



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.ramirez@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



Citation: Begovich, O.; Lizárraga, A.; Ramírez-Treviño, A. Sensor Fusion Architecture for Fault Diagnosis with a Predefined-Time Observer. *Algorithms* **2024**, *17*, 270. <https://doi.org/10.3390/a17060270>

Academic Editor: Francesc Pozo

Received: 22 May 2024

Revised: 11 June 2024

Accepted: 15 June 2024

Published: 20 June 2024



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

1. Introduction

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 holds 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*.

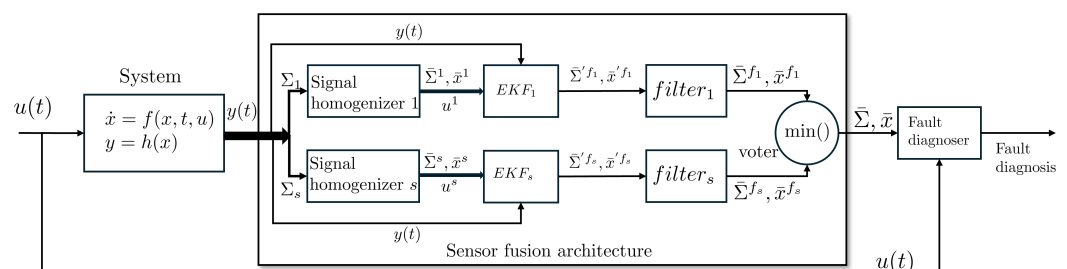


Figure 1. Sensor fusion architecture and the diagnoser applied in a diagnoser scheme. The sensor fusion architecture encompasses the homogenizer, EKF, and voter stages. The homogenizer generates virtual outputs, while the EKF merges real and virtual signals to generate more reliable signals, the filter reduces the noise from the signals, and the voter selects the best signal (in this work, the minimum variance signal). The diagnoser leverages these reliable signals to detect, locate, and identify concurrent and nonconcurrent faults.

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.

Since in this work the output of the proposed fusion architecture is used by a fault diagnoser, we considered the system model introduced in [14]:

$$\dot{x} = f(x) + g(x)u(t) + l(x)m(t), \quad y = h(x) \quad (1)$$

where the state is $x \in \mathbb{R}^n$, the control input is $u \in \mathbb{R}^c$, the output is $y \in \mathbb{R}^z$, and the fault magnitude is $m \in \mathbb{R}^d$, while $f(x) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^{n \times c}$, $l(x) \in \mathbb{R}^{n \times d}$, and $h(x) \in \mathbb{R}^z$ are smooth functions defined on an open set $\Omega \subset \mathbb{R}^n$.

Assumption 1. No one $m(t)$ belongs to the kernel of $l(x)$. In other words, any fault occurrence always affects the system state.

2.1. Sensor Fusion Architecture

The proposed sensor fusion architecture is composed of the *signal homogenizer*, *sensor fusion*, and *filter* stages.

2.1.1. Homogenizer

Modern systems are endowed with several sensors grouped into different classes depending on the type of measured signals. The information provided by sensors, although different in nature, may be redundant. For example, autonomous vehicles include cameras, GPS, lidar, encoders, among other sensors, that register different types of information. 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 $\Sigma = \{y_1, \dots, y_z\}$ be the set of system outputs, and $y = [y_1 \dots y_z]^T$ be the system output vector. Let $\Sigma_i, \Sigma_j \subset \Sigma$ be two arbitrary subsets of Σ . If there exists a function $P_{ij} : \Sigma_i \rightarrow \Sigma_j$ such that from the information provided by the signals in Σ_i , the signals in Σ_j may be estimated, then P_{ij} is the i -th homogenizer for Σ_j . The homogenizer value $\bar{\Sigma}_j^i = P_{ij}(\Sigma_i)$ is named the i -th virtual sensor of Σ_j . If $\Sigma_j = \Sigma$, then $\bar{\Sigma}^i = P_{ij}(\Sigma_i)$, i.e., the whole set of output sensors is reconstructed from Σ_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 Σ_i and Σ_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 Σ_i are used to estimate the system input \bar{u}^i . Afterwards, leveraging \bar{u}^i and the system model, the system outputs $\bar{\Sigma}^i$, due to Σ_i , are estimated, i.e., the proposed homogenizer is capable of computing all the signals in Σ . Thus, the homogenizer is represented as a function $P_i : \Sigma_i \rightarrow \Sigma$. As a notation $\bar{\Sigma}^i$ ($\bar{\Sigma}^i = P_i(\Sigma_i)$) is used to indicate that all system outputs are estimated from the information in Σ_i . In addition to $\bar{\Sigma}^i$, the proposed homogenizer also computes the estimated state \bar{x}^i and the estimated input \bar{u}^i , obtained from Σ_i .

