

Qualitative Event-based Diagnosis with Possible Conflicts Applied to Spacecraft Power Distribution Systems[★]

Matthew Daigle^{*} Anibal Bregon^{**} Indranil Roychoudhury^{***}

^{*} NASA Ames Research Center, Moffett Field, CA, 94035, USA
(e-mail: matthew.j.daigle@nasa.gov).

^{**} Department of Computer Science, University of Valladolid,
Valladolid, 47011, Spain (e-mail: anibal@infor.uva.es)

^{***} SGT Inc., NASA Ames Research Center, Moffett Field, CA,
94035, USA (e-mail: indranil.roychoudhury@nasa.gov)

Abstract: Model-based diagnosis enables efficient and safe operation of engineered systems. In this paper, we describe two algorithms based on a qualitative event-based fault isolation framework augmented with model-based fault identification that are applied to spacecraft power distribution systems. Although based on a common framework, the fundamental difference between the two algorithms is that one uses a global model for residual generation, fault isolation, and fault identification; whereas the other uses a set of minimal submodels computed using Possible Conflicts. We describe the implementation of the two algorithms and compare their diagnosis results on a representative spacecraft power distribution system.

Keywords: model-based diagnosis, spacecraft power systems, possible conflicts.

1. INTRODUCTION

Fault diagnosis methodologies are motivated by the need for increased performance, safety, and reliability of complex engineering systems. This paper presents a model-based, qualitative, event-based fault diagnosis scheme that performs the functions of fault detection, isolation, and identification. The diagnosis scheme has its foundations in the TRANSCEND diagnosis approach (Mosterman and Biswas, 1999), but can diagnose abrupt, incipient, and intermittent single faults. Our diagnosis scheme has two instantiations, QED (Qualitative Event-based Diagnosis), and QED-PC (QED with Possible Conflicts).

The general approach extends the TRANSCEND diagnosis scheme. In this scheme, fault isolation is achieved through analysis of the transients produced by faults, manifesting as deviations in observed behavior from predicted nominal behavior. QED extends TRANSCEND by including relative measurement orderings, which provide a partial ordering of measurement deviations for different faults, leading to an enhanced event-based fault isolation scheme (Daigle et al., 2009). Further, TRANSCEND deals only with abrupt faults, so we incorporate methods for incipient faults (Roychoudhury, 2009) and intermittent faults (Daigle and Roychoudhury, 2010).

QED-PC uses the Possible Conflicts (PCs) diagnosis approach (Pulido and Alonso-González, 2004) within the general QED framework. The PCs approach is a dependency-compilation technique from the DX community similar to the derivation of Analytical Redundancy

Relations from the FDI community. The approach decomposes the global system model into minimal submodels containing sufficient analytical redundancy to generate fault hypotheses from observed measurement deviations. With QED-PC, residuals are computed using the PCs (instead of the global system model) and measurement deviations are analyzed following the TRANSCEND ideas as in QED. For fault identification, the algorithm uses minimal parameter estimators computed from PCs for each faulty parameter, as described in (Bregon et al., 2011b).

As a case study, we adopt a subset of the Advanced Diagnostic and Prognostic Testbed (ADAPT) (Poll et al., 2007), which is functionally representative of a spacecraft power distribution system. We apply our diagnosis algorithms to this system, evaluate the results, and compare and contrast algorithm performance.

The paper is organized as follows. Section 2 overviews the diagnosis approaches. Section 3 provides the system model. Sections 4, 5, and 6 describe fault detection, isolation, and identification, respectively. Section 7 presents the diagnosis results, and Section 8 concludes the paper.

2. DIAGNOSIS APPROACH

Our diagnosis approach performs the tasks of (i) fault detection, i.e., determining if a fault is present in the system, (ii) fault isolation, i.e., determining which fault has occurred, and (iii) fault identification, i.e., estimating the parameters that define the fault behavior.

The diagnosis architecture is shown in Fig. 1, and reflects the implementation of both algorithms. The system receives inputs $\mathbf{u}(t)$ and produces outputs $\mathbf{y}(t)$. The system

[★] Anibal Bregon's work has been funded by the Spanish MCI DPI2008-01996 and TIN2009-11326 grants.

