the virtual sensors $\bar{\Sigma}^i$, an estimate of the state $\bar{x}^i$, and an estimate for the input $\bar{u}^i$.](image)

2.1.2. Sensor Fusion

The reliability of the values of the system signals in $\Sigma$ is increased if there exists at least one homogenizer $P_i$. In this case, the system outputs $y$ and the signals in $\bar{\Sigma}^i$ are smartly merged by using an extended Kalman filter (EKF) to obtain a more accurate value of signals in $\Sigma$ [18]. Note that the EKF algorithm requires a discrete system, so the system in Equation (1) is discretized with the Euler method or any other discretization method. In addition, herein the subscript $k$ is used to denote the discrete time index.

For the sensor fusion, the signals in the output vector $y$ and the virtual sensors $\bar{\Sigma}^i$ are arranged into an augmented measurement vector as follows [19]:

$$\bar{y}_k^i = [(y_k)^T \ (\bar{y}_k^{\bar{\Sigma}^i})^T]^T$$

(4)

where $(\bar{y}_k^{\bar{\Sigma}^i})^T$ is the vector formed with the virtual sensors $\bar{\Sigma}^i$. While the augmented covariance matrix of the measurement noise is given by

$$R_k^i = \begin{pmatrix} R_k^y & 0 \\ 0 & R_k^{\bar{\Sigma}^i} \end{pmatrix}$$

(5)
where $R^R_k$ is the covariance matrix of the noise of the real sensors’ measurements and $R^S_k$ is the covariance matrix of the noise of the virtual sensors obtained from the homogenizer $i$. The entries of the covariance matrix in Equation (5) are selected based on the respective measurement noise variance. Similarly, the entries of the process noise covariance matrix $Q_k$ are chosen. Once the parameters of the EKF are set, the standard EKF algorithm [20] is used to obtain the required sensor fusion.

2.1.3. Filter

The sensor fusion stage generates more reliable signals. These signals are filtered to reduce their noise even more. To this end, a classical second-order low pass filter (LPF) is used:

$$F(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The parameters $\zeta$ and $\omega_n$ are chosen such that a desired transient and noise reduction are achieved.

2.1.4. Voter

Finally, a voter selects the output of the sensor fusion architecture. This is depicted as the min operation in Figure 1. In this case, it selects the signals $\bar{\Sigma}^f_i$ and $\bar{x}^f_i$, corresponding to the signal $\bar{\Sigma}^f_i$ with the lowest variance.

2.2. Diagnoser

The naive approach of solving the unknown faults $m(t)$ from Equation (1) and the estimated state $\bar{x}$ (output from the proposed architecture) is not possible, since it involves the computation of $\dot{\bar{x}}$, increasing the noise and leading to false faults detection. This inconvenience is avoided by using a fault diagnoser; in particular, this work proposes a diagnoser based on a predefined-time observer to swiftly diagnose the fault.

First, we assume that the estimated system state $\bar{x}$ is partitioned as $\bar{x} = [\bar{x}_m \: \bar{x}_\bar{m}]$, where the states in $\bar{x}_m$ are directly affected by the faults, i.e., $l_m(\bar{x})m(t) \neq 0$ when $m(t) \neq 0$, and the states in $\bar{x}_\bar{m}$ are not affected by the faults, i.e., $l_{\bar{m}}(\bar{x})m(t) = 0$ when $m(t) \neq 0$; then, from Equation (1) we obtain

$$\begin{bmatrix} \dot{\bar{x}}_m \\ \dot{\bar{x}}_{\bar{m}} \end{bmatrix} = \begin{bmatrix} f_m(\bar{x}) \\ f_{\bar{m}}(\bar{x}) \end{bmatrix} + \begin{bmatrix} g_m(\bar{x}) \\ g_{\bar{m}}(\bar{x}) \end{bmatrix} u(t) + \begin{bmatrix} l_m(\bar{x}) \\ l_{\bar{m}}(\bar{x}) \end{bmatrix} m(t)$$