In order to compute the input \bar{u}^i , first, the coordinate transformation

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

is applied [15], where $\zeta^i = [(\zeta^{ih_a})^T \dots (\zeta^{ih_w})^T]^T$ and η^i is the internal system dynamic. Notice that ζ^{ih_a} corresponds to the coordinate transformation obtained from the a -th output $h_a(x)$, with $h_a, \dots, h_w \in \Sigma_i$. Then, by the construction of the coordinate transformation, the

states ξ^i and their derivatives $\dot{\xi}^i = [(\dot{\xi}^{ih_a})^T \dots (\dot{\xi}^{ih_w})^T]^T$ can be estimated by higher-order sliding mode (HOSM) differentiators [16] (by computing the derivatives of the outputs of the system), where $\dot{\xi}^{ih_a} = [\dot{\xi}_1^{ih_a} \dots \dot{\xi}_{r_{h_a}}^{ih_a}]^T$, and r_{h_a} denotes the relative degree of the a -th output $h_a(x)$. Once the state ξ^i and its derivative $\dot{\xi}^i$ are estimated, the input \bar{u}^i is computed as reported in [15]:

$$\bar{u}^i = (E^i)^{-1} \left[\begin{pmatrix} \hat{\xi}^{ih_a} \\ \hat{\xi}^{r_{h_a}} \\ \hat{\xi}^{ih_b} \\ \hat{\xi}^{r_{h_b}} \\ \vdots \\ \hat{\xi}^{ih_w} \\ \hat{\xi}^{r_{h_w}} \end{pmatrix} - \begin{pmatrix} L_f^{r_{h_a}} h_a((\Phi^i)^{-1}) \\ L_f^{r_{h_b}} h_b((\Phi^i)^{-1}) \\ \vdots \\ L_f^{r_{h_w}} h_w((\Phi^i)^{-1}) \end{pmatrix} \right] \quad (2)$$

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]

$$E^i(x) = \begin{bmatrix} L_{g_1}(L_f^{(r_{h_a}-1)} h_a) & \dots & L_{g_w}(L_f^{(r_{h_a}-1)} h_a) \\ L_{g_1}(L_f^{(r_{h_b}-1)} h_b) & \dots & L_{g_w}(L_f^{(r_{h_b}-1)} h_b) \\ \vdots & \vdots & \vdots \\ L_{g_1}(L_f^{(r_{h_w}-1)} h_w) & \dots & L_{g_w}(L_f^{(r_{h_w}-1)} h_w) \end{bmatrix} \quad (3)$$

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.

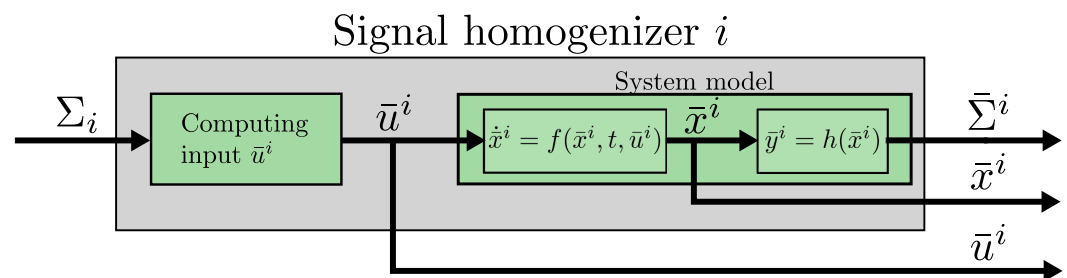


Figure 2. Implementation of the signal homogenizer i to obtain the virtual sensors $\bar{\Sigma}^i$, an estimate of the state \bar{x}^i , and an estimate for the input \bar{u}^i .

2.1.2. Sensor Fusion

The reliability of the values of the system signals in Σ 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_i) to obtain a more accurate value of signals in Σ [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]:

$$\bar{y}_k^i = [(y_k)^T (y_k^{\bar{\Sigma}^i})^T]^T \quad (4)$$

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

$$\bar{R}_k^i = \begin{pmatrix} R_k^y & 0 \\ 0 & R_k^{y^{\bar{\Sigma}^i}} \end{pmatrix} \quad (5)$$

where R_k^y is the covariance matrix of the noise of the real sensors' measurements and $R_k^{\Sigma^i}$ 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^i are chosen. Once the parameters of the EKF_i 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:

$$F_{\Sigma^i}(s) = \frac{\omega_{n_i}^2}{s^2 + 2\zeta_i\omega_{n_i}s + \omega_{n_i}^2} \quad (6)$$

The parameters ζ_i and ω_{n_i} 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 $\bar{\Sigma} = \bar{\Sigma}^{f_i}$ and $\bar{x} = \bar{x}^{f_i}$, 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

$$\begin{bmatrix} \dot{\bar{x}}_m \\ \dot{\bar{x}}_{\bar{m}} \end{bmatrix} = \begin{bmatrix} f_m(\bar{x}) \\ f_{\bar{m}}(\bar{x}) \end{bmatrix} + \begin{bmatrix} g_m(\bar{x}) \\ g_{\bar{m}}(\bar{x}) \end{bmatrix} u(t) + \begin{bmatrix} l_m(\bar{x}) \\ l_{\bar{m}}(\bar{x}) \end{bmatrix} m(t) \quad (7)$$

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

$$\dot{\hat{x}}_m = f_m(\bar{x}) + g_m(\bar{x})u(t) + \Theta \quad (8)$$

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

$$e = \bar{x}_m - \hat{x}_m \quad (9)$$

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

$$\dot{e} = l_m(\bar{x})m(t) - \Theta \quad (10)$$

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

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

$$\dot{x}(t) = f(x, t; \rho) \quad (11)$$

where the system state is $x \in \mathbb{R}^n$, the system parameters are $p \in \mathbb{R}^b$, which are assumed to be constant, and $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is assumed to be a continuous nonlinear function. If there exists a continuous definite-positive radially unbounded function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ such that

$$\dot{V}(x) \leq -\frac{\gamma}{T_c}(\alpha V(x)^p + \beta V(x)^q)^\kappa, \quad \forall x \in \mathbb{R}^n \setminus \{0\} \quad (12)$$

with constants $\beta, \alpha, p, q, \kappa > 0$, $\kappa p < 1$, and $\kappa q > 1$, and the constant γ is given by

$$\gamma = \frac{\Gamma(m_p)\Gamma(m_q)}{\alpha^\kappa\Gamma(\kappa)(q-p)}\left(\frac{\alpha}{\beta}\right)^{m_p} \quad (13)$$

where Γ is the gamma function [22], $m_p = \frac{1-\kappa p}{q-p}$, and $m_q = \frac{\kappa q-1}{q-p}$.