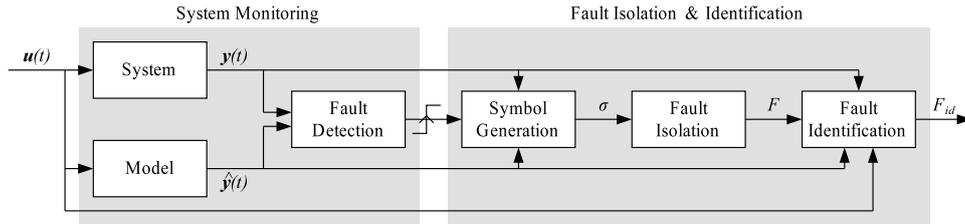


Fig. 1. Diagnosis architecture.

model, given inputs $\mathbf{u}(t)$, computes predicted values $\hat{\mathbf{y}}(t)$. The fault detection module decides whether a measurement has deviated from its nominal value in a statistically significant manner, triggering the fault isolation and identification modules. Measurement deviations, viewed as events, are abstracted into a symbolic representation using the symbol generator. The sequence of these symbols, where a symbol is denoted by σ , is used to isolate faults F . Fault isolation consists of candidate generation at the point of fault detection and hypothesis refinement as new symbols are provided. Each fault $f \in F$ is associated with a component, a fault mode, and a set of fault parameters. Fault identification computes, for each fault $f \in F$, the values of the fault parameters.

For both QED and QED-PC, we compute residuals as the difference between an observation, $\mathbf{y}(t)$, and the predicted nominal behavior of the output, $\hat{\mathbf{y}}(t)$, with the only difference coming from the way the predicted behavior is computed by each algorithm. For QED, the predicted values $\hat{\mathbf{y}}(t)$ are computed based on a global model of the system. For QED-PC, the system model is decomposed, using the Possible Conflicts approach (Pulido and Alonso-González, 2004), into minimal over-determined subsystems, each with a single output, that suffice for fault diagnosis. In this approach, the predicted values $\hat{\mathbf{y}}(t)$ are computed based on these subsystem models. The submodels are decoupled from each other by using measured values as inputs to the submodels.

3. SYSTEM MODELING

Our diagnosis approach is model-based, requiring a model of both nominal and faulty behavior for use throughout the diagnosis process. As described in the previous section, the two algorithms implement the nominal model in a different way. For QED, the nominal model is a global model of the system, \mathcal{M} , and its inputs are those of the global system. For QED-PC, the nominal model is composed of a set of 11 minimal submodels, with each submodel \mathcal{M}_i estimating the value of sensor i using a subset of the system measurements as input variables. In the following, we describe the models of nominal and faulty behavior of the ADAPT system for QED and QED-PC, indicating their similarities and differences.

3.1 Nominal Model

A schematic of the selected subset of ADAPT is given in Fig. 2. Sensors prefixed with an “E” are voltage sensors, those with an “IT” are current sensors, and those with “ISH” or “ESH” are for sensing the states of circuit breakers and relays. TE228 is the battery temperature

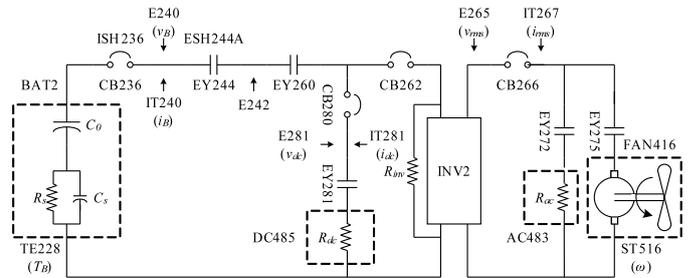


Fig. 2. ADAPT subset schematic.

sensor, and ST516 is the fan speed sensor. Note that the inverter (INV2) converts DC power to AC, and E265 and IT267 provide rms values of the AC waveforms.

