International Journal of Applied Metaheuristic Computing
Volume 15 • Issue 1

Multi-Fault Diagnosis Based on Hybrid Bio-Inspired Algorithm ACO-GA

Abid Sabrina, University Mustapha Stambouli, Mascara, Algeria*
https://orcid.org/0000-0002-1922-3963

Debbat Fatima, University of Mustapha Stambouli, Mascrara, Algeria

ABSTRACT

The fault detection and isolation (FDI) procedure increases the assurance of quality, reliability, and safety of industrial systems. Several faults may appear simultaneously, and the purpose of multi-fault diagnosis is to identify and locate these multiple faults. This work is particularly interested in the diagnosis based on the structural analysis of the system; residuals can be generated and used as fault indicators by model-based fault detection techniques. The isolation is dependent on the structure of the fault signature matrix. A new fault signature that represents the superposition of the fault is produced by simultaneous fault effects, resulting in an additional column in an extended signature matrix. This remedy is rather combinatorial. This research focuses on two methods to isolate multiple faults: (1) A modified enumerative method; (2) A hybrid ant colony optimization algorithm-genetic algorithm (Hybrid ACO-GA) is adapted to the MFD problem which has the advantage of a better research as well as the hybridization with GA.

KEYWORDS

Analytical redundancy relations (ARRs), Fault Detection and Isolation (FDI), Fault Signature matrix, Hybrid Ant Colony Optimization-Genetic Algorithm, Multi-fault diagnosis, Structural analysis

INTRODUCTION

Safety critical systems, such as aircraft, automobiles, nuclear power plants, and space vehicles, are becoming significantly more complex and interconnected. The recent advances in wireless technology, remote communication, computational capabilities, sensor technology, and standardized hardware/software interfaces have further increased the complexity of these systems. This complexity may result in failures of multiple components. Hence, there is a need to develop smart on-board diagnostic algorithms that can determine the most likely set of failure causes in a system, given observed test outcomes over time (Ouyang et al., 2023). Fault detection and isolation (FDI) is an essential aspect of Industry 4.0, which refers to the integration of digital technologies in manufacturing processes. FDI in Industry 4.0 involves the use of advanced monitoring and data analytics techniques to detect and isolate faults or anomalies in industrial systems (Webert et al., 2022; Kang et al., 2022).

DOI: 10.4018/IJAMC.341804 *Corresponding Author

In some systems, many defects may also manifest at once. This circumstance often arises when a first fault occurs, but the system has not been halted, either because the first fault was not serious or had a slow impact. In the meanwhile, a second problem arrives. When a reconfiguration approach is applied, this circumstance occurs more frequently. Reconfiguration techniques modify the control law, enabling production to go on even when a defect occurs.

The presence of two simultaneous faults in the system results in a new fault signature (Koscielny, 1993), which corresponds to the superposition of the two fault effects. According to the linearity hypothesis, processing the isolation of multiple faults using an expanded incidence matrix that includes a new column for each combination of faults may result in a combinatorial solution.

Multi-fault diagnosis (MFD) is a key issue in fault diagnosis technology because multiple faults commonly exist in engineering and complex systems. Zhang et al. (2015) did a survey on fault mechanism, manifestation of symptoms, and approach thinking for MFD. The MFD problem originates in several fields, such as medical diagnosis (Yu et al., 2007), error correcting codes, speech recognition, distributed computer systems, and networks (Odintsova et al., 2005). The MFD problem in largescale systems with unreliable tests was first considered by Shakeri et al. (1998).

The problem at the origin of this study is to study the possibility of the occurrence of several failures (or faults) simultaneously in a system, and that it is not interrupted or corrected after the first fault. The monitoring system in this case must diagnose the presence of different faults by making the best use of the structural properties of the system studied. MFD identifies multiple faults in a system, based on one or more symptoms and can be used as part of an overall diagnostic system or as a separate system. Its importance is evident in modern complex systems because they are systems characterized by the interconnection of several components, and the relationships describing the processes can be of different types (algebraic, differential). On the other hand, the complexity of this type of diagnosis plays an important role in the performance of an overall system.

Specifically, the objective of the research presented in this paper is to determine the possibilities of combining the effects of several defects on the system components. In addition, to study the situation in depth, two algorithms are presented and fully implemented to explore their possibilities and to evaluate the quality of their respective results. One is based on the standard approach that we have reduced, which consists in identifying and testing all possible combinations of faults, as well as the temporal and combinatorial inconsistencies that may occur. The other bio inspired algorithm ACO-GA uses different techniques of research and best exploration strategy to identify appropriate combinations of defects using simple elements.

This choice was guided by the fact that the reduced exact algorithm remains the reference in terms of complete, exhaustive, and exact solutions, and that the ACO-GA offers the best quality/performance ratio, especially when the system is large.

This paper is organized as follows: The first section introduces our research while the second section presents background about different methods for diagnosis and some definitions. Then the proposed approaches are presented in the third section. After that, in the fourth section we describe the steps of the ACO-GA algorithm. In the fifth section, we discuss the results obtained. Finally, a conclusion and references are given.

BACKGROUND

Early detection and isolation of faults are critical tasks in modern process industries. Many research works have been made during last decades to improve fault detection and isolation methods. Existing methods can be grouped into two general categories: model-based methods (Venkatasubramanian et al., 2003) and data-driven methods (Yin et al., 2014).

Data-Driven Diagnosis

Data-driven diagnosis, in general, isolates faults by using classifiers learned from training data using nominal data and data from different faults (Theissler, 2017). However, in many industrial applications, faults are rare events, and available training data from faulty conditions are usually limited (Sankavaram et al., 2015; Dong et al., 2017). Collecting a sufficient amount of data from relevant fault scenarios is a time-consuming and expensive process. Also, if there are faults that do not occur before several years of system operation time, they might not be considered during system development. Therefore, it is desirable that a diagnosis system is not only able to identify and localize known faults as they occur, but it should also be able to identify new types of faults and improve fault classification performance over time as new data are collected (Pulido et al., 2019). One solution to limited training data from different fault scenarios is the use of physical models.

Model-Based Diagnosis

An advantage of model-based methods, with respect to data-driven methods, is that fault isolation performance can be achieved without training data from different faults (Cordier et al., 2004). Even though the fault has not been observed before, it is possible to point out likely fault locations based on residual information and model analysis (Pucel et al., 2009). However, there are often many diagnosis candidates that can explain the triggered residuals, meaning that it can still be difficult to identify the actual fault.

Differences between the estimated and the actual behavior are symptoms or fault indicators. These differences are called residuals. Later, the residuals are evaluated aiming at localizing the fault. These structured residuals can be generated using different methods studied (Abid & Hafid, 2018), as shown in Table 1.