Then, the origin of (11) is predefined-time stable with T_c as its predefined time.

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

$$\Theta_i = \left[\frac{\gamma}{T_c}(\alpha|e_i|^p + \beta|e_i|^q)^k + \xi_i\right]\text{sign}(e_i), \quad i = 1, \dots, d \quad (14)$$

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 $V(e_i) = |e_i|$, $i = 1, \dots, d$. The time derivative of each Lyapunov function $V(e_i)$ yields

$$\begin{aligned} \dot{V}(e_i) &= \text{sign}(e_i)[\Delta_i - \Theta_i] \\ &= -\frac{\gamma}{T_c}(\alpha|e_i|^p + \beta|e_i|^q)^k - \xi_i + \Delta_i\text{sign}(e_i) \\ &\leq -\frac{\gamma}{T_c}(\alpha V(e_i)^p + \beta V(e_i)^q)^k - \xi_i + |\Delta_i\text{sign}(e_i)| \\ &= -\frac{\gamma}{T_c}(\alpha V(e_i)^p + \beta V(e_i)^q)^k - (\xi_i - |\Delta_i\text{sign}(e_i)|) \\ &\leq -\frac{\gamma}{T_c}(\alpha V(e_i)^p + \beta V(e_i)^q)^k \end{aligned} \quad (15)$$

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

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 Θ_{eq} is obtained such that $\dot{e} = 0$ holds. Hence, from Equation (10), Θ_{eq} is found as

$$\begin{aligned} 0 &= l_m(\bar{x})m(t) - \Theta_{eq} \\ \Theta_{eq} &= l_m(\bar{x})m(t) \end{aligned} \quad (16)$$

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

$$\hat{m}(t) = l_m^{-1}(\bar{x})\Theta_{eq} \quad (17)$$

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 Θ_{eq} , second-order LPFs, such as the ones proposed in Section 2.1.3, are required:

$$F_{m_i}(s) = \frac{\omega_{nm_i}^2}{s^2 + 2\zeta_{m_i}\omega_{nm_i}s + \omega_{nm_i}^2}, \quad i = 1, \dots, d \quad (18)$$

where the filter settling time $t_{sm_i} = \frac{4}{\zeta_{m_i} \omega_{nm_i}}$, $i = 1, \dots, d$ 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.

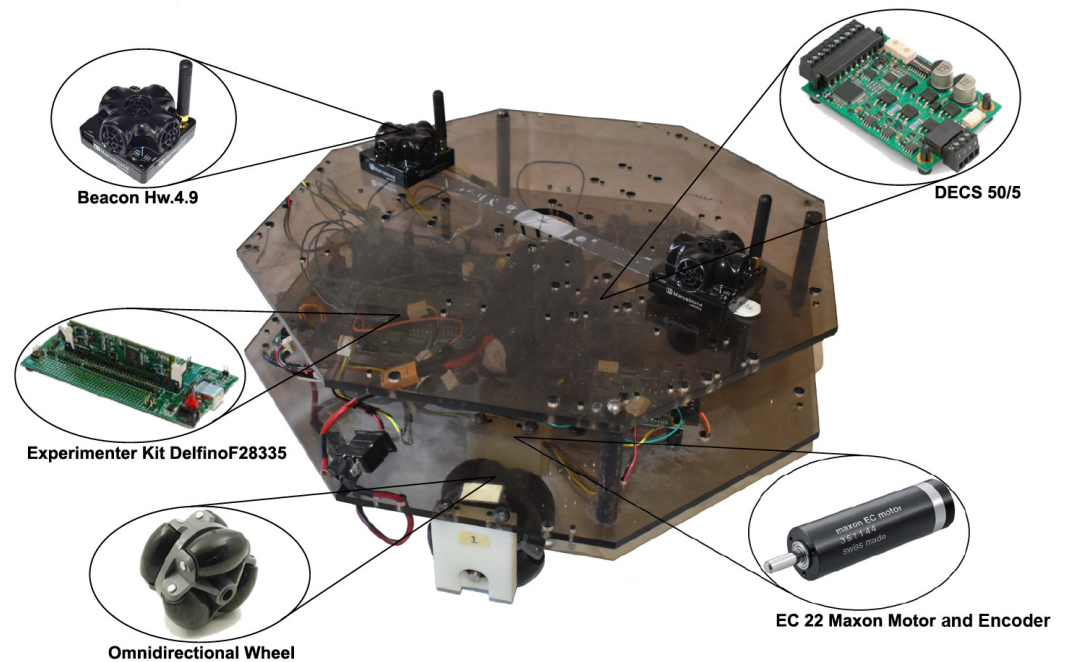


Figure 3. Three-wheeled omnidirectional mobile robot.