Most of the equations of the global model and the corresponding PCs are summarized in Table 1. Details may be found in (Daigle and Roychoudhury, 2010). Here, v_B and i_B are the battery voltage and current, v_0 is the voltage across C_0 , v_s is the voltage across C_s , e is the inverter efficiency, v_{inv} is the inverter voltage on the DC side, R_{inv} is the DC resistance of the inverter, R_{dc} is the DC load resistance, J_{fan} is the fan inertia, and B_{fan} is a damping parameter. Both QED and QED-PC assume TE228, ISH236, and ESH244A are constant. The PCs for E242 and E281 are simply other measured voltages with a bias term added.

Most of the PCs are derived directly from the global model, but in some cases, the PCs have to account for additional dynamics. For example, the fan speed (ω) has no dynamics during nominal operation because it is always operated at the same speed. So, QED models the fan speed as a constant. QED-PC, on the other hand, must model the dynamics, because some faults independent of the fan submodel will cause the fan speed to decrease through a decrease in E265, which is an input to the PC.

A key difference, then, compared to the global model, is that the behavior of each PC has to be nominal not only for the nominal situation, but also for those faulty situations where the fault parameters are independent of a PC. This decoupling requires a more detailed modeling of the system for the QED-PC algorithm. This introduced some modeling difficulties, especially concerning IT240. In nominal operation, the measured value averages around 16 ± 2 A. When faults occur, however, the value takes on a much wider range, and the IT240 PC must accurately predict values in the entire range due to faults that are decoupled from the PC. This made the system identification task more complex. System identification was also more complex for QED-PC because sensor biases had to be considered for the inputs to the PCs.

Table 1. Models for the Case Study.

Global Model (QED)		PCs (QED-PC)	
$\dot{v}_0 = \frac{1}{C_0} (-i_B)$	$\dot{v}_s = \frac{1}{C_s} (i_B R_s - v_s)$	$\dot{v}_0 = \frac{1}{C_0} (-IT240)$	$\dot{v}_s = \frac{1}{C_s} (IT240R_s - v_s)$
$v_B = v_0 - v_s$	$i_{dc} = \frac{v_B}{R_{dc}}$	$\widehat{E240} = v_0 - v_s$	$\widehat{IT281} = \frac{E281}{R_{dc}}$
$i_{inv} = \frac{v_{rms} i_{rms}}{e \cdot v_{inv}} + \frac{v_{inv}}{R_{inv}}$	$i_B = i_{inv} + i_{dc}$	$i_{inv} = \frac{E265 \cdot IT267}{e \cdot E242} + \frac{E242}{R_{inv}}$	$\widehat{IT240} = i_{inv} + IT281$
$i_{rms} = \frac{v_{rms}}{R_{fan}} + \frac{v_{rms}}{R_{ac}}$	$v_{rms} = v_{inv,0}$	$\widehat{IT267} = \frac{E265}{R_{fan}} + \frac{E265}{R_{ac}}$	$\widehat{E265} = v_{inv,0} - \frac{IT267}{R_{rms}}$
$\dot{\omega} = 0$		$\dot{\omega} = \frac{1}{J_{fan}} \left(\frac{E265}{B_{fan}} - \omega \right)$	$\widehat{ST516} = \omega$

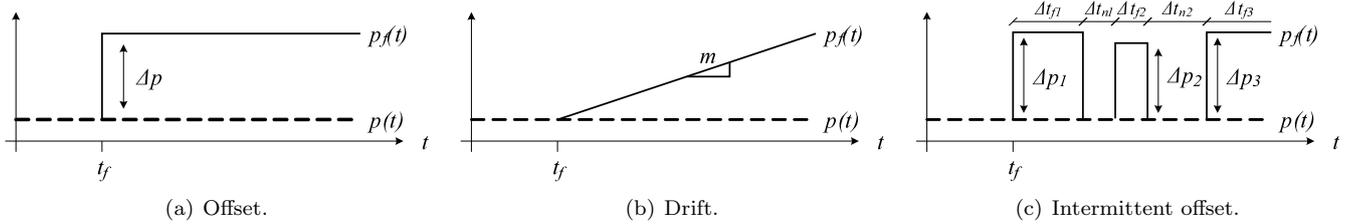


Fig. 3. Fault profiles.

