electronics

Article

Inversion-Based Approach for Detection and Isolation of Faults in Switched Linear Systems

Alexandre Miguel Silveira [1,*] and Rui Esteves Araújo [2]

[1] Superior Institute of Engineering of Porto, Polytechnic Institute of Porto, 4249-015 Porto, Portugal
[2] INESC TEC and Faculty of Engineering, University of Porto, 4200-465 Porto, Portugal; raraujo@fe.up.pt
[*] Correspondence: asi@isep.ipp.pt; Tel.: +351-228-340-500

Received: 25 January 2020; Accepted: 23 March 2020; Published: 27 March 2020

Abstract: This paper addresses the problem of the left inversion of switched linear systems from a diagnostics perspective. The problem of left inversion is to reconstruct the input of a system with the knowledge of its output, whose differentiation is usually required. In the case of this work, the objective is to reconstruct the system's unknown inputs, based on the knowledge of its outputs, switching sequence and known inputs. With the inverse model of the switched linear system, a real-time Fault Detection and Isolation (FDI) algorithm with an integrated Fuzzy Logic System (FLS) that is capable of detecting and isolating abrupt faults occurring in the system is developed. In order to attenuate the effects of unknown disturbances and noise at the output of the inverse model, a smoothing strategy is also used. The results are illustrated with an example. The performance of the method is validated experimentally in a DC-DC boost converter, using a low-cost microcontroller, without any additional components.

Keywords: inversion problems; fault diagnosis; DC-DC converters

1. Introduction

The problem of linear systems inversion is a concept that gathered great interest among the control engineering community since the middle of last century. The inversion of Multiple-Input Multiple-Output (MIMO) state-space systems was principally driven by Reference [1,2] which gave, with their contributions, a lot of insight into the comprehension of the properties of inverse systems. The former, realized with his so-called *structure algorithm*, that an inverse system can be achieved with the same number of differentiators (or delay elements) as the original system. The latter studied principally left inversions and introduced the concept of inherent integration (delay) of continuous (discrete) Linear Time-Invariant (LTI) systems.

Some years later, Reference [3] provided necessary and sufficient conditions for the stable invertibility and introduced the concepts of left and right inverses. In fact, regarding its application, the problem of inversion can be divided into two classes—(*i*) right inversion; and (*ii*) left inversion. While right inversion deals with finding inputs and initial conditions that produce a given output, left inversion is related with the problem of recovering unknown inputs applied to the system from the knowledge of its outputs. As stated by Reference [3], a left inverse for a system $\boldsymbol{\Sigma}$ is a system $\boldsymbol{\Sigma}^{\mathsf{L}}$ which computes the input to $\boldsymbol{\Sigma}$ from the knowledge of its output, such that $\boldsymbol{\Sigma}^{\mathsf{L}}\boldsymbol{\Sigma} = \boldsymbol{I}_m$. One can verify if a system is left invertible by the following definition:

Definition 1 (Left Invertibility)**.** *Let u_1 and u_2 be any two inputs to system $\boldsymbol{\Sigma}$ and let y_1 and y_2 be the corresponding outputs for the same initial state. System $\boldsymbol{\Sigma}$ is said to be left invertible if $y_1(t) = y_2(t)$ for all $t \geq 0$ implies that $u_1(t) = u_2(t)$ for all $t \geq 0$.*

Whilst right inversion deals typically with reference tracking in control applications, an important domain of application for the left inversion problem is the one related to model-based Fault Detection and Isolation (FDI) [4,5].

Hence, the inversion of linear systems plays an important role not only in the areas of digital signal processing [6] and control systems [7,8], but also in the area of fault diagnosis [9–12]. In these latter works, it were explored different approaches to take advantage of the concept of system inversion to tackle the problem of FDI, specially in the linear case. Although there is a close link between input observability and system inversion, the first is preferred when addressing the problem of input reconstruction [13]. Nevertheless, the concept of left inversion has been widely used by the scientific community for unknown input reconstruction [8,10,12–15]. Basically, the idea is to model faults as unknown inputs and to invert the system to reconstruct them.

Since power electronics converters are embedded in almost every modern electric and electronic application, there were published some works related to inversion of Switched Linear Systems (SLSs). The problem of invertibility of SLSs was introduced by Reference [16], where it was addressed the recovery of the switching signal and input from the knowledge of the output and initial state. In their work, they gave the necessary and sufficient conditions for a switched system to be invertible, based on the methodology of inversion proposed by Reference [1] for LTI systems. This algorithm was already adapted for the purpose of accomplish FDI for the case of LTI systems [17,18] and also for the case of switched linear systems using state-space approaches [19]. Giving some continuity to his work, Reference [20] addressed the invertibility problem for switched nonlinear systems. Both of these works were developed from a perspective of control.

Regarding the approach proposed by Reference [2], there are only a few works using it with the purpose of FDI, for example, References [21,22]. Hence, the goal of this paper is twofold. Firstly, the article provides a detailed perspective on the state-space inversion approach, developed based on the work of Sain and Massey [2], for researchers who are familiar with FDI, showing the mathematical explanations and formal proofs. Secondly, the problem of inversion of Switched Linear Systems is introduced to a wide audience, guiding practicing engineers in the field to a new solution to their FDI problems for SLSs.

2. General Setting of the FDI Approach and Previous Results

In this section, we set up the problem and review our previous results. As referred before, the concept of left inversion is close related to fault diagnosis, since it permits to reconstruct the input of the original system from the knowledge of its output. Consequently, the concept can be adapted to reconstruct faults modeled as unknown inputs. Henceforth, the Sain and Massey's [2] state-space inversion approach will be adapted to accommodate faults modeled as unknown inputs, which one are interested in recover. Moreover, this state-space mathematical description will be performed in discrete-time, since it has more advantages regarding its implementation in digital controllers. Consequently, the usual state-space description is replaced by a new one, where it is included an additional unknown input vector and its corresponding matrix.

Consider a system or process described by the following generalized discrete-time state-space dynamical model $\boldsymbol{\Gamma}$, with $\boldsymbol{x}_0 = \boldsymbol{x}(0) = \boldsymbol{0}$:

$$\boldsymbol{\Gamma} = \begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{F}\boldsymbol{\varphi}_k & \text{(1a)} \\ \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k & \text{(1b)} \end{cases}$$

where $k \in \mathbb{N}$ is the natural number standing for the discrete time instant, $\boldsymbol{x}_k = \boldsymbol{x}(kT) \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}_k \in \mathbb{R}^m$ is the known input vector, $\boldsymbol{\varphi}_k \in \mathbb{R}^\ell$ is the unknown input vector, and $\boldsymbol{y}_k \in \mathbb{R}^p$ is the output vector. Moreover, $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$, $\boldsymbol{C} \in \mathbb{R}^{p\times n}$ and $\boldsymbol{F} \in \mathbb{R}^{n\times \ell}$ are the matrices that define system (1). As it is shown, the feed-through $\boldsymbol{D}$ matrix is absent from (1b) since the majority of the modeled physical systems are strictly proper.

From (1), for any given integer $i \geq 0$, one can write its state sequence $\mathcal{X} = \{\boldsymbol{x}_k, \boldsymbol{x}_{k+1}, \ldots, \boldsymbol{x}_{k+i}\}$ over $i+1$ time-steps as:

$$\boldsymbol{x}_k = \boldsymbol{x}_0 \tag{2a}$$

$$\boldsymbol{x}_{k+i} = \boldsymbol{A}^i\boldsymbol{x}_k + \sum_{j=0}^{i-1}\boldsymbol{A}^{i-1-j}\boldsymbol{B}\boldsymbol{u}_{k+j} + \sum_{j=0}^{i-1}\boldsymbol{A}^{i-1-j}\boldsymbol{F}\boldsymbol{\varphi}_{k+j}, \quad i = 1,2,3,\ldots, \tag{2b}$$

where $\mathcal{U} = \{\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_{k+i-1}\}$ is the known input sequence and $\Phi = \{\boldsymbol{\varphi}_k, \boldsymbol{\varphi}_{k+1}, \ldots, \boldsymbol{\varphi}_{k+i-1}\}$ is the unknown input sequence. In the same manner one can write the output sequence $\mathcal{Y} = \{\boldsymbol{y}_k, \boldsymbol{y}_{k+1}, \ldots, \boldsymbol{y}_{k+i}\}$ as:

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k \tag{3a}$$

$$\boldsymbol{y}_{k+i} = \boldsymbol{C}\boldsymbol{A}^i\boldsymbol{x}_k + \boldsymbol{C}\sum_{j=0}^{i-1}\boldsymbol{A}^{i-1-j}\boldsymbol{B}\boldsymbol{u}_{k+j} + \boldsymbol{C}\sum_{j=0}^{i-1}\boldsymbol{A}^{i-1-j}\boldsymbol{F}\boldsymbol{\varphi}_{k+j}, \quad i = 1,2,3,\ldots. \tag{3b}$$

Hence, one are interested in recover the unknown input vector $\boldsymbol{\varphi}_k$ by inverting the system described by the state-space model (1), based on the approach developed initially by References Sain and Massey [2,23] for inverting LTI systems.

Accordingly, system $\boldsymbol{\Gamma}$ is invertible if it has an *L-delay* inverse for some finite L. The least integer L for which an *L-delay inverse* exists is called the inherent delay of the invertible system (L_0). This inherent delay of the invertible system is the least number of differentiations required to recover the original system's unknown input from the knowledge of its input and output. A consequence of not having $\boldsymbol{D}$ matrix is that L_0 will be at least equal to one, since at least one differentiation will be required to reconstruct the unknown input vector $\boldsymbol{\varphi}_k$.

The referred concept of reconstructing an unknown input is depicted in Figure 1. On the top, it is shown the state-space description of a system, modeled with parameter uncertainties that affect both $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices. This system is transformed to one that have these unknown parameter uncertainties modeled as unknown inputs. Since the output equation does not have any information from this vector, one need to perform some differentiations in order to subsequently recover it.

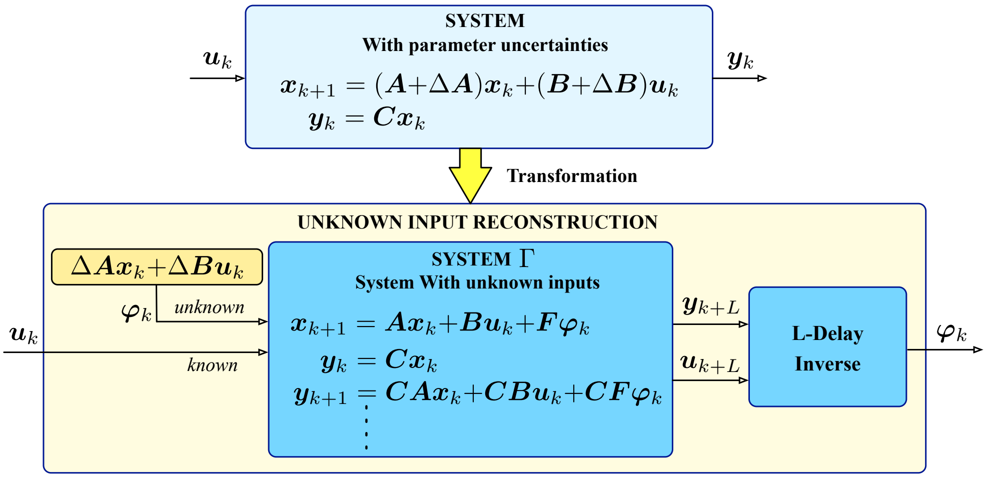

SYSTEM
With parameter uncertainties
$\boldsymbol{u}_k$
$\boldsymbol{x}_{k+1} = (\boldsymbol{A}+\Delta\boldsymbol{A})\boldsymbol{x}_k+(\boldsymbol{B}+\Delta\boldsymbol{B})\boldsymbol{u}_k$
$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k$
$\boldsymbol{y}_k$
Transformation
UNKNOWN INPUT RECONSTRUCTION
$\Delta\boldsymbol{A}\boldsymbol{x}_k+\Delta\boldsymbol{B}\boldsymbol{u}_k$
φ_k
unknown
$\boldsymbol{u}_k$
known
SYSTEM Γ
System With unknown inputs
$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{B}\boldsymbol{u}_k+\boldsymbol{F}\varphi_k$
$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k$
$\boldsymbol{y}_{k+1} = \boldsymbol{C}\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{C}\boldsymbol{B}\boldsymbol{u}_k+\boldsymbol{C}\boldsymbol{F}\varphi_k$
$\boldsymbol{y}_{k+L}$
$\boldsymbol{u}_{k+L}$
L-Delay
Inverse
φ_k

Figure 1. Unknown input reconstruction overview.

Then, the *L-delay inverse* block performs the inversion of the system and reconstructs the unknown input vector after *L-delays.* This *L-delay inverse* block will be shown in more detail afterwards. Note that, for $\boldsymbol{\Gamma}$ to have an *L-delay* inverse, the system must have at least as many outputs as unknown inputs, that is, $p \geq \ell$.

Imbued in this concept of input reconstruction, it is now necessary to adapt the Sain and Massey's inversion algorithm proposed in Reference [2] to this new approach, that is, the system is modeled in discrete-time and there is a new unknown input vector which one want to reconstruct. This matrix $\boldsymbol{F}$ and vector $\boldsymbol{\varphi}_k$ that are added to the state equation are the main challenges faced doing this adaptation.