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:

$$\begin{aligned}
\dot{x}_1 &= x_4 \\
\dot{x}_2 &= x_5 \\
\dot{x}_3 &= x_6 \\
\dot{x}_4 &= a_1 x_4 - a_2 x_5 x_6 - b_1 \cos(x_3) v_1 - \sqrt{3} b_1 \sin(x_3) v_2 \\
\dot{x}_5 &= a_2 x_4 x_6 + a_1 x_5 - b_1 \sin(x_3) v_1 + \sqrt{3} b_1 \cos(x_3) v_2 \\
\dot{x}_6 &= a_3 x_6 + b_2 v_3 \\
\dot{x}_7 &= -\lambda_1 (x_7 - 18000 u_1 - Y \text{sign}(\Omega_1)) \\
\dot{x}_8 &= -\lambda_2 (x_8 - 18000 u_2 - Y \text{sign}(\Omega_2)) \\
\dot{x}_9 &= -\lambda_3 (x_9 - 18000 u_3 - Y \text{sign}(\Omega_3))
\end{aligned} \tag{19}$$

where

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

the signals u_i are the inputs to the system, Ω_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 φ 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 $\dot{\varphi}$ 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.

Parameter	Value	Parameter	Value
a_1	−10.3949	b_1	2.9695
a_2	2.5026	b_2	29.5783
a_3	−15.2935		

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 $y_3 = \varphi$, are the (X, Y) position in a global framework and the orientation φ 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 Σ , 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 $\tilde{\zeta}_1^{2h_4} = y_4, \tilde{\zeta}_1^{2h_5} = y_5$, and $\tilde{\zeta}_1^{2h_6} = y_6$. So, the estimated input \tilde{u}^2 (see Figure 2) is computed from Equation (2) as

$$\tilde{u}^2 = (E^2)^{-1} \left[\begin{pmatrix} \dot{\tilde{\zeta}}_1^{2h_4} \\ \dot{\tilde{\zeta}}_1^{2h_5} \\ \dot{\tilde{\zeta}}_1^{2h_6} \end{pmatrix} - \begin{pmatrix} -\lambda_1 (\tilde{\zeta}_1^{2h_4} - Y \text{sign}(\Omega_1)) \\ -\lambda_2 (\tilde{\zeta}_1^{2h_5} - Y \text{sign}(\Omega_2)) \\ -\lambda_3 (\tilde{\zeta}_1^{2h_6} - Y \text{sign}(\Omega_3)) \end{pmatrix} \right] \tag{20}$$

where $\hat{\xi}_1^{2h_4}, \hat{\xi}_1^{2h_5}, \hat{\xi}_1^{2h_6}$ and their derivatives are estimated by using HOSM differentiators [16]. The matrix E^2 is given by

$$E^2 = \begin{pmatrix} 18000\lambda_1 & 0 & 0 \\ 0 & 18000\lambda_2 & 0 \\ 0 & 0 & 18000\lambda_3 \end{pmatrix} \quad (21)$$

Then, using the system model in Equation (19), $\bar{\Sigma}^2$, the virtual output associated with Σ_2 is computed. In a similar way, the homogenizer P_1 and $\bar{\Sigma}^1$ are computed.

3.4. Sensor Fusion

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

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

$$\bar{y}_k^2 = [(y_k)^T (y_k^{\bar{\Sigma}^2})^T]^T \quad (22)$$

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 $y_k^{\bar{\Sigma}^2} = [\bar{y}_{1k}^2 \ \bar{y}_{2k}^2 \ \bar{y}_{3k}^2 \ \bar{y}_{4k}^2 \ \bar{y}_{5k}^2 \ \bar{y}_{6k}^2]^T$ are the virtual outputs taken from $\bar{\Sigma}^2$. Based on Equation (5), the augmented covariance of the measurement noise is

$$\bar{R}_k^2 = \begin{pmatrix} R_k^y & 0 \\ 0 & R_k^{\bar{\Sigma}^2} \end{pmatrix} \quad (23)$$

where

$$R_k^y = \text{diag}(0.0517, 0.0138, 0.1035, 4237, 5269, 5408) \quad (24)$$

$$R_k^{\bar{\Sigma}^2} = \text{diag}(0.0279, 0.0057, 0.0214, 27777, 27306, 39973) \quad (25)$$

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

$$Q_k^2 = \text{diag}(0.005 \ 0.005 \ 0.005 \ 0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5 \ 0.5) \quad (26)$$

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 $\bar{\Sigma}^{f_2}$.

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 $\omega_{n_2} = 60$ rad/s and $\zeta_2 = 0.8$, such that a good compromise between the noise reduction and the settling time of the filter is achieved. So, the derived filter is

$$F_{\Sigma^2}(s) = \frac{40000}{s^2 + 320s + 40000} \quad (27)$$

3.6. Voter

Finally, the voter selects $\bar{\Sigma}^{f_2}$ and \bar{x}^{f_2} , since $\bar{\Sigma}^{f_2}$ has the lowest noise variance. The obtained noise variances after the filter stage are reported in Table 2.

Table 2. Noise variance comparison.

$\bar{\Sigma}^{f_1}$ (GPS)	Variance	$\bar{\Sigma}^{f_2}$ (Encoders)	Variance
$\bar{\Sigma}_1^{f_1}$ (x)	0.0772	$\bar{\Sigma}_1^{f_2}$ (x)	0.0353
$\bar{\Sigma}_2^{f_1}$ (y)	0.0173	$\bar{\Sigma}_2^{f_2}$ (y)	0.0076
$\bar{\Sigma}_3^{f_1}$ (φ)	0.1245	$\bar{\Sigma}_3^{f_2}$ (φ)	0.0308
$\bar{\Sigma}_4^{f_1}$ (Ω_1)	327,610	$\bar{\Sigma}_4^{f_2}$ (Ω_1)	26,072
$\bar{\Sigma}_5^{f_1}$ (Ω_2)	333,730	$\bar{\Sigma}_5^{f_2}$ (Ω_2)	25,752
$\bar{\Sigma}_6^{f_1}$ (Ω_3)	582,470	$\bar{\Sigma}_6^{f_2}$ (Ω_3)	38,149
$\bar{x}_4^{f_1}$ (\dot{x})	0.0022	$\bar{x}_4^{f_2}$ (\dot{x})	0.0005
$\bar{x}_5^{f_1}$ (\dot{y})	0.0018	$\bar{x}_5^{f_2}$ (\dot{y})	0.0003
$\bar{x}_6^{f_1}$ ($\dot{\varphi}$)	0.0083	$\bar{x}_6^{f_2}$ ($\dot{\varphi}$)	0.0024