The diagnoser is proposed as the following predefined-time observer:

$$\dot{\hat{x}}_m = f_m(\bar{x}) + g_m(\bar{x}) u(t) + \Theta$$

where $\Theta$ (that is related to the fault value) is the observer feedback law that must be selected to ensure that $\hat{x}_m$ converges to $x_m$ at a predefined-time $T_c$. The estimation error is

$$e = \bar{x}_m - \hat{x}_m$$

where the measurements of $x_m$ are in $\bar{\Sigma}$. The time derivative of the error in Equation (9) is

$$\dot{e} = l_m(\bar{x})m(t) - \Theta$$

Proposition 1 shows how to compute $\Theta$. The proof of this proposition is based on the next theorem, presented in [21].

**Theorem 1** ([21]). Consider the nonlinear system

$$\dot{x}(t) = f(x, t; \rho)$$


Then, the origin of (11) is predefined-time stable with $T_c$, where the pseudoinverse $\Gamma$ is computed using Equation (13). Thus, by using Theorem 1, every component of each estimated fault $\hat{e}_i$ is computed using Equation (13). Therefore, the origin of the error system (10) is predefined-time stable with $T_c$ as its predefined time.

**Proposition 1.** Let $\Delta = l_m(x)m(t)$, where its $i$-th component is bounded by a positive constant, i.e., $|\Delta| \leq \bar{\zeta}_i < \infty$, $\bar{\zeta}_i \in \mathbb{R}^+$. If the $i$-th entry $\Theta_i$ of the observer feedback law $\Theta$ is chosen as

$$\Theta_i = \left[ \frac{\gamma}{T_c} (\alpha |e_i|^p + \beta |e_i|^q)^k + \bar{\zeta}_i \right] \text{sign}(e_i), \quad i = 1, \ldots, d$$

then, the origin of the error system (10) is predefined-time stable with $T_c$ as its predefined time.

**Proof.** For every estimation error $e_i$, consider the continuous radially unbounded Lyapunov function $V(e_i) = |e_i|$, $i = 1, \ldots, d$. The time derivative of each Lyapunov function $V(e_i)$ yields

$$\dot{V}(e_i) = \text{sign}(e_i) |\Delta_i - \Theta_i|$$

$$= -\frac{\gamma}{T_c} (\alpha |e_i|^p + \beta |e_i|^q)^k - \bar{\zeta}_i + |\Delta_i \text{sign}(e_i)|$$

$$\leq -\frac{\gamma}{T_c} (\alpha |e_i|^p + \beta |e_i|^q)^k - \bar{\zeta}_i + |\Delta_i \text{sign}(e_i)|$$

$$= -\frac{\gamma}{T_c} (\alpha |e_i|^p + \beta |e_i|^q)^k - (\bar{\zeta}_i - |\Delta_i \text{sign}(e_i)|)$$

$$\leq -\frac{\gamma}{T_c} (\alpha |e_i|^p + \beta |e_i|^q)^k$$

where $\gamma$ is computed using Equation (13). Thus, by using Theorem 1, every $e_i$ is predefined-time stable, $i = 1, \ldots, d$. Therefore, the origin of the error system (10) is predefined-time stable with $T_c$ as its predefined time.

Since the origin of (10) is reached in a predefined time, it implies that $e(t) = 0$, and $\dot{e}(t) = 0$ for $t \geq T_c$. Then, the equivalent observer feedback law $\Theta_{eq}$ is obtained such that $\dot{e} = 0$ holds. Hence, from Equation (10), $\Theta_{eq}$ is found as

$$0 = l_m(x)m(t) - \Theta_{eq}$$

$$\Theta_{eq} = l_m(x)m(t)$$

(16)

So, if $e = 0$, $\dot{e} = 0$, and Assumption 1 holds, then the estimation of the fault magnitudes $m(t)$ can be computed from Equation (16) as

$$\dot{m}(t) = l_m^{-1}(x)\Theta_{eq}$$

where the pseudoinverse $l_m^{-1}$ exists by Assumption 1. In order to remove the high-frequency components of each estimated fault $\hat{m}_i$, due to the high-frequency signals of $\Theta_{eq}$, second-order LPFs, such as the ones proposed in Section 2.1.3, are required:

$$F_{m_i}(s) = \frac{\omega_{nm_i}^2}{s^2 + 2\zeta_m\omega_{nm_i}s + \omega_{nm_i}^2}, \quad i = 1, \ldots, d$$

(18)
where the filter settling time \( t_{sm_i} = \frac{4}{\zeta_{mi} \omega_{nmi}} \), \( i = 1, \ldots, d \) should be selected to be approximately equal to \( T_c \) to ensure a fast convergence to the real fault \( m_i(t) \).