Firstly, let $\boldsymbol{u}_{[k:k+L-1]} \in \mathbb{R}^{mL}$, $\boldsymbol{\varphi}_{[k:k+L]} \in \mathbb{R}^{\ell L}$ and $\boldsymbol{y}_{[k:k+L]} \in \mathbb{R}^{p(1+L)}$ denote the vectors of sequences over $L+1$ time units, for the known inputs, unknown inputs and outputs, respectively, which are given by:

$$\boldsymbol{y}_{[k:k+L]} := \begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{y}_{k+1} \\ \vdots \\ \boldsymbol{y}_{k+L} \end{bmatrix}, \boldsymbol{u}_{[k:k+L-1]} := \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{u}_{k+1} \\ \vdots \\ \boldsymbol{u}_{k+L-1} \end{bmatrix} \text{ and } \boldsymbol{\varphi}_{[k:k+L-1]} := \begin{bmatrix} \boldsymbol{\varphi}_k \\ \boldsymbol{\varphi}_{k+1} \\ \vdots \\ \boldsymbol{\varphi}_{k+L-1} \end{bmatrix}.$$

From (3), assuming known initial state (so that its effects can be subtracted out), it can readily be checked that:

$$\boldsymbol{y}_{[k:k+L]} = \boldsymbol{\mathcal{O}}_L \boldsymbol{x}_k + \boldsymbol{\mathcal{H}}_L \boldsymbol{u}_{[k:k+L-1]} + \boldsymbol{\mathcal{M}}_L \boldsymbol{\varphi}_{[k:k+L-1]}, \tag{4}$$

where matrices $\boldsymbol{\mathcal{O}}_L \in \mathbb{R}^{p(1+L)\times n}$, $\boldsymbol{\mathcal{H}}_L \in \mathbb{R}^{p(1+L)\times mL}$ and $\boldsymbol{\mathcal{M}}_L \in \mathbb{R}^{p(1+L)\times \ell L}$ can be defined recursively (neglecting the right $\mathbf{0}$ column) for $i \geq 1$ as:

$$\boldsymbol{\mathcal{O}}_0 = \boldsymbol{C} \tag{5a}$$

$$\boldsymbol{\mathcal{O}}_i = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{\mathcal{O}}_{i-1}\boldsymbol{A} \end{bmatrix}, \quad i = 1, 2, \ldots \tag{5b}$$

$$\boldsymbol{\mathcal{O}}_i = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA}^i \end{bmatrix}, \tag{5c}$$

$$\boldsymbol{\mathcal{H}}_0 = \mathbf{0} \tag{6a}$$

$$\boldsymbol{\mathcal{H}}_i = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{\mathcal{O}}_{i-1}\boldsymbol{B} & \boldsymbol{\mathcal{H}}_{i-1} \end{bmatrix}, \quad i = 1, 2, \ldots \tag{6b}$$

$$\boldsymbol{\mathcal{H}}_i = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{CB} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{CAB} & \boldsymbol{CB} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{i-1}\boldsymbol{B} & \boldsymbol{CA}^{i-2}\boldsymbol{B} & \boldsymbol{CA}^{i-3}\boldsymbol{B} & \cdots & \boldsymbol{CB} \end{bmatrix}, \tag{6c}$$

and

$$\boldsymbol{\mathcal{M}}_0 = \mathbf{0} \tag{7a}$$

$$\boldsymbol{\mathcal{M}}_i = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{\mathcal{O}}_{i-1}\boldsymbol{B} & \boldsymbol{\mathcal{M}}_{i-1} \end{bmatrix}, \quad i = 1, 2, \ldots \tag{7b}$$

$$\boldsymbol{\mathcal{M}}_i = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{CF} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \boldsymbol{CAF} & \boldsymbol{CF} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{i-1}\boldsymbol{F} & \boldsymbol{CA}^{i-2}\boldsymbol{F} & \boldsymbol{CA}^{i-3}\boldsymbol{F} & \cdots & \boldsymbol{CF} \end{bmatrix}. \tag{7c}$$

As referred to by Reference [2], system (1) has an inverse with delay L if and only if (4) can be solved uniquely for $\boldsymbol{\varphi}_k$. Accordingly, the following theorem can be written:

Theorem 1. *System* $\boldsymbol{\Gamma}$ *has an L-delay inverse if and only if*

$$\operatorname{rank}(\boldsymbol{\mathcal{M}}_L) - \operatorname{rank}(\boldsymbol{\mathcal{M}}_{L-1}) = \ell, \tag{8}$$

where $\operatorname{rank}(\boldsymbol{\mathcal{M}}_{-1}) := 0$.

Proof of Theorem 1 will be given using Lemma 1.

Lemma 1. *An* $\ell \times p(1+L)$ *matrix* $\boldsymbol{\mathcal{K}}$ *satisfying* $\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{M}}_L = \breve{\boldsymbol{I}}_\ell$*, with* $\breve{\boldsymbol{I}}_\ell := [\boldsymbol{I}_\ell \;\; \boldsymbol{0}] \in \mathbb{R}^{\ell \times \ell(1+L)}$*, exists if and only if* $\operatorname{rank}(\boldsymbol{\mathcal{M}}_L) - \operatorname{rank}(\boldsymbol{\mathcal{M}}_{L\text{-}1}) = \ell$.

Proof. The existence of an L-integral inverse implies that (4) can be solved uniquely for $\boldsymbol{\varphi}_k$, given $\boldsymbol{x}_k$, $\boldsymbol{y}_{[k:k+L]}$ and $\boldsymbol{u}_{[k:k+L-1]}$. It follows that there is a matrix $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{\ell \times p(1+L)}$ satisfying $\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{M}}_L = \breve{\boldsymbol{I}}_\ell$ if and only if the matrix $\breve{\boldsymbol{I}}_\ell = [\boldsymbol{I}_\ell \;\; \boldsymbol{0}] \in \mathbb{R}^{\ell \times \ell(1+L)}$ lies in the rowspace of $\boldsymbol{\mathcal{M}}_L$. This is equivalent to the condition:

$$\operatorname{rank}\left(\begin{bmatrix} \breve{\boldsymbol{I}}_\ell \\ \boldsymbol{\mathcal{M}}_L \end{bmatrix}\right) = \operatorname{rank}(\boldsymbol{\mathcal{M}}_L).$$

Note that:

$$\begin{aligned}\operatorname{rank}\left(\begin{bmatrix} \breve{\boldsymbol{I}}_\ell \\ \boldsymbol{\mathcal{M}}_L \end{bmatrix}\right) &= \operatorname{rank}\left(\begin{bmatrix} \boldsymbol{I}_\ell & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{O}}_{L-1}\boldsymbol{B} & \boldsymbol{\mathcal{M}}_{L-1} \end{bmatrix}\right) \\ &= \ell + \operatorname{rank}(\boldsymbol{\mathcal{M}}_{L-1}).\end{aligned}$$

□

Thus, proof of Theorem 1 can now be given [24]:

Proof. As referred, inverting a system corresponds to determine $\boldsymbol{\varphi}_k$ as a linear combination of $\boldsymbol{x}_k$, $\boldsymbol{u}_{[k:k+L-1]}$ and $\boldsymbol{y}_{[k:k+L]}$ in the form:

$$\boldsymbol{\mathcal{U}}_L\boldsymbol{x}_k + \boldsymbol{\mathcal{V}}_L\boldsymbol{u}_{[k:k+L-1]} + \boldsymbol{\mathcal{W}}_L\boldsymbol{y}_{[k:k+L]} = \boldsymbol{\varphi}_k. \tag{9}$$

Substituting (4) into (9) one get:

$$(\boldsymbol{\mathcal{U}}_L + \boldsymbol{\mathcal{W}}_L\boldsymbol{\mathcal{O}}_L)\boldsymbol{x}_k + (\boldsymbol{\mathcal{V}}_L + \boldsymbol{\mathcal{W}}_L\boldsymbol{\mathcal{H}}_L)\boldsymbol{u}_{[k:k+L-1]} + (\boldsymbol{\mathcal{W}}_L\boldsymbol{\mathcal{M}}_L)\boldsymbol{\varphi}_{[k:k+L]} = \boldsymbol{\varphi}_k, \tag{10}$$

which is valid for all $\boldsymbol{x}_k$, $\boldsymbol{u}_{[k:k+L-1]}$ and all $\boldsymbol{y}_{[k:k+L]}$ if and only if:

$$\boldsymbol{\mathcal{U}}_L = -\boldsymbol{\mathcal{W}}_L\boldsymbol{\mathcal{O}}_L, \tag{11}$$

$$\boldsymbol{\mathcal{V}}_L = -\boldsymbol{\mathcal{W}}_L\boldsymbol{\mathcal{H}}_L \tag{12}$$

and

$$\boldsymbol{\mathcal{W}}_L\boldsymbol{\mathcal{M}}_L = \breve{\boldsymbol{I}}_\ell. \tag{13}$$

It follows from Lemma 1 that a matrix $\boldsymbol{\mathcal{W}}_L$ satisfying (13) exists if and only if condition (8) of Theorem 1 holds. □

Note that if $\operatorname{rank}(\boldsymbol{\mathcal{M}}_L) - \operatorname{rank}(\boldsymbol{\mathcal{M}}_{L-1}) = \ell$, then the first ℓ columns of $\boldsymbol{\mathcal{M}}_L$ are linearly independent of each other and the remaining ℓL columns.

A consequence of Theorem 1 and Lemma 1 is that there exist $L+1$ $(\ell \times p)$-matrices $\boldsymbol{K}_0$, $\boldsymbol{K}_1$, ..., $\boldsymbol{K}_L$ such that:

$$\sum_{i=0}^{j} \boldsymbol{K}_i \boldsymbol{J}_{j-i} = \boldsymbol{0}, \qquad j = 0, 1, 2, \ldots, L-1 \tag{14a}$$

$$\sum_{i=0}^{L} \boldsymbol{K}_i \boldsymbol{J}_{L-i} = \boldsymbol{I}_\ell, \tag{14b}$$

where

$$\boldsymbol{J}_0 = \boldsymbol{0}, \tag{15a}$$

$$\boldsymbol{J}_i = \boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{B}. \tag{15b}$$

Multiplying (from the left) both sides of (4) by $\boldsymbol{\mathcal{K}}$, in accordance to Lemma 1 one can write:

$$\boldsymbol{\mathcal{K}}\boldsymbol{y}_{[k:k+L]} = \boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{O}}_L \boldsymbol{x}_k + \boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{H}}_L \boldsymbol{u}_{[k:k+L-1]} + \boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{M}}_L \boldsymbol{\varphi}_{[k:k+L-1]} \tag{16}$$

where $\boldsymbol{\mathcal{K}}$ is partitioned as in (14), compatible with the partition of $\boldsymbol{y}_{[k:k+L]}$. Since:

$$\boldsymbol{\mathcal{M}}_L = \begin{bmatrix} \boldsymbol{J}_0 & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{J}_1 & \boldsymbol{J}_0 & \ldots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{J}_L & \boldsymbol{J}_{L-1} & \ldots & \boldsymbol{J}_0 \end{bmatrix} \tag{17}$$

it follows from (14) that:

$$\begin{aligned} \boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{M}}_L \boldsymbol{\varphi}_{[k:k+L]} &= (\boldsymbol{K}_L \boldsymbol{J}_0 + \boldsymbol{K}_{L-1}\boldsymbol{J}_1 + \ldots + \boldsymbol{K}_0 \boldsymbol{J}_L)\boldsymbol{\varphi}_k + \\ &+ (\boldsymbol{K}_{L-1}\boldsymbol{J}_0 + \boldsymbol{K}_{L-2}\boldsymbol{J}_1 + \ldots + \boldsymbol{K}_0 \boldsymbol{J}_{L-1})\boldsymbol{\varphi}_{k+1} + \ldots + \\ &+ (\boldsymbol{K}_1 \boldsymbol{J}_0 + \boldsymbol{K}_0 \boldsymbol{J}_1)\boldsymbol{\varphi}_{k+L-1} = \boldsymbol{\varphi}_k \end{aligned} \tag{18}$$

Thus, substituting (18) into (16) results in the following equation:

$$\boldsymbol{\varphi}_k = -\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{O}}_L \boldsymbol{x}_k - \boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{H}}_L \boldsymbol{u}_{[k:k+L-1]} + \boldsymbol{\mathcal{K}}\boldsymbol{y}_{[k:k+L]} \tag{19}$$

which permits to reconstruct the unknown input vector $\boldsymbol{\varphi}_k$ from the knowledge of $\boldsymbol{x}_k$, $\boldsymbol{u}_{[k:k+L-1]}$ and $\boldsymbol{y}_{[k:k+L]}$.

It is important to realize that in order to obtain the state-space inverse system, it is firstly necessary to determine the matrix $\boldsymbol{\mathcal{K}}$ satisfying $\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{M}}_L = \check{\boldsymbol{I}}_\ell$. For that, one enunciate the following theorem:

Theorem 2. *If condition (8) is verified, the input $\boldsymbol{\varphi}_k$ of $\boldsymbol{\Gamma}$ can be reconstructed as a linear combination of $\boldsymbol{y}_{[k:k+L]}$, $\boldsymbol{u}_{[k:k+L-1]}$ and $\boldsymbol{x}_k$. The reconstruction is given by (19) and $\boldsymbol{\mathcal{K}}$ must satisfy:*

$$\boldsymbol{\mathcal{K}} = \check{\boldsymbol{I}}_\ell \boldsymbol{\mathcal{M}}_L^{(1)} + \boldsymbol{Y}_L \boldsymbol{Z}_L \tag{20}$$

where $\boldsymbol{Y}_L$ is an $\ell \times p(1+L)$ arbitrary parameter matrix and $\boldsymbol{Z}_L := \boldsymbol{I}_{p(1+L)} - \boldsymbol{\mathcal{M}}_L \boldsymbol{\mathcal{M}}_L^{(1)}$, where $\boldsymbol{\mathcal{M}}_L^{(1)}$ is the generalized inverse or {1}-inverse of matrix $\boldsymbol{\mathcal{M}}_L$ [24].

