Residual selection for fault detection and isolation using convex optimization

Daniel Jung and Erik Frisk

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-151295

N.B.: When citing this work, cite the original publication.

Original publication available at:
https://doi.org/10.1016/j.automatica.2018.08.006

Copyright: Elsevier
http://www.elsevier.com/
Residual Selection for Fault Detection and Isolation Using Convex Optimization

Daniel Jung \textsuperscript{a,b} and Erik Frisk \textsuperscript{a}

\textsuperscript{a}Linköping University, Linköping, Sweden
\textsuperscript{b}The Ohio State University, Columbus, Ohio, USA

Abstract

In model-based diagnosis there are often more candidate residual generators than what is needed and residual selection is therefore an important step in the design of model-based diagnosis systems. The availability of computer-aided tools for automatic generation of residual generators have made it easier to generate a large set of candidate residual generators for fault detection and isolation. Fault detection performance varies significantly between different candidates due to the impact of model uncertainties and measurement noise. Thus, to achieve satisfactory fault detection and isolation performance, these factors must be taken into consideration when formulating the residual selection problem. Here, a convex optimization problem is formulated as a residual selection approach, utilizing both structural information about the different residuals and training data from different fault scenarios. The optimal solution corresponds to a minimal set of residual generators with guaranteed performance. Measurement data and residual generators from an internal combustion engine test-bed is used as a case study to illustrate the usefulness of the proposed method.

Key words: Fault detection and isolation, feature selection, model-based diagnosis, convex optimization, computer-aided design tools

1 Introduction

A model-based diagnosis system is typically based on a set of residual generators, sometimes referred to as monitors, to detect if faults have occurred or not [3]. Each residual generator is designed to monitor a specific part of the system and then, based on which residuals that trigger, a set of diagnosis candidates (fault hypotheses) can be computed [6].

There are two main motivational observations for this work. First, the number of possible residual generator candidates in general grows exponentially with the degree of redundancy of the model [18]. This means that in many cases there are significantly more candidates possible than what is needed to detect and isolate the faults. A second observation is that in realistic scenarios all candidate residual generators do not perform equally well, mainly due to the inherent uncertainties in the model and measurement noise. Fig. 1 shows a typical situation with a set of residuals that are all sensitive to the same fault. In an ideal case, all residuals in the plot should react in the gray regions, but clearly the detection performance varies and some has no clear reaction at all, making them less useful for this particular fault. Thus, selecting an appropriate subset of residual generators is a key step in the design process to ensure that satisfactory detection and isolation performance can be achieved at low computational cost.

![Fig. 1. A comparison of residuals sensitive to the same fault but with different detection performance. The gray-shaded intervals indicate where the fault is active.](image)

Even though residual selection is important to achieve
satisfactory fault detection and isolation performance, it has received relatively little attention compared to other steps in the model-based diagnosis system design, e.g., sensor selection [2,19,21] and residual generator design [1,10,29]. In previous works, for example [21,23,27], the residual generators are assumed ideal when formulating the residual selection problem. Residual selection by optimization has been proposed in [21] using a Binary Integer Linear Programming approach, in [27] using a greedy heuristic, and in adaptive on-line solutions in [5,20], also here assuming ideal performance. A main limitation with these methods is that quantitative residual performance is not taken into consideration in the residual selection process, i.e., assuming that the detection performance of all residuals in Fig. 1 are equal which is clearly not the case.

The main property to consider in the selection process is robustness in the detector with respect to model uncertainties and noise. One approach would be to model noise and model uncertainty using, e.g., probabilistic methods, see for example [7,30]. However, in general this is difficult unless uncertainties are well modeled by stationary random processes. The approach adopted here is to let measured data model the uncertainties and the effects of different faults.

