Design and Selection of Additional Residuals To Enhance Fault Isolation of A Turbocharged Spark Ignited Engine System*

Kok Yew Ng[1,3], Erik Frisk[2], and Mattias Krysander[2]

Abstract— **This paper presents a method to enhance fault isolation without adding physical sensors on a turbocharged spark ignited petrol engine system by designing additional residuals from an initial observer-based residuals setup. The best candidates from all potential additional residuals are selected using the concept of sequential residual generation to ensure best fault isolation performance for the least number of additional residuals required. A simulation testbed is used to generate realistic engine data for the design of the additional residuals and the fault isolation performance is verified using structural analysis method.**

I. INTRODUCTION

Fault diagnosis has always been an interesting and exciting area of research, especially with the advancements in automation and manufacturing [1]. It is crucial for these autonomous systems, be it robotic systems in a manufacturing plant or a self-driving vehicle, to know the health of the systems such that corrective measures can be carried out in the event of a failure. As such, a fault diagnosis scheme must be able to perform two main tasks: fault detection, i.e. the ability to determine if a fault is present in the system, and fault isolation, i.e. to locate the root cause of the fault [2].

These fault diagnosis schemes are usually designed using either hardware redundancy or analytical redundancy methods. The hardware redundancy method utilizes multiple identical sensors to measure the same variable of the system. A voting mechanism is then applied to determine the presence of a fault [3]. One of the main challenges of diagnostic systems is to improve fault isolation without adding physical sensors in order to reduce weight and the overall cost of the system. As a result, many of modern diagnostic systems are designed using model-based or analytical redundancy methods. These methods usually use observers, which are constructed using the mathematical equations describing the system dynamics, to estimate states of the system [4]. Using the same control input to drive both observer and the actual system, the difference between the actual outputs of the system and the estimated outputs of the observer is computed to produce the residuals, which are then processed to perform fault diagnosis [5]. One of the limitations to these techniques

[1]K. Y. Ng is with the Engineering Research Institute, Ulster University, Newtownabbey, BT37 0QB, UK. Emails: `mark.ng@ulster.ac.uk`

[2]E. Frisk and M. Krysander are with the Department of Electrical Engineering, Linköping University, Linköping 58183, Sweden. Emails: {`erik.frisk, mattias.krysander`}`@liu.se`

[3]K. Y. Ng is also affiliated with the School of Engineering, Monash University, 47500 Selangor, Malaysia. Email: `kok.yew.ng@monash.edu`

is that the number of sensors available would affect the quality of the diagnosis, i.e. more sensors (and hence, residuals) would lead to better fault isolation performance [6].

This paper proposes to use the concept of sequential residual generation reported in [7] to design and select additional residuals for a vehicular turbocharged spark ignited engine system with data obtained using the simulation testbed in [8]. The purpose is to improve fault isolation without adding physical sensors onto the engine system.

This paper is organized as follows: Section II introduces the engine system and the problem statement; Section III presents the design and generation of residuals using a conventional model-based method; Section IV provides some backgrounds on the design and generation of additional residuals; Section V shows the application of additional residuals on the engine system, which includes simulation results and discussions; and Section VI provides some conclusions.

II. PROBLEM STATEMENT

This paper addresses the issue of fault isolation in a vehicular turbocharged spark ignited (TCSI) engine system. Fig. 1 shows the engine test bench used for data collection in the lab. The engine system has 13 states: $T_{af}, p_{af}, T_c, p_c, T_{ic}, p_{ic}, T_{im}, p_{im}, T_{em}, p_{em}, T_t, p_t$, and W_{af}, which represent the temperatures (K) and pressures (Pa) at the air filter, compressor, intercooler, intake manifold, exhaust manifold, turbine, as well as the mass flow (kg/s) at the air filter, respectively. The engine system has six actuators: $A_{th}, u_{wg}, \omega_{eREF}, \lambda, p_{amb}$, and T_{amb}, which represent the throttle position area (m^2), wastegate input ([0...1]), reference engine speed (rad/s), air-fuel ratio, ambient pressure, and ambient temperature, respectively. It is assumed that the engine has seven measured outputs: $T_c, p_c, T_{ic}, p_{ic}, T_{im}, p_{im}$, and W_{af}. The specifications and parameters of the reference engine system can be found in [8].

