AN INTEGRATED MODEL-BASED APPROACH FOR REAL-TIME ON-LINE DIAGNOSIS OF COMPLEX SYSTEMS

Frederic D. McKenzie

University of Central Florida

fdm@engr.ucf.edu

Avelino J. Gonzalez

University of Central Florida

ajg@ece.engr.ucf.edu

Robert Morris

Florida Institute of Technology

morris@cs.fit.edu

ABSTRACT

Model-based diagnostic programs have been shown to be useful in isolating unpredictable faults in various types of systems. Due to the complex nature of many of these systems, models used by these programs to represent monitored systems have traditionally imposed restrictions on domain representations. These restrictions can make it difficult (and often impossible) to model a domain whose behavior is *global* in nature. By global, we mean behavior that affects system variables in parts of the system not directly related to the component in question. Analog electrical circuits and hydraulic circuits are only but a few examples of such global systems. Accurate modelling of the behavior of these global systems is very often essential for obtaining a correct diagnosis. In complex systems such as those typically found in the electrical power distribution domain, global behavior can be observed when voltages and currents throughout an entire system are affected by local load fluctuations, transient disturbances, faults, or circuit re-configurations, even when these are in remote parts of the circuit. Traditional models used in diagnosis have not been able to easily reflect these global interactions, and as a result, monitoring and diagnostic capabilities of model-based systems dependent upon such models are degraded significantly. This paper presents an implementation that can correctly simulate power systems and other such complex systems by overcoming the problem of representing global behavior while preserving the diagnostic abilities of structure-function models in model-based reasoning methodologies. This paper describes the integration of *robust models*, within the conventional device-centered models. These robust models are mathematically-accurate system models normally used in quantitative simulation for the purpose of system analysis. If used, within the conventional device-centered models, they can provide the functionality needed in a structure-function model-based diagnostic paradigm and therefore eliminate the problem of representing global behaviors in diagnosis. This paper further describes a conflict-oriented diagnostic technique used in conjunction with robust models to obtain real-time on-line FDIR (Fault Diagnosis, Isolation, and Recovery).

1.0 INTRODUCTION

The autonomous operation and control of systems is an important topic in domains where constant supervision of critical values is needed, large amounts of data are processed, and rapid response to various phenomena is a must. One area associated with the autonomous operation and maintenance of systems involves the handling of faulty system parts. Faulty parts must be detected and the system must be restored to an acceptable state through *fault diagnosis*, *isolation*, and *recovery* (FDIR) or by troubleshooting and repair. In those systems that must remain on-line for sustained periods of time before repair is possible, redundancy is usually designed into the system. This enables the system to remain operational by establishing an alternate pathway should the normal pathway contain a faulty component. This action is called *recovery*. The location and incapacitation of the pathway that contains the fault is called fault *isolation*. This will minimize potential damage to other devices in the system and prevent the faulty pathway from interfering with the normal operation of a restored system. Once the system has been restored to normal operation, repair can take place at the convenience of the technician. The act of identifying the faulty component whether by manual methods (troubleshooting) or automated methods (diagnostic programs) is called *diagnosis*.

1.1 Approaches to Real-time Diagnosis

Diagnostic knowledge for complex systems can be found in several forms. The first of these is termed *shallow reasoning* and is most often used in reasoning methodologies that are associative in nature. Associative systems rely on experience-based knowledge as typically found in human experts. This expertise is most often represented in the form of rules in the traditional rule-based knowledge-based systems. For simple systems, this method can be quite effective. However, as monitored systems become more complex, the large number of knowledge bits (i.e., rules, frames) to be generated during development as well as executed at run time, can result in a slow and computationally expensive system.

Furthermore, associative systems cannot generally adapt to unforeseen situations, as all fault modes must be predefined by the developer within a *fault model*.

There are numerous diagnostic systems described in the literature which employ associative techniques, and an exhaustive review of these is not warranted here. Nevertheless, some notable ones exist that are closely related to our work. One of these is a diagnostic tool called the *Fault Recovery and Management Expert System* (FRAMES) [3], [4], [26], [27]. FRAMES is a largely associative system implemented in LISP which performs FDIR on a spacecraft power distribution system. While it performs isolation and recovery quite capably and quickly, it depends on the interrupting device itself for primary fault detection and isolation.

Gonzalez et al [1986] developed a purely rule-based system called *GenAID* to diagnose abnormal conditions in real time for a turbine-generator system. The magnitude of the knowledge base (around 6,000 to 7,000 rules) speaks to the significant development effort required for such large systems. Nevertheless, the domain of generator diagnostics is not easily defined through first principles and the real-time requirements of such a system are in the order of minutes which is supportive of a slow diagnostic paradigm. Furthermore, this system is advisory in nature and does not unilaterally carry out any control functions. GenAID has been in continuous commercial operation since 1985.