Residual selection is closely related to the feature selection problem in machine learning [4,11,14]. Different feature selection algorithms for data-driven fault diagnosis have been proposed, for example [15,16]. Performance of feature selection algorithms depends on the quality of available training data [28]. Collecting representative data from different faults is time-consuming, costly, and often infeasible since it is not known exactly how different faults manifest. This means that available data from different faults is often limited and not representative of all fault modes and, most importantly, it is also assumed that data is limited and not representative of all realizations of each fault. A main contribution of this work is systematic utilization of the analytic model in the data-driven feature selection process, alleviating the fundamental problem of limited training data from different fault scenarios. The proposed residual selection algorithm can handle both single-fault and multiple-fault isolation. To illustrate the proposed algorithm, it is applied to a real industrial use-case with data from an internal combustion engine.

### 2 Model-based diagnosis

Before defining the residual selection problem, a summary of some model-based diagnosis notions needed is given in this section. Structural properties of residual generators are defined which will be used to formulate the fault isolability constraints in the residual selection problem. An ideal residual generator is defined as

**Definition 1 (Ideal residual generator)** An ideal residual generator $r_k(z)$ for a given system is a function of sensor and actuator data $z$ where a fault-free system implies the residual output $r_k(z) = 0$.

An ideal residual generator $r_k(z)$ is said to be **sensitive** to a fault $f_i$ if there exists a realization of the fault that implies that the residual output $r_k(z) \neq 0$ [27]. Information about which set of faults each residual is sensitive to can be summarized in a Fault Signature Matrix (FSM). An example is shown in Fig. 1 where a mark at position $(k,l)$ means that residual $r_k$ is sensitive to fault $f_l$. A fault $f_l$ is said to be **decoupled** in $r_k$ if the residual is not sensitive to that fault.

Instead of discussing single-faults and multiple-faults, the term **fault-mode** is used to describe the system state. A fault mode $F \subseteq F$ describes which faults that are present in the system and the no-fault case $F = \emptyset$ is denoted NF. Based on fault modes, the following definition of fault detectability and isolability will be used to formulate the residual selection problem [27].

![Table 1](image)

**Table 1** Fault signature matrix of residual set $R^*$.

<table>
<thead>
<tr>
<th>Residual</th>
<th>$f_{waf}$</th>
<th>$f_{pin}$</th>
<th>$f_{dec}$</th>
<th>$f_{Tic}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_2$</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>$r_{19}$</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>$r_{26}$</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_{27}$</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_{29}$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$r_{30}$</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Definition 2 (Fault detectability and isolability)
Let \( R \subseteq R_{\text{all}} \) denote a set of residual generators. A fault mode \( F_i \) is detectable in \( R \) if there exists a residual \( r_k \in \mathcal{R} \) that is sensitive to at least one fault \( f_l \in F_i \). A fault mode \( F_j \) is isolable from another fault mode \( F_i \) if there exists a residual \( r_k \in \mathcal{R} \) that is sensitive to at least one fault \( f_l \in F_i \) but not any fault \( f_j \in F_j \).

To determine if any of the residuals has deviated from its nominal behavior, different test quantities are used, such as thresholded residuals or cumulative sum (CUSUM) tests [22].

3 Problem formulation

A first thing to observe is that for a given model there can be many possible residual generators. In general, the number of candidates grows exponentially with the degree of model redundancy [18]. To illustrate this, consider the small example

\[
x = g(u), \quad y_i = x, \quad i = 1, \ldots, n
\]

where \( u \) is a known control input and there are \( n \) measurements of the unknown variable \( x \). With \( n = 1 \) there is only one possible residual generator, i.e., \( r = y_1 - g(u) \), but with an increasing \( n \) the number of possibilities increases. It is straightforward to realize that the number of residual generators based on a minimal number of equations is given by

\[
|\{\text{minimal residual generators}\}| = \binom{n + 1}{2}
\]

since any pair of two equations, from the set of \( n + 1 \) equations, can be used to compute a residual. This simple observation generalizes to more general models [18].