3.2 Fault Modeling

We consider both *parametric* faults, defined as unexpected changes in system parameter values, and *discrete* faults, defined as unexpected changes in the operating mode of a component. Parametric faults include changes in the AC and DC resistances, R_{ac} and R_{dc} , and additive terms to sensor equations. Discrete faults include stuck faults of the relays and circuit breakers, inverter failure, load failure, fan overspeed and underspeed faults, and sensor stuck faults. Note that sensor stuck faults are defined as $y(t) = c$, where c is a constant, and sensor noise is absent.

Parametric faults may assume offset, drift, and intermittent offset profiles, as defined in Fig. 3 (t_f denotes the time of fault occurrence). For an offset fault, we identify the offset Δp ; for a drift fault, we identify the slope m ; and for an intermittent fault, we identify the mean offset $\mu_{\Delta p}$, i.e. $\text{mean}(\Delta p_1, \Delta p_2, \dots)$, the mean faulty time μ_f , i.e. $\text{mean}(\Delta t_{f1}, \Delta t_{f2}, \dots)$, and the mean time it is nominal μ_n , i.e. $\text{mean}(\Delta t_{n1}, \Delta t_{n2}, \dots)$.

4. FAULT DETECTION

QED and QED-PC use the same approach for fault detection and symbol generation. Each sensor is assigned a fault detector. For each sensor output $y(t)$, we define the residual as $r(t) = y(t) - \hat{y}(t)$, where $\hat{y}(t)$ is the model-predicted output signal. As described in the previous section, for QED, $\hat{y}(t)$ is computed using the global model, whereas for QED-PC, it is computed using the corresponding PC. Statistically significant nonzero residual signals indicate faults. Details may be found in (Daigle and Roychoudhury, 2010).

Fault detection works by applying a Z-test to the residual values. A threshold based on the Z-test is computed, and to account for modeling error, an additional error term E is added to the threshold. When the absolute value of the mean residual value over a small window (e.g., 5 samples) is over this combined threshold, a fault is detected.

Generally, fault detectors for the global model can be more sensitive than those for the PCs. This is because with PCs, noisy sensor values are used as inputs to the submodels, and this adds additional noise to the predicted values. In our implementation, this was captured by larger E values for the fault detectors for QED-PC than the values for QED.

5. FAULT ISOLATION

We utilize a qualitative diagnosis methodology that isolates faults based on the transients they cause in system behavior, manifesting as deviations in residual values (Mosterman and Biswas, 1999). In both QED and QED-PC, we define residuals, as described in the previous section, with respect to a particular sensor.

The transients produced by faults are abstracted using qualitative + (increase), - (decrease), and 0 (no change) values and N (zero to nonzero), Z (nonzero to zero), and X (no discrete change) values to form *fault signatures*. Fault signatures represent these measurement deviations from nominal behavior as the immediate (discontinuous) change in magnitude, the first nonzero derivative change, and discrete zero/nonzero value changes in the measurement from the estimate caused by discrete faults. These symbols are computed from the residuals using symbol generation, as described in (Daigle and Roychoudhury, 2010).

In addition to signatures, for QED we also capture the temporal order of measurement deviations, termed *relative measurement orderings* (Daigle et al., 2009). Within the QED-PC algorithm, since measurement orderings may be defined only within a given submodel, we cannot define any orderings between residuals of two different PCs, because they are decoupled. The predicted fault signatures and measurement orderings can be computed manually or automatically from a system model (Mosterman and Biswas, 1999; Daigle, 2008). The predicted signatures and orderings are compared with observed signatures and orderings in order to isolate faults.

Table 2. Selected Fault Signatures for the QED algorithm for ADAPT

Fault	E240	E242	IT281	IT267	ST516
AC483 $\Delta p > 0$	+0X	+0X	+0X	-0X	00X
DC485 $\Delta p > 0$	+0X	+0X	-0X	00X	00X
E240 $\Delta p > 0$	+0X	00X	00X	00X	00X
E240 $m > 0$	0+X	00X	00X	00X	00X
E240 $\mu_{\Delta p} > 0$	+0X	00X	00X	00X	00X
EY244 stuck open	+0X	-0Z	-0Z	-0Z	0-X
FAN416 underspeed	+0X	+0X	00X	-0X	-0X