Note that, since the matrix $\boldsymbol{Y}_L$ can be freely chosen, the matrix $\boldsymbol{\mathcal{K}}$ in (20) is, in general, not unique.

Substituting (19) into the state Equation (1a) of $\boldsymbol{\Gamma}$ yields the following L-delay inverse system with input $\boldsymbol{y}_{[k:k+L]}$, $\boldsymbol{u}_{[k:k+L-1]}$ and output $\boldsymbol{\varphi}_k$:

$$\boldsymbol{\Gamma}^{-\mathsf{L}} = \begin{cases} \boldsymbol{x}_{k+1} = (\boldsymbol{A} - \boldsymbol{F}\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{O}}_L)\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k - \boldsymbol{F}\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{H}}_L\boldsymbol{u}_{[k:k+L-1]} + \boldsymbol{F}\boldsymbol{\mathcal{K}}\boldsymbol{y}_{[k:k+L]} \\ \boldsymbol{\varphi}_k = -\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{O}}_L\boldsymbol{x}_k - \boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{H}}_L\boldsymbol{u}_{[k:k+L-1]} + \boldsymbol{\mathcal{K}}\boldsymbol{y}_{[k:k+L]}. \end{cases} \tag{21}$$

The structure of the L-delay inverse system that corresponds to this state-space system of equations is depicted in Figure 2. Since there is a vector of known inputs, it will have to be considered in both equations and, consequently, introduces some complexity to the diagram. This added complexity is fundamentally based on the bank of delays necessary to obtain the parcel $\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{H}}_L\boldsymbol{u}_{[k:k+L-1]}$, which has to be subtracted for the reconstruction of $\boldsymbol{\varphi}_k$ and also in the state equation pre-multiplied by the matrix $\boldsymbol{F}$. Note that the matrices $\boldsymbol{H}_0$, $\boldsymbol{H}_1$, $\boldsymbol{H}_{L-1}$ and $\boldsymbol{H}_L$ correspond to the L^{th} column of $\boldsymbol{\mathcal{H}}_L$.

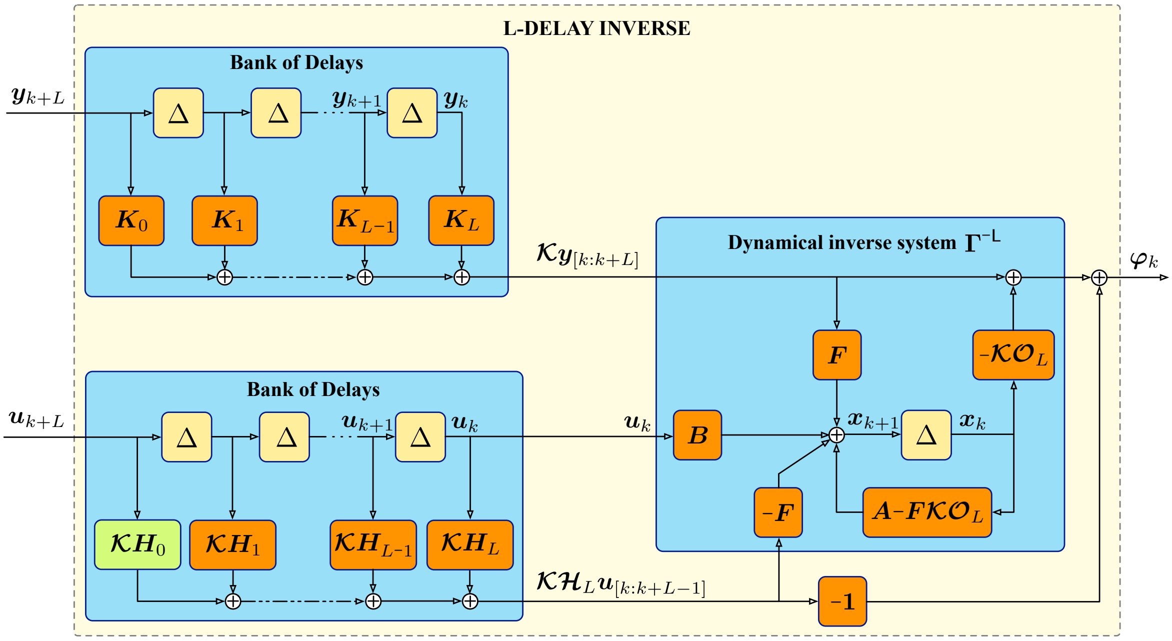


Figure 2. Structure of the L-delay inverse system.

3. Inversion of Switched Linear Systems

In general, switched systems form a particular class of *hybrid systems* where the occurrence of a discrete event leads to a change in system's mode. A switched system is composed of a family of dynamical subsystems and a rule (switching law) that orchestrates the switching among them. A wide range of biologic, physical and engineering systems, can be modeled in this framework. Some examples of real processes that can be modeled as switched systems are, for example, autonomous vehicles, flight control systems and power electronics circuits, like converters.

The focus here will be on a particular class of switched systems where all the subsystems are LTI and the switching signal is governed by a deterministic process. These are known as Switched Linear Systems (SLSs), and an overview of their principle of operation is illustrated in Figure 3. As it is shown, besides the subsystems, this SLS also consists of a switching device usually called *supervisor*. The *supervisor* generates the switching rule $\sigma(k)$ which orchestrates the switching between subsystems. Consequently, there is an active subsystem, enabled by the switching rule, while the other $q-1$ subsystems are not active.

According to References [19,25,26], one can represent the SLS dynamic behavior by the following generalized discrete-time switched state-space model $\Gamma_{\sigma(k)}$ with $\boldsymbol{x}_0 = \boldsymbol{x}(0) = \boldsymbol{0}$:

$$\Gamma_{\sigma(k)} = \begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{A}_{\sigma(k)}\boldsymbol{x}_k + \boldsymbol{B}_{\sigma(k)}\boldsymbol{u}_k + \boldsymbol{F}_{\sigma(k)}\boldsymbol{\varphi}_k & \text{(22a)} \\ \boldsymbol{y}_k = \boldsymbol{C}_{\sigma(k)}\boldsymbol{x}_k & \text{(22b)} \end{cases}$$

where $k \in \mathbb{N}_0$ is a time index, $\boldsymbol{x}_0$ is the initial state, $\boldsymbol{x}_k \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}_k \in \mathbb{R}^m$ is the known input vector, $\boldsymbol{\varphi}_k \in \mathbb{R}^\ell$ is the unknown input vector, $\boldsymbol{y}_k \in \mathbb{R}^p$ is the output vector, all at time k, $\sigma(k) : \mathbb{N}_0 \mapsto \mathcal{Q}$ is a piecewise switching signal which indicates the active subsystem at any time instant $k = \{0, 1, 2, \ldots\}$ and takes values from the finite index set $\mathcal{Q} := \{1, 2, \ldots, q\}$. Moreover, $\boldsymbol{A}_q \in \mathbb{R}^{n\times n}$, $\boldsymbol{B}_q \in \mathbb{R}^{n\times m}$, $\boldsymbol{C}_q \in \mathbb{R}^{p\times n}$ and $\boldsymbol{F}_q \in \mathbb{R}^{n\times \ell}$, with $q \in \mathcal{Q}$, are constant matrices that define the subsystems in (22). One are assuming that $\sigma(\cdot) := \{\sigma(k)\}$ is a piecewise constant function, meaning that it takes a constant value every interval between two consecutive switchings. It is also important to note that, this switching signal is an external input which can be selected freely from $\mathcal{Q}$ without any constraint.

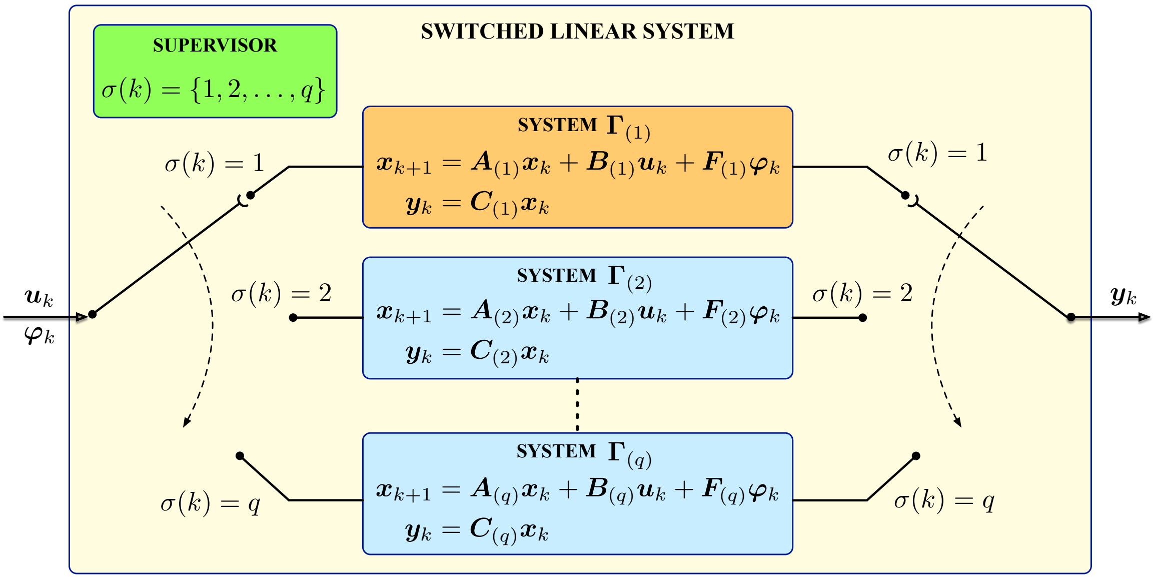

SWITCHED LINEAR SYSTEM
SUPERVISOR
$\sigma(k) = \{1, 2, \ldots, q\}$
SYSTEM $\boldsymbol{\Gamma}_{(1)}$
$\boldsymbol{x}_{k+1} = \boldsymbol{A}_{(1)}\boldsymbol{x}_k + \boldsymbol{B}_{(1)}\boldsymbol{u}_k + \boldsymbol{F}_{(1)}\boldsymbol{\varphi}_k$
$\boldsymbol{y}_k = \boldsymbol{C}_{(1)}\boldsymbol{x}_k$
SYSTEM $\boldsymbol{\Gamma}_{(2)}$
$\boldsymbol{x}_{k+1} = \boldsymbol{A}_{(2)}\boldsymbol{x}_k + \boldsymbol{B}_{(2)}\boldsymbol{u}_k + \boldsymbol{F}_{(2)}\boldsymbol{\varphi}_k$
$\boldsymbol{y}_k = \boldsymbol{C}_{(2)}\boldsymbol{x}_k$
SYSTEM $\boldsymbol{\Gamma}_{(q)}$
$\boldsymbol{x}_{k+1} = \boldsymbol{A}_{(q)}\boldsymbol{x}_k + \boldsymbol{B}_{(q)}\boldsymbol{u}_k + \boldsymbol{F}_{(q)}\boldsymbol{\varphi}_k$
$\boldsymbol{y}_k = \boldsymbol{C}_{(q)}\boldsymbol{x}_k$
$\sigma(k) = 1$
$\sigma(k) = 2$
$\sigma(k) = q$
$\sigma(k) = 1$
$\sigma(k) = 2$
$\sigma(k) = q$
$\boldsymbol{u}_k$
$\boldsymbol{\varphi}_k$
$\boldsymbol{y}_k$

Figure 3. Switched linear system overview.

The concept of inversion applied to SLSs is an approach with few publications. According to Reference [16,19], the problem of inverting such systems, is to recover the switching signal and the input, given uniquely the output and the initial state. However, since one are considering that the switching signal is known, this problem is reduced merely to recovering the unknown input vector of the system. For that, as it is shown in References [16,19], it is only necessary to verify that each subsystem of the SLS is invertible. If all subsystems are invertible, then the SLS is invertible.

Accordingly, first it is necessary to verify the invertibility condition given in Theorem 1 for each subsystem. If all subsystems are invertible, it is possible to obtain the discrete state-space description of the inverse system for each subsystem from the set $\mathcal{Q} = \{1, 2, \ldots, q\}$, which is given by:

$$\boldsymbol{\Gamma}^{-\mathsf{L}}_{\sigma(k)} = \begin{cases} \boldsymbol{x}_{k+1} = (\boldsymbol{A} - \boldsymbol{F}\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{O}}_L)_{\sigma(k)}\boldsymbol{x}_k - (\boldsymbol{F}\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{H}}_L)_{\sigma(k)}\boldsymbol{u}_{[k:k+L-1]} + \\ \qquad + \boldsymbol{B}_{\sigma(k)}\boldsymbol{u}_k + (\boldsymbol{F}\boldsymbol{\mathcal{K}})_{\sigma(k)}\boldsymbol{y}_{[k:k+L]} & \text{(23a)} \\ \boldsymbol{\varphi}_k = -(\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{O}}_L)_{\sigma(k)}\boldsymbol{x}_k - (\boldsymbol{\mathcal{K}}\boldsymbol{\mathcal{H}}_L)_{\sigma(k)}\boldsymbol{u}_{[k:k+L-1]} + \boldsymbol{\mathcal{K}}_{\sigma(k)}\boldsymbol{y}_{[k:k+L]}. & \text{(23b)} \end{cases}$$

Since the number of inverse subsystems is q, it is also necessary to obtain q-times all matrices in (23), one for each configuration. The procedure of constructing such matrices is the same as the one explained in previous section, which is not going to be repeated here.