Most methods for residual generation are based on fault detection filter observers (Zhong et al., 2015; Chen & Speyer, 2000), parity space (Gertler, 1997; Medvedev, 1995), parameter estimation, and graphical and mathematical approaches (Medvedev, 1995; Jha et al., 2017). Essentially these methods compensate the lack of sensors by taking into account the dynamic nature of the considered system.

Fault isolation, or locating the defects in the system, is a crucial step in the fault diagnosis of industrial systems; with increasing system complexity, the system must be dependable and capable of early defect detection to function autonomously to reduce expenses and the threat of possible hazards.

In model-based diagnosis, fault isolation is mainly performed by matching the set of triggered residual generators with the different fault signatures to compute diagnosis candidates.

Despite the general acknowledgment of the two steps in FDI methods and the importance of each step, most publications focus on the residual generation step. This work focuses on the isolation step.

The major methods for fault isolation are based on the handling of multiple faults; one column for each possible multiple fault can be added to the isolation table. This approach was suggested in Gertler (1998). However, if N is the number of components, the number of columns needed is $2^N - 1$, which makes this approach computationally intractable for anything but small systems.

The second problem, multiple fault types per component, can be addressed by adding separate columns for each fault type (Nyberg, 2002, 2006). When combining this with a consideration of multiple faults, and letting M denote the number of fault types per component, the number of columns needed becomes $(M + 1)^N - 1$. Thus the computational burden is increased even further.

Multi-Fault Diagnosis Problem

Large-scale, complex, and dynamic systems that are described by a large number of equations, both linear and non-linear, can be studied using the broad framework of structural analysis. Equations are abstracted by retaining only their connections to variables. As a result, it disregards the specifics of parameter values in favor of basing the analysis on the system's structure using effective graph-based techniques (Cassar & Staroswiecki, 1997). Consequently, structural analysis has the primary benefit of

Table 1. A sample of the review papers of model-based methods

Reference	Type of analysis	The model used	Fault detection and isolation
Bouziani L., Haffaf H. (2010)	Structural analysis	structural model	multi fault isolation
Liu J. & al. (2016), Sidhu A. & al. (2015) Chatti N.& al. (2016), Jha, M.& al. (2017)	mathematical methods	mathematical model	Fault detection
Zhong, M. & al. (2010), Zhong, M. (2015)	Fault Detection Filter (FDF)	linear discrete time-varying system	
Hammouri H., Kabore P. (1998)		bilinear systems	Fault detection
Tidriri K.& al. (2018)	Structural analysis	Tennessee Eastman process	Fault detection
Chatti N. & al. (2014)	traction system of an intelligent and autonomous vehicle	structural analysis	multiple faults diagnosis
Yang Y. & al. (2015)	Observer based method	nonlinear systems	fault detection
Gertler J. (1997)	parity relations		Fault detection and isolation
Medvedev A. (1995)	parity space method	numerical example	isolation of sensor and actuator faults
Xu A., Zhang Q. (2004)	high gain observer and a recently developed linear adaptive observer	nonlinear systems	
Besancon G. (2003)	high-gain observer	three-tank system benchmark	fault detection
Periasamy V. (2019)	non-linear observer	Solid Oxide Fuel Cell (SOFC) systems	Fault identification

being applicable to uncertain systems for which the analytical model is not fully understood (Cassar & Staroswiecki, 1997; Düstegör et al., 2006).

Analytical redundancy is used in the fundamental MFD techniques, such as parity equations (Gertler, 1997) and state and output observers (Patton & Chen, 1997), to achieve the fault isolation function. M residuals r_j $(i = 1 \ldots M)$ are generated. When a fault F_j $(j = 1 \ldots N)$ occurs, some residuals stay close to zero, and others become different from zero. The $D(M \times N)$ signature matrix typically translates the residual state into Boolean terms 0 or 1 using a threshold.

$$D\left(M \times N\right) = \begin{pmatrix} F_1 & \cdots & F_j & \cdots & F_N \\ 1 & \cdots & 0 & \cdots & 1 \\ 0 & \cdots & 1 & \cdots & \cdots \\ 1 & \cdots & 0 & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \end{pmatrix} \begin{matrix} r_1 \\ r_2 \\ \vdots \\ r_i \\ r_M \end{matrix}$$

The isolation function's goal is to identify where in the system fault F_j first manifests itself. Column *j* of the signature matrix displays the fault F_j signature. The signature matrix structure affects the capacity to isolate faults (Gertler & Anderson, 1992).

It is not a simple matter to compare the columns of the Boolean incidence matrix with the residual vector.

In real industrial systems, several failures or faults can occur simultaneously. In general, this situation occurs when the system has not been interrupted or corrected after the first occurrence of a fault. For this reason, MFD can be used as a part of a global diagnostic system or as a complete system depending on the type of system in which it is used. Its purpose is to identify multiple faults in a specific system based on one or more symptoms.

Modeling for MFD

In the case of MFD, the diagnosis matrix must include the signatures of the multiple faults so that they can be considered in the decision phase. However, the number of possibilities evolves exponentially with the number of faults or components faults. Moreover, multiple faults in dynamic systems are difficult to detect because they can mask or offset each other's effects. Any assumption of a single fault can lead to a misdiagnosis when multiple faults occur (Daigle et al., 2006).

Algorithms for diagnosing multiple faults are therefore more complex than those for single faults for two reasons: First, the effects of one fault may be masked or compensated for (also called protected) by the effects of another fault. The second difficulty is that the same multiple faults may manifest themselves in different ways depending on the order in which they occurred.

According to Weber et al. (1999) and under the assumption of linearity, the isolation of multiple faults can be handled by employing an extended signature table, including a new column for each fault combination, leading to a combinatorial solution. The number of columns of the extended incidence matrix can be equal to $2^n - 1$ if all multiple faults are considered (n being the number of simple components of the system).

The theoretical signature of a multiple fault is usually obtained from the signatures of the simple faults. Let us consider that F_j is a multiple fault corresponding to the simultaneous occurrence of k simple faults $F_1, \ldots F_k$, then, the entries of the signature vector of F_j is given by:

$$\begin{cases} S_{ij} = 0 & if\ S_{i1} = S_{i2} = \ldots = S_{ik} = 0 \\ S_{ij} = 1 & else\ i.e.\ \exists S_{il}\ with\ l \in \{1,\ldots,k\}\ such\ as\ S_{il} = 0 \end{cases} \quad (1)$$

Thus, the effects of the faults are added, and the new signature produced by the multiple faults is given by a logical " OR " between the signatures of the different single faults (Weber et al., 1999). Moreover, the fact that the signatures of the multiple faults are obtained from the signatures of the single faults expresses the intuitive idea that a multiple fault can only affect an ARR if, and only if, at least one of the single faults of which it is composed affects that ARR. This means that the scope of a multiple fault is the union of the scopes of its single fault constituents.