Now, consider a set of \( n_r \) residual generator candidates \( R_{\text{all}} = \{ r_1, r_2, \ldots, r_{n_r} \} \) that is sensitive to a set of \( n_f \) faults \( F = \{ f_1, f_2, \ldots, f_{n_f} \} \). Each residual generator is, if the model is perfect, sensitive to a subset of the faults. As stated above, it is assumed that there are training data available from all faults in \( F \) but data can not be assumed to be representative of all possible realizations of each fault. To illustrate this assumption, consider the use-case in Section 6. In the experimental test-bed, a set of specific fault realizations, for example different biases on sensors, are implemented and measurement data is obtained. However, these data are generally not representative for other fault realizations, e.g., intermittent faults or fault realizations with dynamic profiles.

The residual selection problem has a set of \( n_p \) performance requirements, including both fault detection and isolation requirements. Each requirement \( l \) will be associated with a performance function denoted as \( \Phi_l(R) \) where \( R \subseteq R_{\text{all}} \). The function \( \Phi_l(R) \) uses training data, but for brevity this dependence is implicit in the notation. The larger the value of \( \Phi_l(R) \), the better the residual set \( R \) is for performance property \( l \).

Utilization of the model structure in the formulation of the optimization problem, i.e., which faults each residual is sensitive to, will show to be beneficial. All candidate residual generators are not useful for each performance requirement and the fault signature matrix contains this information. Let the set \( R_l \) denote the set of candidate residual generators useful for property \( l \). For example, if the performance criterion is a detection property, \( R_l \) is all residuals structurally sensitive to that fault. If the performance criterion is an isolation property, \( R_l \) is all residuals that structurally isolate the fault [12].

The residual selection problem seeks a minimal set of residuals, given some objective function \( \Omega(R) \), that satisfies a set of performance constraints and formulated as:

\[
\begin{align*}
\min_{R \subseteq R_{\text{all}}} & \quad \Omega(R) \\
\text{s. t.} & \quad \Phi_l(R \cap R_l) \geq C_l, \quad l = 1, 2, \ldots, n_p
\end{align*}
\]

where each performance requirement \( l \) is bounded from below by \( C_l \). For minimal cardinality solutions, the objective function is \( \Omega(R) = |R| \). However, this choice makes (1) an NP-complete combinatorial problem which is not suitable for direct implementation. A key contribution of this work is a convex re-formulation of (1), that takes both residual detection performance and structural fault isolability of the residual candidates into consideration. This convex problem can then be efficiently solved using general-purpose solvers.

4 Evaluating residual performance using logistic regression

In the optimization problem (1) the residual set performance functions \( \Phi_l() \) play crucial roles. Here, the approach to measure the performance of a set of residual generators \( R \) is based on how well they can distinguish faulty data from fault-free data. The data-driven technique logistic regression [11] is used to evaluate classification performance. Regularized logistic regression models have been shown useful for feature selection since they can be formulated as convex optimization problems [17].

Let \( r[t] = (r_1[t], r_2[t], \ldots, r_{|R|}[t]) \) denote the sample of all residuals in \( R \) at time \( t \). The logistic regression model is composed of a linear combination of the independent variables, here the residuals, as \( \hat{r}[t] = \sum_{k=1}^{|R|} r_k[t] \beta_k + \beta_0 \) and the logit-function. The classifier parameters \( \beta = (\beta_1, \beta_2, \ldots, \beta_{|R|})^T \) and \( \beta_0 \) are the residual weights and the bias, respectively. Since \( \hat{r}[t] = \hat{r}[t] \beta \), where \( \hat{r}[t] = (r[t], 1) \) and \( \beta = (\beta^T, \beta_0)^T \), the logistic regression model
can be written as
\[ P(\Psi = \psi[t]|r[t]; \beta, \beta_0) = \frac{1}{1 + e^{-\psi(t)r[t]\beta}} \]  
where \( \psi[t] = 1 \) corresponds to that there is a fault \( f_i \) at time \( t \), \( \psi[t] = -1 \) that there is no fault, and \( \beta, \beta_0 \) are the tuning parameters of the classifier.