3. Case Study

The three-wheeled omnidirectional mobile robot depicted in Figure 3 is considered for this case study.

![Figure 3](image-url) Three-wheeled omnidirectional mobile robot.

3.1. Prototype

Our prototype, depicted in Figure 3, is a three-wheeled omnidirectional mobile robot. It has three omnidirectional wheels driven by EC 22 brushless Maxon motors connected to gearbox reducers with a ratio of 109:1. Each motor shaft speed and rotational direction is controlled by the Maxon DECS 50/5 driver. As a computational unit, the 32-bit Experimenter Kit Delfino F28335 Texas Instruments microcontroller is used.

The position and orientation of the mobile robot are measured by a Marvelmind Starter set HW 4.9 indoor GPS system, while every motor shaft velocity is obtained by Maxon encoders with a resolution of 512 counts per turn. The sensors have a frequency of 100 Hz and the sensors’ data are collected with a sampling time of 0.025 s.

3.2. Model of the Three-Wheeled Omnidirectional Mobile Robot

The state space representation of a three-wheeled omnidirectional mobile robot in global coordinates is taken from [23]; then, the motors’ dynamics are added such that the following state space model is derived:
\[ \begin{align*}
    x_1 &= x_4 \\
    x_2 &= x_5 \\
    x_3 &= x_6 \\
    x_4 &= a_1 x_4 - a_2 x_3 x_6 - b_1 \cos(x_3) v_1 - \sqrt{3} b_1 \sin(x_3) v_2 \\
    x_5 &= a_2 x_4 x_6 + a_1 x_3 - b_1 \sin(x_3) v_1 + \sqrt{3} b_1 \cos(x_3) v_2 \\
    x_6 &= a_3 x_6 + b_2 v_3 \\
    x_7 &= -\lambda_1 (x_7 - 18000 u_1 - Y \text{sign}(\Omega_1)) \\
    x_8 &= -\lambda_2 (x_8 - 18000 u_2 - Y \text{sign}(\Omega_2)) \\
    x_9 &= -\lambda_3 (x_9 - 18000 u_3 - Y \text{sign}(\Omega_3))
\end{align*} \]

where

\[
\begin{align*}
    v_1 &= \frac{x_7 - Y \text{sign}(\Omega_1)}{18000} + \frac{x_8 - Y \text{sign}(\Omega_2)}{18000} - \frac{2 x_9 - Y \text{sign}(\Omega_3)}{18000} \\
    v_2 &= \frac{x_7 - Y \text{sign}(\Omega_1)}{18000} - \frac{x_8 - Y \text{sign}(\Omega_2)}{18000} \\
    v_3 &= \frac{x_7 - Y \text{sign}(\Omega_1)}{18000} + \frac{x_8 - Y \text{sign}(\Omega_2)}{18000} + \frac{x_9 - Y \text{sign}(\Omega_3)}{18000}
\end{align*}
\]

the signals \( u_i \) are the inputs to the system, \( \Omega_i \) represent the shaft velocity of the \( i \)-th motor, \( i = 1, 2, 3 \); the states \( x_1, x_2, \) and \( x_3 \) are the positions at the coordinates (x, y), and the orientation \( \phi \) of the mobile robot, respectively. While \( x_4, x_5 \) are the mobile robot linear velocities \( v_x, v_y \), respectively, and \( x_6 \) is the angular velocity \( \dot{\phi} \) of the vehicle about its center of mass. The states \( x_7, x_8, \) and \( x_9 \) represent the motor velocities (RPM). Since real motors have fast dynamics, the parameters \( \lambda_1 = 100 \), \( \lambda_2 = 100 \), and \( \lambda_3 = 100 \) are chosen such that a fast convergence is achieved, whilst the parameter \( Y = 1000 \) (RPM) is selected according to the datasheet of the motors. The other system parameters are identified by using the least squares approach in [24]. The derived parameters are reported in Table 1.