In this work, we will focus on model-based diagnosis. The model-based diagnosis is based on the use of a model of the physical system (Abid & Hafid, 2012, 2013). This model is described using a formalism language adapted to the needs of the application. When the data of the model differ from those of the physical system, an anomaly is detected. The detected error is then explained by providing a diagnostic space (a diagnostic being a set of faulty components).

Definition 1: A component ***x*** is protected by the failure of component ***y***, if the fault signature of ***x*** is included in that of ***y*** (Odintsova et al., 2004).

PROPOSED APPROACHES DESCRIPTION

The complexity of solving an MFD problem leads to a combinatorial difficulty, because the discovery of a solution that best explains all of the symptoms cannot be done in a "reasonable time."

The multi-fault localization problem is a difficult problem that cannot be solved in an exact way in an acceptable computation time. This section presents two approaches: The first one describes the modified exact methods for faults localization, and the second is an adaptation of the modified ACO with GA parameters to this multi-fault localization problem. The overall approach consists in iteratively modifying a set of initial solutions while hoping to reach a final solution respecting the constraints of the problem.

Problem Formulation

To create a mathematical formulation for multi-fault diagnosis, we need to define the problem in terms of variables, constraints, and objectives. Below is a simplified mathematical formulation that we can adapt based on the specifics of the system and diagnostic approach:

Variables

- $\boldsymbol{F_i}$: binary variable indicating whether fault *i* is present ($\boldsymbol{F_i} = 1$) or not ($\boldsymbol{F_i} = 0$)
- $\boldsymbol{D_i}$: binary decision variable indicating whether fault *i* is diagnosed ($\boldsymbol{D_i} = 1$) or not ($\boldsymbol{D_i} = 0$).
- $\boldsymbol{O_j}$: Binary variable indicating whether observation *j* is true ($\boldsymbol{O_j} = 1$) or false ($\boldsymbol{O_j} = 0$).

Objective

- Minimize the total cost, including the cost of undetected faults, the cost of false diagnoses, and any other relevant costs

Constraints

1. Fault detection constraints:
   - If a fault is present, it should be detected: $F_i \leq D_i \quad \forall i$
   - If a fault is diagnosed, it must be present: $D_i \leq F_i \quad \forall i$
2. Consistency constraints:
   - Ensure consistency between faults and observations: $F_i \leq \sum_j O_j \quad \forall i$. This constraint enforces that if a fault is present, at least one observation related to that fault should be true.
   - Ensure consistency between diagnosed faults and observations: $D_i \leq \sum_j O_j \quad \forall i$. This constraint ensures that a fault cannot be diagnosed unless there is at least one observation supporting it.
3. Limit on diagnosis faults:
   - Limit the number of diagnosed faults to a certain threshold: $\sum_i D_i \leq threshold$

Optimized Exact Method Approach

We optimized the enumerative method in order to validate the results by comparing them with the proposed GA-ACO approach.

The modification brought to the algorithm is the reduction of the FSM (with the use of the criteria 1 and 2, in the fourth section), which allows to reduce the possible combinations of faults.

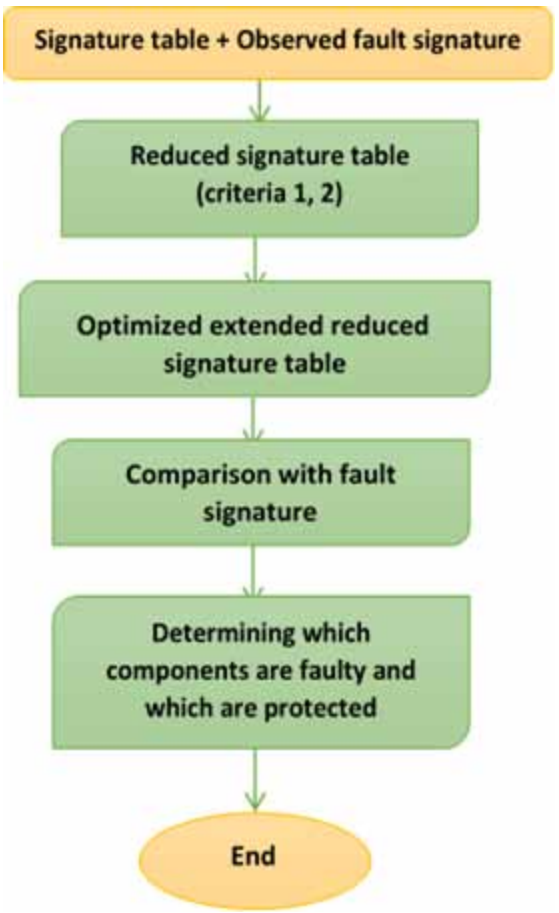
Fault isolation from the reduced FSM typically involves identifying the faulty components or subsystems in a system based on the observed fault signatures. The process generally follows these steps:

1. A fault signature matrix is constructed by observing the system's behavior under various fault conditions.
2. An observed fault signature is given.
3. Reduce the fault signature matrix with involving criteria 1 and 2 (see Section 4) then generate an extended FSM with all combinations of faults and calculate their signatures (see Equation 1).
4. Compare signatures: For a given observed fault signature, compare it with the extended reduced FSM. Measure the similarity or distance between the observed signature and the extended FSM.
5. Identify the faulty components or subsystems based on the comparison results.
6. Apply the principle of parsimony.

Figure 1 represents the approach adopted for the resolution of the fault location problem.
Data:

```
n= Number of simple components,
Bits=String of n characters (0 or 1)
```

Figure 1. The general flowchart of optimization by the optimized exact algorithm



```
            Table of components TAB = Table of simple component
names,
            New_Cmp = New combination of components to insert in
TAB,
```

Results

```
TAB table of the different possible combinations.
Begin
  // Initialization of Bits to n '0
   For i=1 to n
   Bits ← Bits + '0'
   // Possibility calculation loop
   Repeat
           Bits ← Bits + '1'         // Addition (+) dans le sens
binaire
   // The new component is the combination of those that are 1 in
Bits
          for i =1 to n
                 if Bits[i] = '1'
New_Cmp ← New_Cmp + TAB[i]
// Insertion
                           Increase TAB length
           TAB ←TAB + New_Cmp
// Stop conditions: 2^n possibilities
    until Bits contains only '1's
   End.
```

Hybrid Bio Inspired ACO-GA

For the ACO-GA approach, we have modified the basic version of the ACO algorithm by introducing blind and nonblind search in order to improve the exploration elaborated for the localization of defects, and we combined the genetic operators of crossing and mutation with the ACO algorithm.

This section presents an adaptation of the ACO-GA to the problem of fault location through the fault signature matrix. For this purpose, it evolves a population of several solutions in order to improve the individuals and, at each generation, a new set of individuals is proposed. We thus obtain a set of solutions (Figure 2).

