Solving the ADAPT Benchmark Problem - A Student Project Study

Erik Almqvist, Daniel Eriksson, Andreas Lundberg, Emil Nilsson, Niklas Wahlström, Erik Frisk[1], and Mattias Krysander[1]

[1] *Department of Electrical Engineering, Linköping University*
SE-581 83 Linköping, Sweden
{frisk,matkr}@isy.liu.se

ABSTRACT

This paper describes a solution to the Advanced Diagnosis and Prognostics testbed (ADAPT) diagnosis benchmark problem. One main objective was to study and discuss how engineering students, with no diagnosis research background, would solve a challenging diagnosis problem. The study was performed within the framework of a final year project course for control engineering students. A main contribution of the work is the discussion on the development process used by the students.

The solution is based on physical models of components and includes common techniques from control theory, like observers and parameter estimators, together with established algorithms for consistency based fault isolation. The system is fully implemented in C++ and evaluated, using the DXC software platform, with good diagnosis performance.

1 INTRODUCTION

Designing diagnosis systems for monitoring technical systems is a versatile task and often requires knowledge and techniques from more than one discipline. This heterogeneous aspect is also reflected in the scientific community in that researchers from many different fields are working in developing theory, methods, and tools. The main contribution of this paper is to evaluate how simple and sound engineering approaches from control engineering and AI can be used to solve a diagnosis problem.

This investigation is done in the form of a student project, initiated by Erik Frisk and Mattias Krysander, in a course for final year master students at the Department of Electrical Engineering at Linköping University, Sweden. The rules in the course were, 5 students should meet project objectives using maximum 1200 hours during 1 semester. In these 1200 hours, additional course requirements and administrative duties require about 250 hours which means that 950-1000 hours could be dedicated to solving the technical problem. One fundamental *rule* of the game was that very little tutoring was allowed from teachers. The students were final year master students, where 3 had attended an FDI-oriented introductory course in model-based diagnosis and the other two had no previous experience in the field. All 5 students had an electrical engineering background with a clear control theory focus and this is also reflected in the type of solution which closely relates to FDI type solutions as in for example (Blanke *et al.*, 2006; Gertler, 1998). However, also consistency based diagnosis techniques from the AI community is used and the solution is well founded in the theory described in (de Kleer *et al.*, 1992; de Kleer and Williams, 1987).

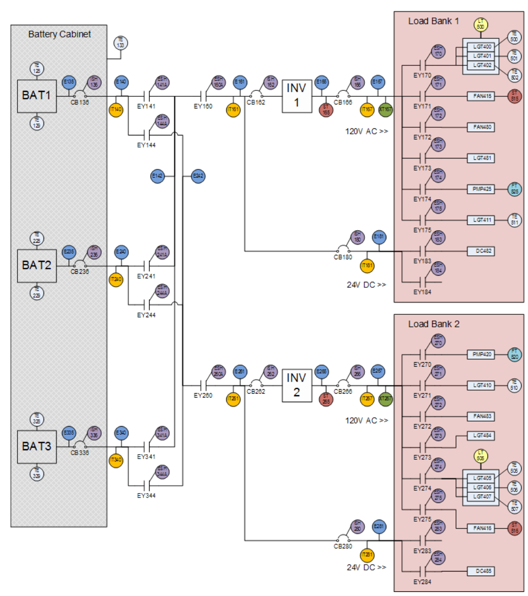

Figure 1: Overview of the ADAPT electrical system.

In short, the project objectives were to solve the Advanced Diagnosis and Prognosis Testbed (ADAPT), industrial track, of the First International Diagnostic Competition[1]. The ADAPT problem is well suited for a student project, it has full documentation, a well developed software platform, evaluation metrics, and still offers a challenging problem. The ADAPT system is an electrical power system sketched in Figure 1. All circles indicate measurements and more or less all components can fail in one or more ways.

An interesting consequence of the project participants background was that the students did not have access to theory and tools available to researchers. Thus, they approached the problem in a more engineering based way. Since, for example, no automatic design tools were available, they implemented a well chosen set of precompiled tests by hand. Due to the rich instrumentation of the process, the number of possible tests is huge and therefore

[1] `http://www.dx-competition.org/`

methods to choose which tests to design were needed. Main contributions of this work, besides the interesting aspect on how the participants approached and solved the problem, is the modeling work and also the design principle using component-local tests in the solution which had some interesting consequences.

Section 2 describes, some parts of, the modeling work done and Section 3 then describes the method used for designing the diagnosis system. A key part of the development process, and also of the evaluation, is the isolability analysis described in Section 4. A discussion on the methods properties, robustness, and evaluation score is found in Section 5 and some conclusions in Section 6.

2 MODELING FOR DIAGNOSIS

When developing a model based diagnosis system, an important cornerstone is to have accurate models of the components within the system. This section describes the mathematical models of some selected components in the system, and also some additional comments about the nature of these components and choices made in the modeling. In addition to the models mentioned below, models of the remaining components such as inverters, sensors, relays and circuit-breakers were also implemented.