This research considers 11 faults of interest located in various subsystems of the engine system, where six are variable faults ($f_{p_{af}}, f_{C_{vol}}, f_{W_{af}}, f_{W_c}, f_{W_{ic}}, f_{W_{th}}$), one actuator fault ($f_{x_{th}}$), and four sensor measurement faults ($f_{y_{W_{af}}}, f_{y_{p_{im}}}, f_{y_{p_{ic}}}, f_{y_{T_{ic}}}$). Table I shows the faults of interest and their descriptions. Only single fault scenarios are considered in this paper.

Given that the engine system is highly nonlinear with many interconnected subsystems, diagnostic systems usually monitor multiple components simultaneously, although they are quite independent from each other. As a result, a fault that is present in the engine system can trigger several monitors or manifest into other types of faults, hence affecting fault
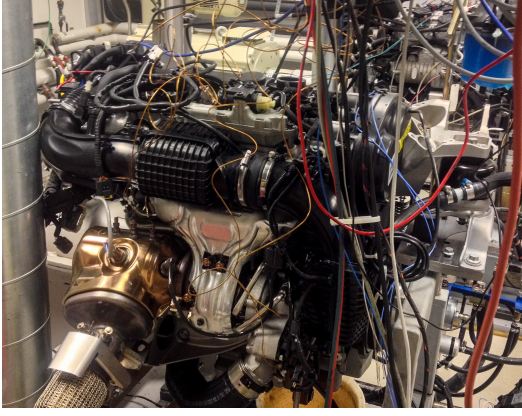
Fig. 1: The engine test bench used for data collection. Photo used with permission from [8].

TABLE I: Faults of interest and their descriptions.

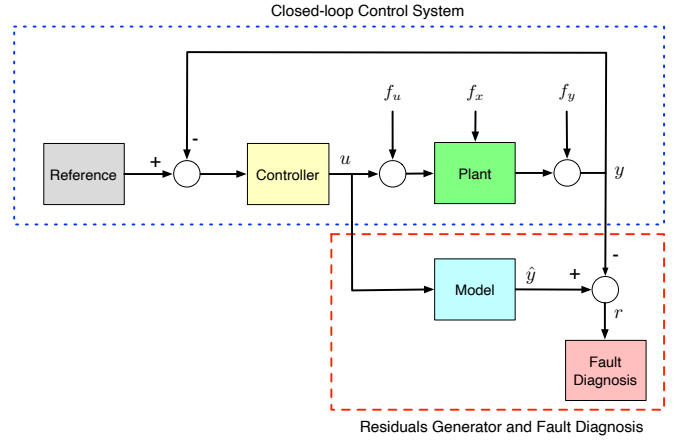| Fault	Description
$f_{p_{af}}$	Loss of pressure in the air filter
$f_{C_{vol}}$	Intake valve timing stuck at arbitrary position
$f_{W_{af}}$	Air leakage between the air filter and the compressor
f_{W_c}	Air leakage between the compressor and the intercooler
$f_{W_{ic}}$	Air leakage between the intercooler and the throttle
$f_{W_{th}}$	Air leakage after the throttle in the intake manifold
$f_{x_{th}}$	Throttle position actuator error
$f_{y_{W_{af}}}$	Air filter flow sensor fault
$f_{y_{p_{im}}}$	Intake manifold pressure sensor fault
$f_{y_{p_{ic}}}$	Intercooler pressure sensor fault
$f_{y_{T_{ic}}}$	Intercooler temperature sensor fault

Fig. 2: Block diagram representation of the closed-loop control system and the residuals generator.

TABLE II: Default "Original 7" residuals generated for fault detection based on the sensors setup of the engine system.

Residual	Description
r_{T_c}	Residual for Compressor Temperature Sensor
r_{p_c}	Residual for Compressor Pressure Sensor
$r_{T_{ic}}$	Residual for Intercooler Temperature Sensor
$r_{p_{ic}}$	Residual for Intercooler Pressure Sensor
$r_{T_{im}}$	Residual for Intake Manifold Temperature Sensor
$r_{p_{im}}$	Residual for Intake Manifold Pressure Sensor
$r_{W_{af}}$	Residual for Air Filter Mass Flow Sensor

isolation performance of the diagnostic systems [9]. This is critical as the ability to identify and isolate the root fault from the manifested faults enables the replacement of the correct faulty components to enhance the reliability of the overall system [10]. This also helps to ensure the safety of the occupants onboard the vehicle as well as other road users.

The goal is to design and generate additional useful residuals that would be useful to improve fault isolation without adding physical sensors to the systems.