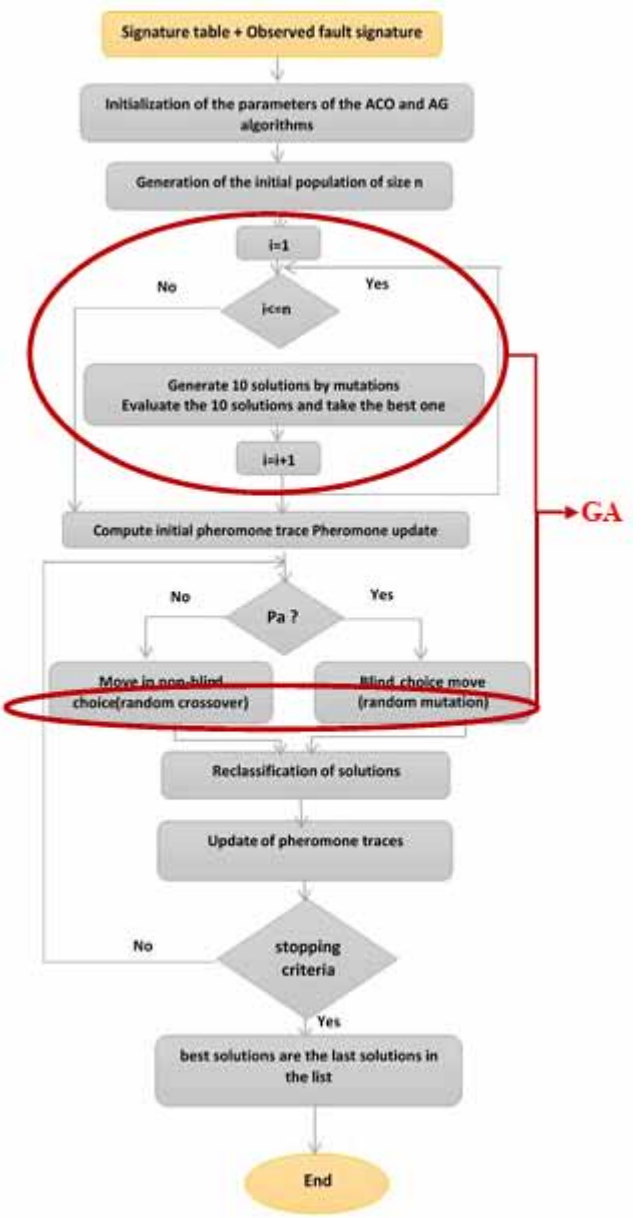
In our case, we will try to find the solutions that are closest to the observed fault signature, and for that we will start with a population of simple fault signatures. These signatures will be compared with the observation, and the ones that come closest (according to the Hamming distance) will be selected. Then, a crossing between faults will be performed in order to generate new signatures that will be compared, and the best ones will be selected. This operation must be repeated a number of times defined in the algorithm parameters. Several improvements can be made to this algorithm in order to optimize the results and the calculation time. In the following we will present these different improvements using the following example, which is based on a signature table of twenty-five (***25***) components and six (***06***) ARRs (Figure 3). This FSM is reduced using criteria 1 and 2 shown in Figure 4.

Table 2 shows the fault signature that will be used.

Notations Used in the Adaptation

Before describing the different steps of our algorithm, we need some notations:

Figure 2. The general flowchart of optimization by ACO-GA



- f: Number of ants, we identify in the following the ants and their solutions, stored in a table sorted by total decreasing cost, the best solution is thus the last one.
- ρ: Pheromone track persistence rate $0 \leq \rho < 1$.
- P_a: Probability of blind movement, ignoring pheromone traces.
- μ: Index of the ant.
- F_μ: Constant weight assigned to the ant number μ.
- L_μ: Cost of the solution found by the μ ants.

Figure 3. Binary fault signature matrix

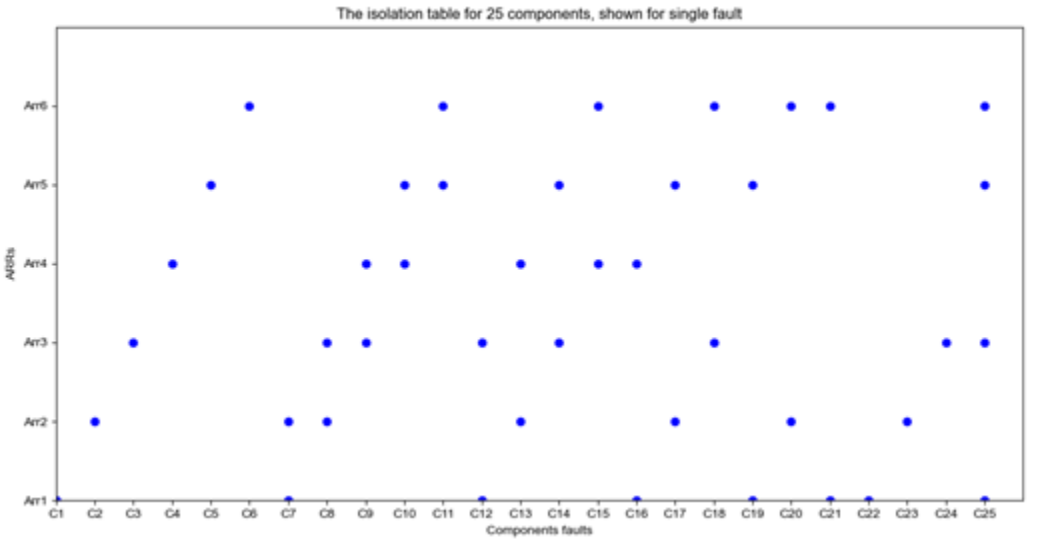
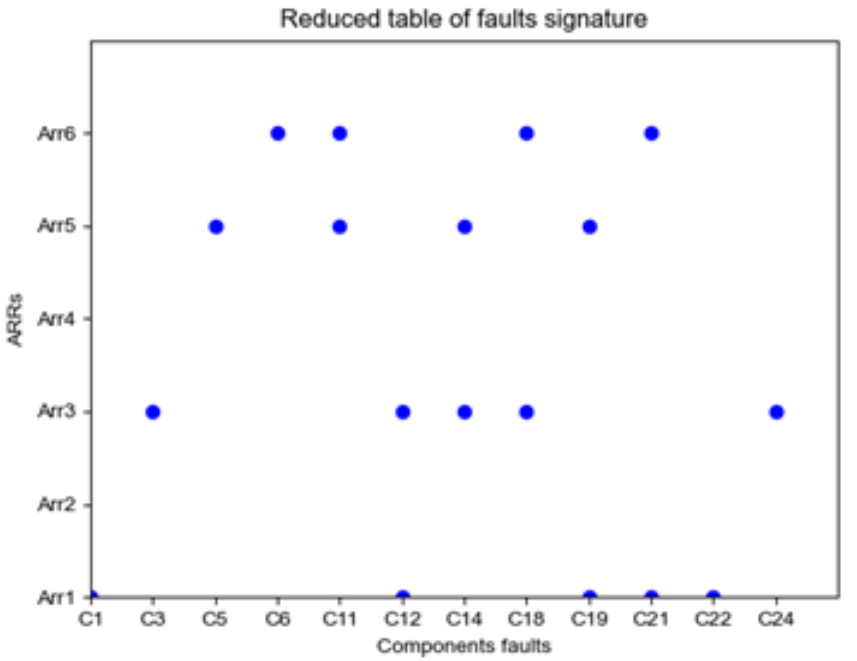