2.1 Battery

The model chosen for the battery is that of an ideal voltage source generating a voltage V_0 in series with an internal resistance R_i that depends on the current. With V as the battery terminal voltage, and I as the battery current, the internal resistance can be calculated from measurements according to

$$R_i = \frac{V_0 - V}{I} \tag{1}$$

The voltage V_0 is equal to the open circuit battery voltage and is estimated from such measurements. The voltage and current measurements indicate some dynamics that are neglected here since most of the available data are from stationary working points. The stationary dependence of R_i by I is modeled as

$$R_i = \frac{B}{I^e}, \tag{2}$$

where B and e are model parameters. The structure of (2) was chosen based on empirical observations.

The reason that the model of the battery is focused on its internal resistance is that when the battery is degraded (which is the fault mode) its internal resistance is increased.

The three batteries can be seen to the left in Figure 1 where one is of another brand and model than the other two. Their internal resistances are modeled according to (2) and can all be described by the same model parameters.

The parameters are $B = 0.23$ and $e = 0.51$. This model is only considered valid for currents above 2.0 A, since no data with lower currents was available, and the internal resistance estimates varied more for low currents than for high currents, implying that the model might be increasingly inaccurate for currents below 2.0 A. As a result of this model limitation, no diagnostic statement about the battery is made when currents are below 2.0 A.

The fact that the model does not describe system dynamics, and thereby no diagnostic statements are allowed during transients, is not a significant limitation since transitions between stationary modes takes little time compared to the time the system is in stationary mode.

Even though not included in the model, the open circuit voltage is affected by the charge level of the battery and the battery is discharged during the course of a scenario resulting in a decrease in V_0. This effect can be seen in long load characterization scenarios, but in the 240 seconds long scenarios this effect can be neglected. A model that could be used to describe how V_0 decreases is

$$V_0(t_2) = V_0(t_1) - K T_{\text{abs}} \frac{1}{Q_{\text{nom}}} \int_{t_1}^{t_2} I(t)dt \tag{3}$$

where T_{abs} is absolute battery temperature , Q_{nom} is nominal battery capacity, I(t) is the current out of the battery, and K is a design parameter.

Available measurement data indicates that the current out from one battery sometimes lowers the voltage output of other batteries, even though they according to the system lay-out and relay configurations not are connected to each other. Possible reason for this phenomenon is electromagnetic fields from the wires and/or other components. Because of this phenomenon V_0 has to be determined for all of the three batteries at times when there is no current drawn from any of the batteries. Fortunately all relays are open at the start of the scenarios, so the scenarios always start in a situation where V_0 can be determined for each battery.

Although it is likely that the internal resistance depends on battery temperature, no such connection was found. This was probably due to lack of training data, since the temperature for all data sets was basically the same and fairly constant throughout each data set.

2.2 AC-Loads

The load banks can be seen in the right colored boxes in Figure 1. According to given data, a good model for all AC-loads is to assume that their complex impedance's Z are constant. If voltage, current and phase shift of the load is known, the impedance can be estimated using Ohm's law.

Furthermore, the AC loads are connected in parallel in two separate load banks and only the total admittance for each of these load banks can be measured. For a group of loads connected in parallel their total impedance, is given by

$$\frac{1}{Z_{tot}} = \sum_i \frac{1}{Z_i} \tag{4}$$

Therefore, the AC loads have been defined by their admittance Y, which is the reciprocal of the impedance $Y = Z^{-1}$, in order to make the calculation of the total admittance linear.

Each mode of each load has its own characteristic admittance which can be represented as a point with an associated confidence interval in the complex plane. Figure 2 shows the estimated admittances and confidence intervals.

Some of the loads also have sensors measuring quantities which are affected by the power output of the load. These are light and temperature sensors for some light bulbs, speed transmitters for some fans, and flow transmitters for the pumps. The relations between these quantities and the power output will also be described with models. Unlike the admittances, these systems prove to have a

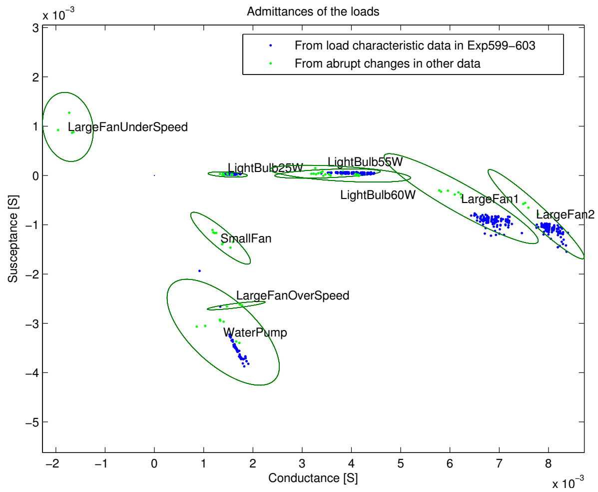

Figure 2: Measurements and models of the admittances for the AC loads with a confidence interval of four standard deviations