4. Inversion-Based FDI

This section shows simulation results regarding the implementation of the inversion approach to detect and isolate faults in a DC-DC boost converter. Moreover, it is derived the converter's dynamic state-space model and the corresponding inverse model, which is used to recover the unknown input vector, which, in turn, includes faults, injected into the converter and sensors. Furthermore, it is described the proposed FDI approach in detail.

4.1. Boost Power Converter

To illustrate the procedure discussed in previous sections, it is considered the boost converter shown in the top of Figure 4. As referred before, a power converter is a switched linear system with a finite set of subsystems and a set of logic rules that determine the transition between those subsystems, one for each state of the switch, resulting in two active subsystems $\mathcal{Q} = \{1, 2\}$ in this case, since one are assuming that the converter operates only in Continuous Conduction Mode (CCM). Moreover, $s_{\sigma(k)}$ is a Boolean variable that indicates the state of the semiconductor switch: when S_1 is ON then $s_{\sigma(k)} = 1$; when S_1 is OFF then $s_{\sigma(k)} = 0$. This step-up DC-DC converter is constituted of a power source, an inductor L (not ideal), a switch S_1, a diode D_1, a filtering capacitor C_1 and a load resistance R_o at the output.

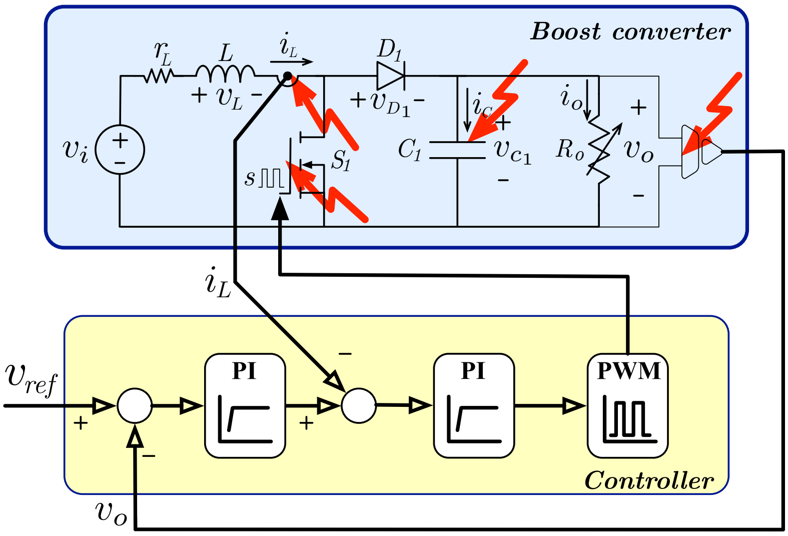


Figure 4. Boost converter and injected faults.

As illustrated, the studied boost converter is operating in closed-loop, in order to regulate the output voltage of the converter. Moreover, one are considering the possibility of occurrence of four different types of faults in the converter (shown also in the figure): (*i*) switch open fault; (*ii*) capacitor degradation; (*iii*) voltage sensor fault; and (*iv*) current sensor fault.

Since the converter is operating in closed-loop, all the considered faults will manifest themselves in the unknown input vector of (22). Thus, the dynamics of the boost converter with the inclusion of faults can be modeled by the following generalized discrete-time state-space system $\langle \Gamma_{\sigma(k)} \rangle$:

$$\boldsymbol{\Gamma}_{\sigma(k)} = \begin{cases} \boldsymbol{x}_{(k+1)} = \begin{bmatrix} 1-\frac{r_L}{L}T & -\frac{T}{L}(1-s_{\sigma(k)}) \\ \frac{T}{C}(1-s_{\sigma(k)}) & 1-\frac{T}{R_oC} \end{bmatrix} \boldsymbol{x}_{(k)} + \begin{bmatrix} \frac{T}{L} \\ 0 \end{bmatrix} v_{i(k)} + \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} \phi_{1(k)} \\ \phi_{2(k)} \end{bmatrix} & \text{(24a)} \\ \boldsymbol{y}_{(k)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_{(k)} & \text{(24b)} \end{cases}$$

which was discretized applying Euler's forward method of discretization, using T as *sampling period*. Moreover, it is considered that v_i is known and that the state vector $\boldsymbol{x}=[i_L \;\; v_C]^{\mathsf{T}}$ can be measured.

Consequently, applying the methodology described in Section 2, one can obtain the state-space equations for the inverse system $\langle \boldsymbol{\Gamma}^{-\mathsf{L}}_{\sigma(k)} \rangle$, which can be written as:

$$\begin{cases} \boldsymbol{z}_{(k+1)} = -\begin{bmatrix} \frac{T}{L} \\ 0 \end{bmatrix} v_{i(k)} + \begin{bmatrix} \frac{T}{L} \\ 0 \end{bmatrix} v_{i(k)} + \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{y}_{(k,k+1)} \\ \begin{bmatrix} \phi_{1(k)} \\ \phi_{2(k)} \end{bmatrix} = \begin{bmatrix} \frac{r_L}{L}-\frac{1}{T} & \frac{1}{L}(1-s_{\sigma(k)}) \\ -\frac{1}{C}(1-s_{\sigma(k)}) & \frac{1}{R_oC}-\frac{1}{T} \end{bmatrix} \boldsymbol{z}_{(k)} \\ \qquad -\begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_{i(k)} + \begin{bmatrix} 0 & 0 & \frac{1}{T} & 0 \\ 0 & 0 & 0 & \frac{1}{T} \end{bmatrix} \boldsymbol{y}_{(k,k+1)} \end{cases} \tag{25}$$

where the unknown inputs are recovered with one step delay. Moreover, beyond the switching sequence, the inputs to this inverse system are $\boldsymbol{y}_{(k,k+1)} = \begin{bmatrix} i_{L(k)} & v_{C(k)} & i_{L(k+1)} & v_{C(k+1)} \end{bmatrix}^{\mathsf{T}}$ and $v_{i(k)}$. Note that, in this case, it is only possible to recover two signals from the inverse system: $\phi_{1(k)}$ and $\phi_{2(k)}$, which are the unknown additive inputs of (24).

4.2. Symptoms Generation for Fault Detection

The approach followed in this paper relies upon the use of the left inversion of deterministic systems in state-space form, where the objective is to exactly reconstruct the input applied to the system from knowledge of its output. It is clear that the fault signals ϕ_1 and ϕ_2 contain useful information (symptoms). However, as it will be clear later, with only the two signals ϕ_1 and ϕ_2, it is not possible to detect all four types of faults occurring in the converter. In order to distinguish between different faults, it is necessary to create two additional special symptoms from measurements and reconstructed inputs. In the search for obtaining a mathematical description for the sensor faults, it was found that a good approximation to each sensor gain fault can be obtained by:

$$\delta_{1(k)} \approx \frac{\phi_{1(k)}}{\tilde{i}_{L(k)} - \phi_{1(k)}} \tag{26}$$

and

$$\delta_{2(k)} \approx \frac{\phi_{2(k)}}{\tilde{v}_{C(k)} - \phi_{2(k)}}, \tag{27}$$

where $\tilde{i}_{L(k)}$ and $\tilde{v}_{C(k)}$ are the discrete-time derivatives of i_L and v_C, respectively. Thus, in order to detect the four types of faults, it was developed a delta block, which generates these two extra signals: δ_1 and δ_2, as illustrated in Figure 5. As shown, the FD system consists of an inversion block which generates the signals ϕ_1 and ϕ_2, a δ block which generates the signals δ_1 and δ_2, followed by a smoothing filter and a threshold checking mechanism that generates a fault flag indicating the presence of a fault in the converter. It is assumed that these four signals change significantly when a fault appears, making it possible to generate a fault indicator signal, which shows that there is a fault present in the system.

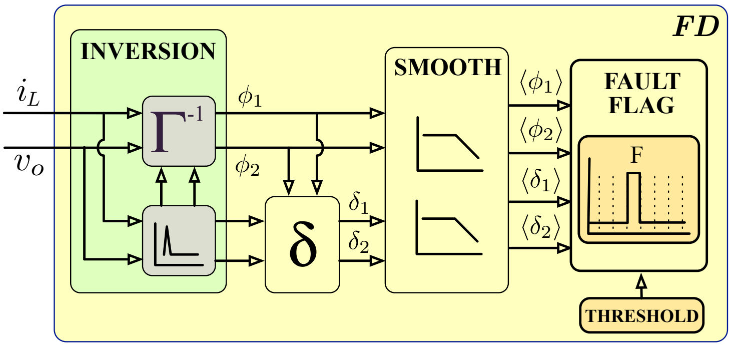


Figure 5. Overview of the implemented Fault Detection (FD) system.

4.3. Fault Isolation Using a Fuzzy Logic System

As it is easy to understand, using the previously described Fault Detection (FD) system, faults can only be detected and not isolated. Once the fault detection mechanism perceives a condition of fault occurrence, that fault has to be classified. Like referred before, with the signals obtained from the inversion and delta blocks it is possible to detect the presence of faults in the converter. However, it is intended not only to detect them, but also to have the knowledge of the component or sensor that has a problem. This is the task of isolation in a FDI system.

The problem of FDI is not easy to achieve, since there are uncertainties inherent to every modeling process. Moreover, in the presence of faults, the faulty measurements influence the closed-loop behavior and the state estimation becomes corrupted. Also, since the measured variables are only two, it is not possible to fully isolate four types of faults, that is, the isolation is only possible with some known constraints.

In fact, with only two measured variables one can obtain only two outputs from the inverse system, as already stated. Thus it is not possible to achieve complete fault decoupling. To overcome this problem, it was developed a new approach that, with some restrictions, turns it possible to isolate the four types of faults, achieving an acceptable FDI performance. An overview of that implementation is illustrated in Figure 6.

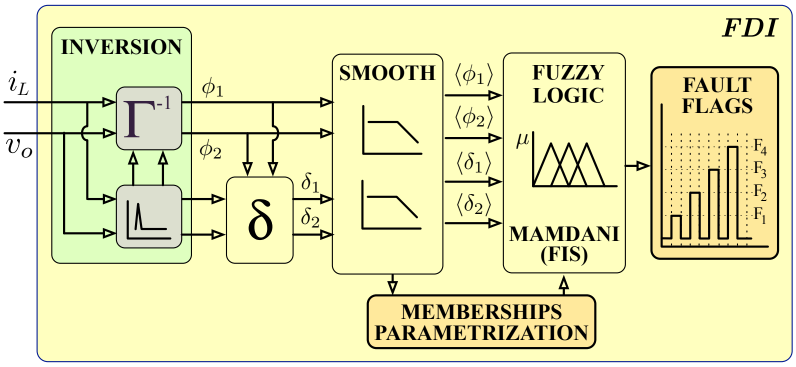


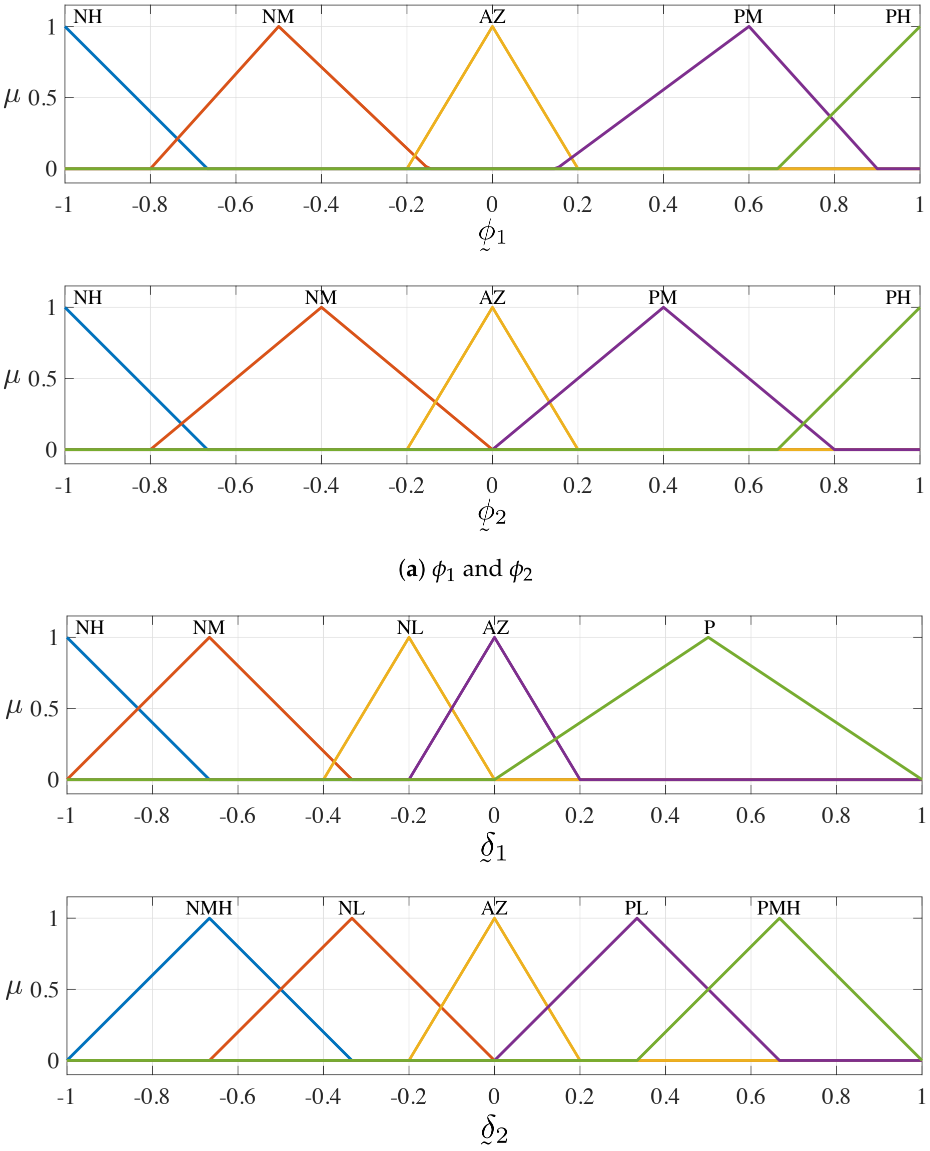
Figure 6. Overview of the implemented Fault Detection and Isolation (FDI) block.