Figure 4. Reduced table of faults signature



DESCRIPTION STEPS OF ACO-GA ALGORITHM

The goal of the ACO-GA algorithm is not to find all possible solutions, as in the exact approach, but to find solutions that are as close as possible to the observation's signature and that involve as few components as possible. Indeed, the more components that are involved in the generation of a

Table 2. Fault signature observation

				Fault Signature
A	R	R	1	1
A	R	R	2	0
A	R	R	3	1
A	R	R	4	0
A	R	R	5	1
A	R	R	6	1

signature, the more the protection effect is amplified and therefore the further away from the real solution. These two objectives, i.e., to get closer to the fault signature and to minimize the number of components per solution, define the objective function of the ACO-GA algorithm that we wish to use.

Components Eligible to Participate in Solutions

When possible, it would be more interesting to generate only population components respecting the constraints. This will accelerate the convergence.

Criterion One

We perform a global search of the signature table for components with the same signature as the observation, before generating the initial population. These components will be excluded from the generation of individuals because their signature is the same as the one of the observation. Thus, any variation would not lead to a better solution, and any combination with another component would not bring any improvement to the individual but, on the contrary, would protect this newly added component. These components, which are of no help to the individual, are displayed at the end of the algorithm as exact solutions to the problem.

In the example we are dealing with, the component ***C25*** will be excluded from any participation to an individual of the population from the beginning. It will be re-displayed at the end as an exact result.

Criterion Two

On the other hand, when inserting a component into an individual, one must ensure that the signature of the component is a subset of the signature of the fault. That is, the component must not be sensitive to an ARR to which the fault is not sensitive. To implement this constraint, we will associate each component with its binary sum, which is the sum of the bits of its signature. Thus the signature of $\boldsymbol{C_1}$, which is only sensitive to the first ARR, will be the binary number (***100000***). We will use the function $\boldsymbol{\sigma_B}$ to calculate this number for each component. Thus, the condition quoted before, and for any component, $\boldsymbol{C_i}$ to be added to the individual, will be written:

$$\sigma_B\left(C_i\right) + \sigma_B\left(fault\right) = \sigma_B\left(fault\right) \tag{2}$$

Where the operator (+) represents the binary AND. The components $\boldsymbol{C_2}$, $\boldsymbol{C_4}$, $\boldsymbol{C_7}$, $\boldsymbol{C_8}$, $\boldsymbol{C_9}$, $\boldsymbol{C_{10}}$, $\boldsymbol{C_{13}}$, $\boldsymbol{C_{15}}$, $\boldsymbol{C_{16}}$, $\boldsymbol{C_{17}}$, $\boldsymbol{C_{20}}$, $\boldsymbol{C_{23}}$ do not satisfy this condition and will therefore be excluded from any participation in the generation of individuals.

The Coding of Individuals

Each solution of the treated problem represents an individual of the population. The individual is a set of components of the system, and each individual is represented as a sequence of bits (binary coding), where each one bit (***01***) indicates that the corresponding component is used by the individual. Thus, an individual ***I*** is represented by a chromosome ***100010100000***, which indicates that this individual uses the first, fifth, and seventh component in the list of components accepted as a solution. The number of bits of the chromosome is equal to the number of components admitted to participate in the solutions. Indeed, we will see later that not all the components can be used (admitted). On the other hand, the signature of an individual is the (binary) sum of the signatures of its different components. Thus,

$$Sign\left(I\right) = Sign\left(C_1\right) + Sign\left(C_5\right) + Sign\left(C_7\right) \tag{3}$$

Generation of Initial Solutions

The starting solutions of an ACO algorithm are in general random, each ant solution represents a sub-set of failed components, and the obtained solutions are on average poor but well dispersed in the search space.

To reinforce the initial population, we adopt the following procedure:

- for each solution of the initial population:
  - generate ***10*** solutions by performing permutations in the solution vector
  - evaluate the ***10*** solutions with the fitness function the best solution among the 10 will be kept

These initial solutions allow to compute the initial pheromone trace in the search space.

To generate the initial population, the number of individuals is a parameter of the algorithm that must be specified at the beginning. In our example, we set it to six (***06***) (Table 3). This number will increase with a high number of components. For implementation constraints, and in order to process the individuals pair by pair in order to apply the genetic operations to them, we require that the number of individuals in the population be an even number.

In our example, we will randomly generate individuals by drawing uniformly among the accepted components of the system. This choice was made in a completely random way in order to make the most of the performance of the ACO algorithm.

Population Evaluation

Once the initial population has been created, we will select the most promising individuals, i.e., those who will contribute to the improvement of the population. To do this, we will assign a "score," or quality index, to each of our individuals. This score is calculated through an Objective function. This function allows to show the notion of dominance between individuals. Each individual of the population is defined by two (02) functions:

- Δ_{H}: which represents the Hamming distance between its signature and the fault signature (its value varies between 0 and $\left|ARR\right|$). In other words, this function calculates the number of different bits between the individual's signature and the fault signature.
- ***Nb***: which gives the number of elements of which the individual is composed (its value varies between 02 and $\left|ARR\right|$). The maximum value of the function Nb is conditioned by the fact that

a combination of more than $|ARR|$ elements in an individual would be non-localizable by the initial signature table of the system.

The individual is more interesting if the value of Δ_H is close to ***0*** (i.e., the same signature as the fault) and if the value of ***Nb*** is close to ***2*** (i.e., the minimum number of elements). Therefore, the algorithm aims to achieve these two objectives.

The Global Objective Function

To determine the Objective function for each individual, we will use the two functions Δ_H and ***Nb*** in a new Fitness function, such that:

$$fitness(I) = \frac{|ARR| - \Delta_H(I)}{|ARR|} \times \frac{|ARR| - Nb(I) + 1}{|ARR|} \quad (4)$$

where: $|ARR|$ is the maximum number of ARRs in the signature table.