3.7. Diagnoser

Here, the predefined-time observer described in Section 2.2 is designed. First, the states of the system in Equation (19) are separated as follows: $\bar{x}_m = [\bar{x}_7 \ \bar{x}_8 \ \bar{x}_9]^T$ and $\bar{x}_{\bar{m}} = [\bar{x}_1 \ \bar{x}_2 \ \bar{x}_3 \ \bar{x}_4 \ \bar{x}_5 \ \bar{x}_6]^T$. So, the dynamics of the states \bar{x}_m are

$$\begin{aligned}\dot{\bar{x}}_7 &= -\lambda_1(\bar{x}_7 - 18000[u_1(t) + m_1(t)] - Y\text{sign}(\Omega_1)) \\ \dot{\bar{x}}_8 &= -\lambda_2(\bar{x}_8 - 18000[u_2(t) + m_2(t)] - Y\text{sign}(\Omega_2)) \\ \dot{\bar{x}}_9 &= -\lambda_3(\bar{x}_9 - 18000[u_3(t) + m_3(t)] - Y\text{sign}(\Omega_3))\end{aligned}\quad (28)$$

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):

$$\begin{aligned}\dot{\hat{x}}_7 &= -\lambda_1(\bar{x}_7 - 18000u_1(t) - Y\text{sign}(\Omega_1)) + \Theta_1 \\ \dot{\hat{x}}_8 &= -\lambda_2(\bar{x}_8 - 18000u_2(t) - Y\text{sign}(\Omega_2)) + \Theta_2 \\ \dot{\hat{x}}_9 &= -\lambda_3(\bar{x}_9 - 18000u_3(t) - Y\text{sign}(\Omega_3)) + \Theta_3\end{aligned}\quad (29)$$

The observer estimation errors are $e_1 = \bar{x}_7 - \hat{x}_7$, $e_2 = \bar{x}_8 - \hat{x}_8$, and $e_3 = \bar{x}_9 - \hat{x}_9$. In order to drive the estimations errors to zero in a predefined-time T_c , the feedbacks Θ_1 , Θ_2 , and Θ_3 are chosen, as in Proposition 1:

$$\begin{aligned}\Theta_1 &= \left[\frac{\gamma}{T_c}(\alpha|e_1|^p + \beta|e_1|^q) + \zeta_1\right]\text{sign}(e_1) \\ \Theta_2 &= \left[\frac{\gamma}{T_c}(\alpha|e_2|^p + \beta|e_2|^q) + \zeta_2\right]\text{sign}(e_2) \\ \Theta_3 &= \left[\frac{\gamma}{T_c}(\alpha|e_3|^p + \beta|e_3|^q) + \zeta_3\right]\text{sign}(e_3)\end{aligned}\quad (30)$$

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):

$$\begin{bmatrix} \hat{m}_1 \\ \hat{m}_2 \\ \hat{m}_3 \end{bmatrix} = \begin{pmatrix} \frac{1}{\lambda_1 18000} & 0 & 0 \\ 0 & \frac{1}{\lambda_2 18000} & 0 \\ 0 & 0 & \frac{1}{\lambda_3 18000} \end{pmatrix} \begin{bmatrix} \Theta_{1eq} \\ \Theta_{2eq} \\ \Theta_{3eq} \end{bmatrix}\quad (31)$$

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 $t_s = 0.125$ s:

$$F_{m_i}(s) = \frac{1600}{s^2 + 64s + 1600} \quad \forall i = 1, 2, 3. \quad (32)$$

Table 3. Selected parameters for the diagnoser.

Parameter	Value	Parameter	Value
γ	4.9968	q	3
T_c	0.15	k	0.5
α	1	ξ_1	100000
β	2	ξ_2	100000
p	1.5	ξ_3	300000

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 τ 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 ± 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.

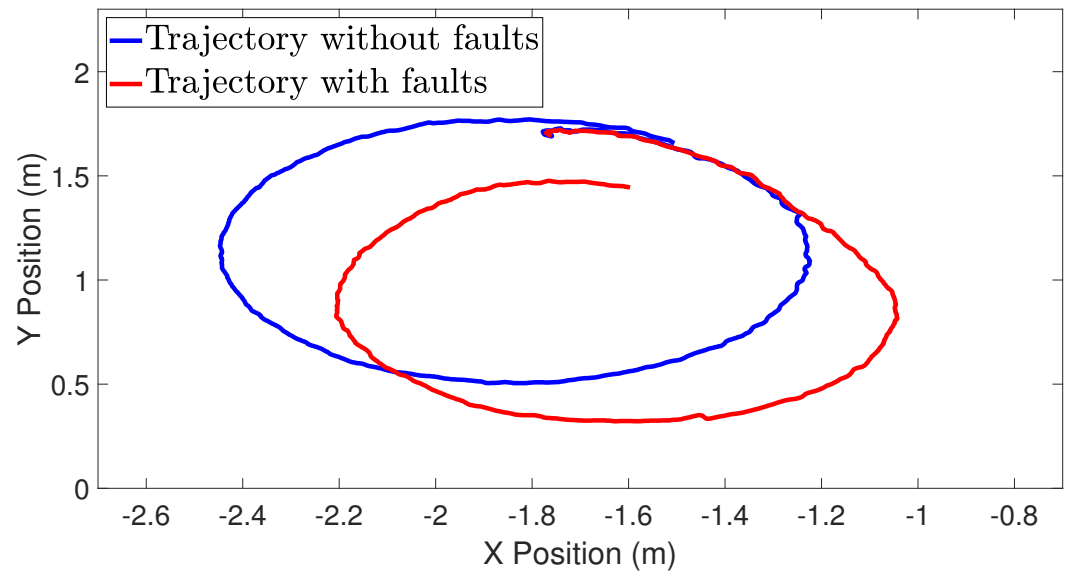


Figure 4. Mobile robot trajectory comparison.