III. DESIGN AND GENERATION OF RESIDUALS

Fig. 2 shows a typical block diagram of the closed-loop feedback system with the residuals generator. The blocks within the blue dotted box form the closed-loop control system while the blocks within the red dashed box form the residuals generator and the fault diagnosis scheme. Both plant and model are driven by the same control input, where the plant would then produce the output y while the model would produce an estimate of the output $\hat{y}$. The difference between the actual output and its estimate is then used to generate the residual for fault detection, i.e. $r = \hat{y} - y$. During nominal fault-free scenario, the residual r would have a mean of zero. In the presence of a fault, the residual r would have a nonzero mean and hence, a fault has been detected.

As stated in Section II, the reference engine system has seven measured outputs: $T_c, p_c, T_{ic}, p_{ic}, T_{im}, p_{im},$ and W_{af}. As such, there are seven residuals that can be generated. These residuals are called the "Original 7" (see Table II).

Fig. 3 shows the "Original 7" for a nominal fault-free scenario during the Worldwide Harmonized Light Vehicles Test Procedure (WLTP) run using the simulation testbed in [8] while Fig. 4 shows the "Original 7" for a fault leading to a pressure drop in the air filter $f_{p_{af}}$ during the same driving cycle. The horizontal red dashed lines are the thresholds set for fault detection. A fault is said to have been detected when one or more residuals exceed the thresholds, i.e. $|r| > J$ where the thresholds were initially established as $J = 5$. The gray regions in Fig. 4 show the the duration that the fault is induced during the simulation. Fig. 4 also shows that five of the "Original 7" residuals have triggered due to the presence of the fault: $r_{T_c}, r_{p_c}, r_{p_{ic}}, r_{p_{im}},$ and $r_{W_{af}}$. The simulation was repeated with the other faults of interest in Table I and the states of the residuals were recorded in a fault sensitivity matrix (FSM) as shown using the unshaded rows in Table III, i.e. residuals that are sensitive to a particular fault and have triggered are given a state of "1", and "0" otherwise.

Using data from the unshaded rows in Table III, a fault isolation matrix (FIM) could then be generated to analyse the fault isolation performance of using only the "Original 7" residuals (see Fig. 5). Fig. 5 shows that only $f_{x_{th}}$ could be isolated from the other faults. As such, additional residuals can be designed to improve fault isolation. See [11] for some background studies on structural model, FSM, and FIM.

TABLE III: The FSM for the "Original 7" residuals (unshaded rows) and the additional residuals (shaded rows).

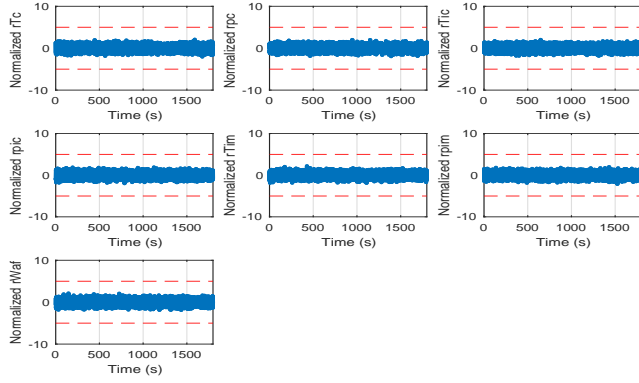| Residuals	$f_{p_{af}}$	$f_{C_{vol}}$	$f_{W_{af}}$	f_{W_c}	$f_{W_{ic}}$	$f_{W_{th}}$	$f_{x_{th}}$	$f_{y_{p_{ic}}}$	$f_{y_{p_{im}}}$	$f_{y_{T_{ic}}}$	$f_{y_{W_{af}}}$
r_{T_c}	1	1	0	1	0	0	0	0	0	0	0
r_{p_c}	1	1	0	1	1	1	0	0	0	0	0
$r_{T_{ic}}$	0	1	0	1	0	0	0	0	0	1	0
$r_{p_{ic}}$	1	1	0	1	1	1	0	1	0	0	0
$r_{T_{im}}$	0	1	0	1	0	0	0	0	0	0	0
$r_{p_{im}}$	1	1	0	1	1	1	0	0	1	0	0
$r_{W_{af}}$	1	1	1	1	1	1	0	0	0	0	1
p_{im}-W_{af}	0	1	0	1	0	1	0	0	1	0	1
T_{im}-W_{af}	0	1	0	1	0	1	0	0	0	0	0
p_{ic}-W_{af}	0	1	0	1	0	1	1	1	0	0	1
T_{ic}-W_{af}	0	1	0	1	0	1	0	0	0	1	0
W_{af}-p_{im}	1	1	1	1	1	1	1	0	1	0	1
T_{im}-p_{im}	1	1	0	1	1	0	0	0	1	0	0
p_{ic}-p_{im}	1	1	0	1	1	1	0	1	1	0	0
T_{ic}-p_{im}	0	1	0	0	0	0	0	0	0	1	0
W_{af}-T_{im}	1	1	1	1	1	1	0	0	0	0	1
W_{af}-$T_{ic}p_{ic}T_{im}p_{im}$	1	1	1	1	1	1	1	1	1	0	1
W_{af}-$p_cT_{im}p_{im}$	1	1	1	1	1	1	1	1	1	0	1
W_{af}-$p_cT_{ic}p_{ic}$	1	1	1	1	1	1	1	1	0	1	1
p_{ic}-$T_cp_cT_{im}p_{im}$	0	0	0	0	1	0	0	1	1	0	0
T_{ic}-$T_cp_cT_{im}p_{im}$	0	0	0	0	1	0	0	0	1	1	0