It is clear in this case, that a Fitness equal to one (01) corresponds to a simple and exact solution of the problem (i.e., ***Nb = 1 and*** Δ_H ***= 0***). We thus obtain the following results for our example:

General Iteration Steps

Updating the Pheromone Traces

At the beginning, for any individual ***i*** (set of failed components), the amount of pheromone τ_i is zero. We update it at the beginning of each iteration according to the following formula, which includes one term for evaporation and one for enhancement.

$$\tau_i = \rho\tau_i + \sum_{i=1}^{f} \Delta\tau_i \quad (5)$$

where: $\rho\tau_i$ Evaporation term

$\sum_{i=1}^{f} \Delta\tau_i$ Rein forcement term

With $\Delta\tau_i = \frac{F^{\mu}}{L^{\mu}}$

Table 3. Calculation of the evaluation functions on the example

Individuals	Chromosomes	Signatures	Hamming distance (Δ_H)	Number of elements (*Nb*)	Evaluation function
1	101000000000	100010	2	2	0,555556
2	000000001000	100010	2	1	0,666667
3	000010000100	100011	1	2	0,694444
4	011000100101	101011	0	5	0,333333
5	010000001000	101010	1	3	0,555556
6	000000000111	101001	1	3	0,555556

A simple idea is to assign equal weights $F^{\mu} = 1$ to the ants, but then the quality of the solution is ignored in this additional weighting of the solutions. Since the ***f*** solutions are sorted by decreasing cost, a better option is to use the rank of the ant by posing $F^{\mu} = \mu$: the rank of the ant.

Moving Ants to Build New Solutions

Each ant μ constructs a solution by choosing a set of failed components in a probabilistic way (ignoring the pheromone traces) or by taking the traces into account; blind choices are necessary to explore new solutions.

Structure of the Algorithm

The population of ***f*** ants actually consists of ***fe*** "elitist" ants and ***f –fe*** "non-elitist" ants. The elitists ensure the convergence of the algorithm, while the non-elitists explore the search space to maintain the diversity of solutions and prevent premature convergence. In the table of ***f*** solutions, the last solution of an elitist ant is replaced by the new one only in case of improvement. On the other hand, the last solution of a non-elitist ant is always replaced by its new solution, whether there is an improvement or degradation.

- **Blind choice move:**
  - For each ant in the list, we follow this approach if the probability condition is verified and ignoring the pheromone traces:
    1. Randomly choose two elements from the selected solution (ant).
    2. Make mutation between the two elements of the solution.
    3. Evaluate the obtained solution.
    4. If the ant is non-elitist, then replace it with the new one if there is an improvement.
    5. Reclassify the list of ants.
    6. Update the pheromones.
- **Move in nonblind choice**
  - For each ant in the list, if the probability condition is not verified, we follow the following approach taking into account the pheromone traces and the problem constraint:
    1. Randomly choose a cell from the selected solution (ant).
    2. Change the switch assigned to the cell by the switch that has the best pheromone concentration for this cell and compared to any solution in the list.
    3. Evaluate the resulting solution.
    4. Replace the solution if the ant is non-elitist Otherwise Replace the solution (elitist ant) if there is an improvement.
    5. Reclassify the solutions in the list according to the value of the cost function.
    6. Update the pheromone traces.

SIMULATION AND RESULTS

Parsimony is a principle that consists in using only the minimum number of elementary causes to explain a phenomenon. For example, in fault diagnosis, this principle favors the occurrence of a minimum number of faults (even a single fault) to explain the observed symptoms. In an MFD, this principle is used to simplify the diagnosis procedure. It is also called "principle of simplicity," or "principle of economy," because it excludes the multiplication of reasons and demonstrations within a logical construction. Thus, during MFD, the principle of parsimony is applied in case of difficulty in determining which faults are the most probable: These are a priori those that are the simplest, and we prefer to minimize the set of combinations of components in faults.

Exact Method With Optimized Extended Reduced Signature Table

Once the components (or groups of components) with the corresponding signature are identified, they can be examined in detail to simplify them. It is preferable to keep the simplest combinations at the expense of those that contain many components.

Any component whose signature does not bring any new element to the solution of the problem is considered as protected, even if its signature resembles that of the fault. This optimization method can be applied to any column containing a large number of possible combinations.

This execution time was reduced after modification of the exact algorithm by introducing the reduced signature table (criteria 1 and 2). The classical exact algorithm could not give results for 25 components because of the time and space complexity.

ACO-GA Algorithm

The ACO-GA algorithm converges quickly to the best solution compared to the exact method; the ACO-GA algorithm gave us the same results as the exact algorithm in a minimum of execution time in only 10 iterations, which explains the importance of the design of our ACO-GA algorithm (initial population that was studied with genetic parameters, the objective function, and the new search method adopted by the ants: blind and nonblind).

CONCLUSION

The problem of MFD consists of identifying one or more problems or faults that are most likely responsible for a given set of symptoms. Due to its complexity, this problem cannot currently be solved exactly, especially for large instantiations.

Table 4. Global list of faulty components

Failed components	Protected components	Components that cannot be part of the solution
C_{25} C_{11} C_{18} (C_{12}, C_{14}) (C_{19}, C_{21}) (C_3, C_{24}) (C_1, C_{22})	C_5 C_6	$(C_2, C_4, C_7, C_8, C_9, C_{10}, C_{13}, C_{15},$ $C_{16}, C_{17}, C_{20}, C_{23})$