dynamic behavior. A first order system has been used

$$\dot{y}(t) = \frac{1}{k_t}(y_P(P(t)) - y(t)) \tag{5}$$

where k_t is a proportionality constant, deciding the swiftness of the system, and $y_P(P(t))$ is the working point of the measured quantity as a function of the output power. For the relation between the working point and the output power, a quasi-linear relation can be used,

$$(y_P(P(t)) - y_0)^p = k_0 \cdot P(t). \tag{6}$$

Here, k_0 is a proportionality constant, y_0 the value of the measured quantity without any power output from the corresponding load, and p is a characteristic exponent coupling the measured quantity with the output power. Since the power has a quadratic relation to speed and flows, $p = 2$ has been chosen for these quantities. For other quantities, a linear relation between the power and the measured quantity, i.e. $p = 1$, is assumed.

3 DIAGNOSIS METHOD

The basic idea of the diagnosis algorithm is to process observations from different parts of the system and determine the set of candidates that are consistent with observations and the model. In particular the minimal diagnoses (de Kleer *et al.*, 1992) are of interest. There are many possibilities when choosing a method for performing this. Here, desirable properties are simplicity, predictability in computing effort, low memory foot print, and good single fault isolability. Based on this, diagnosis is performed with a set of precompiled tests generating conflicts and then a hitting-set based procedure is used to generate the diagnoses. Figure 3 shows the general outline of the approach.

The test quantities correspond to signal processing algorithms together with a threshold to generate an alarm and a corresponding conflict. The fault isolator is then a hitting-set procedure as in, e.g., (de Kleer and Williams, 1987; de Kleer *et al.*, 1992) with a modification to handle multiple component fault modes and negative conflicts. The diagnosis procedure used is described in (Nyberg, 2006).

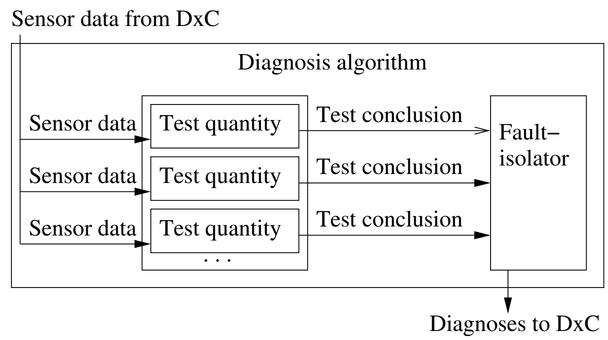

Figure 3: An overview of the diagnosis approach.

One main design effort is thus to choose which tests to design, i.e., which conflicts to detect to achieve a suitable isolability performance. A second step in the design is to, for each conflict, design a suitable test quantity and a corresponding threshold such that an alarm is generated in case of a fault. In addition, the test quantities need to be robust against sensor noise and model uncertainty and still give fast and reliable fault detection.

3.1 Choosing Test Quantities

The choice of test quantities is crucial for the capability of the resulting diagnosis system to isolate between different faults. Since the ADAPT system includes a large number of sensors, there is a lot of analytical redundancy available for performing diagnosis and isolating faults. However, all redundancy cannot be exploited when designing tests by hand, since the number of possible conflicts and therefore also the number of different tests grows exponentially in the number of sensors. Since a complete solution is not feasible, the question is then which tests to design in order to get the most diagnosis performance out of the engineering time spent on test design.

As a guide for selecting tests, the following design philosophy was formulated.

1. For each component in the system, design as many test quantities as possible using only sensors directly connected to that component. Since these tests are devoted to a single component, they will be called component-local tests.
2. The single fault isolability provided by the component-local tests is analysed, and if there are faults not isolable from other faults, tests based on more than one component are added to enhance the overall fault isolability.

Note that single fault isolability is the focus in both design steps above. The suggested design philosophy provides guidance in test selection and design if component-local tests can be constructed. For the ADAPT-system, component-local tests can be constructed for most components, since both the inputs and outputs of these components are measured. This means that much of the fault isolability capability can be achieved by component-local tests and the design work in step 2 becomes limited to minor improvements. In this section the focus will be on the first step, the details of the second step will be described in Section 4. Designing tests according to the proposed design philosophy has several nice properties that will be discussed next.

First, component-local tests can easily be developed in parallel where component experts are responsible for the corresponding component-local tests. Only when the isolability requirements put demands on tests involving several components, different component experts must collaborate. This makes the proposed design philosophy efficient from an organizational point of view.

Second, each component-local test is by construction sensitive to a small or even a minimal set of faults. This is important in combination with the single fault isolability focus. Maximum single fault isolability can be achieved with different test sets, but if tests sensitive to few faults are used to achieve maximum single fault isolability then multiple fault isolability is to a greater extent achieved for free. As an extreme example, consider tests sensitive to exactly one fault. To isolate all single faults using such tests, one test for each fault is needed. However, this set of tests also achieve full multiple fault isolability.