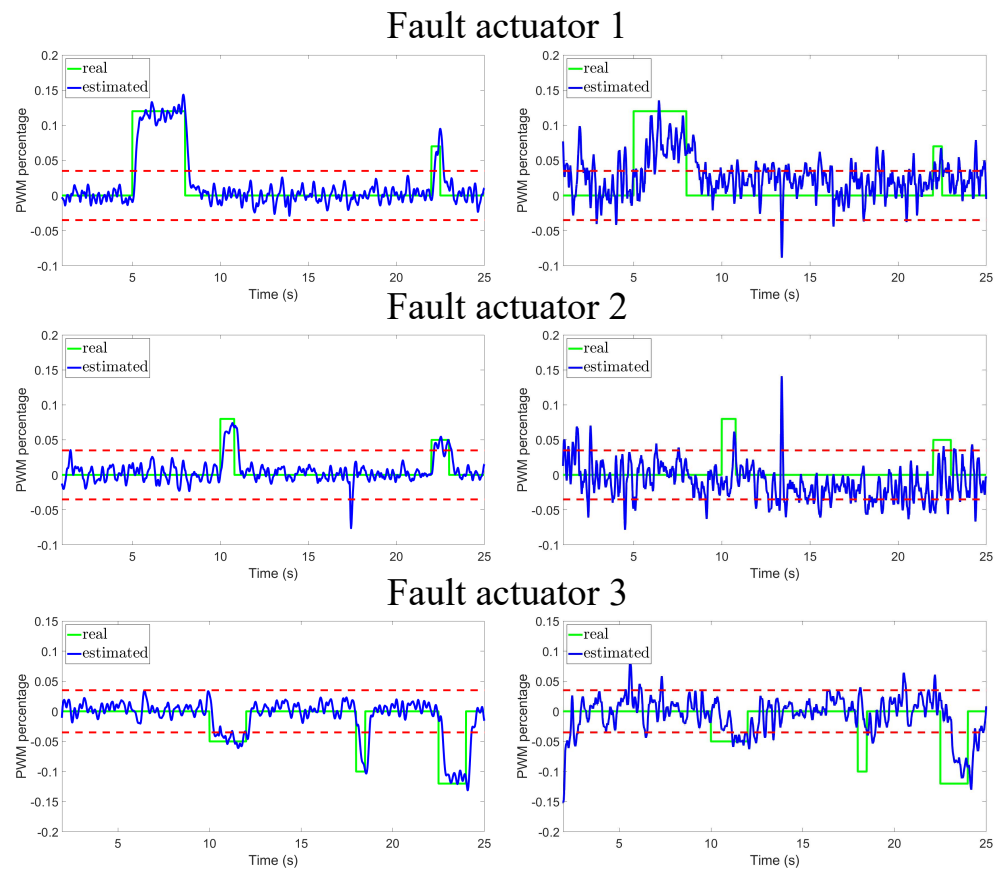


Figure 5. The proposed diagnoser based on a predefined-time observer is used, with (left) and without (right) the proposed sensor fusion architecture.