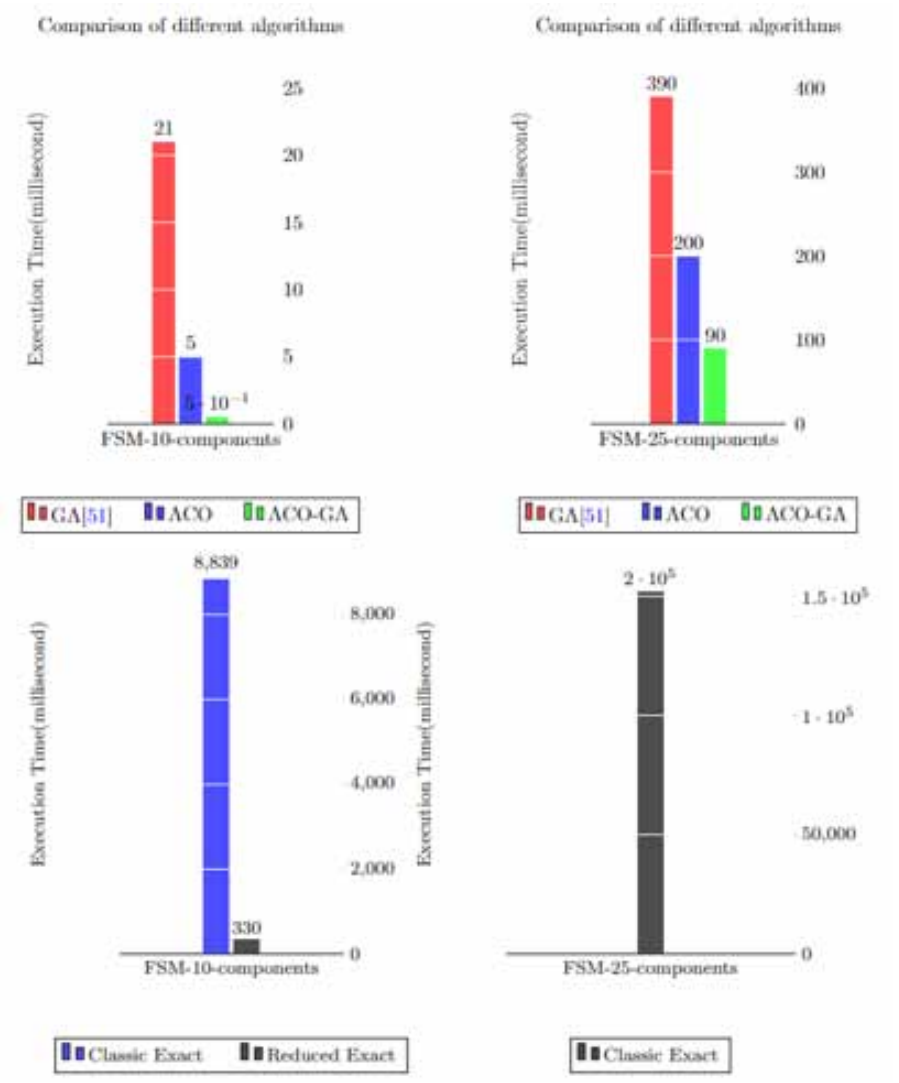
Table 5. ACO-GA parameters

Parameters	Value
population size P_a evaporation rate iteration number crossover Mutation	*10* *0.8* *0.3* *10* single point random one bit

Table 6. Global list of faulty components

Failed components	Protected components	Components that cannot be part of the solution
C_{25} C_{11} C_{18} (C_{12}, C_{14}) (C_{19}, C_{21}) C_{24} C_{18}	C_1, C_3 C_5, C_6 C_{22}	$(C_2, C_4, C_7, C_8, C_9,$ $C_{10}, C_{13}, C_{15}, C_{16},$ $C_{17}, C_{20}, C_{23})$

Figure 5. Comparison of different algorithms



In this paper, the adapted heuristic is the modified ant colony algorithm with GA. It was inspired by the field of swarm intelligence, which studies the collective intelligent behaviors of animals and uses sophisticated exploration techniques to avoid blocking at a local minimum.

The implementation of the method consisted in

- reduce the number of components with two criteria to encoding individuals.
- reinforce the initial population with permutations.
- define a series of moves (blind and nonblind search) for the solutions. These moves aim at improving the cost of the solution and restoring its feasibility. This led us to define a payoff structure with updated procedures to efficiently choose the best solution at each iteration and better explore the search space.

The results obtained are satisfactory and show the efficiency of the adopted method. Our implementation has also been tested against different parameters of the ACO-GA method to improve its robustness and efficiency. Thus, we studied, among others, the influence of the population size (number of ants), the number of iterations, and the type of mutation and crossover. Finally, the results were compared with those obtained by the reduced exact algorithm applied to the same example, which confirmed the efficiency of the adaptation.

Despite the satisfactory results obtained, the performance of our approach strongly depends on the choice of parameters. These are not easy to choose, and the values used in the program are values that give globally good results, but which are not necessarily the best for a given type of problem.

ACKNOWLEDGMENTS

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

REFERENCES

Abid, S., & Hafid, H. (2012). Sensor placement optimization for FDI: Graph tripartite approach. *International Journal of Systems Control*, *3*(1), 26–35.

Abid, S., & Hafid, H. (2013). A tripartite graph approach for optimizing sensor placement problem. *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC),* (pp. 108-113). IEEE. doi:10.1109/IWCMC.2013.6583543

Abid, S., & Hafid, H. (2018). A tripartite graph approach for optimal sensor diagnosis placement. *Electrotehnica, Electronica, Automatica (EEA), 66*(3), 95-105.

Besancon, G. (2003). High gain observation with disturbance attenuation and application to robust fault detection. *Automatica*, *39*(6), 1095–1102. doi:10.1016/S0005-1098(03)00059-1

Cassar, J., & Staroswiecki, M. (1997). A structural approach for the design of failure detection and identification systems. In IFAC conference on control of industrial systems (pp. 841-846). Science Direct. doi:10.1016/S1474-6670(17)43469-0

Chatti, N., Guyonneau, R., Hardouin, L., Verron, S., & Lagrange, S. (2016). Model based approach for fault diagnosis using set membership formulation. *Engineering Applications of Artificial Intelligence*, *55*, 307–319. doi:10.1016/j.engappai.2016.08.001

Chatti, N., Ould-Bouamama, B., Gehin, A.-L., & Merzouki, R. (2014). Signed bond graph for multiple faults diagnosis. *Engineering Applications of Artificial Intelligence*, *36*, 134–147. doi:10.1016/j.engappai.2014.07.018

Chen, R., & Speyer, J. (2000). A generalized least squares fault detection filter. *International Journal of Adaptive Control and Signal Processing*, *14*(7), 747–757. doi:10.1002/1099-1115(200011)14:7<747::AID-ACS619>3.0.CO;2-F

Cordier, M. O., Dague, P., Levy, F., Montmain, J., Staroswiecki, M., & Trave-Massuyes, L. (2004). Conflicts versus analytical redundancy relations: A comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, *34*(5), 2163–2177. doi:10.1109/TSMCB.2004.835010 PMID:15503514

Daigle, M., Koutsoukos, X., & Biswas, G. (2006). *Multiple fault diagnosis in complex physical systems*. Vanderbilt University.

Dong, L., Shulin, L., & Zhang, H. (2017). A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples. *Pattern Recognition*, *64*, 374–385. doi:10.1016/j.patcog.2016.11.026

Düstegör, D., Frisk, E., Cocquempot, V., Krysander, M., & Staroswiecki, M. (2006). Structural analysis of fault isolability in the damadics benchmark. *Control Engineering Practice*, *14*(6), 597–608. doi:10.1016/j.conengprac.2005.04.008

Gertler, J. (1997). Fault detection and isolation using parity relations. *Control Engineering Practice*, *5*(5), 653–661. doi:10.1016/S0967-0661(97)00047-6

Gertler, J. (1998). *Fault detection and diagnosis in engineering systems*. Marcel Dekker Inc.

Gertler, J., & Anderson, K. C. (1992). An evidential reasoning extension to qualitative model based failure diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, *22*(2), 275–288. doi:10.1109/21.148430

Ghilane, B. L., & Hafid, H. (2010). *Algorithmes et optimisations pour le diagnostic multi faute*. [Doctorate thesis, Oran University Es Senia].

Hammouri, H., & Kabore, P. (1998). A geometric approach to fault detection and isolation for bilinear systems, 2010. *IEEE Trans*.