Table 3. Selected Fault Signatures for the QED-PC algorithm for ADAPT

Fault	E240	E242	IT281	IT267	ST516
AC483 $\Delta p > 0$	00X	00X	00X	-0X	00X
DC485 $\Delta p > 0$	00X	00X	-0X	00X	00X
E240 $\Delta p > 0$	+0X	-0X	00X	00X	00X
E240 $m > 0$	0+X	0-X	00X	00X	00X
E240 $\mu_{\Delta p} > 0$	+0X	-*X	00X	00X	00X
EY244 stuck open	00X	-0Z	00X	00X	00X
FAN416 underspeed	00X	00X	00X	-0X	-0X

Selected fault signatures for ADAPT are shown in Table 2 for QED and Table 3 for QED-PC, where the first symbol is the immediate change in magnitude, the second is the slope, and the third is the discrete change. For example, a positive offset in E240 will cause an abrupt increase in the E240 residual with no change in slope, and no discrete change behavior (+0X). No other sensors are affected (00X) by this fault in the QED approach, but for the QED-PC algorithm, an abrupt decrease in the E242 residual with no change in slope, and no discrete change behavior (-0X) is also caused, because sensor E240 is used as input for the PC that estimates E242. A resistance offset in AC483 causes multiple deviations for QED but only one in IT267 for QED-PC, because the corresponding parameter, R_{ac} , is present only in the PC for IT267.

The system is not diagnosable based only on signatures and orderings. First, there are four pairs of faults that produce exactly the same quantitative behavior on the given measurements: failures in CB262 and INV2, failures in EY281 and DC485, failures in EY272 and AC483, and failures in EY275 and FAN416. Second, offset and intermittent faults produce the same initial transients, therefore they can only be distinguished by their quantitative effects. The fault identification module can handle that issue. Third, for QED, a sensor fault affects only a single residual, so when a sensor fault occurs, we, in theory, have to wait infinitely long to confirm that no other residuals deviate. For QED-PC, on the other hand, sensor faults affect many residuals, but due to the decoupling introduced by PCs, some nonsensor faults, e.g., a fault in AC483 (see Table 3), affect only a single measurement, and we run into the same issue.

To resolve this third issue, we introduce heuristic isolation rules based on timing information. We expect that residuals, if they will deviate, will do so within a certain time since fault detection. If not, we assume a 00X signature for that residual, and this allows us to isolate sensor faults

for QED and nonsensor faults for QED-PC in finite time. For example, for QED we expect that nonsensor faults will affect multiple residuals within 60 s of fault detection. Note that due to these diagnosability properties, in general, QED will be faster at isolating nonsensor faults and QED-PC will be faster at isolating sensor faults, because these classes of faults produce many deviations in residuals, allowing the faults to be quickly isolated.

We also introduce several other heuristic isolation rules, mostly based on timing information, in order to improve fault isolation times and generally enhance diagnosability. For example, for both QED and QED-PC, fan faults should affect ST516 within 30 s, and relay/circuit breaker faults should affect their position sensors within 2 s.

6. FAULT IDENTIFICATION

Fault identification takes the set of faults F produced by the fault isolation module, and computes the fault parameters for each fault, producing a new fault set F_{id} that includes this information. In some cases, fault identification may change the fault mode for a fault based on identification results, so F and F_{id} do not always have an exact correspondence. Identification is initiated immediately after the initial set of fault candidates is produced after fault detection. An identification routine is run for each fault candidate, which updates at each time step. Identification terminates for a given fault candidate when the fault isolation module determines the candidate is no longer consistent.

The faults that require identification are faults in the resistances R_{ac} and R_{dc} , and sensor faults. In addition, equivalent resistance values are computed for relays EY272 and EY281. For sensor faults of the stuck profile, the stuck value is simply computed as the most recent sensor measurement, and the candidate is eliminated as soon as two consecutive measurement values are different. Each of the remaining faults is directly associated in the model with a parameter change Δp .