Another similar system called the *Generator Expert Monitoring System* (GEMS) [18] employs Bayesian Belief networks to represent the same associative knowledge for on-line diagnosis of electrical generators. There are many other diagnostic systems that employ shallow reasoning or fault models[5], [12], [16], [22], [23], [29], [30], [31].

These serious disadvantages of associative systems described above, however, have caused researchers to search for more robust ways to carry out the diagnostic function. This has led to the technique known as the *first principles* approach which is said to use *deep knowledge*. This approach encompasses what is known as *model-based reasoning* (MBR), and is often used in encoding the transfer (input-to-output) equations or constraints in *structure-behavior* models. These structure-behavior models are designed to predict and/or simulate the behavior of the monitored system. The structure-behavior approach has traditionally required an assignment of directionality to its objects and the use of the *principle of locality* that suggests causal relationships between component neighbors. These restrictions give rise to the problem of representing global behaviors. Both associative and first principle-based reasoning methodologies employ either a qualitative or quantitative approach to representing measured values throughout the modelled system. Diagnostic systems with qualitative methodologies [9], [21], [31], however, are susceptible to faults that may be undetectable in macro-sensitive representations.

Adamovits & Pagurek [1993] employ a hybrid method in the system called Multiple Fault Diagnostic System (MFDS) also intended to perform FDIR on electric power systems. The authors combine a robust first principles model of a domain with shallow reasoning diagnostic control structures. Here, global behavior is handled by shallow reasoning procedures during a *hypothesis space pruning* step which weeds out many of the hypothesis that would have been due to global phenomena [1]. This step, as it is shallow reasoning, requires fault models. Thus, any global behavior exhibited during the malfunction of the device is handled by shallow reasoning procedures rather than by the robust model. Later, during a testing phase, the robust model is used to determine newly calculated values for global behaviors that occur with changes to the resistive equivalence of the electric circuit due to structure changes from switches opening and closing.

The resulting implementation is a successful strategy that effectively handles multiple faults. However, the implementation suffers from the problems brought on by fault models, inefficient candidate reduction, and much off-line testing. The time complexity involved is in the order of minutes with the majority of time being spent in the Prolog-implemented reasoning procedures [1].

Another system described in the literature uses both model-based and associative methods to reduce candidates. Xiang & Srihari [1986] use a strategy for diagnostic reasoning that encompasses both shallow and deep reasoning (i.e., reasoning from experience and reasoning from first principles). The approach used in the design of this system is to first diagnose using empirical knowledge and then, if need be, using a model-based approach. When using the model-based approach, reasoning takes place qualitatively at first; then, if need be, quantitatively.

A fault identification method that uses a structure-behavior model eliminates the problem of unforeseen faults since these would be modelled implicitly in the objects or devices themselves and the knowledge base involved would not grow based on the number of fault patterns of a system but rather would be based on the number of devices in the system. This is one significant benefit of structure-behavior systems.

Fesq et al. [1992] employs a constraint suspension technique (see [7]) in developing a system called Marple. When a fault occurs, the offending device propagates the error and the behavior of a number of objects will then become inconsistent with the modelled system. The lowest component is ignored or suspended and its predecessor (or the device from which this component gets its input) is observed. If an inconsistency still exists, then the next predecessor is suspended and so on until there are no more inconsistencies between the remaining devices and the model. The last component to be suspended is found to be the fault.

Suspects are viewed as components that are causally connected upstream of a discrepancy. In the case of many device-centered models, causality is modelled as local behaviors. Behaviors that are nonlocal (i.e., global) are difficult or impossible to represent in device-centered models. At times these *global behaviors* do not follow any perceived physical connections such as in a bridge fault where the actual physical connectivity of a device is accidentally compromised by a foreign influence such as flying solder in a circuit board. In the candidate generation step, such a multiple fault scenario would require hypothesis sets that included the two components in order that the diagnoser could reason that the two failed at the same time. In theory, causalities can occur among every multiple of components possible.

To account for all probabilities, every component in the system should be a suspect. However, this is not computationally acceptable. Therefore, diagnostic tool designers discount small probability scenarios in order to handle the majority of faults. For example, device-centered models tag the components upstream of a discrepancy as its suspects.