Third, the component-local tests for a specific component type can be reused for each location in the system where this component type is connected with the same sensor setup. For example, the component local test for the relay EY141 using the voltage sensors E140 and E142 and the position sensor ESH141A can directly be applied also to the relays EY160, EY144, EY241, EY244, EY341, EY344, and EY260.

Fourth, in the first step when designing component-local tests, the isolability performance analysis can be restricted to the faults and tests in the considered component. This local view gives an easy performance overview of the isolability where for example the effects of adding and removing tests can easily be seen.

3.2 Test Design Principles

A general design principle when creating tests has been to tune the tests such that the probability of false-alarms is negligible. No mechanism is then needed to retract falsely generated conflicts which makes the fault isolation procedure significantly easier.

The designed tests, see examples below, can be categorized in three different groups:

- Residual tests. A residual can be designed by comparing an expected magnitude of a certain quantity with the measured/observed version of this quantity. For example in the battery the internal resistance is computed as in (1) and compared with its expected value. If this deviation is sufficiently large, the test quantity will alarm.
- Classification tests. Here, the magnitude of a change due to introduced faults is of interest in order to perform a direct isolation/classification. This approach has been used with the estimated admittances described in Section 2.2. When an abrupt change in the estimated admittance has been detected, this change will correspond to the admittance of the load that has been added and/or removed from the load bank, since parallel coupling of admittances fulfill (4). This admittance change will be compared with all admittances of all loads that are turned on at that load bank in order to perform the isolation.
- Logical tests. These tests deals with Boolean signals and analyse if the signals contradict each other. No signal noise has to be assumed and tests with 100% detection accuracy could be designed. These tests can be found in the components affected by Boolean signals, i.e. the relays and the circuit breaker where the actuator is compared with the command signal of these components.

A brief illustration of the set of tests for a relay and a light bulb now follows. For the relay EY141 there exists two different tests

- **Actuator test** - This is a logical test comparing the actuator, u_{act}, with the command signal, x_{com} as
$$\text{if not } u_{\text{act}} = x_{\text{com}} \text{ then fault} \qquad (7)$$
- **Voltage test** - When the relay is closed, $x_{\text{st}} = 1$, the voltage ΔV across should be zero. The residual is computed and thresholded as
$$r(t) = x_{\text{st}}\Delta V < J_r \qquad (8)$$
and an alarm indicates that the relay is stuck open, see Figure 4.

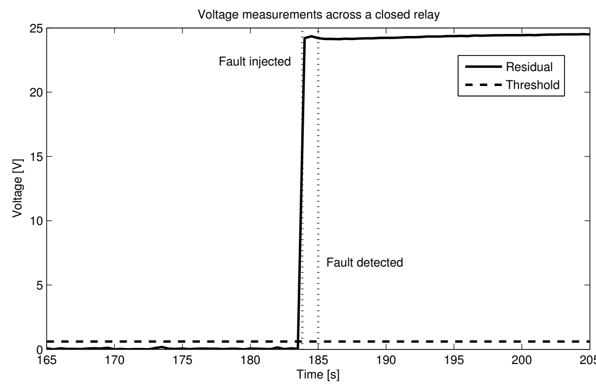

Figure 4: A voltage test reacts after a the relay EY141 has been stuck open at t=183.83 s.

For the 25W light bulb LGT400 there exist three different tests

- **Admittance test** - In this classification test a detected admittance change will be compared with the admittance of this light bulb (see Section 2.2). In order to detect a admittance change, a Kalman filter $\mathcal{KF}$ estimates the admittance $\hat{Y}$ and its covariance P_Y
$$(\hat{Y}_k, P_k^Y) = \mathcal{KF}(Y_k). \qquad (9)$$
This Kalman filter simply computes a low-pass filtered version of the estimated admittance Y. An admittance change ΔY_k with belonging covariance matrix $P_k^{\Delta Y}$ can now be defined as
$$\Delta Y_k = \hat{Y}_{k-1} - Y_k \qquad (10)$$
$$P_k^{\Delta Y} = P_{k-1} + R, \qquad (11)$$
where R is the covariance of the measurement noise. When no fault $\Delta Y_k \approx 0$, however, when the 25W light bulb fails off $\Delta Y_k \approx -Y_{LB}$, where Y_{LB} is the admittance of the 25W light bulb. This is used in a classification test by thresholding the following test quantity
$$T = \|\Delta Y_k - (-Y_{LB})\|_{\Sigma_{LB} + P_k^{\Delta Y}}, \qquad (12)$$
where Σ_{LB} is the covariance matrix of Y_{LB} (represented as an ellipse in Figure 2).

- **Temperature test** - With the temperature measurement the power $\hat{P}$ is observed with a Kalman filter by using (5) and (6) and compared with the nominal 25W. If $\hat{P}$ falls below a certain threshold $\hat{P} < J$ this test will alarm, see Figure 5.
- **Light test** - Compares the estimated power of the three 25W light bulb with the nominal 75W using the light sensor measurement of that light bulb group. This is done in the same way as the temperature test, however $k_t \approx 0$ in (5) since this system does not show any significant dynamics.