For a given set of training data, the optimal choice of parameters \( \beta \) and \( \beta_0 \) can be selected using Maximum Likelihood (ML) which is a convex problem [11]. The log-likelihood function is
\[ \ell(\beta, \beta_0; \psi, R) = -\sum_{t=1}^{N} \log \left( 1 + e^{-\psi(t)r[t]\beta} \right) \]  
where \( R \) is a matrix where the rows consist of the different samples \( r[t] \) and \( \psi \) is a response vector with the same number of rows as \( R \). The ML estimation of \( \beta \) and \( \beta_0 \) is achieved by minimizing the negative log-likelihood, e.g., using a Newton method where the gradient \( g \) and Hessian \( H \) of (3) can be computed as
\[ g = \frac{\partial \ell}{\partial \beta} = -R^T(p - \psi), \quad H = \frac{\partial^2 \ell}{\partial \beta \partial \beta^T} = -R^TWR \]  
where \( p \) is a column vector and element \( t \) is \( p[\bar{r}[t]; \bar{\beta}] = P(\bar{Y} = \bar{\psi}[t]|\bar{r}[t]; \beta, \beta_0) \) and \( W \) is a diagonal matrix where the diagonal element at position \((t,t)\) is \( p(\bar{r}(t); \bar{\beta})(1-p(\bar{r}(t); \bar{\beta})) \) [11].

5 A convex formulation of the residual selection problem

A convex formulation of the residual selection problem (1) is presented using \( L_1 \)-regularized logistic regression where multiple fault detection and isolation constraints are taken into consideration. First, the residual selection problem is considered for a single performance requirement where it is described how fault detection and isolation constraints are formulated for each requirement. Then, the global optimization problem is formulated including multiple requirements.

Let \( R^{(i)} \in \mathbb{R}^{N_i \times n_r} \), where \( i = 1, 2, \ldots, n_f \), denote sets of residual training data with \( N_i \) samples including both nominal data and data when fault \( f_i \) is present. The corresponding response vector is denoted \( \psi^{(i)} \in \mathbb{R}^{N_i} \).

Fault detection performance of a set of residuals to detect a fault \( f_i \) is evaluated using the subset of residual data \( R_i \) that contains the columns of \( R^{(i)} \) corresponding to \( R_i \subseteq R^{(i)} \). The residual set \( R_i \) is used to formulate the fault isolability requirement depending on which residual generators that are included in the set.

To ensure that the solution set is able to isolate a fault \( f_i \) from another fault \( f_j \), the set of residual generator candidates is defined such that \( f_j \) is decoupled in all candidates. Let \( J_i = \{ p = 1, 2, \ldots, r_p : f_j \not\in R_p \} \) denote the indices of the subset of residual generators where \( f_j \) is decoupled. Then, the candidate set is given by \( \mathcal{R}_i = \{ r_k \}_{k \in J_i} \) and the data set \( R_i \) is given by the corresponding columns in \( R^{(i)} \).

Consider first a single performance criteria. Finding a minimal subset of \( \mathcal{R}_i \) corresponds to finding \( \beta_0 \) and a sparse vector \( \beta \) in (3) such that the log-likelihood exceeds some lower bound \( C_I \). It is assumed that the performance requirement \( C_I \) is defined such that there exists a feasible solution. The parameter \( C_I \) can be selected, for example, by tuning a logistic regression model (3), for all individual residual candidates, and then select a lower bound that corresponds to a satisfactory residual detection performance. Finding a minimal subset is a combinatorial problem but it is possible to force sparsity to an optimization problem by imposing \( L_1 \)-regularization [25]. Thus, the residual selection problem is formulated as an \( L_1 \)-regularized logistic regression problem [11]
\[ \min_{\beta, \beta_0} \| \beta \|_1 \quad \text{s.t.} \quad \ell(\beta, \beta_0; R_i) \geq C_I. \]  
The fault detection performance constraint \( C_I \) will determine the sparsity of the solution, i.e., how many residuals are required. A lower \( C_I \), i.e., a less restrictive constraint, will give a more sparse solution and vice versa. Note that \( \mathcal{R}_i \) contains residuals that are not sensitive to fault \( f_i \) because if noise in the residuals are correlated, fault detection performance can be improved by using residuals not sensitive to the fault as well. Note that residuals are typically correlated since noise originating from model errors are common to several residuals.