**Table 1. Identified parameters of the system.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_1 )</td>
<td>-10.3949</td>
<td>( b_1 )</td>
<td>2.9695</td>
</tr>
<tr>
<td>( a_2 )</td>
<td>2.5026</td>
<td>( b_2 )</td>
<td>29.5783</td>
</tr>
<tr>
<td>( a_3 )</td>
<td>-15.2935</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.3. Signal Homogenizer

The system output vector is \( y = [y_1, y_2, y_3, y_4, y_5, y_6]^T \), where the GPS output signals \( y_1 = x, y_2 = y, \) and \( y_3 = \phi \), are the (X,Y) position in a global framework and the orientation \( \phi \) of the mobile robot, respectively; the encoders’ output signals are \( y_4 = \Omega_1, y_5 = \Omega_2, \) and \( y_6 = \Omega_3 \). Hence, the set of system outputs is \( \Sigma = \{y_1, y_2, y_3, y_4, y_5, y_6\} \). From the set \( \Sigma \), two subsets are generated: \( \Sigma_1 = \{y_1, y_2, y_3\} \), which contains the GPS outputs; and \( \Sigma_2 = \{y_4, y_5, y_6\} \) for the encoder’s outputs.

In order to generate the homogenizers \( P_1 : \Sigma_1 \rightarrow \Sigma \) and \( P_2 : \Sigma_2 \rightarrow \Sigma \), the structure depicted in Figure 2 is applied. For the homogenizer \( P_2 \), the coordinate transformation is chosen as \( \xi_1^{2h_4} = y_4, \xi_1^{2h_5} = y_5, \) and \( \xi_1^{2h_6} = y_6 \). So, the estimated input \( \hat{u}^2 \) (see Figure 2) is computed from Equation (2) as

\[
\hat{u}^2 = (E^2)^{-1} \left[ \begin{bmatrix}
\xi_1^{2h_4} \\
\xi_1^{2h_5} \\
\xi_1^{2h_6}
\end{bmatrix} - \begin{bmatrix}
-\lambda_1 (\xi_1^{2h_4} - Y \text{sign}(\Omega_1)) \\
-\lambda_2 (\xi_1^{2h_5} - Y \text{sign}(\Omega_2)) \\
-\lambda_3 (\xi_1^{2h_6} - Y \text{sign}(\Omega_3))
\end{bmatrix} \right]
\]
where $\hat{\xi}_1^2, \hat{\xi}_5^2, \hat{\xi}_6^2$ and their derivatives are estimated by using HOSM differentiators [16]. The matrix $E^2$ is given by

$$
E^2 = \begin{pmatrix}
18000\lambda_1 & 0 & 0 \\
0 & 18000\lambda_2 & 0 \\
0 & 0 & 18000\lambda_3
\end{pmatrix}
$$

(21)

Then, using the system model in Equation (19), $\hat{\Sigma}^2$, the virtual output associated with $\Sigma^2$ is computed. In a similar way, the homogenizer $P_1$ and $\hat{\Sigma}^1$ are computed.

3.4. Sensor Fusion

This subsection shows how to compute the merged output of the Kalman filter $\hat{\Sigma}^f_2$; a similar procedure should be used for $\hat{\Sigma}^f_1$.