In each case, we have a submodel that computes the value of $\Delta p(t)$ at a given time t based on the sensor measurements $\mathbf{y}(t)$. The resistance value R_{dc} is computed using the PC for IT281, solved for R_{dc} . The resistance value R_{ac} is given by

$$R_{ac}(t) = \frac{v_{ac}(t)}{\sqrt{i_{ac}(t)^2 - \left(\frac{v_{ac}(t)}{Z_{fan}} \sin \phi\right)^2 - \frac{v_{ac}(t)}{Z_{fan}} \cos \phi}},$$

where for $v_{ac}(t)$ we use $\sqrt{2}$ E265, and for $i_{ac}(t)$ we use $\sqrt{2}$ IT267. Here, ϕ is the phase offset introduced by the fan load, and Z_{fan} is its equivalent impedance. The nominal values of Z_{fan} and ϕ were calculated by solving the following expression at steady state using two different values of R_{ac} and measured values of i_{ac} and v_{ac} :

$$|i_{ac}| = \left| \frac{v_{ac}}{Z_{fan}} (\cos \phi + j \sin \phi) + \frac{v_{ac}}{R_{ac}} \right|.$$

This equation is derived from the complex impedance expressions for the fan and R_{ac} .

For sensor faults, $\Delta p(t)$ is computed as the residual $y(t) - \hat{y}(t)$ for output y using the predictive model. In the case

of QED, $\hat{y}(t)$ is computed using the global model, but for QED-PC, it is computed using only the PC that is associated with y .

The time profile of the parameter change Δp over $[t_d, t]$, where t_d is the detection time and t is the current time, may be an offset, a drift, or an intermittent offset. For the offset profile, we must identify the magnitude of the offset; for the drift profile, we must identify the slope; and for the intermittent profile, we must identify the average time the parameter value is faulty, the average offset value during this time period, and the average time the parameter value is nominal.

The identification procedure keeps the history of both the predicted nominal values p and the computed faulty values p_f . We compute the fault parameters as follows:

- For offset faults, we compute the offset simply as the mean of $\Delta p = p_f - p$.
- For drift faults, we compute the slope of Δp as $(\Delta p(t_2) - \Delta p(t_1))/(t_2 - t_1)$ over three different time intervals with $[t_1, t_2]$ as $[t_d, t]$, $[t_d, (t + t_d)/2]$, and $[(t + t_d)/2, t]$ where t is the current time, and take the median of these values as the slope, as described in (Daigle and Roychoudhury, 2010). We improve on the robustness of this computation by computing the average value over a small window of $\Delta p(t_1)$ and $\Delta p(t_2)$.
- For intermittent offset faults, we define a limit l above which $\Delta p(t)$ is considered faulty, and below which is considered nominal. The limit l is typically chosen within 1-2% of the nominal value of $y(t)$ or $p(t)$. We step through the signal $\Delta p(t)$ to determine the average nominal time μ_n , the average faulty time μ_f , and the average faulty value $\mu_{\Delta p}$. The full procedure is described in (Daigle and Roychoudhury, 2010).

Fault identification is also used to improve fault isolation, either by eliminating candidates whose identification results are inconsistent with the supposed candidate, e.g., a stuck fault candidate is not actually stuck, or by changing the fault mode of the candidate to be consistent with the identification results, e.g., if an offset fault looks like a drift fault, then the fault mode of the candidate will be changed to drift. This type of change is meant to correct errors in fault isolation caused by incorrect symbol generation. For example, a 0+ or 0- signature must be generated in order to hypothesize a drift fault as a candidate, but, for very small drift faults, the slope may not be able to be confidently calculated so instead an offset fault is generated. The identification results are used to correct that error. Another example is distinguishing between offset and intermittent faults, because they produce the same fault signatures at the time of fault occurrence.

To generally account for this type of error, for each fault we identify the parameters for each possible fault profile, and define tests that determine which fault profile is valid. For example, if μ_n or μ_f are less than 0.5, we conclude that the fault is not intermittent. For slowly developing drift faults on noisy sensors, the computed value of Δp can fluctuate above and below the intermittent threshold l , so the fault may look intermittent. So we select three points, as in the computation of drift, and check that the relative change