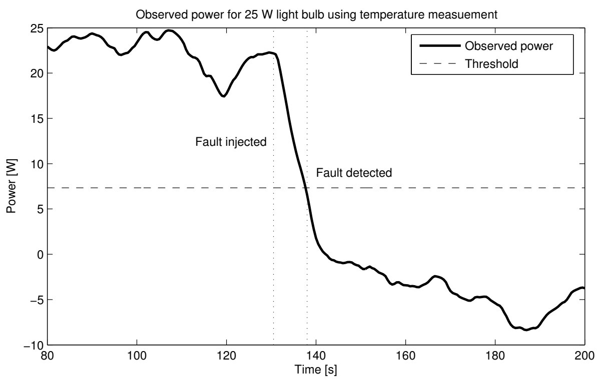

Figure 5: A light bulb fails off at t=130.51 s and a temperature test alarms accordingly 8 s later. Here the threshold is $J = 25/3$ which corresponds to one third of the nominal value 25.

3.3 Fault Isolation Algorithm

As previously stated, the purpose of the fault isolation algorithm is to compute the diagnoses given the conflicts generated from triggered tests. The fault isolation algorithm used here (Nyberg, 2006) assume that the set of conflicts is extended over time and not in conflict with each other. To ensure that this assumption is fulfilled several issues must be handled when considering ADAPT-like systems.

First, one reason for getting non-valid conflicts is if these are generated based on false alarms. As stated in the previous section, this is handled by making tests with very low probability of false alarm. Second, another reason for getting non-valid conflicts is if the mode of the system has switched since the conflict was generated. For the ADAPT-system, fault free components can become faulty but not vice versa, i.e., there are no intermittent or self-healing faults. This is handled by designing tests such that no test can reject a fault mode of a component without also rejecting the fault free mode of the component. This restricts the possible tests that can be included in the diagnosis system but at the same time significantly simplifies computational complexity of running the fault isolation algorithm.

Another simplifying assumption regarding the fault isolation algorithm is that all faults in all components have equal a priori probability to occur. The minimal diagnoses which contain the least number of faults, i.e. the minimum cardinality diagnoses, are the most probable diagnoses and all of these have equal a priori probability. These diagnoses are an order of magnitude more likely than the other diagnoses including more number of faulty components and a simplification is made by considering only the minimal cardinality diagnoses as possible diagnoses. Hence, if there are n minimum cardinality diagnoses the probability of each of these diagnoses is approximated to be n^{-1}. If a new test quantity reacts which updates the diagnoses such that the cardinality of the new minimum cardinality diagnoses increases, the probability is again equally split between the new minimum cardinality diagnoses. When there is only one minimum cardinality diagnosis, the faults are considered to be isolated. In this way the status of the diagnosis system is updated as new data is processed.

3.4 Isolability Performance and Design Choices

One significant design choice for the fault isolation algorithm was that all past test conclusions from alarming test quantities affects the diagnosis statement that the diagnosis system outputs. This increased the importance of keeping the false alarm rate for individual test quantities low, since false test conclusions would not be forgotten, thereby corrupting the diagnosis statement for the rest of the particular scenario. Since the test quantities design had low false alarm rates in focus, some of the test quantities are rather slow. One example of this is the test quantities that alarm if a scalar sensor is stuck or offset. These test quantities could have been designed to detect faults faster, at the price of a higher false alarm rate.

The method by which test quantities was created was not automatic, i.e. they were not generated automatically and exhaustively from the mathematical descriptions of each component and component modes, see examples in Section 5.3. This of course resulted in that the number of test quantities the diagnosis system contains is considerably lower than it could have been, which in turn affects the isolability performance negatively. Still, the straightforward method of creating test quantities based on the behavior of one component at a time gives a fairly good isolability performance for this system.

4 ISOLABILITY ANALYSIS

A detectability and isolability analysis is quite naturally an important tool to verify that the isolability properties of a given diagnosis system meets the requirements. In particular, it is interesting to evaluate what kind of performance is possible, given the choice to only design component-local test quantities. In addition, when developing a diagnosis system based on precompiled tests, isolability analysis is an important tool for providing suggestions for new test quantities to design to enhance performance when a current set of tests is not sufficient. These analyses are thus an integrated part of the diagnosis system development process where inadequate isolability performance is discovered as early as possible.

4.1 Isolability Properties of the System

The analysis is based on the fundamentals in (de Kleer *et al.*, 1992; Reiter, 1987), but the computational approach is inspired by (Krysander and Frisk, 2008) but is also closely related to (Travé-Massuyès *et al.*, 2006). The approach is in line with (Pucel *et al.*, 2009) which could easily be used to extend the analysis to multiple fault analysis. The following definitions, which follows the theory in (Krysander and Frisk, 2008), are used:

Definition 1 (Isolability for a set of tests $\mathcal{T}$)**.** *A fault mode b_i is isolable from a fault mode b_j if there exists a test $T \in \mathcal{T}$ such that*

$$b_i \notin H^0,\ b_j \in H^0$$

where H^0 is the null hypothesis for test T.