Consequently, a Fuzzy Inference System (FIS) was developed to cope with the problem of fault isolation. This fuzzy inference system was developed in Matlab®, using its fuzzy control toolbox [27]. The use of this toolbox proved to be an easy way to design the membership functions for the inputs and output of our FDI system.

As fuzzy-logic has similarities with human reasoning, it has been widely used for fault diagnosis [28]. Given a set of symptom variables, it is first necessary to define membership functions that map this numeric data into linguistic variables. The process where a variable is represented by a membership function is known as *fuzzification*. In our case, each input is a crisp numerical value limited to the universe of disclosure of the input variable, that is, the universe $X \in [-1, 1]$. Consequently, it were used linguistic variables to describe the behavior of the physical variables, which are composed of 5 terms each, corresponding to membership functions.

According to Reference [29], the uniform triangular membership functions describing negative high (NH), negative medium (NM), negative small (NS), almost zero (AZ), positive small (PS), positive medium (PM) and positive high (PH), all in absolute values, are very effective for describing the behavior of input and output variables. In our case, it were made small changes to these membership functions, maintaining the coverage of the range of the input variables to obtain better performance of the FDI system. In some cases it were introduced the following terms: positive medium/high (PMH), negative medium/high (NMH) and positive (P). The membership functions for all variables are illustrated in Figure 7a,b, and were obtained after some simplifications of the above mentioned set referred to in Reference [29].

The proposed fuzzy fault detection and isolation block FDI has four inputs: $\langle \phi_1 \rangle$, $\langle \phi_2 \rangle$, $\langle \delta_1 \rangle$ and $\langle \delta_2 \rangle$, which are the outputs of the smoothing block. Each input is then *fuzzified* or mapped to a value between 0 and 1 defined by the membership functions referred before.



(**a**) ϕ_1 and ϕ_2

(**b**) δ_1 and δ_2

Figure 7. Membership functions of input variables.

After the definition of the membership functions for all input variables, it is necessary to implement a set of linguistic fuzzy rules, which consists of a collection of IF-THEN rules. In this fuzzy inference system it was used a set with seven fuzzy IF-THEN rules.

After each rule is evaluated, it is generated a number that is then applied to the output, which consists of the set of membership functions depicted in Figure 8. The terms for these membership functions are related with the presence of faults and their isolation. Thus, they are defined as: no fault (NF), switch fault (SW), current sensor fault (CS), voltage sensor fault (VS) and capacitor fault (C). As referred before, the indication of switch fault means an open circuit in the semiconductor switch, and the indication of capacitor fault means a degradation of its capacitance.

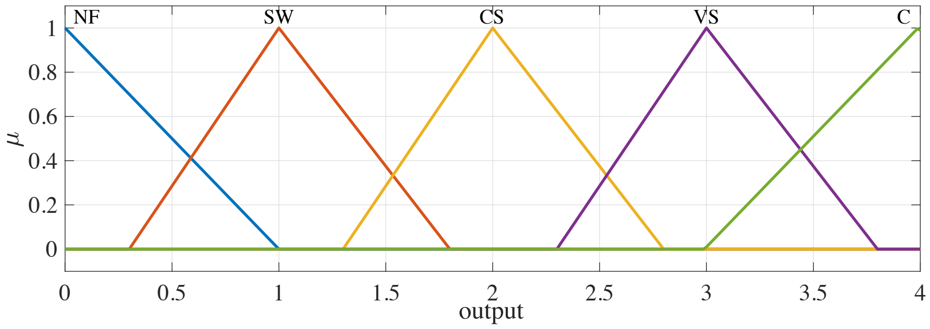


Figure 8. Membership functions of output.

After the evaluation of all rules, their outputs are combined in a process called aggregation which, in this work, is computed using the 'maximum' method. Finally, it is necessary to *deffuzify* the output to obtain a crisp value. This process of inference can be viewed in more detail in the specialty literature, for example, in Reference [27]. It is important to note that the output of the fuzzy block is an integer number indicating what is the fault present at a certain moment. Consequently, if a fault occurs, a flag will be set and it will trigger a visual indicator showing what component or sensor has a problem.

4.4. Illustrative Simulation Studies of the Inversion-Based Method

This section shows simulation results obtained from the implementation of the inversion-based approach to detect and isolate faults, in a boost DC-DC converter. The studied converter is simulated in Matlab®/Simulink® using the parameters and other important values listed in Table 1.

Table 1. Boost converter's values used in simulation.

Description	Value
Input voltage – V_i	24 V
Output voltage – V_o	50 V
Inductor inductance – L	4.7 mH
Inductor resistance – r_L	1 Ω
Capacitor capacitance – C_1	1100 µF
Load resistance – R_o	35 Ω to 65 Ω
Switching frequency – f_s	5 kHz
Sampling time – T	1×10^{-5} s

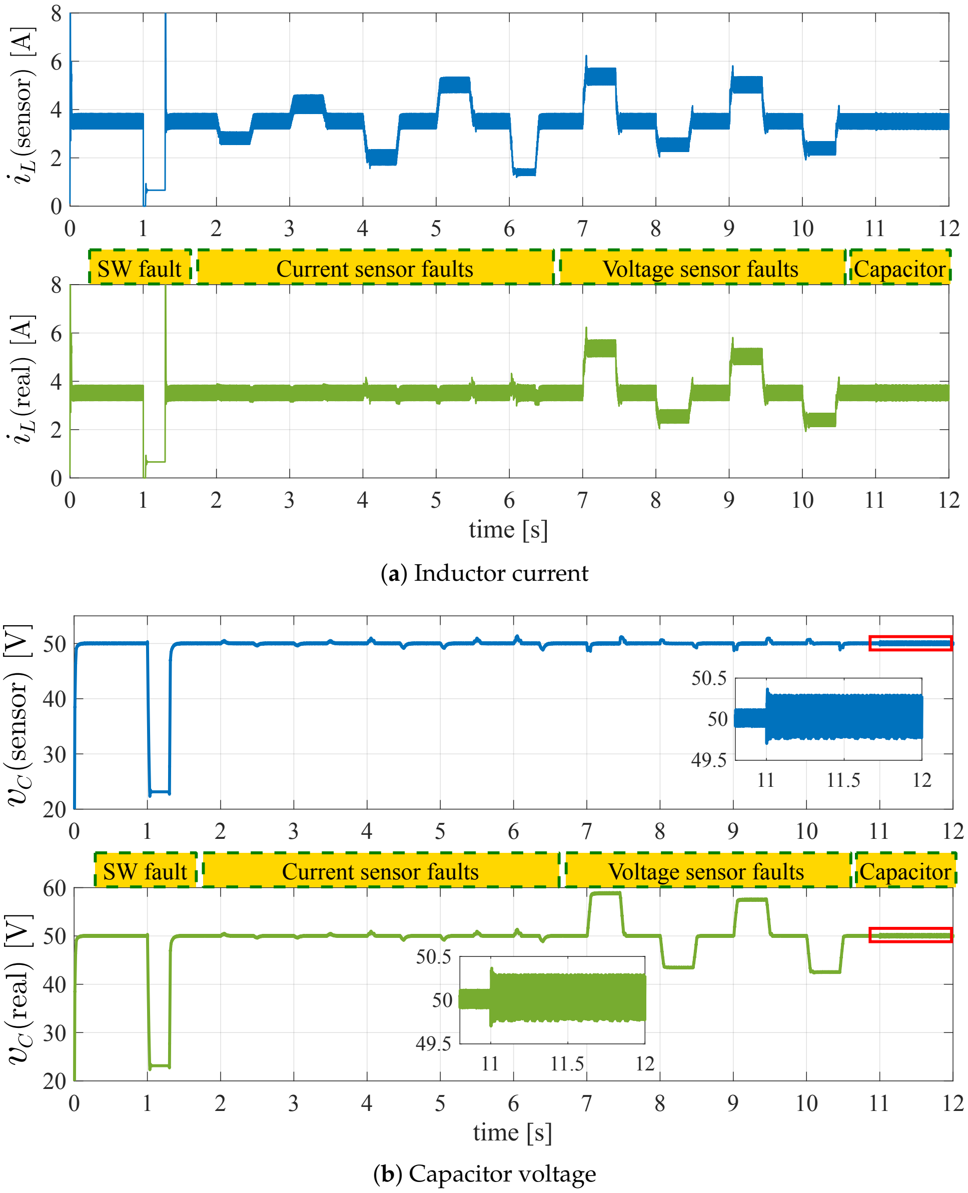
The developed Matlab®/Simulink® model consists of a boost converter constructed using the Simulink's simscape library, an inverse model to recover the unknown inputs, and blocks to inject the faults in the switch, sensors and capacitor. As already referred, the converter is working in closed-loop to regulate its output voltage to $V_o = 50$ V. The *controller* has two PI controllers whose gains where calculated using a small-signal model of the converter, as explained in Reference [30], and then fine tuned by trial and error. Moreover, it were made simulations for different load values, varying between 35 Ω to 65 Ω.

The injection of faults in the converter consists in applying step changes to the above mentioned components and sensors. In the case of sensors, it consists in changing their gain δ_g and offset δ_o values, while in the capacitor consists in degrading the value of its capacitance by δ_C.

Regarding the current sensor, its gain δ_{g_i} is changed to 0.8 at $t = 3$ s and to 1.2 at $t = 4$ s. Moreover, it is injected a more drastic fault at $t = 6$ s, where its gain is changed to 0.4. In the case of the offset faults δ_{o_i} it is added -1 A at $t = 4$ s and 1 A at $t = 5$ s.

For the voltage sensor, its gain δ_{g_v} changes between -15% at $t = 7$ s and $+15\%$ at $t = 8$ s. In the case of the offset faults δ_{o_v}, it is added -7.5 V at $t = 9$ s and 7.5 V at $t = 10$ s. Finally, the value of the capacitor's capacitance will drop to 40% of its nominal value at $t = 11$ s.

One of the first results obtained from simulation is illustrated in Figure 9 for a load resistance of $R_o = 35\,\Omega$. It is easy to spot the time instant when the switch is OPEN, as the current drops to zero and the output voltage drops to the value of the input. Note that the measured values (sensor) are different from the real ones and that the controller and FDI block receive information from the sensors, which, although being highly reliable, are susceptible to show incorrect measurements.



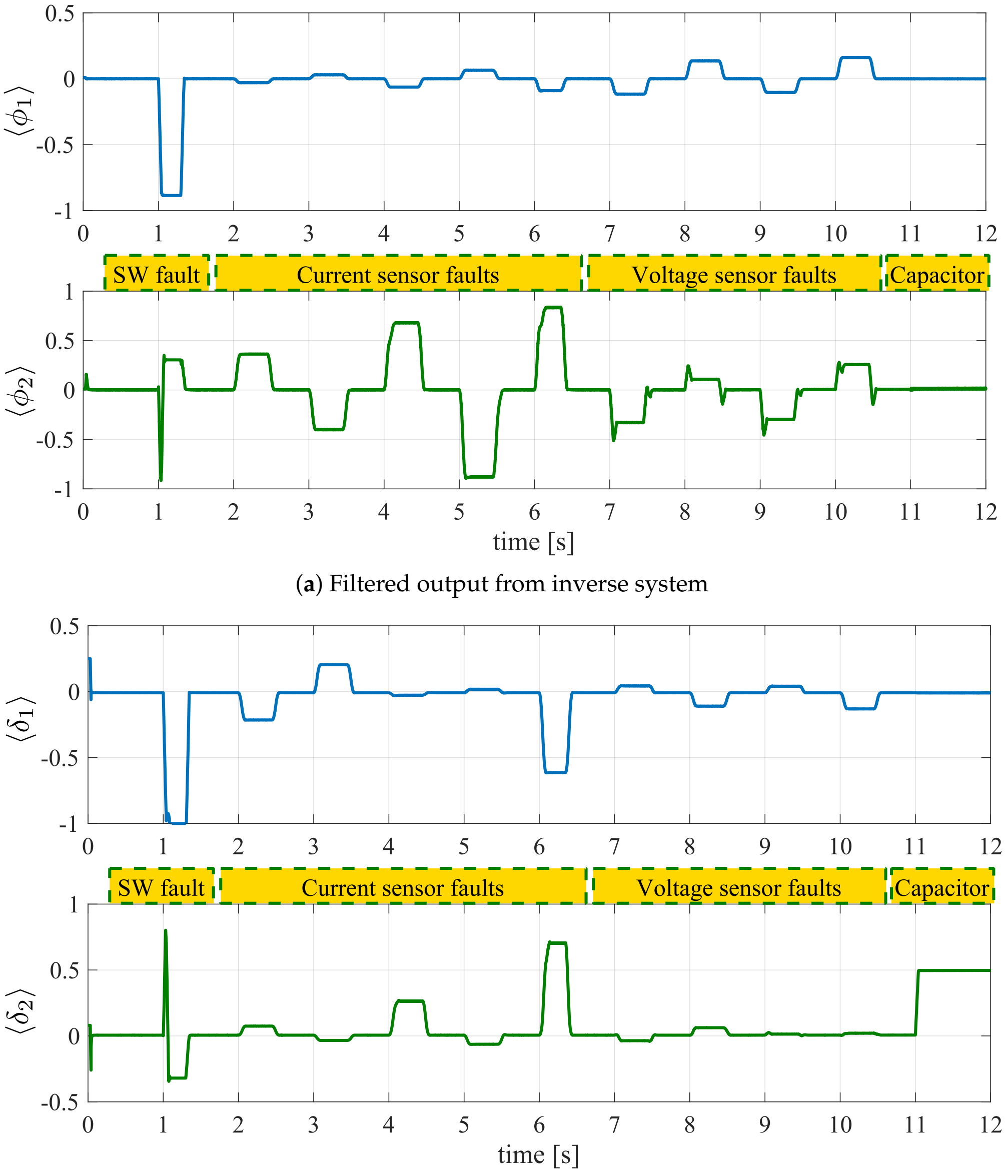
(**a**) Inductor current

(**b**) Capacitor voltage

Figure 9. Boost converter simulation: real and measured states for $R_o = 35\,\Omega$.