The objective is to find a minimal set of residual generators fulfilling a set of \( n_p \) performance requirements. One approach is to solve (4) for each requirement and then select the global solution as the union of the solutions to each individual problem [17]. However, this approach does not utilize that some residual generators could be used to solve multiple constraints and thus reducing the total number of residual generators even though the solution is not minimal for each individual constraint.

To distinguish the optimization variables for the \( n_p \) different requirements, the parameters for each requirement \( f_i \) are denoted \( \beta_i \) and \( \beta_{0i} \). A new cost variable \( \alpha_i \) is added where each element \( \alpha_i \) is given by
\[ \alpha_i = \max \{ |\beta_i| : r_k \in \mathcal{R}_i, \forall l = 1, 2, \ldots, n_p \}. \]  
Equation (5) defines the cost \( \alpha_i \) as the maximum parameter value \( |\beta_i| \) in all constraints where residual generator \( r_k \) is a candidate. This means that if \( r_k \) is used to fulfill one performance constraint, it is free to use for
other performance constraints, i.e., the total cost of using \( r_k \) in other performance constraints is not affected as long as the maximum value is not changed.

The global residual selection problem (1) can be formulated as the following convex optimization problem:

\[
\min_{\alpha, \beta} \sum_{k=1}^{n_c} \alpha_k \quad \text{s. t.} \quad \Phi_l(R \cap R_l; \beta, \beta') \geq C_l \\
- \alpha_j \leq \beta_j \leq \alpha_j \\
\forall l = 1, 2, \ldots, n_p
\]

where \( \Phi_l(R \cap R_l; \beta, \beta') = \ell(\beta, \beta'; R_l) \), \( \leq \) denotes elementwise \( \leq \), and \( \alpha_j \) is a column vector containing the elements in vector \( \alpha \) at indices \( J_l \). The objective function \( \Omega(R) \) is given by \( \sum_{k=1}^{n_c} \alpha_k \) where the solution set can be determined as \( R^* = \{ r_k \in R_{all} : \alpha_k > 0 \} \). Note that multiple-fault isolability can be included in the optimization problem as additional constraints by defining candidate residual sets \( R_l \) where the multiple faults are decoupled.

Let \( \alpha^* \) denote the solution found by the optimization problem. Since \( \alpha^* \) is likely, due to numerical reasons, to contain values close to zero instead of exactly zero, the solution residual generator set \( R^* \subseteq R_{all} \) is determined by thresholding each element in \( \alpha^* \) as \( \alpha^*_k \geq \epsilon \) where \( \epsilon \geq 0 \) is a threshold.

For efficient implementation of an interior-point method, the gradient and Hessian of the non-linear constraints (7) can be formulated as \( \bar{g}^T = (0^n, g_h^T, g_g^T, \ldots, g_f^T) \) and \( \bar{H} = \text{diag}(0_{n_x}, H_1, H_2, \ldots, H_{n_p}) \), respectively. The Hessian of (7) is block-diagonal and since the matrix \( \bar{H} \) can be large if there are large training data sets and many performance requirements in (7), memory is saved by using a sparse representation.

6 Case study

To evaluate the residual selection algorithm a set of residual generator candidates is generated to monitor a passenger car four cylinder turbo-charged internal combustion engine [17].