Thus, a fault b_i is isolable from a fault b_j if test T will never alarm for fault b_j and there exists a fault instantiation for mode b_i such that the test alarms. A simple way to represent single fault isolability properties is by a matrix $\mathcal{I}$ where the position (i, j) is defined as

$$\mathcal{I}(i,j) = \begin{cases} 0 & \text{fault } j \text{ is isolable from fault } i \\ 1 & \text{otherwise} \end{cases}$$

Note that the isolability matrix is an optimistic estimation of the properties of the diagnosis system. This means that even if the detectability or isolability analysis shows that all possible faults *can* be detected, it is not the same statement as that they *will* be detected in a particular situation. Examples of this include test quantities operating on the sensors and whether or not they are within their normal working intervals. If not then there is an offset fault detected. However, if the offset does not make the sensor leave its working interval then the fault is not detected. Other examples include faults in the batteries and the inverters. The faults are detectable by only one and two test quantities respectively for each type of component. Isolability of these faults requires good detectability from these test quantities.

The isolability matrix for the designed diagnosis system is shown in Figure 6. In an ideal case, where each

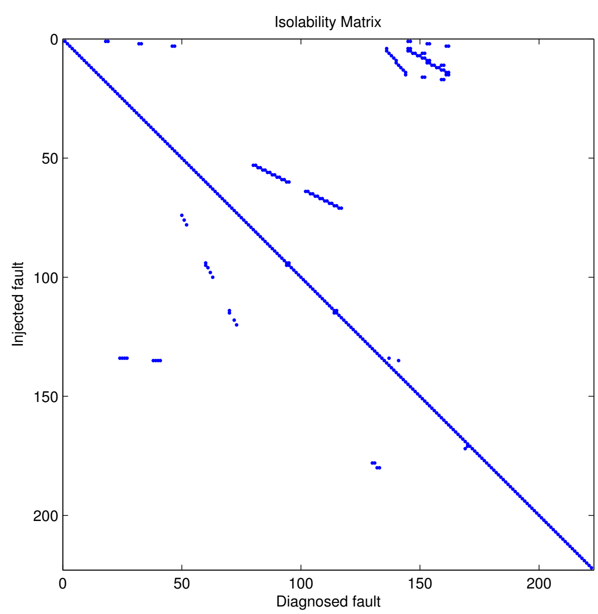

Figure 6: The isolability matrix of the diagnosis system.

fault is uniquely diagnosable, the 222×222 matrix in Figure 6 should be an identity matrix. Unfortunately, this is not the case for the diagnosis system at hand and a more detailed analysis of the non-diagnosable pairs is included below. But first, a brief comment on how the analysis were used during the development process.

During the development process, computation of an isolability matrix directly gives indications of which new tests that need to be designed to increase performance. For example, if fault b_i is not isolable from b_j, i.e., position (i, j) in matrix $\mathcal{I}$ is 1. To introduce corresponding isolability, a test that is sensitive to fault b_i and *not* fault b_j is needed, i.e. there is direct and concrete feedback to the diagnosis systems designer which test to design.

Note that even though the above analysis is based on a single fault assumption, it is clear that the set of tests also can detect and isolate multiple faults. In summary, the diagnosis system consists of 315 different test quantities, all with different possible test-conclusions, and can detect 222 different faults which is the total amount for the electrical power system.

4.2 Non Isolable Pairs of Faults

As noted above, the isolability matrix in Figure 6 is not an identity matrix and each off-diagonal non-zero element corresponds to a non-isolable pair of faults. A more detailed analysis of each off-diagonal element in $\mathcal{I}$ reveals that there are 9 fundamental reasons. Below is a brief discussion for the 9 situations that lead to non-isolable pairs of faults.

1. **Non-isolable fault in battery**. According to the battery model, degradation only affects the internal resistance but not, for example, the temperature. This results in that battery degradation cannot be isolated from faults in the sensors used to compute the internal resistance.
2. **Non-isolable faults in the circuit breakers**. Since the circuit breaker can be closed or open while being in a nominal mode (Nominal or Tripped), the only way to detect a fault in a circuit breaker is with current measurements, so only one test can be constructed and therefore faults in the circuit breaker cannot be isolated from each other.
3. **Non-isolable stuck in load relays**. Because there are no current or voltage sensors for each individual load, it is not possible to isolate faults in the relay position sensor for the relay of the load from faults in the relay itself.
4. **Non-isolable stuck in load relays without loads**. There exists empty slots in the load banks, i.e. relays that are not controlling any load. Therefore the only possible test for the relay and its position sensor is the one comparing the relay command and the sensor reading. This makes it possible to detect faults in any of these two components, but none of the two faults can be isolated from the other.
5. **Non-isolable StuckClosed in relays except load relays**. For relays that are not directly connected to loads, measurements of current and voltage are available. In addition to the test that is possible for load relays, this makes it possible also to construct tests that detects faults in the relay position sensor but not faults in the relays. This results in that only faults in the relay is non-isolable from positions sensor faults, while the reverse is no longer non-isolable.
6. **Non-isolable FailedOff in inverter**. Since it is not possible, according to the model of the inverter, to detect that the inverter is off when it should be on, without using current and voltage sensors related to the inverter, the fault mode FailedOff will not be isolable from the above mentioned sensors' faults.