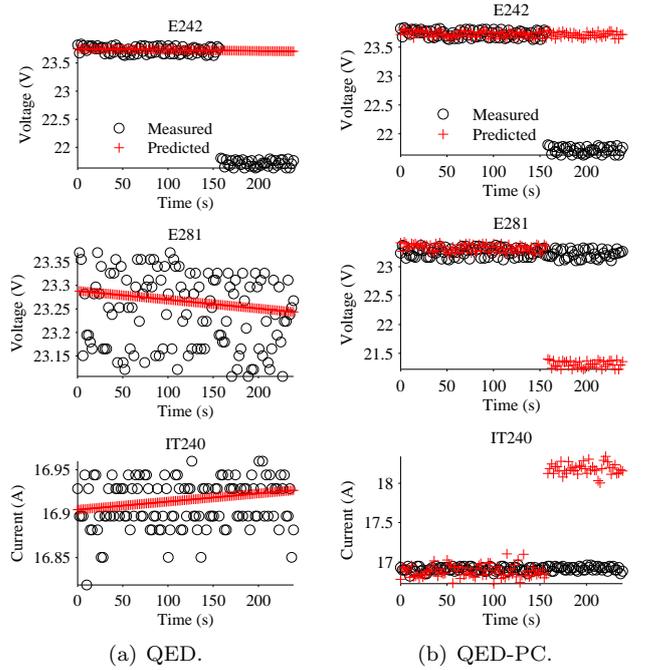


Fig. 4. Selected measured and predicted values for E242 offset fault with $\Delta p = -2.0$.

between Δp at these three points is greater than some threshold (e.g., 15%), and that the absolute differences get larger over these three points. If so, the fault is indeed a drift, and not intermittent.

To determine whether a drift fault is in fact a drift, we use the same test for checking whether an intermittent fault is actually a drift fault, only with lower thresholds. We use lower thresholds because if fault isolation provided the drift fault candidate, then it is because a smooth change was detected, and a positive result on the slope test has high confidence. To determine whether an offset fault is in fact an offset fault, we check that both the intermittent and drift tests fail.

7. EXPERIMENTAL RESULTS

We validate our diagnosis approach on an experimental data set from the ADAPT subset. The data set is from the Second International Diagnostic Competition (DXC'10) (Poll et al., 2010). We first provide an example to demonstrate the approach and the differences between the two algorithms. We then evaluate the algorithms on the complete competition data set.

As an example, we consider an offset fault in E242, with the offset being -2.0 (see Fig. 4). The fault is injected at 158 s, and QED detects the decrease in the measured value almost immediately, at 158.1 s. The candidate set is reduced to a failure in CB236; a failure in EY244; and offset, drift, intermittent offset, and stuck faults of E242. At the next new measured value of E242, the E242 stuck fault is eliminated. At 158.5 s, the symbol generation module determines that E242 did not undergo a zero/nonzero transition, eliminating the failure in CB236 and the failure in EY244. At 168.2 s, the symbol generation module determines that the slope on E242 is 0, eliminating

Table 4. QED Diagnosis Results

Algorithm	M_{fd} (s)	M_{fn}	M_{fp}	M_{da}	M_{fi} (s)	M_{err}
QED	12.63	0.015	0	0.987	125.996	10
QED-PC	17.53	0.023	0	0.980	129.334	12

the E242 drift fault. At 200.5 s, the intermittent faults are eliminated since the identified parameters are inconsistent, leaving only the offset fault in E242. The offset is computed as -1.998 .

QED-PC detects the fault a little later at 158.5 s with deviations in the PCs for E242 (decrease), E281 (increase), and IT240 (decrease). A deviation is detected in IT240 because the PC computes its output based on the measured value of E242, and since the sensor is faulty, the PC predicts a different value for IT240 than the measured value. The initial candidate list consists only of drift, offset, and stuck faults in E242, which is a much smaller candidate set than that initially generated by QED. At 169.5 s, the symbol generation module determines that the slopes on E242, E281, and IT240 are 0, leaving only an offset fault in E242 as a candidate. At 200.5 s, the offset fault mode is confirmed with the offset computed to be -1.995 .

Table 4 summarizes the performance of QED and QED-PC on ADAPT. The metrics consist of the mean time to detect faults M_{fd} , the mean false negative rate M_{fn} , the mean false positive rate M_{fp} , detection accuracy M_{da} , the mean time to isolate faults M_{fi} , and the number of classification errors M_{err} .