6.1 System description and data collection

The available measurements from the engine are the following eight sensor signals: pressure before throttle \( y_{pic} \), pressure in intake manifold \( y_{pm} \), ambient pressure \( y_{pamb} \), temperature before throttle \( y_{t_ic} \), ambient temperature \( y_{t_amb} \), air mass flow after air filter \( y_{Waf} \), engine speed \( y_s \), and throttle position \( y_{pos} \), and two actuator signals: wastegate actuator \( u_{w} \) and injected fuel mass into the cylinders \( u_{mf} \).

A mathematical model that describes the air flow through the engine is used with a similar model structure as described in [8], and is based on six control volumes and mass and energy flows given by restrictions. The model is a non-linear DAE and has 14 states. A schematic illustration of the model is shown in Fig. 2.

The proposed method can be applied to any type of faults, including system faults (leakages, clogging, etc.), sensor faults, and actuator faults. In this case study, 4 sensor faults are considered: A fault in the sensor measuring the air mass flow \( y_{Waf} \), the pressures at the intercooler \( y_{pic} \), and the intake manifold \( y_{pm} \), and the temperature at the intercooler \( y_{t_ic} \).

The engine is controlled to follow a load cycle corresponding to a Highway Fuel Economy Test Cycle (HWFET). Intermittent sensor faults are injected one by one in the engine control unit when the engine is running. The faults \( f_{Waf}, f_{pic}, \) and \( f_{pm} \) are injected as multiplicative faults \( y_i(t) = (1 + f_i)x_i(t) \) with a 20% change in the measured value and the fault \( f_{t_ic} \) as a sensor bias \( y_{t_ic}(t) = x_{t_ic}(t) + f_{t_ic} \) of 20°C. Note that some of the sensor faults affect the system operation. For example, the control system compensates for a change in sensor \( y_{Waf} \). An example of sensor data from \( y_{pm} \) is shown in Fig. 3 with an intermittent fault \( f_{pm} \).
6.2 Residual selection

A set of \( n_r = 64 \) residual generators is automatically generated from the engine model as a candidate set using the Fault diagnosis toolbox [13]. The residual generators are implemented in a sequential form, i.e., the set of model equations used in each residual generator is solved sequentially where the final equation is used as residual equation [26]. The FSM of the 64 residual generator candidates is shown in Fig. 4.

The residual selection problem (6)-(9) is formulated to find a minimal residual set such that all single-faults can be isolated from each other. This results in 12 isolation requirements. The four candidate sets representing the different residual sets where each fault is decoupled, are shown in Fig. 4. The residual selection problem is implemented in Matlab and solved using the general-purpose interior-point method available in \texttt{fmincon}.

Two sets of constraints, i.e., different values of \( C_i \) for each requirement (7), are evaluated, see Table 2. Position \((i,j)\) in the table shows the values of \( C_i \) to isolate fault \( f_i \) from fault \( f_j \). The two values of \( C_i \) in each position represent the two sets to be evaluated. The first set has lower values of the different \( C_i \) that represents less restrictive performance requirements while the second set has higher values representing tougher requirements. To make sure that there exists a feasible solution, each value \( C_i \) is selected within the range of values achieved when tuning a logistic regression model (3) for each of the residual candidates, separately.

![Fig. 4. Fault signature matrix for all residual generator candidates. Each plot illustrate the subset of the candidates where each fault is decoupled, respectively.](image)

![Table 2](image)

<table>
<thead>
<tr>
<th>( f_{Waf} )</th>
<th>( f_{pim} )</th>
<th>( f_{pic} )</th>
<th>( f_{Tic} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>((-1000, -373))</td>
<td>((-1000, -275))</td>
<td>((-1000, -210))</td>
<td>((-1000, -696))</td>
</tr>
<tr>
<td>((-1000, -733))</td>
<td>((-1000, -885))</td>
<td>(-1000, -846)</td>
<td></td>
</tr>
<tr>
<td>((-1000, -184))</td>
<td>((-1000, -240))</td>
<td>((-1000, -184))</td>
<td></td>
</tr>
</tbody>
</table>