Fig. 3: Normalized plots of the "Original 7" for a fault-free scenario. The dashed lines are the fault detection thresholds.
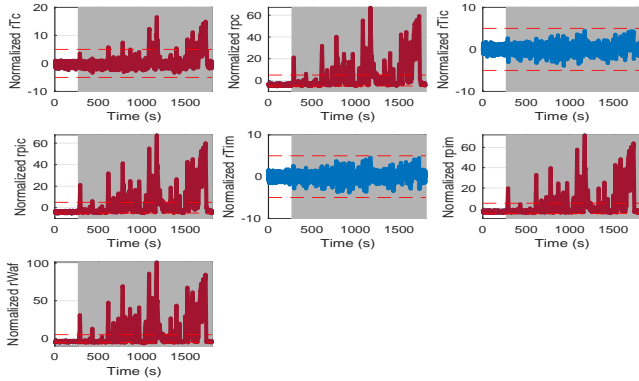

Fig. 4: Normalized plots of the "Original 7" for a $f_{p_{af}}$ fault. The dashed lines are the fault detection thresholds. The plots in red are the residuals that have triggered. The shaded regions show the duration when the fault is active.

IV. DESIGN AND GENERATION OF ADDITIONAL RESIDUALS

With reference to [7], this section presents some backgrounds on the design of the additional residuals.
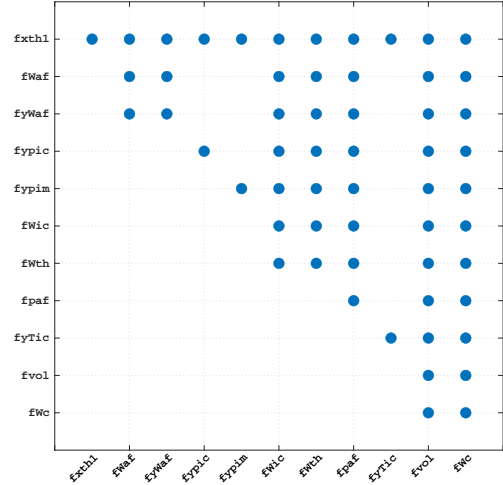

Fig. 5: The FIM for the FSM of the "Original 7" in Table III (unshaded rows only).

First, let's assume a system can be described using the following differential equations

$$\dot{x}_1 = -x_1 + u + f_u, \tag{1}$$

$$\dot{x}_2 = x_1 - x_2, \tag{2}$$

$$y_1 = x_1 + f_1, \tag{3}$$

$$y_2 = x_2 + f_2, \tag{4}$$

where x_i are the states, u the measurable input, y_i the measurable outputs, f_u the input fault, and $\{f_1, f_2\}$ are the output faults.

The relationship between the input u and the outputs $\{y_1, y_2\}$ for the system in (1)–(4) can be traced through the states $\{x_1, x_2\}$, as shown in Fig. 6a. Since there are 2 outputs, i.e. y_1 and y_2, a model can be formulated to produce 2 residual signals, r_1 and r_2, respectively. Fig. 6a also shows that each residual is generated by taking different routes from
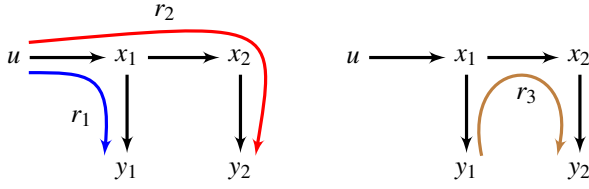
(a) Tracing from u to $\{y_1, y_2\}$ through $\{x_1, x_2\}$ to generate $\{r_1, r_2\}$.