Inspecting Figure 9a, it is visible that when faults are injected in the current sensor, they manifest themselves in the measured value but, as would be expected, not in the real one. Contrarily, in the case of the voltage sensor, since the controller adjusts the output voltage to the reference, the faults do not show up in the measured values, as it is illustrated in Figure 9b. However, looking at the real values one can see that the output voltage changes to accommodate the injected fault in the sensor. At the same time, as the voltage changes, the current also adapts to the new situation.

With model uncertainties and possibility of faults in the converter, it is difficult at first place to obtain good output from the inverse system and also to perform FDI from the available data. Consequently, it was opted first to filter the output of the inverse system to obtain a cleaner and more perceptible information. Figure 10a shows the evolution of the two output variables from the inverse system after passing through that filter. From the figure, it is clear that the signal $\langle \phi_1 \rangle$ is more sensible to switch and voltage sensor faults and that current sensor gain and offset faults are more noticeable in signal $\langle \phi_2 \rangle$.



(**a**) Filtered output from inverse system

(**b**) Filtered output from δ block

Figure 10. Boost converter simulation: generated signals for $R_o = 35\,\Omega$.

Inspecting Figure 10b, It is clear that, in one hand, the top signal $\langle\delta_1\rangle$ is more sensible to switch and current sensor gain faults, while, on the other hand, the bottom signal $\langle\delta_2\rangle$ is more sensible to capacitor's faults. With these four recovered signals it is possible to detect and isolate all the referred faults using the fuzzy inference system presented before. Drastic gain and offset current sensor faults are also evident in these figures. From simulations it was also found that $\langle\delta_2\rangle$ signal can be used to detect and isolate the capacitor's fault.

As already stated, it were made simulations for different load values to validate the FDI approach, which, due to space limitations, are not included here. In fact, for the inversion algorithm to be insensitive to the load, its resistance value is computed during the simulations, which inserts some transients in the results. Nevertheless, there are advantages from implementing this additional calculation.

The generated flags for fault detection and isolation are illustrated in Figure 11 for $R_o = 35\,\Omega$. As it can be seen, the FDI system has an acceptable performance, being able to isolate the four types of faults. Moreover, it proved to be robust to load variations. It is also important to refer that the system continues to operate, although with degraded performance, in the presence of sensor and capacitor faults.

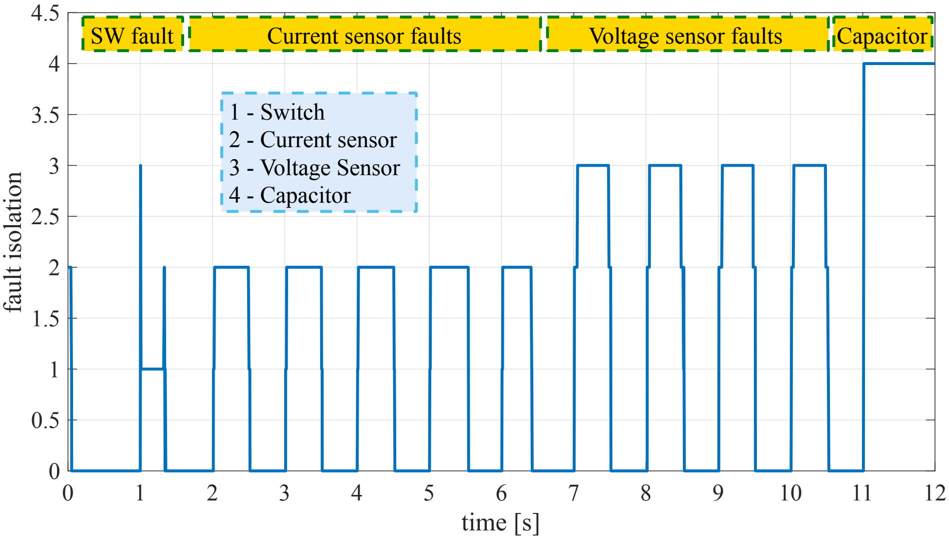


Figure 11. Boost simulation: flags for FDI with $R_o = 35\,\Omega$.

5. Experimental Results

In order to validate the previously exposed and simulated FDI approach, it was developed an experimental setup whose general overview is illustrated in Figure 12. This setup consists of a boost converter, current (CS) and voltage (VS) sensors, a gate driver (d), a microcontroller and a computer-based dashboard. A 24 V power source v_i feeds the boost converter, which is connected to a load resistance R_o. To control the converter it is used a microcontroller, which is connected to a computer that has a Graphical User Interface (GUI) (*dashboard*) to inject faults and visualize signals. This *dashboard* also permits to access the values of variables and to make measurements in real-time. Moreover, permitted to implement some visual indicators that light up 'green' in the case of normal operation and turn 'red' in the presence of faults in the converter or sensors.

The microcontroller receives through its Analog to Digital Converters (ADCs) the measurements from the sensors installed in the boost converter, i_{sens} and v_{sens}. After acquiring these signals, they pass through a software-based filter to reduce inherent noise and are used in the controller's current and voltage PI loops and also in the FDI block. The PWM signal is generated by the *controller*, based on the acquired current and voltage, being then sent to the gate driver, which controls the converter's switch S_1 ON and OFF state. The proportional and integral gains were calculated using the small-signal model of the converter, as explained in Reference [30], and then fine tuned making experimental tests.

Those signals are also used by the *FDI* block to detect and isolate the faults in the converter. Firstly, the filtered values of the current and voltage are used to reconstruct the additive unknown inputs using the inverse model of the converter in the *Inversion* block. These, together with the discrete derivatives of the voltage and current are used to compute δ_1 and δ_2, which are then passed through another filter to smooth them. Finally, these four filtered signals are used by the *Fuzzy Logic* block to perform FDI. The injection of faults and visualization of the FDI output is made in the *dashboard*, as illustrated in Figure 12.

In order to implement the algorithm which controls the output voltage of the converter and the inversion-based FDI algorithm explained before, it was used the Infineon DAVE$^{\text{TM}}$ software, together with some built-in APPs. DAVE$^{\text{TM}}$ is a free eclipse based IDE offering code repository and graphical system design methods using APPs, which automatically generates code for XMC microcontrollers using cortex-M processors. In addition to DAVE$^{\text{TM}}$ it was also used the µC/Probe$^{\text{TM}}$ software, whose working window is depicted in Figure 13. This software works in communication with DAVE$^{\text{TM}}$ and has some interesting functionalities, like visualizing and changing variable's values. It permits to develop, for example, a dashboard that shows information of the variables defined in DAVE's C code in real-time. In what refers to this work, it allows to visualize values of variables, inject faults in sensors by changing their gain and offset values, and visualize *boolean* variables in a LED-like form, that is, 'green' for no-fault and 'red' when a fault occurs. Moreover, it permits to work as a scope to see the evolution of variables in real-time, although with some communication constraints. To better understand the software implementation, Figure 14 illustrates its execution flowchart, where it is depicted the main loop, 'MAIN', and the three interrupt routines: 'INTERRUPT_0', 'INTERRUPT_1', and 'ADC MEAS.'.

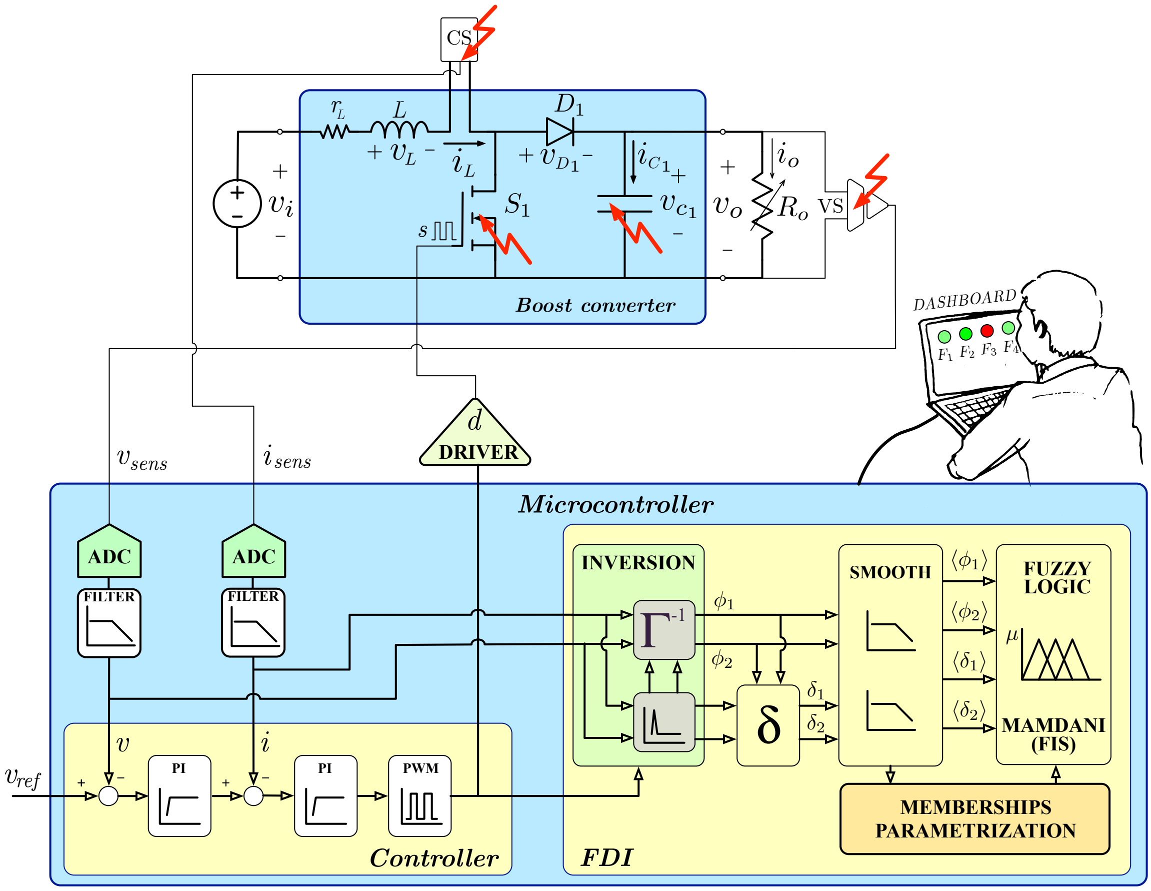


Figure 12. General overview of the implemented solution.

To validate the simulation results, it was constructed an experimental setup whose layout is shown in Figure 15. Its main components, are identified with labels in the figure, and are mainly the following: power source, converter, microcontroller and associated electronics, converter's load and computer (*dashboard*).

As already stated, sensor faults were injected by software, using the dashboard constructed in μC/Probe^{TM}. As illustrated in Figure 13, the injection of a fault is performed by using an 'option button', permitting to inject gain and offset faults for each sensor. The results obtained by the practical experience can be divided in two groups: the first consisting only of the first images of the real and measured voltages, and the second consisting of the images of the signals from the inverse and δ blocks.

Thus, regarding the first, Figure 16 shows the evolution of the real output voltage and the sensor reading voltage when faults are injected in the voltage sensor. It also shows the value of the flag that indicates the presence of a fault in the sensor.

At instant ①, there is a change in the reference voltage v_{ref} from 40 V to 50 V. As shown, both real and sensor voltage (acquired by controller) follow the reference value and the fault flag keeps its value. At instant ② it is injected a $-15\,\%$ gain fault/-7.5 V offset fault in voltage sensor. As the sensor information is less than the real value, the voltage at the output increases to compensate that fault. Then, the algorithm detects the existence of that fault and the corresponding flag changes to one, indicating the existence of a voltage sensor fault. At instant ③ the fault is cleared and the system returns to normal operation, without any faults. At instant ④ it is injected a 15 % gain fault/7.5 V offset fault in the sensor. Contrarily to the occurred before, the output voltage is reduced to compensate for the fault and, at the same time, the algorithm detects the existence of the fault by changing the value of its flag. When the fault is cleared at instant ⑤, the converter returns to normal operation and the fault flag changes to zero.