At the sensor fusion stage, the entries of $\Sigma$ are smartly merged with $\hat{\Sigma}^2$ by using the EKF. The augmented measurement vector is

$$
\hat{y}_k^2 = [(y_k)\ T\ (\hat{y}_k^2)]^T
$$

(22)

where $y_k = [y_{1k} \ y_{2k} \ y_{3k} \ y_{4k} \ y_{5k} \ y_{6k}]^T$ are both the real GPS and encoder’s outputs, and $\hat{y}_k^2 = [\hat{y}_{1k}^2 \ \hat{y}_{2k}^2 \ \hat{y}_{3k}^2 \ \hat{y}_{4k}^2 \ \hat{y}_{5k}^2 \ \hat{y}_{6k}^2]^T$ are the virtual outputs taken from $\hat{\Sigma}^2$. Based on Equation (5), the augmented covariance of the measurement noise is

$$
\hat{R}_k^2 = \begin{pmatrix}
R^y_k & 0 \\
0 & R^{\Sigma^2}_k
\end{pmatrix}
$$

(23)

where

$$
R^y_k = \text{diag}(0.0517, 0.0138, 0.1035, 4237, 5269, 5408)
$$

(24)

$$
R^{\Sigma^2}_k = \text{diag}(0.0279, 0.0057, 0.0214, 27777, 27306, 39973)
$$

(25)

Note that the entries of Equation (23) are selected based on the variance noise, as well as the entries of the parameter $Q^2_k$, which in this case are chosen as

$$
Q^2_k = \text{diag}(0.005 \ 0.005 \ 0.005 \ 0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5)
$$

(26)

Then, the augmented measurement vector in Equation (22), the augmented measurement noise covariance matrix in Equation (23), and the process noise covariance matrix $Q_k^2$ in Equation (26) are used by the known EKF algorithm to produce the merged signal $\hat{\Sigma}^f_2$.

3.5. Filter

The signals from the sensor fusion stage are filtered by the second-order LPF described in Equation (6). The parameters of the filter are chosen as $\omega_n = 60 \text{ rad/s}$ and $\zeta = 0.8$, such that a good compromise between the noise reduction and the settling time of the filter is achieved. So, the derived filter is

$$
F_{\Sigma^2}(s) = \frac{40000}{s^2 + 320s + 40000}
$$

(27)

3.6. Voter

Finally, the voter selects $\Sigma^f_2$ and $\hat{x}^f_2$, since $\Sigma^f_2$ has the lowest noise variance. The obtained noise variances after the filter stage are reported in Table 2.
where the system. Then, the predefined-time observer is designed as in Equation (8):

\[
\dot{x}_m = -\lambda_1 (x_7 - 18000[y_1(t) - m_1(t)]) - Y\text{sign}(\Omega_1))
\]

\[
\dot{x}_8 = -\lambda_2 (x_8 - 18000[y_2(t) + m_2(t)]) - Y\text{sign}(\Omega_2))
\]

\[
\dot{x}_9 = -\lambda_3 (x_9 - 18000[y_3(t) + m_3(t)]) - Y\text{sign}(\Omega_3))
\]

where \(m(t) = [m_1 m_2 m_3]^T\) represents the actuator faults at the first, second, and third actuator, respectively, and \(u(t) = [u_1(t) u_2(t) u_3(t)]^T\) is the vector of the control inputs to the system. Then, the predefined-time observer is designed as in Equation (8):

\[
\dot{x}_7 = -\lambda_1 (x_7 - 18000u_1(t) - Y\text{sign}(\Omega_1)) + \Theta_1
\]

\[
\dot{x}_8 = -\lambda_2 (x_8 - 18000u_2(t) - Y\text{sign}(\Omega_2)) + \Theta_2
\]

\[
\dot{x}_9 = -\lambda_3 (x_9 - 18000u_3(t) - Y\text{sign}(\Omega_3)) + \Theta_3
\]

The observer estimation errors are \(e_1 = x_7 - \dot{x}_7, e_2 = x_8 - \dot{x}_8, \) and \(e_3 = x_9 - \dot{x}_9.\) In order to drive the estimations errors to zero in a predefined-time \(T_c,\) the feedbacks \(\Theta_1, \Theta_2,\) and \(\Theta_3\) are chosen, as in Proposition 1:

\[
\Theta_1 = \left[\frac{\gamma}{T_c} (a|e_1|^p + \beta|e_1|^q) + \xi_1]\text{sign}(e_1)
\]

\[
\Theta_2 = \left[\frac{\gamma}{T_c} (a|e_2|^p + \beta|e_2|^q) + \xi_2]\text{sign}(e_2)
\]

\[
\Theta_3 = \left[\frac{\gamma}{T_c} (a|e_3|^p + \beta|e_3|^q) + \xi_3]\text{sign}(e_3)
\]