(b) A third and new residual, r_3, which is used to estimate y_2, is generated by tracing y_1 to y_2.

Fig. 6: Design of residuals by tracing paths from other measurable variables to the variables to be estimated.

TABLE IV: The FSM for $\{r_1, r_2\}$ (unshaded rows), and the additional residual r_3 (shaded row).

Residual	f_1	f_2	f_u
r_1	1	0	1
r_2	0	1	1
r_3	1	1	0

u to y_1 and y_2, respectively. As such, residual r_1 can be designed to estimate y_1 using

$$\dot{\hat{x}}_1 = -\hat{x}_1 + u, \tag{5}$$
$$\hat{y}_1 = \hat{x}_1, \tag{6}$$
$$r_1 = y_1 - \hat{y}_1, \tag{7}$$

and residual r_2 can be designed to estimate y_2 using

$$\dot{\hat{x}}_2 = \hat{x}_1 - \hat{x}_2, \tag{8}$$
$$\hat{y}_2 = \hat{x}_2, \tag{9}$$
$$r_2 = y_2 - \hat{y}_2, \tag{10}$$

where $\hat{x}_i$ and $\hat{y}_i$ are the estimates of the states x_i and the outputs y_i, respectively.

Simulating this system with the residual generators in Matlab/Simulink where the faults are injected as sinusoidal signals; $f_u = 2sin(t)$, $f_1 = 2sin(t + \frac{\pi}{4})$, and $f_2 = 2sin(t + \frac{\pi}{2})$, Fig. 7a shows that residual r_1 is sensitive to f_1 and f_u while r_2 is sensitive to f_2 and f_u. As a result, the FSM for $\{r_1, r_2\}$ can be represented by the unshaded rows in Table IV. The unshaded rows in Table IV show that the sensor faults can be successfully isolated from each other. However, the fault sensitivity does not allow for the elimination of f_u from the sets of diagnosis, i.e. the detection of either f_1 or f_2 also includes a detection of f_u. Therefore, a new residual has to be designed to achieve better fault isolation.
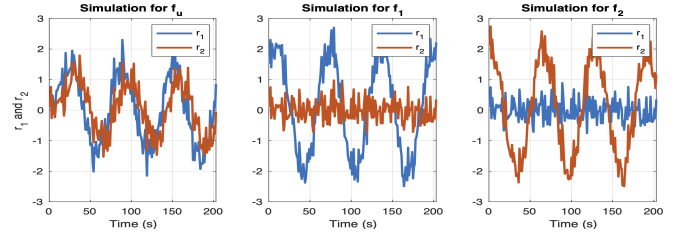
Let's reconsider the system (1)–(4). Assume now that there exists another route for the estimation of y_2. Removing the equation for y_1 in (3) and using $\hat{x}_1$ to estimate y_1 as shown in (11), a new path can be traced to y_2 (see Fig. 6b), which results in the generation of a third residual, r_3 in (14).
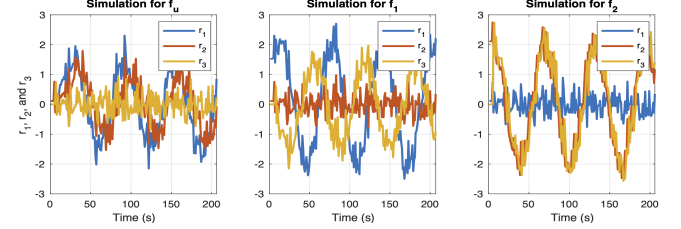
$$\hat{x}_1 = y_1, \tag{11}$$
$$\dot{\hat{x}}_2 = \hat{x}_1 - \hat{x}_2, \tag{12}$$
$$\hat{y}_2 = \hat{x}_2, \tag{13}$$
$$r_3 = y_2 - \hat{y}_2. \tag{14}$$

(a) Sensitivity of only the original residuals $\{r_1, r_2\}$.

(b) Sensitivity of the original and additional residuals $\{r_1, r_2, r_3\}$.