The optimal solution vector \( \alpha^* \) for the less restrictive requirements is shown in the left plot in Fig. 5. The significant non-zero values in the vector, here defined when \( \alpha^*_k \geq 0.001 \), gives the solution set \( R^* = \{r_2, r_{19}, r_{26}, r_{27}, r_{29}, r_{30}\} \) containing six residuals and the corresponding FSM is shown in Table 1.

The solution set in Table 1 is compared to the solution when applying the residual selection algorithm proposed in [17]. The algorithm is implemented to select the single best residual generator for each requirement \( l \) to find a minimal solution set. The resulting solution set \( R' = \{r_{19}, r_{26}, r_{27}, r_{29}, r_{30}, r_{34}, r_{62}\} \) is found by taking the union of the selected residual generators for all requirements. When comparing \( R^* \) and \( R' \), five residuals are the same in both sets but the proposed residual selection strategy is able to find a smaller solution.

The solution set \( R^* \) is evaluated using data from each of the four faults and the different residual outputs are shown in Figs. 6-9, respectively. The gray areas represent the intervals when the fault is present and residuals that are sensitive to each fault are highlighted in red. The dashed lines represent thresholds tuned based on nominal data to illustrate nominal residual behavior. Most residuals react as expected when a fault occurs, except \( r_{19} \) in Fig. 6 which does not change significantly when \( f_{Waf} \) occurs. However, \( r_{19} \) is still useful since it is used to detect and isolate fault \( f_{pic} \), see Fig. 8.

![Fig. 5. The solutions \( \alpha^* \) to (6)-(9) for each set of requirements in Table 2, respectively. Each solution set corresponds to the non-zero elements in \( \alpha^* \). In both cases, the solution \( \alpha^*_k \geq 0.001 \) for all \( k > 35 \).](image)

![Fig. 6. Evaluation of residuals to data with fault \( f_{Waf} \). The grey areas represents intervals when fault is present and residuals sensitive to the faults are colored red.](image)
As a second case, the set of tougher performance constraints is selected. This results in a larger number of non-zero elements in the optimal vector $\alpha^*$ which is visible in the right plot in Fig. 5. The corresponding solution set is then $\mathcal{R}^* = \{r_2, r_{19}, r_{24}, r_{26}, r_{27}, r_{29}, r_{30}, r_{32}\}$. The solution set contains a larger set of residual generators to fulfill the tougher performance constraints.

Fig. 10 shows the solution vector $\alpha_k$ after each iteration of the interior-point method. The elements $\alpha_k$ that, eventually, are part of the solution $\alpha^*$ are highlighted in the plots. It is visible that the significant elements in vector $\alpha$ can be identified already after about 1000 iterations in these two cases while the other elements are decreasing in a step-wise manner. However, the selected interior-point method requires additional iterations to converge within set tolerances. The robustness of the optimization is evaluated by trying different randomly selected starting points. The solution converges in all tested cases and for this case study each optimization takes around seven minutes on a standard desktop computer.

7 Conclusions

The engine case study illustrates the importance of residual selection to achieve satisfactory performance of a diagnosis system. By including structural fault sensitivity information about the candidate residual generators in the residual selection problem, it is possible to fulfill isolability requirements even though available training data are limited. This is important in many fault diagnosis applications where collecting data can be both time-consuming and expensive. A key contribution is that the residual selection problem is formulated as a convex optimization problem where the optimal solution corresponds to a small set of residual generators that fulfills multiple fault isolation and detection performance constraints simultaneously with guaranteed performance. The residual selection approach can handle both single and multiple-fault isolation performance requirements and is successfully applied to an industrially relevant use-case which illustrates the efficacy of the approach.

References