7. **Non-isolable faults in DC loads**. Unlike the AC loads where a test using the phase sensor can be designed, only one test for detecting FailedOff can be made for DC loads, a test using a current sensor. Therefore it is not possible to isolate faults in the load from faults on the current sensor.

8. **Non-isolable FailedOff in loads**. There are pairs of loads, in the same load bank, that have similar admittance, making the admittance change test unable to separate the FailedOff faults in these loads from each other. However, in these cases, one load in the pair of loads has some other sensor connected to it, resulting in that the other load's FailedOff mode is non-isolable from FailedOff in the load with the additional sensor.

9. **Non-isolable FlowBlocked in Water Pump**. Since blocked flow do not manifest itself by any change of admittance, the only way to detect blocked flow is by using the flow sensor. Therefore the blocked flow fault cannot be isolated from faults in the sensor.

5 METHOD DISCUSSION AND EVALUATION

A discussion now follows on the design process, how the tests were selected, and also the important topic of robustness is treated.

5.1 Component-local Tests and the Design Process

As presented in Sections 2 and 3, the modeling and diagnosis method is based on the specific component and sensor configuration of this system. In addition, the diagnosis system development, and in particular the test design, is close to the modeling effort due to the component-local approach.

For each component/sensor group, its corresponding position in the circuit topology, how it interacts with other components, and what is measured influences the possible diagnostic performance. Therefore, due to the chosen approach, one can expect a possible limitation in diagnosis performance. But there are also advantages with the approach. In the implementation, for each component there is a local function that implements all tests for that component. Since the tests are made manually, a high degree of flexibility in method is allowed while still keeping a local view of the system. For example, it is straightforward to mix tests that are based on basic logic expressions, Kalman filters to monitor dynamic systems, and also include tests based on inequalities etc. Another nice property of the approach is that, since both models and tests are local, if the same type of component appears somewhere else in the system, the engineering effort can be directly reused. Note that this is not the case for more general types of tests that utilizes the circuit topology and utilizes interaction between component. Further, the local property also makes the development process easier since engineers can work with a high degree of independence.

However, this loss in generality makes it difficult to handle changes in the component/sensor configuration since the implementation is strongly connected to this configuration. In addition, as a consequence of the component based diagnoser design, the work of tuning thresholds parameter is more cumbersome since each test quantity will have its own special parameter weighting different performance properties. Due to loss of generality they are often difficult to relate to each other.

5.2 Test Selection

As described in Section 3.1, no automatic procedure has been used for neither developing nor choosing which tests to implement. The choices have been made by engineering intuition and with guidance from the isolability analysis as described in Section 4 during the work process. For instance, in Section 4.2 it is clearly stated where efforts should be made in order to increase isolation performance, either by making the models of these components better or by creating new redundant test quantities. This constant feedback from the isolability analysis has proved to be an effective approach, and is easy to incorporate in the component based design work.

The number of possible test quantities grows exponentially with the degree of redundancy in the system, which means more or less the number of sensors. For a system like the ADAPT, with a relatively large number of sensors, to systematically find all possible test quantities is not feasible. Fortunately, only a small fraction of these possible tests are needed to reach the required isolability performance and this is typically due to the fact that many possible tests react to the same, or similar, set of faults. To find an optimal, in some sense, choice of tests is difficult since many aspects have to be considered, not only the ideal isolability performance. Here, engineering intuition supported by the isolability analysis gave that about 300 tests, most of them very simple, very sufficient to reach stated performance requirements of the diagnosis system.

5.3 Robustness Analysis

This section contains a discussion of the robustness properties of the diagnosis system, i.e. how sensitive the different test quantities are to modeling errors and measurement noise. Because of reasons stated earlier, it was important to keep the false alarm rate for each test quantity low, and this affected the test quantity design, which in turn affected the robustness of the complete diagnosis system. The robustness is for the tests, corresponding to the components mentioned in Section 2.

1. **Battery**. The battery is a rather complex system, and the model used does not capture all of its behavior, making it difficult to make precise test conclusions. Thus, to achieve good robustness properties, thresholds had to be set rather high resulting in good robustness properties but at the cost of lowered detection performance.