Fig. 7: Simulations showing the performance of the original residuals $\{r_1, r_2\}$ and then with the additional residual r_3 towards the faults in system (1)–(4).
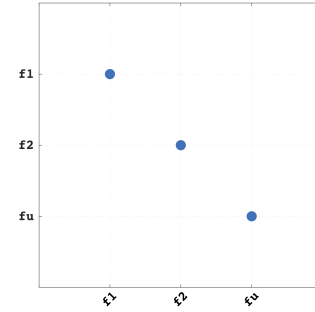

Fig. 8: The FIM with the inclusion of r_3.

TABLE V: The triggered residuals and the diagnosis decisions for the isolated faults.

Triggered Residuals	Diagnosis Decision
r_1 and r_2	f_u
r_1 and r_3	f_1
r_2 and r_3	f_2

The omission of u from this path removes the sensitivity of r_3 towards f_u. As a result, the FSM for the residuals can be updated to include the shaded row in Table IV. The updated FSM shows that f_u can be isolated from the sensor faults using the new residual and this is reflected in the FIM in Fig. 8. Table V shows the relationships between the triggered residuals and the diagnosis decisions.

Another possible additional residual, r_4, can be generated by removing the equation for y_2 in (4) and using $\hat{x}_2$ to estimate y_2. Hence, a path can be traced to y_1 (the opposite direction of r_3) to provide an estimate of y_1. However, it is not utilized in this example as all faults can already be isolated from each other with the the original residuals and only one additional residual, r_3.

V. APPLICATION TO A TCSI ENGINE SYSTEM

For the TCSI engine, it is desired to have more residuals than the "Original 7" to make the fault isolation investigation relevant. As such, the procedure discussed in Section IV is applied to design additional residuals for the engine model. However, due to the higher number of original residuals used by the engine (7 residuals) compared to the ones used in the example (2 residuals) in Section IV, there are more routes/paths that can be chosen to generate the additional residuals. For example, a system with n original residuals, there are $(n \times 2^{n-1}) - n$ possible candidates as additional residuals. Therefore, the TCSI engine system with the "Original 7" residuals has 441 potential candidates as additional residuals. Some of these additional residuals would enhance fault isolation performance while the remaining ones, although being able to provide estimation of the measurable variables, do not necessarily improve fault isolation. To reiterate, for the example presented in Section IV, r_3 is sufficient to enhance fault isolation performance while r_4 is not necessary although it would still provide an estimate of y_1. Therefore, the challenge is to determine the minimum number of additional residuals that would improve the fault isolation performance most. This is critical as the requirements for computing power increase with the number of additional residuals generated.

With reference to the sequential residuals generation method in [7], a Matlab-based algorithm is used to generate all potential candidates for additional residuals, as well as to search for the best candidates and to determine the least number of additional residuals that can guarantee the best fault isolation performance.

Algorithm 1 first explains the procedure to generate all possible additional residuals from the current residuals setup. Given that the TCSI engine system has 7 original residuals ($n = 7$), there are $(2^n - 1)$ combinations in residuals setup using the variable `cycle`, where 'ON' and 'OFF' states of these residuals are represented in binary. The main loop is then used to generate all $(n \times 2^{n-1}) - n$ possible candidates by sequentially removing the corresponding equations and rows from the relations matrix of the structural model. Algorithm 2 would then utilize the results from Algorithm 1 to select the minimum best candidates for additional residuals to enhance fault isolation performance. This is done by running all original residuals and potential candidates through an optimization function `autoSelect()`, where each iteration would choose a potential candidate as one of the best candidates and update the overall FIM of all residuals. The algorithm would terminate when the optimization function returns no best candidate.

The resultant FIM for the original residuals together with this new set of best additional candidates would provide the updated analysis on fault isolation performance.

A. RESULTS AND DISCUSSIONS

For the TCSI engine system, Algorithm 1 would generate the additional 441 potential candidates as well as their ideal FSM. Using these results, Algorithm 2 returned 14 best

---

Algorithm 1 Generation of all possible additional residuals from the "Original 7".