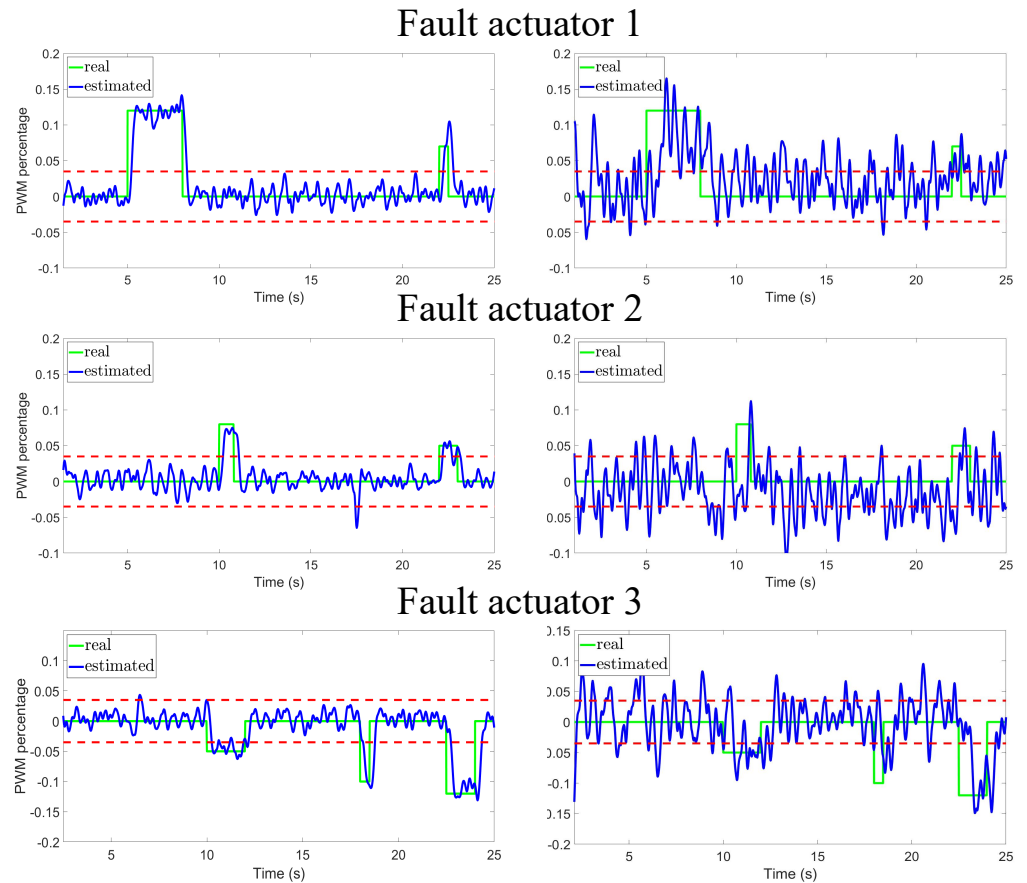


Figure 6. A previously reported diagnoser based on an HOSM differentiator is used, with (left) and without (right) the proposed sensor fusion architecture.

Finally, Figure 7 shows that it is relevant to select the virtual sensors with the lowest variance. In fact, when Σ^{f_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 Σ^{f_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.

Diagnoser	\hat{m}_1	\hat{m}_2	\hat{m}_3
Predefined-Time	0.0201	0.0147	0.0247
HOSM	0.0233	0.0166	0.0268

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

Diagnoser	\hat{m}_1	\hat{m}_2	\hat{m}_3
Predefined-Time	0.0822	0.0808	0.0636
HOSM	0.0421	0.0428	0.0463

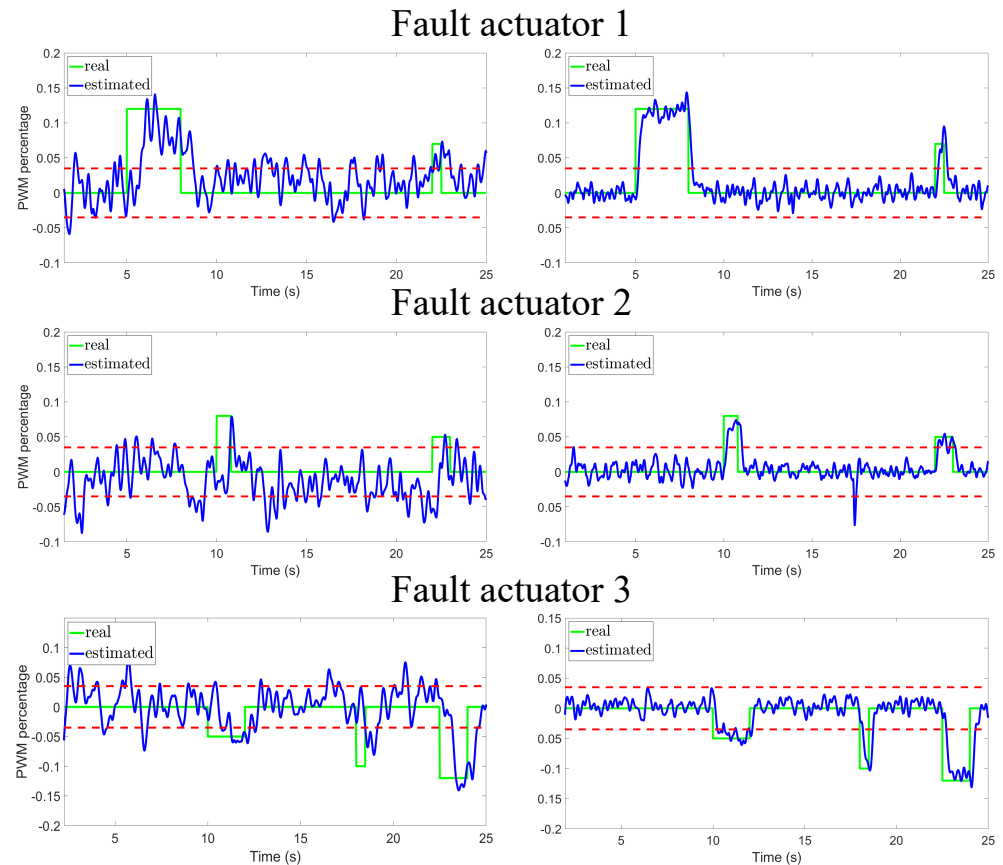


Figure 7. Comparison of the diagnosis with Σ^1 (left) and Σ^2 (right).

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.

Author Contributions: Conceptualization, O.B., A.L. and A.R.-T.; methodology, O.B., A.L. and A.R.-T.; software, A.L.; validation, O.B., A.L. and A.R.-T.; formal analysis, O.B., A.L. and A.R.-T.; investigation, O.B., A.L. and A.R.-T.; writing—original draft preparation, O.B., A.L. and A.R.-T.; writing—review and editing, O.B., A.L. and A.R.-T. All authors have read and agreed to the published version of the manuscript.