The selected parameters are reported in Table 3. Now, the estimations of the fault magnitudes \(\hat{m}_1, \hat{m}_2,\) and \(\hat{m}_3\) are computed as in Equation (17):

\[
\begin{bmatrix}
\hat{m}_1 \\
\hat{m}_2 \\
\hat{m}_3
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{\lambda_11800} & 0 & 0 \\
0 & \frac{1}{\lambda_21800} & 0 \\
0 & 0 & \frac{1}{\lambda_31800}
\end{bmatrix}
\begin{bmatrix}
\Theta_1 \\
\Theta_2 \\
\Theta_3
\end{bmatrix}
\]

The filters to extract the desired information from the high-frequency signals \(\hat{m}_i,\) \(i = 1, 2, 3,\) are designed as in Equation (18), such that their settling time is \(t_s = 0.125 \text{ s}:

\[
F_{m_i}(s) = \frac{1600}{s^2 + 64s + 1600} \quad \forall i = 1, 2, 3.
\]
Table 3. Selected parameters for the diagnoser.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\gamma$</td>
<td>4.9968</td>
<td>$q$</td>
<td>3</td>
</tr>
<tr>
<td>$T_c$</td>
<td>0.15</td>
<td>$k$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1</td>
<td>$\zeta_1$</td>
<td>100000</td>
</tr>
<tr>
<td>$\beta$</td>
<td>2</td>
<td>$\zeta_2$</td>
<td>100000</td>
</tr>
<tr>
<td>$p$</td>
<td>1.5</td>
<td>$\zeta_3$</td>
<td>300000</td>
</tr>
</tbody>
</table>

4. Results

The fusion architecture proposed here is supplied with real data from a mobile robot and assessed using two diagnosers: one based on HOSM [25,26], and the other based on the proposed extension to a predefined-time observer.

Experimental Results

The architecture and the diagnosers are executed in Simulink, with the real data obtained from the prototype shown in Figure 3. The sampling time in Simulink is 0.001 s and the selected solver is ode3 (Bogacki–Shampine). The HOSM parameters are chosen as $\lambda = [3, 1.5, 1.1, 3, 1.5, 1.1, 3, 1.5, 1.1]^T$ and $L = [50,000, 50,000, 50,000]$, according to [15]. The chosen parameters for the proposed predefined-time diagnoser are reported in Table 3, while the selected parameters for the sensor fusion architecture are reported in Section 3. When the proposed sensor fusion architecture is not applied, the GPS signals are used by the diagnosers and the differentiator $\dot{z} = -\tau(z - y)$ is computed for the predefined-time diagnoser, where $\dot{z}$ is the estimated derivative of the signal $y$, and $\tau$ is a designing parameter which is chosen as $\tau = 5$.

The considered fault scenario consists of abrupt faults occurring either simultaneously or non-simultaneously at the actuators of the mobile robot. The detection fault threshold is $\pm 0.035$ and it is represented as a red line in the graphics of the experiments. The time intervals and magnitudes of fault occurrences are:

- Faults at the first actuator: $m_1 = 0.12$ during the interval [5 s, 8 s] and $m_1 = 0.7$ during the interval [22 s, 22.5 s].
- Faults at the second actuator: $m_2 = 0.8$ during the interval [10 s, 10.8 s] and $m_2 = 0.5$ during the interval [22 s, 23 s].
- Faults at the third actuator: $m_3 = -0.5$ during the interval [10 s, 12 s], $m_3 = -0.1$ during the interval [18 s, 18.5 s] and $m_3 = -0.12$ during the interval [22.5 s, 24 s].

Notice that the faults $m_1$ and $m_2$ are concurrent during the interval [22 s, 23 s], also $m_2$ and $m_3$ occur simultaneously during the interval [10 s, 10.8 s]. At the considered fault scenario, the faults correspond to an actuator loss of effectiveness of around 20%, 50%, and 70%. Also, the faults occur in small intervals of time such as 0.5 s. Although the system is affected by the faults during small intervals of time, Figure 4 shows that the faults cause a significant deviation from the fault-free trajectory.