Input: $n, StructuralModel, FSM$
1: $cycle \leftarrow binary([1 : (2^n - 1)])$ ▷ All residuals
2: combinations
3: **for** $i \leftarrow 1$ **to** $(2^n - 1)$ **do** ▷ Main loop
4:     $sumSens \leftarrow getSumSensON(cycle(i))$
5:     $posSens \leftarrow getPosSensON(cycle(i))$
6:     $nameSens \leftarrow getNameSensON(posSens)$
7:     $tempRelMat \leftarrow getRelMat(StructModel)$
8:     **for** $j \leftarrow 1$ **to** $sumSens$ **do** ▷ Remove corresponding
9:          eqs. from relations matrix
10:         $eqNo \leftarrow getEqNo(StructModel, nameSens)$
11:         **remove** $tempRelMat(eqNo)$
12:     **end for**
13:     $genNewRes(tempRelMat)$
14: **end for**
Output: $allAddRes, allAddResFSM$

Algorithm 2 Selection of minimum additional residuals from all possible additional residuals to enhance fault isolation.

Input: $n, nFaults, allAddResFSM$
1: $bestFIM \leftarrow ones(nFaults)$
2: **repeat**
3:     $selRes \leftarrow autoSelect(bestFIM, allAddResFSM)$
4:     $bestAddRes \leftarrow [bestAddRes; selRes]$
5:     $bestFIM \leftarrow FSMtoFIM(allAddResFSM(bestRes))$
6: **until** $isEmpty(selRes)$
Output: $bestAddRes$
7:
8: **function** AUTOSELECT(FIM, FSM)
9:     **for** $i \leftarrow 1$ **to** $(n \times 2^{n-1})$ **do**
10:         $zerosIdx \leftarrow getPosZeros(FSM(i))$
11:         $onesIdx \leftarrow getPosOnes(FSM(i))$
12:         **if** $sum(FIM(onesIdx, zerosIdx)) > 0$ **then**
13:             $selRes \leftarrow i$
14:         **else**
15:             $selRes \leftarrow [\ ]$
16:         **end if**
17:     **end for**
18:     **return** $selRes$
19: **end function**

---

candidates as additional residuals that could be used to enhance fault isolability. The simulations were run again with the additional residuals and the shaded rows in Table III show their sensitivity towards the faults, i.e. if they have triggered or otherwise (Figs. 9 and 10 show some simulation results).

Computing the FIM for the FSM of the original residuals together with this new set of best additional candidates produced a much desirable result as shown in Fig. 11, where better fault isolation could be achieved compared to Fig. 5. With the additional residuals, five faults: $f_{W_{af}}, f_{x_{th}}, f_{y_{T_{ic}}}, f_{y_{p_{ic}}}$, and $f_{y_{p_{im}}}$ can now be isolated from the other faults.

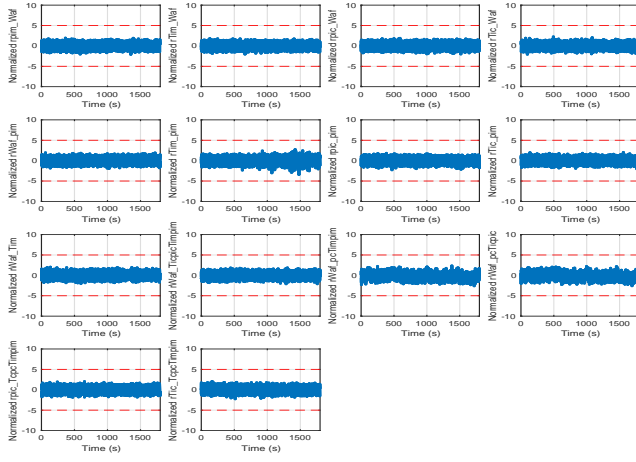Furthermore, the FIM also shows that the detection of any

Fig. 9: Normalized plots of the 14 additional residuals for a fault-free scenario. The dashed lines are the fault detection thresholds.
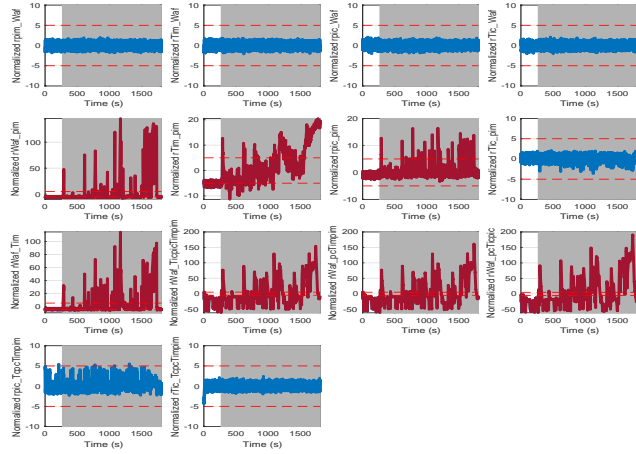

Fig. 10: Normalized plots of the 14 additional residuals for a $f_{p_{af}}$ fault. The dashed lines are the fault detection thresholds. The plots in red are the residuals that have triggered. The shaded region shows the duration when the fault is active.