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

1. Flores-León, H.; Begovich, O.; Ruiz-León, J.; Ramírez-Treviño, A. Active fault detection and isolation in linear time-invariant systems: A geometric approach. *Asian J. Control* **2023**, *25*, 710–721. [\[CrossRef\]](#)
2. Chen, J.; Patton, R.J. *Robust Model-Based Fault Diagnosis for Dynamic Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 3.
3. Delgado-Aguíñaga, J.; Besancon, G.; Begovich, O.; Carvajal, J. Multi-leak diagnosis in pipelines based on Extended Kalman Filter. *Control Eng. Pract.* **2016**, *49*, 139–148. [\[CrossRef\]](#)
4. Lizarraga, A.; Begovich, O.; Ramírez, A. Real-Time Sliding Mode Fault Diagnosis for a Three-Wheeled Omnidirectional Mobile Robot. In Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), Mexico City, Mexico, 20–24 August 2022; IEEE: New York, NY, USA, 2022; pp. 1953–1958.
5. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques-Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3757–3767. [\[CrossRef\]](#)
6. Zhou, D.; Zhao, Y.; Wang, Z.; He, X.; Gao, M. Review on diagnosis techniques for intermittent faults in dynamic systems. *IEEE Trans. Ind. Electron.* **2019**, *67*, 2337–2347. [\[CrossRef\]](#)
7. Delgado-Aguíñaga, J.A.; Begovich, O. Water leak diagnosis in pressurized pipelines: A real case study. In *Modeling and Monitoring of Pipelines and Networks*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 235–262.
8. Koch, W. *Tracking and Sensor Data Fusion*; Springer: Berlin/Heidelberg, Germany, 2016.
9. Garcia-Huerta, R.A.; González-Jiménez, L.E.; Villalon-Turrubiates, I.E. Sensor Fusion Algorithm Using a Model-Based Kalman Filter for the Position and Attitude Estimation of Precision Aerial Delivery Systems. *Sensors* **2020**, *20*, 5227. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Bader, K.; Lussier, B.; Schön, W. A fault tolerant architecture for data fusion: A real application of Kalman filters for mobile robot localization. *Robot. Auton. Syst.* **2017**, *88*, 11–23. [\[CrossRef\]](#)
11. Hachemi, L.; Guatni, M.; Nemra, A. Fault diagnosis and reconfiguration for mobile robot localization based on multi-sensors data fusion. *Unmanned Syst.* **2022**, *10*, 69–91. [\[CrossRef\]](#)
12. Ifqir, S.; Combastel, C.; Zolghadri, A.; Alcalay, G.; Goupil, P.; Merlet, S. Fault tolerant multi-sensor data fusion for autonomous navigation in future civil aviation operations. *Control Eng. Pract.* **2022**, *123*, 105132. [\[CrossRef\]](#)
13. Abid, A.; Khan, M.T. Multi-sensor, multi-level data fusion and behavioral analysis based fault detection and isolation in mobile robots. In Proceedings of the 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, USA, 3–5 October, 2017; IEEE: New York, NY, USA, 2017; pp. 40–45.
14. Zhang, Z.; He, X. Geometric Approach Based Fault Detection and Isolation: A Review. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; IEEE: New York, NY, USA, 2022; pp. 4136–4141.
15. Fridman, L.; Shtessel, Y.; Edwards, C.; Yan, X.G. Higher-order sliding-mode observer for state estimation and input reconstruction in nonlinear systems. *Int. J. Robust Nonlinear Control IFAC-Affil. J.* **2008**, *18*, 399–412. [\[CrossRef\]](#)
16. Shtessel, Y.; Edwards, C.; Fridman, L.; Levant, A. *Sliding Mode Control and Observation*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 10.
17. Isidori, A. *Nonlinear Control Systems*; Springer: Berlin/Heidelberg, Germany, 1995.
18. Ma, H.; Yan, L.; Xia, Y.; Fu, M. *Kalman Filtering and Information Fusion*; Springer: Berlin/Heidelberg, Germany, 2020.
19. Gao, J.; Harris, C. Some remarks on Kalman filters for the multisensor fusion. *Inf. Fusion* **2002**, *3*, 191–201. [\[CrossRef\]](#)
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. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Aldana-López, R.; Gómez-Gutiérrez, D.; Jiménez-Rodríguez, E.; Sánchez-Torres, J.D.; Defoort, M. Enhancing the settling time estimation of a class of fixed-time stable systems. *Int. J. Robust Nonlinear Control* **2019**, *29*, 4135–4148. [\[CrossRef\]](#)
22. Trujillo, M.A.; Aldana-López, R.; Gómez-Gutiérrez, D.; Defoort, M.; Ruiz-León, J.; Becerra, H. Autonomous and non-autonomous fixed-time leader–follower consensus for second-order multi-agent systems. *Nonlinear Dyn.* **2020**, *102*, 2669–2686. [\[CrossRef\]](#)
23. Tzafestas, S.G. *Introduction to Mobile Robot Control*; Elsevier: Amsterdam, The Netherlands, 2013.
24. Nguyen, N.T. Model-reference adaptive control. In *Model-Reference Adaptive Control*; Springer: Berlin/Heidelberg, Germany, 2018; Chapter 6, pp. 126–127.
25. Iqbal, M.; Bhatti, A.; Iqbal, S.; Khan, Q. Parameter estimation based fault diagnosis of uncertain nonlinear three tank system using HOSM differentiator observer. In Proceedings of the 2009 IEEE 13th International Multitopic Conference, Islamabad, Pakistan, 14–15 December 2009; IEEE: New York, NY, USA, 2009; pp. 1–6.
26. Iqbal, M.; Bhatti, A.I.; Ayubi, S.I.; Khan, Q. Robust parameter estimation of nonlinear systems using sliding-mode differentiator observer. *IEEE Trans. Ind. Electron.* **2010**, *58*, 680–689. [\[CrossRef\]](#)

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