2. **Load**. The admittance tests detect changes at the load bank by comparing new measured values of the current and the phase with the predicted values from the previous iteration. If this difference is sufficiently large, a fault is assumed to have been detected. A tuning parameter in the algorithm weighs how large this difference must be, based on the variance of the measurement signals, in order to raise an alarm. Thus, this parameter weighs robustness (to signal noise) against detectability. Furthermore, all loads have been modeled to behave as a constant impedance. This assumption is more correct for some loads than for others. However, the modeling procedure does allow the user to specify variances for the model parameters, making it possible to, in a straightforward way, handle these uncertainties. There is also an algorithm parameter deciding the confidence interval (number of standard deviations) to be used (see Figure 3). Thus, this parameter weighs robustness (to model errors) against isolability. With these two

parameters the test can be made arbitrary robust at expense of detect-/isolability performance.

5.4 Result compared to participants in competition

To evaluate the performance of the developed diagnosis algorithm an evaluator application is available. The score of running the evaluator with the competition data can be seen in Table 1. The numbers in parenthesis shows the position comparing to the 6 participating solutions in the ADAPT-competition 2009. Data of isolation accuracy did not have a corresponding post in the competition result and could therefore not be ranked.[2]

Table 1: The score of the diagnosis algorithm when running the evaluator with competition data.

	Score	Position
False Positives Rate:	0.2553	(3)
False Negatives Rate:	0.0685	(2)
Detection Accuracy:	0.8583	(2)
Isolation Accuracy:	0.9929	(?)
Mean Time To Detect:	6788.8971 ms	(4)
Mean Time To Isolate:	29081.1733 ms	(4)

The results confirms the achievement of low false-alarm rate at the cost of long detection and isolation mean time. A longer mean time to isolate compared to the time to detect is probably caused by some test quantities that need long time for detection, e.g. the battery tests and inverter tests. The false positives rate is also high as a consequence of preferring missed detections to false-alarms.

6 CONCLUSIONS

A student project solution to the ADAPT-challenge has been presented. The diagnosis system was completed with five last year master students working in total 1200 h, out of which approximately 250 h were allocated to administrative and other course requirements. Some of the students had attended an introductory FDI-oriented course in model-based diagnosis and the other students had no previous experience in this field. In spite of the limited prior experiences in diagnosis, the group succeeded in systematically designing a good diagnosis system within the given time-limit.

The design methodology included several interesting ideas worth mentioning. Since test design was not automatized, only a limited number of handmade tests could be designed during the project. To achieve good diagnosis performance, these tests needed to be of high quality and properly selected. A parallel design process was also needed to utilize all group members efficiently.

The proposed solution is a component-based approach where each student is responsible for one type of component. With this approach both modeling and test design is done locally for one component at a time and by the same person. The component-local approach makes the design task comprehensible and modeling and test design become naturally integrated to first achieve a desired diagnosis performance on the component-level.

[2]Other criterions in the competition were evaluation of CPU time and memory usage. These could not be evaluated in Linux, which was used during the development of the diagnosis algorithm, and are therefore not presented.

Furthermore, component-local tests can be reused for all locations in the system where the component-type together with the needed sensors appear. This is not the case for more general test approaches taking into account the specific system topology.

The isolability analysis was an integrated part of diagnosis system design by pointing out which isolability properties that was not achieved using only component-local tests. By targeting the design of more complex tests, monitoring more than one component, to achieve only the missing isolability properties, the joint efforts including several group members could be kept to a minimum.

The resulting 314 tests range over a wide spectrum of different kinds of tests including typical residual based tests, classification tests, and logical tests. This is a result of applying appropriate methods for each specific test case and component, for example in some cases dynamics is important, in other cases noise is the most important characteristics.

The developed methodology fits into a standard consistency-based diagnosis framework and modified hitting-set based isolation algorithms can be directly applied. Evaluating the developed diagnosis algorithm on competition data shows good detection accuracy and low false alarm rate but with a quite long detection and isolation time. In summary the diagnosis algorithm has a competitive performance compared to DXC'09 competitors.

REFERENCES

(Blanke *et al.*, 2006) M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, 2006.

(de Kleer and Williams, 1987) J. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

(de Kleer *et al.*, 1992) J. de Kleer, A. Mackworth, and R. Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56, 1992.

(Gertler, 1998) J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, Inc., 1998.

(Krysander and Frisk, 2008) Mattias Krysander and Erik Frisk. Sensor placement for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(6):1398–1410, 2008.

(Nyberg, 2006) Mattias Nyberg. A fault isolation algorithm for the case of multiple faults and multiple fault types. In *Proceedings of IFAC Safeprocess'06*, Beijing, China, 2006. Extended version to Appear in IEEE transaction on systems Man and Cybernetics - Part A.

(Pucel *et al.*, 2009) X. Pucel, W. Mayer, and M. Stumptner. Diagnosability analysis without fault models. In *20th International Workshop on Principles of Diagnosis (DX-09)*, pages 67–74, Stockholm, Sweden, 2009.

(Reiter, 1987) R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.

(Travé-Massuyès *et al.*, 2006) L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Transactions on Systems Man and Cybernetics - Part A*, 36(6), 2006.