Fig. 11: The FIM for the "Original 7" and the 14 additional residuals in Table III (all rows including the shaded rows).

the engine system, hence reducing the size and manufacturing cost of the system while ensuring safety and reliability. Future developments of this research include but not limited to further improving fault isolability using fault ranking, a combination of model-based and data-driven methods, or hybrid/hierarchical machine learning algorithms.

actual fault involves less possible detection of other faults as potential candidates compared to Fig. 5, hence improving the overall fault isolability.

VI. CONCLUSION

This paper has presented the application of using additional residuals to enhance fault isolation on a vehicular engine system. The algorithm used to select a minimum number of best candidates for additional residuals helps to reduce the computation requirements to process the residuals, which allows for diagnosis to be performed onboard the vehicle in addition to cloud-based diagnosis architectures that many automakers are adopting nowadays. Also, by using structural analysis method, it has been shown that the presented method is able to improve fault isolability without adding physical sensors to the system. This would enable automakers to reduce hardware-redundancy in the design of
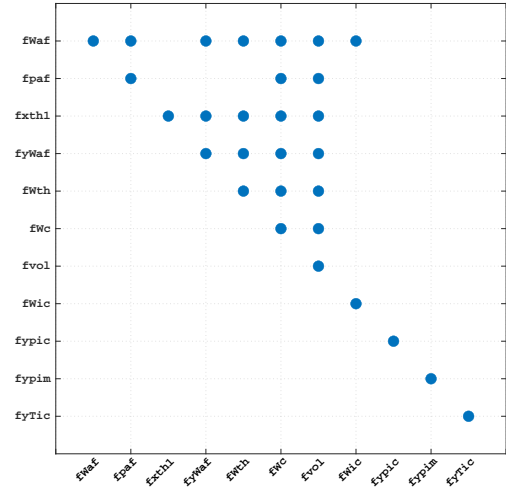
REFERENCES

[1] M. Ge, Y. Xu, and R. Du, "An intelligent online monitoring and diagnostic system for manufacturing automation," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 1, pp. 127–139, 2008.

[2] A. S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatica*, vol. 12, no. 6, pp. 601–611, 1976.

[3] M. Radu, D. Pitica, and C. Posteuca, "Reliability and failure analysis of voting circuits in hardware redundant design," in *International Symposium on Electronic Materials and Packaging (EMAP2000)*, Nov 2000, pp. 421–423.

[4] Z. Gao, C. Cecati, and S. X. Ding, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, June 2015.

[5] S. X. Ding, *Model-based Fault Diagnosis Techniques: Design Schemes, Algorithms, and Tools*. Springer, 2008.

[6] E. Frisk, M. Krysander, and J. Åslund, "Sensor placement for fault isolation in linear differential-algebraic systems," *Automatica*, vol. 45, no. 2, pp. 364–371, 2009.

[7] C. Svärd, M. Nyberg, and E. Frisk, "Realizability constrained selection of residual generators for fault diagnosis with an automotive engine application," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, pp. 1354–1369, Nov 2013.

[8] K. Y. Ng, E. Frisk, M. Krysander, and L. Eriksson, "A realistic simulation testbed of a turbocharged spark ignited engine system: A platform for the evaluation of fault diagnosis algorithms and strategies," *IEEE Control Systems Magazine (In press)*, 2020.

[9] A. E. Goodloe and L. Pike, "Monitoring distributed real-time systems: A survey and future directions," *NASA Technical Reports Server*, 2010.

[10] A. Scacchioli, G. Rizzoni, and P. Pisu, "Model-based fault detection and isolation in automotive electrical systems," in *Proceedings of the ASME 2006 International Mechanical Engineering Congress and Exposition. Dynamic Systems and Control, Parts A and B.* Chicago, Illinois, USA: ASME, Nov. 2006, pp. 315–324.

[11] D. Düştegör, E. Frisk, V. Cocquempot, M. Krysander, and M. Staroswiecki, "Structural analysis of fault isolability in the DAMADICS benchmark," *Control Engineering Practice*, vol. 14, no. 6, pp. 597–608, 2006.