Both QED and QED-PC were tuned such that no false alarms were detected. This, however, did result in some missed detections. QED missed a small offset fault in IT281 and a small drift fault in ST516. QED-PC missed these faults as well, in addition to a small offset fault in ST516, because its detector for ST516 had to be less sensitive due to the PC having to model the transients in fan speed. QED-PC has a larger average detection time due to the decreased sensitivity as described in Section 4.

Of the experiments, there were 8 expected classification errors due to faults that could not be distinguished, e.g., DC485 failing and its relay failing produce the same observations. QED had the two missed detections which resulted in 10 total errors, and QED-PC had the 3 missed detections in addition to another case where a fault in E265 could not be eliminated from consideration from a fan fault. This is because the relationship of the fan speed on inverter voltage was not properly modeled, and therefore we could not drop E265 in that case. This resulted in 12 total classification errors.

8. CONCLUSIONS

In this work, we presented two algorithms, QED and QED-PC, which incorporate principles of qualitative event-based fault diagnosis, and we compared the implementation and performances of these two algorithms.

We found that with the PC approach, additional modeling was required and fault detection sensitivity had to be decreased. Relative measurement orderings cannot be used with a purely PC-based approach, however the decoupling

introduced by the PCs enhances diagnosability in its own way. Overall, isolation times were fairly well-matched, because these were computed based on the time of the last change to the candidate set, and the final decision on what the correct fault mode is (e.g., intermittent vs. drift) was often made very late when a large amount of data was available to declare the correct fault mode with high confidence. Generally, QED-PC was faster at isolating sensor faults, whereas QED was faster at isolating nonsensor faults, for the reasons explained in Section 5. Therefore, a new approach that uses residuals from both the global model and PCs would increase robustness, improve overall diagnosability, and enable many of the heuristic fault isolation rules to be eliminated.

In the future, we would like to apply these algorithms to larger systems, e.g., the complete ADAPT system. ADAPT is a hybrid system, and can have multiple faults. Hence, in the future, we will be extending the current approaches to diagnosis of hybrid systems, and multiple fault diagnosis, foundations of which have been laid in (Bregon et al., 2011a) and (Daigle, 2008), respectively.

REFERENCES

- Bregon, A., Alonso, C., Biswas, G., Pulido, B., and Moya, N. (2011). Hybrid systems fault diagnosis with possible conflicts. In *Proc. of the 22nd International Workshop on Principles of Diagnosis*, 36–43.
- Bregon, A., Biswas, G., and Pulido, B. (2012). A decomposition method for nonlinear parameter estimation in TRANSCEND. *IEEE Trans. on Systems, Man, and Cybernetics, Part A*. doi:10.1109/TSMCA.2011.2170065.
- Daigle, M. (2008). *A Qualitative Event-based Approach to Fault Diagnosis of Hybrid Systems*. Ph.D. thesis, Vanderbilt University.
- Daigle, M. and Roychoudhury, I. (2010). Qualitative event-based diagnosis: Case study on the second international diagnostic competition. In *Proc. of the 21st International Workshop on Principles of Diagnosis*, 371–378.
- Daigle, M.J., Koutsoukos, X., and Biswas, G. (2009). A qualitative event-based approach to continuous systems diagnosis. *IEEE Trans. on Control Systems Technology*, 17(4), 780–793.
- Mosterman, P.J. and Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *IEEE Trans. on Systems, Man and Cybernetics, Part A*, 29(6), 554–565.
- Poll, S., de Kleer, J., Feldman, A., Garcia, D., Kurtoglu, T., and Narasimhan, S. (2010). Second international diagnostics competition – DXC’10. In *Proc. of the 21st International Workshop on Principles of Diagnosis*, 355–369.
- Poll, S. et al. (2007). Evaluation, selection, and application of model-based diagnosis tools and approaches. In *AIAA Infotech@Aerospace 2007 Conference and Exhibit*.
- Pulido, B. and Alonso-González, C. (2004). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 34(5), 2192–2206.
- Roychoudhury, I. (2009). *Distributed Diagnosis of Continuous Systems: Global Diagnosis Through Local Analysis*. Ph.D. thesis, Vanderbilt University.