Jha, M. S., Chatti, N., & Declerck, P. (2017). Robust fault detection in bond graph framework using interval analysis and fouriermotzkin elimination technique. *Mechanical Systems and Signal Processing*, *93*, 494–514. doi:10.1016/j.ymssp.2017.02.022

Kang, B., Na, W., & Lee, H. (2022). Model-based fault analysis and diagnosis of PEM fuel cell control system. *Applied Sciences (Basel, Switzerland)*, *12*(24), 12733. doi:10.3390/app122412733

Koscielny, J. M. (1993). Method of fault isolation for industrial processes. *Diagnostic et surete de fonctionnement. 3*(2), 205-220.

Liu, J., Luo, W., Yang, X., & Wu, L. (2016). Robust model-based fault diagnosis for pem fuel cell air-feed system. *IEEE Transactions on Industrial Electronics*, *63*(5), 3261–3270. doi:10.1109/TIE.2016.2535118

Medvedev, A. (1995). Fault detection and isolation by a continuous parity space method. *Automatica*, *31*(7), 1039–1044. doi:10.1016/0005-1098(95)00008-K

Nyberg, M. (2002). Model-based diagnosis of an automotive engine using several types of fault models. *IEEE Transactions on Control Systems Technology*, *10*(5), 679–689. doi:10.1109/TCST.2002.801873

Nyberg, M. (2006). A fault isolation algorithm for the case of multiple faults and multiple fault types. *IFAC Proceedings, 39*(13), 873-878.

Odintsova, N., Rish, I., & Ma, S. (2004). *Multi fault diagnosis in dynamic systems. IBM T. J*. Watson Research Center.

Odintsova, N., Rish, I., & Ma, S. (2005). Multifault diagnosis in dynamic systems. *Proceedings of IM*.

Ouyang, H., Zou, E., Wang, C., & Lin, Y. (2023). Fault detection and isolation for a class of uncertain nonlinear systems: A dual monitoring model-based approach. *Journal of the Franklin Institute, 360*(1), 109-135.

Patton, R. I., & Chen, J. (1997). Observer based fault detection and isolation: Robustness and application. *Control Engineering Practice*, *5*(5), 671–682. doi:10.1016/S0967-0661(97)00049-X

Pucel, X., Mayer, W., & Stumptner, M. (2009). Diagnosability analysis without fault models. In *20th International Workshop on Principles of Diagnosis* (pp. 67–74). IEEE.

Pulido, B., Zamarreno, J. M., Merino, A., & Bregon, A. (2019). State space neural networks and model decomposition methods for fault diagnosis of complex industrial systems. *Engineering Applications of Artificial Intelligence*, *79*, 67–86. doi:10.1016/j.engappai.2018.12.007

Sankavaram, C., Kodali, A., Pattipati, K., & Singh, S. (2015). Incremental classifiers for data-driven fault diagnosis applied to automotive systems. *IEEE Access : Practical Innovations, Open Solutions*, *3*, 407–419. doi:10.1109/ACCESS.2015.2422833

Shakeri, M., Pattipati, K. R., Raghavan, V., & Patterson-Hine, A. (1998). Optimal and near-optimal algorithms for multiple fault diagnosis with unreliable tests. *IEEE Transactions on Systems, Man, and Cybernetics. Part C, Applications and Reviews*, *28*(3), 431–440. doi:10.1109/5326.704583

Sidhu, A., Izadian, A., & Anwar, S. (2015). Adaptive nonlinear model-based fault diagnosis of li-ion batteries. *IEEE Transactions on Industrial Electronics*, *62*(2), 1002–1011. doi:10.1109/TIE.2014.2336599

Theissler, A. (2017). Detecting known and unknown faults in automotive systems using ensemble based anomaly detection. *Knowledge-Based Systems*, *123*, 163–173. doi:10.1016/j.knosys.2017.02.023

Tidriri, K., Chatti, N., Verron, S., & Tiplica, T. (2018). Model-based fault detection and diagnosis of complex chemical processes: A case study of the Tennessee Eastman process. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*. doi:10.1177/0959651818764510

Venkatasubramanian, V., Rengaswamy, R., Yin, K., & Kavuri, S. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model based methods. *Computers & Chemical Engineering*, *27*(3), 293–311. doi:10.1016/S0098-1354(02)00160-6

Vijay, P., Tadé, M. O., & Shao, Z. (2019). Adaptive observer based approach for the fault diagnosis in solid oxide fuel cells. *Journal of Process Control*, *84*, 101–114. doi:10.1016/j.jprocont.2019.09.010

Weber, P., & Gentil, S. Ripoll, P., & Foulloy, L. (1999). *Multiple fault detection and isolation*. 14th World Congress of IFAC, Beijing.

Webert, H., Döb, T., Kaupp, L., & Simons, S. (2022). *Fault handling in industry 4.0: Definition, process and applications*.

Xu, A., & Zhang, Q. (2004). Nonlinear system fault diagnosis based on adaptive estimation. *Automatica*, *40*(7), 1181–1193. doi:10.1016/j.automatica.2004.02.018

Yang, Y., Ding, S. X., & Li, L. (2015). On observer-based fault detection for nonlinear systems. *Systems & Control Letters*, *82*, 18–25. doi:10.1016/j.sysconle.2015.05.004

Yin, S., Ding, S., Xie, X., & Luo, H. (2014). A review on basic data-driven approaches for industrial process monitoring. *IEEE Transactions on Industrial Electronics*, *61*(11), 6418–6428. doi:10.1109/TIE.2014.2301773

Yu, F., Tu, F., Tu, H., & Pattipati, K. (2007). Multiple disease (fault) diagnosis with applications to the QMR-DT problem. *Proceedings of Computing Communication and Control Technology International Conference*. IEEE.

Zhang, K., Zhou, D. H., & Chai, Y. (2015). Review of multiple fault diagnosis methods. 10.7641/CTA.2015.50262

Zhong, M., Song, Y., & Ding, S. X. (2015). Parity space based fault detection for linear discrete time-varying systems with unknown input. *Automatica*, *59*, 120–126. doi:10.1016/j.automatica.2015.06.013

Zhong, M., Zhou, D., & Ding, S. (2010). On designingH1 fault detection filter for linear discrete time-varying systems. *IEEE Transactions on Automatic Control*, *55*(7), 1689–1695. doi:10.1109/TAC.2010.2046921

Sabrina Abid was born in Mascara, Algeria in 1984 She received a doctorate degree in 2019, from Mascara University, Algeria. Currently she has a position as an conference professor at the Department of Computer Science Mascara University. My current research interests include diagnosis, MFD, sensor placement optimization, Artificial intelligence, deep learning. e-mail: abidsabrina2006@yahoo.fr

received her Master of Sciences in Space Technologies in 2002 from the Space Techniques Center (CTS) Algeria, 2007 from the University of Tlemcen, technical and Ph.D. in Electronics in Algeria. She is currently Associate Professor in the Department of computer Science at Mascara University, Algeria. Her research interests include Artificial Intelligence Applications, Optimization, Mobile Robotics Control, and Wireless Networks.