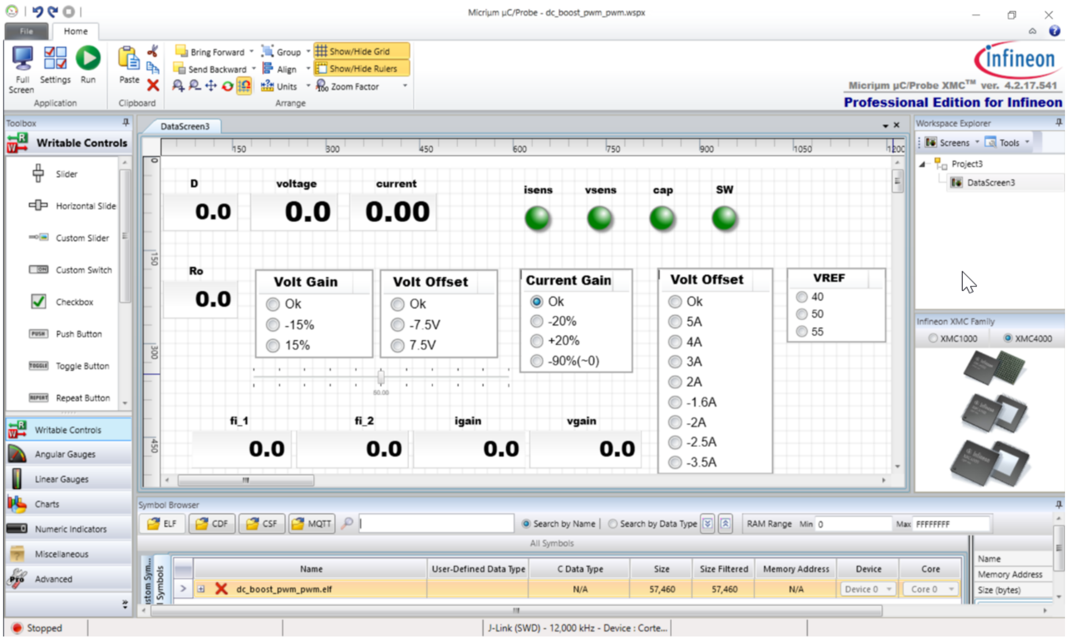


Figure 13. Micrium working window.

Regarding the second experience, the time injection of faults is performed like follows: at instant ① it is injected a -15% gain fault in voltage sensor and a -20% gain fault in current sensor; at instant ② it is injected a 15% gain fault in voltage sensor and a 20% gain fault in current sensor; at instant ③ it is injected a -7.5 V offset fault in voltage sensor and a -2 A gain fault in current sensor; at instant ④ it is injected a 7.5 V offset fault in voltage sensor and a 2 A gain fault in current sensor; at instant ⑤ it is injected a -90% gain fault in current sensor.

Figure 17 shows the FDI of faults in the voltage and current sensors obtained from reasoning about the variables ϕ_1, ϕ_2, δ_1 and δ_2. At this point it is worth mentioning that these variables passed through an average filter implemented by software and that, even though Figure 17 shows only the evolution of ϕ_1 and ϕ_2, the other two variables are also used to obtain FDI.

Inspecting both figures, it is noticeable that variable ϕ_1 is more sensible to voltage sensor faults whereas ϕ_2 is more sensible to current sensor faults. Moreover, as it is clear from figures, the FDI algorithm has a good performance and is capable of detecting and isolating the injected faults. It is also important to realize that these results show similarities to those obtained by simulations and shown in the previous section.

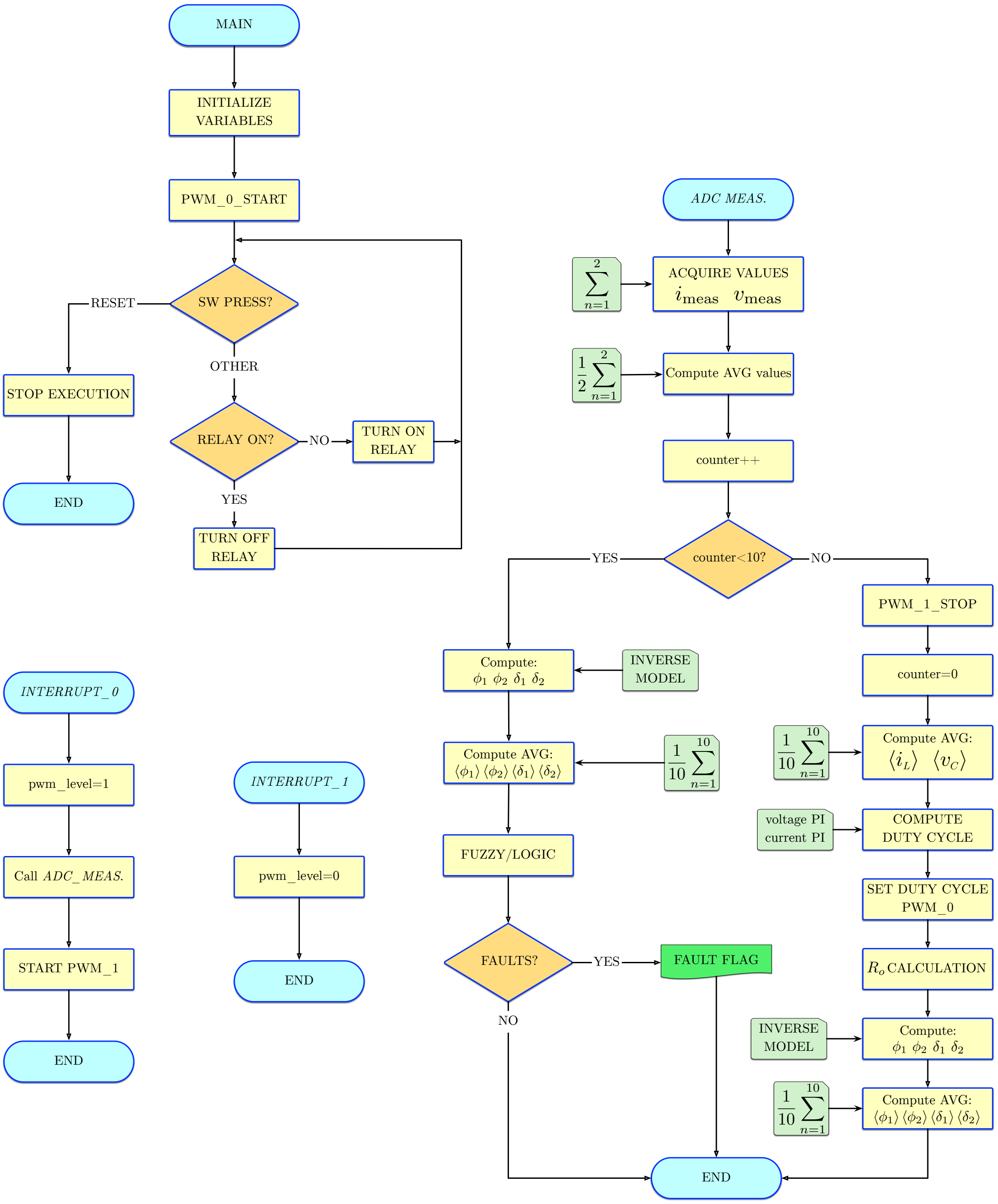

MAIN
INITIALIZE VARIABLES
PWM_0_START
SW PRESS?
RESET
STOP EXECUTION
END
OTHER
RELAY ON?
NO
TURN ON RELAY
YES
TURN OFF RELAY
ADC MEAS.
$\sum_{n=1}^{2}$
ACQUIRE VALUES
i_{meas} v_{meas}
$\frac{1}{2}\sum_{n=1}^{2}$
Compute AVG values
counter++
counter<10?
YES
NO
PWM_1_STOP
counter=0
Compute: ϕ_1 ϕ_2 δ_1 δ_2
INVERSE MODEL
Compute AVG: $\langle\phi_1\rangle$ $\langle\phi_2\rangle$ $\langle\delta_1\rangle$ $\langle\delta_2\rangle$
$\frac{1}{10}\sum_{n=1}^{10}$
$\frac{1}{10}\sum_{n=1}^{10}$
Compute AVG: $\langle i_L\rangle$ $\langle v_C\rangle$
voltage PI current PI
COMPUTE DUTY CYCLE
SET DUTY CYCLE PWM_0
FUZZY/LOGIC
FAULTS?
YES
FAULT FLAG
NO
R_o CALCULATION
INVERSE MODEL
Compute: ϕ_1 ϕ_2 δ_1 δ_2
$\frac{1}{10}\sum_{n=1}^{10}$
Compute AVG: $\langle\phi_1\rangle$ $\langle\phi_2\rangle$ $\langle\delta_1\rangle$ $\langle\delta_2\rangle$
END
INTERRUPT_0
pwm_level=1
Call ADC_MEAS.
START PWM_1
END
INTERRUPT_1
pwm_level=0
END

Figure 14. Software execution flowchart.

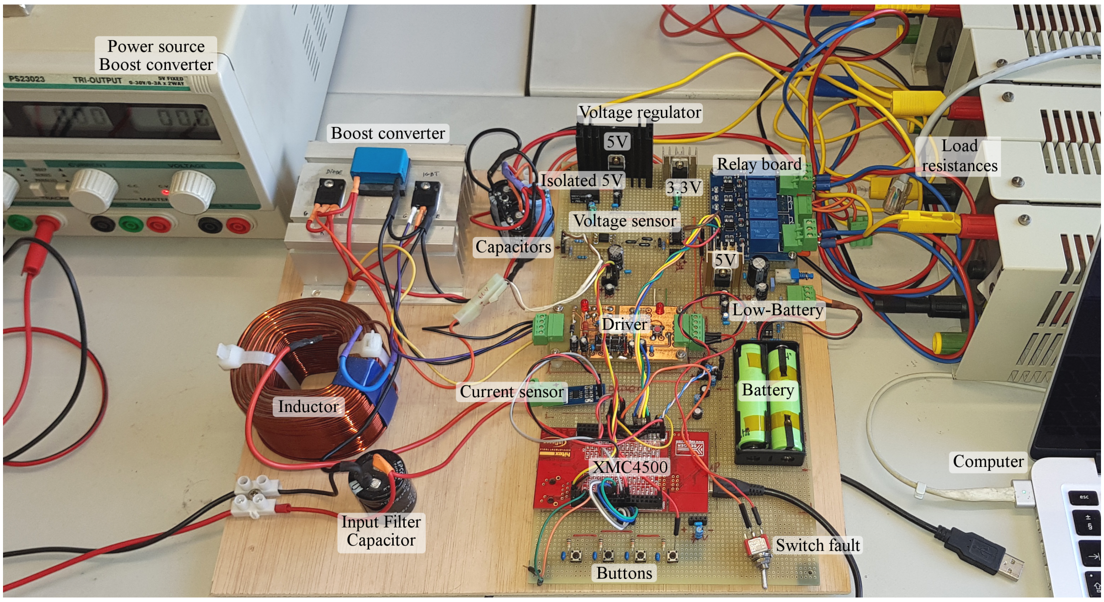
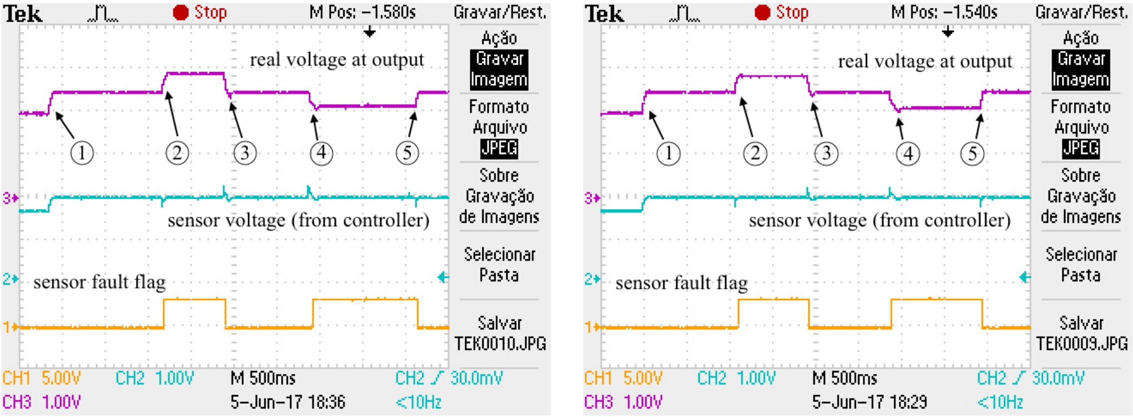
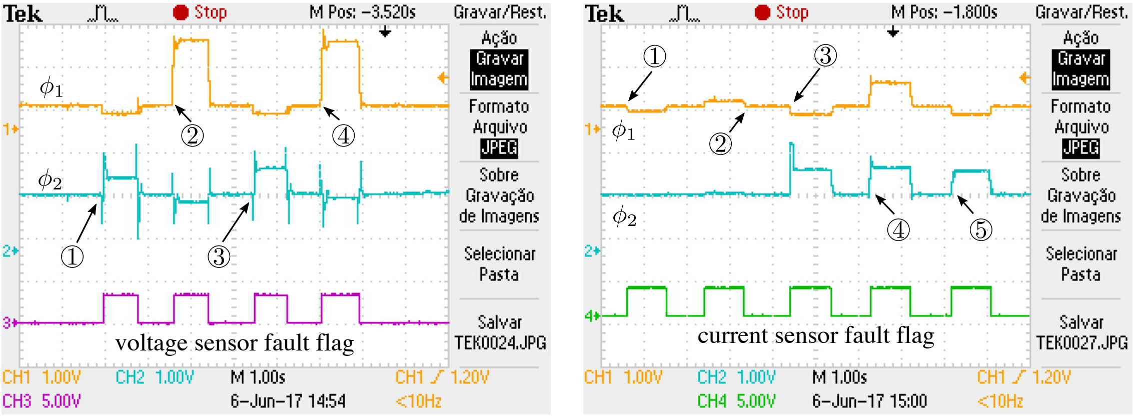


Figure 15. Experimental setup at the laboratory.