Figures 5 and 6 present the fault detection using the proposed predefined-time diagnoser and the HOSM diagnoser reported in [26], respectively. In both figures, the simulations on the left correspond to the case where the proposed sensor fusion architecture is used, and on the right when it is not used. We highlight that the use of the proposed sensor fusion architecture is of major importance, in fact when this architecture is omitted, the graphics in the right columns of Figures 5 and 6 show a poor performance in fault detection, exacerbating the number of false fault detections. In addition, the predefined-time diagnoser estimates faults faster than the HOSM, it also reduces false fault detections. For example, HOSM (Figure 6) detects a false fault detection in actuator 3, when the estimated fault reaches the threshold of 0.035 at 6.45 s.
**Figure 4.** Mobile robot trajectory comparison.

**Figure 5.** The proposed diagnoser based on a predefined-time observer is used, with (left) and without (right) the proposed sensor fusion architecture.
Figure 6. A previously reported diagnoser based on an HOSM differentiator is used, with (left) and without (right) the proposed sensor fusion architecture.

Finally, Figure 7 shows that it is relevant to select the virtual sensors with the lowest variance. In fact, when $\bar{\Sigma}^1$ is used (the one with the biggest variance), the graphics in the left column of Figure 7 show a poor performance compared with the graphics in the right column, when $\bar{\Sigma}^2$ is used.

Table 4 reports the RMS fault identification errors of the diagnosers when the sensor fusion architecture is applied, while Table 5 reports the RMS errors when the sensor fusion architecture is not used. From the comparison of the two tables, the proposed diagnoser architecture provides the lowest fault estimation error. Moreover, the tables show that the sensor fusion architecture reduces the fault estimation errors of the considered diagnosers.

Table 4. Sensor fusion architecture RMS fault identification errors.

<table>
<thead>
<tr>
<th>Diagnoser</th>
<th>$\hat{m}_1$</th>
<th>$\hat{m}_2$</th>
<th>$\hat{m}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predefined-Time</td>
<td>0.0201</td>
<td>0.0147</td>
<td>0.0247</td>
</tr>
<tr>
<td>HOSM</td>
<td>0.0233</td>
<td>0.0166</td>
<td>0.0268</td>
</tr>
</tbody>
</table>

Table 5. RMS fault identification errors when the sensor fusion architecture is not applied.

<table>
<thead>
<tr>
<th>Diagnoser</th>
<th>$\hat{m}_1$</th>
<th>$\hat{m}_2$</th>
<th>$\hat{m}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predefined-Time</td>
<td>0.0822</td>
<td>0.0808</td>
<td>0.0636</td>
</tr>
<tr>
<td>HOSM</td>
<td>0.0421</td>
<td>0.0428</td>
<td>0.0463</td>
</tr>
</tbody>
</table>
5. Conclusions

A fusion sensor architecture is introduced, comprising a homogenizer, sensor fusion stage, and voter. It has a dual purpose: generating virtual sensors comparable to the system output and providing reliable estimated signals (state and system outputs) for diagnosers, controllers, etc. To test this architecture, a case study focusing on diagnosing faults in actuators of a mobile robot was conducted, utilizing two diagnosers. In both cases, the results demonstrate that employing the proposed architecture enhances the efficiency of the diagnosis process. Specifically, false positive/negative fault detections are drastically reduced, and diagnoser convergence time is minimized.

As mentioned above, the reported diagnoser requires the input observability property from the system. Additionally, using a Kalman filter demands fast CPUs for real-time applications. In some cases, these could be considered limitations to be overcome.

In future endeavors, this architecture will be applied in various domains, including controller design, and particle filters will be considered.


Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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


20. Hamadi, H.; Lussier, B.; Fantoni, I.; Francis, C. Data fusion fault tolerant strategy for a quadrotor UAV under sensors and software faults. *ISA Trans.* **2022**, *29*, 520–539. [CrossRef] [PubMed]


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