(**a**) Gain (**b**) Offset

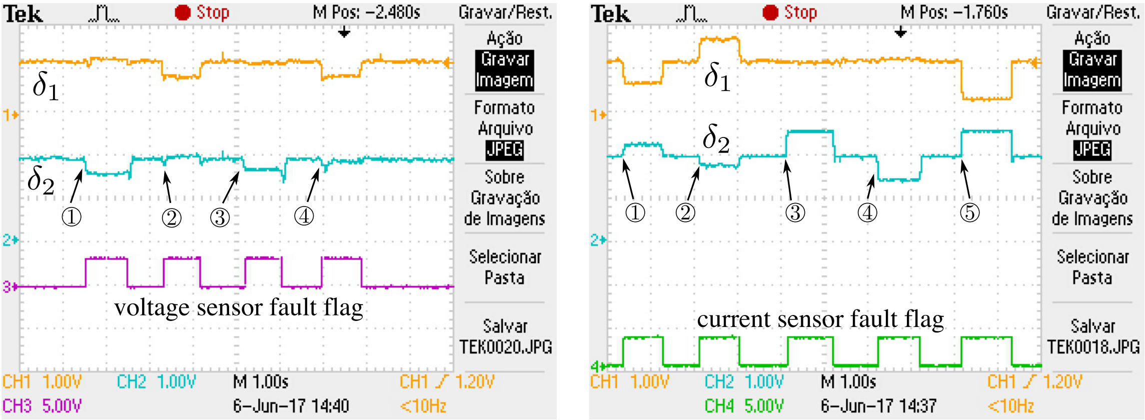
Figure 16. Experimental detection and isolation of faults in voltage sensor.



(**a**) Voltage sensor faults (**b**) Current sensor faults

Figure 17. Experimental FDI in sensors using ϕ_1 and ϕ_2 signals.

Figure 18, it shows the evolution in time of variables δ_1 and δ_2, when faults are injected in voltage and current sensors. The time injection of faults is the same as before, as described above. Inspecting both figures, one can see that variables δ_1 and δ_2 are more sensitive to current sensor faults, as shown by simulations in the previous section. Moreover, as shown in both figures, all injected faults were detected and the corresponding flags were triggered. Once again, it is important to note that in order to isolate these faults the algorithm must also have information of the variables ϕ_1 and ϕ_2.

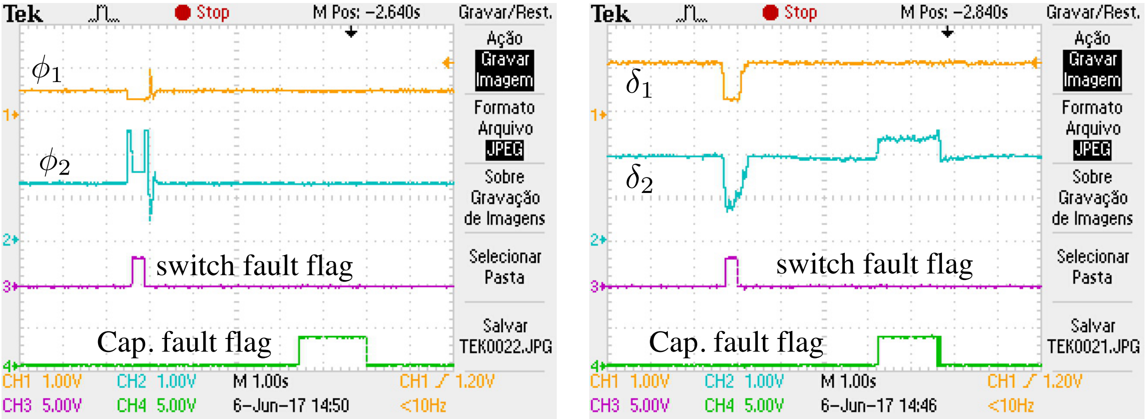


(**a**) Voltage sensor faults (**b**) Current sensor faults

Figure 18. Experimental FDI in sensors using δ_1 and δ_2 signals.

Regarding switch and capacitor faults, they were injected by activating physical switches on the main-board. For injecting the switch open fault, the signal from its gate was disconnected for a couple of seconds. The injection of the capacitor fault was done by pressing a 'pressure switch' in the board, which is connected to the microcontroller. The latter send the information to the 'relay board' that disconnects some of the capacitors (in parallel), which correspond to approximately 80% of its capacitance. It is worth mentioning that the disconnection of that capacitance value does not mean it stops working. In fact, the converter maintains its normal operation, only with a little more ripple magnitude, as illustrated in simulations.

Figure 19 shows the detection and isolation of the switch and capacitor faults. Again, it is necessary to make some reasoning considering the four signals to perform FDI. Nevertheless, as it is shown, the algorithm can isolate both faults with the considered signals. As referred previously, all variables used for isolation of faults are affected by faults occurring in the system. Thus, one can see some variation in all variables when a fault occurs. From the shown figures, one can see that, for each fault, each variable shows a particular symptom. The knowledge of all symptoms permits to infer about which fault is occurring at a given instant.



(**a**) ϕ_1 and ϕ_2 (**b**) δ_1 and δ_2

Figure 19. Experimental detection and isolation of faults in switch and capacitor.

6. Conclusions

This article has explored the combined used of two approaches with the purpose of fault detection and isolation applied to a particular case of a switched linear system. The first is related to the concept of left inversion of linear deterministic systems used in a perspective of fault detection. The latter is the application of a fuzzy logic system, which is able to infer about the faults in the system. Together they were used in a new approach for fault detection and isolation in switched linear systems. Particularly, an alternative formulation of a left inversion published firstly by Sain and Massey was introduced. Here the inversion algorithm consists of a bank of delay elements followed by a dynamical system for unknown input reconstruction. The effectiveness of the whole scheme in detecting and isolating the presence of four faults has been experimentally tested in a DC-DC boost converter.

Author Contributions: Conceptualization: A.M.S. and R.E.A.; formal analysis: A.M.S. and R.E.A.; investigation: A.M.S. and R.E.A.; methodology: A.M.S. and R.E.A.; software: A.M.S.; validation: A.M.S. and R.E.A.; resources: R.E.A.; supervision: R.E.A.; writing—original draft preparation: A.M.S.; writing—review and editing: A.M.S. and R.E.A.; funding acquisition: R.E.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by FCT—Fundação para a Ciência e a Tecnologia, within the project SAICTPAC/0004/2015—POCI-01-0145-FEDER-016434.

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

Abbreviations

The following abbreviations are used in this manuscript:

ADC	Analog to Digital Converter
FD	Fault Detection
FDI	Fault Detection and Isolation
FIS	Fuzzy Inference System
GUI	Graphical User Interface
IDE	Integrated Development Environment
MIMO	Multiple-Input Multiple-Output

References

1. Silverman, L. Inversion of multivariable linear systems. *IEEE Trans. Autom. Control* **1969**, *14*, 270–276. [CrossRef]
2. Sain, M.K.; Massey, J.L. Invertibility of linear time-invariant dynamical systems. *IEEE Trans. Autom. Control* **1969**, *14*, 141–149. [CrossRef]
3. Moylan, P. Stable inversion of linear systems. *IEEE Trans. Autom. Control* **1977**, *22*, 74–78. [CrossRef]
4. Bonilla Estrada, M.; Figueroa Garcia, M.; Malabre, M.; Martínez García, J.C. Left invertibility and duality for linear systems. *Linear Algebra Its Appl.* **2007**, *425*, 345–373. [CrossRef]
5. Lewis, F.L.; Christodoulou, M.A.; Mertzios, B.G. System inversion using orthogonal functions. *Circuits Syst. Signal Process.* **1987**, *6*, 347–362. [CrossRef]
6. Santamarina, J.C.; Fratta, D. *Discrete Signals and Inverse Problems: An Introduction for Engineers and Scientists*; John Wiley: Hoboken, NJ, USA, 2005; p. 350.
7. Bianco, C.G.L.; Piazzi, A. A servo control system design using dynamic inversion. *Control Eng. Pract.* **2002**, *10*, 847–855. [CrossRef]
8. Naderi, E.; Khorasani, K. Inversion-based output tracking and unknown input reconstruction of square discrete-time linear systems. *Automatica* **2018**, *95*, 44–53. [CrossRef]
9. Dong, J.; Verhaegen, M. Identification of fault estimation filter from I/O data for systems with stable inversion. *IEEE Trans. Autom. Control* **2012**, *57*, 1347–1361. [CrossRef]
10. Sundaram, S.; Hadjicostis, C.N. Delayed observers for linear systems with unknown inputs. *IEEE Trans. Autom. Control* **2007**, *52*, 334–339. [CrossRef]

11. Edelmayer, A.; Bokor, J.; Szabó, Z. Inversion-based residual generation for robust detection and isolation of faults by means of estimation of the inverse dynamics in linear dynamical systems. *Int. J. Control* **2009**, *82*, 1526–1538. [CrossRef]
12. Szigeti, F.; Bokor, J.; Edelmayer, A. Input Reconstruction by Means of System Inversion: Application to Fault Detection and Isolation. *IFAC Proc. Vol.* **2002**, *35*, 13–18. [CrossRef]
13. Hou, M.; Patton, R.J. Input observability and input reconstruction. *Automatica* **1998**, *34*, 789–794. [CrossRef]
14. Palanthandalam-Madapusi, H.J.; Bernstein, D.S. A subspace algorithm for simultaneous identification and input reconstruction. In Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 4956–4961. [CrossRef]
15. Kirtikar, S.; Palanthandalam-Madapusi, H.; Zattoni, E.; Bernstein, D.S. L-delay input and initial-state reconstruction for discrete-time linear systems. *Circuits Syst. Signal Process.* **2011**, *30*, 233–262. [CrossRef]
16. Vu, L.; Liberzon, D. Invertibility of switched linear systems. *Automatica* **2008**, *44*, 949–958. [CrossRef]
17. Martínez-Hernández, R.; Rincón-Pasaye, J.J. Fault diagnosis in linear multivariable systems: An inversion approach. In Proceedings of the 2011 IEEE Electronics, Robotics and Automotive Mechanics Conference (CERMA 2011), Cuernavaca, Morelos, Mexico, 15–18 November 2011; pp. 252–257. [CrossRef]
18. Szigeti, F.; Vera, C.; Bokor, J.; Edelmayer, A. Inversion based fault detection and isolation. In Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228), Orlando, FL, USA, 4–7 December 2001; Volume 2, pp. 1005–1010. [CrossRef]
19. Tanwani, A.; Domínguez-garcía, A.D.; Liberzon, D. An Inversion-Based Approach to Fault Detection and Isolation in Switching Electrical Networks. *IEEE Trans. Control Syst. Technol.* **2011**, *19*, 1059–1074. [CrossRef]
20. Tanwani, A.; Liberzon, D. Invertibility of Nonlinear Switched Systems. In Proceedings of the 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008. [CrossRef]
21. Pinheiro, V.; Araújo, R.E. Evaluation of applicability of system inversion to fault detection and isolation on switched power converters. In Proceedings of the Conference on Control and Fault-Tolerant Systems, SysTol, Nice, France, 9–11 October 2013; pp. 784–789. [CrossRef]
22. Silveira, A.; Araujo, R.E.; Ulson, J. Comparative study of inversion-based and observer-based approaches for fault diagnosis in DC-DC converters. In Proceedings of the 2017 IEEE 8th International Symposium on Power Electronics for Distributed Generation Systems (PEDG), Florianopolis, Brazil, 17–20 April 2017; pp. 1–6. [CrossRef]
23. Sain, M.; Massey, J. A modified inverse for linear dynamical systems. In Proceedings of the IEEE Symposium on Adaptive Processes (8th) Decision and Control, University Park, PA, USA, 17–19 November 1969; p. 51. [CrossRef]
24. Wang, G.; Wei, Y.M.; Qiao, S. *Generalized Inverses: Theory and Computations*; Science Press: Beijing, China, 2018; p. 378.
25. Du, D.; Jiang, B.; Shi, P. *Fault Tolerant Control for Switched Linear Systems*; Studies in Systems, Decision and Control; Springer International Publishing: Cham, Switzerland, 2015; Volume 21. [CrossRef]
26. Sundaram, S.; Hadjicostis, C.N. Designing Stable Inverters and State Observers for Switched Linear Systems with Unknown Inputs. In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 13–15 December 2006; pp. 4105–4110. [CrossRef]
27. Mathworks. *Fuzzy Logic Toolbox^{TM} User's Guide R2018b*; Technical Report; MathWorks, Inc.: Natick, MA, USA, 2018.
28. Isermann, R. *Fault-Diagnosis Systems—An Introduction from Fault Detection to Fault Tolerance*; Springer: Berlin/Heidelberg, Germany, 2006. [CrossRef]
29. Chen, G.; Pham, T. *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*; CRC Press: Boca Raton, FL, USA, 2001; p. 316.
30. Kazimierczuk, M.K. *Pulse-Width Modulated DC-DC Power Converters*, 2nd ed.; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2016; p. 963.