Typically, the output values of modelled components are predicted by propagating primary input values downstream until the final downstream components are met. These end components are usually sensors. The predicted values for these sensors are compared with the actual measured values. If any of these comparisons reveal a discrepancy, the diagnostic procedure is initiated. Suspects are obtained from the upstream components of the discrepant sensor or sensors. If these upstream components were dependent upon further upstream components, the further upstream components are also added to the suspect list. These dependencies are continuously checked upstream until the primary inputs are reached. This process can be simplified by keeping a dependency record for each predicted sensor value. The combination of the dependency records (structural connectivity of the components) for all discrepant sensor readings forms the basis for a suspect list. The suspect list may then be reduced by combining sets of the dependency records. Intuitively, (as well as by proof as in [8]), the intersection of the dependency records (assuming more than one discrepant sensor, i.e., more than one dependency record) will contain the culprit (failed component). This is only valid, however, when assuming unidirectional causal models because a faulty behavior in a particular branch that causes discrepancies in disjointed or higher branches cannot be identified by intersection. This method of generating suspects is termed *dependency based* [15].

A third method, called the *conflict-oriented view*, is "a more general framework than the intuitive notion of upstream tracing" [15]. Predictions of the behavior of components are generated. These predictions are made with the assumption that all of the components are operating correctly. Discrepancies indicate that the assumption of correct operation is false for at least one of the components.

Hamscher & Davis [1987] indicate that "in domains for which components' causal direction is the sole source of dependencies, there is little distinction between the 'upstream tracing' and the 'conflict-oriented' views." The benefit of the conflict-oriented view is that it more readily accommodates components that are non-directional in nature because distinctions between inputs and outputs need not be made [15]. The disadvantage of the conflict-oriented view is that it requires sensors between all components so that conflicts can be localized at the component level. A possible fix to this is to supplement this approach with a *fault envisionment* [15] method that will allow conflicts to be extrapolated to nearby sensors. Hamscher & Davis [1987], however, note that fault envisionment requires a predefined set of misbehaviors for each component so that simulation may be possible. The potential exponential growth of these predefined misbehaviors is the significant disadvantage of fault envisionment.

Other work in model-based diagnostics of electric power systems FDIR was carried out by Lee [1993], Adams [1986], and Blasdel [1987], with varying degrees of success.

2.0 ON-LINE, REAL-TIME, DIAGNOSIS USING CONFLICT SETS AND ROBUST MODELS

The objective of the work described here is to devise, implement and test diagnostic techniques that can enable real-time FDIR without the disadvantages of having to pre-determine all potential faults. We utilize Reiter's [1987] conflict set paradigm to produce a conflict-oriented diagnostic algorithm together with a robust modeling approach, integrated within a structure-behavior model. This system meets the requirements stated above without requiring fault envisionment or other such auxiliary techniques. The global scope of the diagnostic engine employed allows for the detection of defective sensors as well as multiple simultaneous independent faults.

2.1 Approach to real-time model-based FDIR

The problem addressed by this investigation was to enhance structure-behavior MBR in order to allow it to be applied to systems which exhibit global behavior and whose response time is exceedingly fast (<< 1 second) such as electric power systems.

In order to address this problem, the research effort focused on two issues: 1) the application of a diagnostic reasoning technique which does not require directional current flow, and; 2) modelling the target system in a more robust fashion than that allowed by traditional device-centered models. These two issues go hand in hand, as they are both needed to solve the problem.

The first issue was resolved by applying a conflict-oriented approach to an existing device-centered framework (e.g., the *Knowledge-based Autonomous Test Engineer* (KATE) model-based diagnostic and control system [28]) which originally used the constraint suspension approach. The efficient conflict-oriented approach replaced the slow diagnostic algorithms used by constraint suspension that employed inversion and iterative propagation. This enhanced the speed of the diagnosis considerably but it still had the slow and cumbersome method of representing the global behavior of the domain of electric power systems. The solution of the second issue was comparatively easier in that robust models of electric power systems have existed for years, and are quite mature. The difficulty lay in integrating these with the device-centered paradigm.

These two solutions are further described below.

3.0 THE CONFLICT-ORIENTED APPROACH

To incorporate robust models in a device-centered model-based diagnostic reasoner (which would provide the needed speed improvement in isolating faults and eliminates cumbersome meta-object strategies for representing global behavior [20]), a diagnostic method was required that improves upon the constraint suspension method used in KATE which relies upon time consuming substitution and recalculation (re-prediction) functions.

Referring to Figure 1, one employing constraint suspension would say, "Let me forget about applying the value to the transfer function of B at node 1 so that a value for node 2 can be obtained (which is the normal way propagation would take place). Instead, let me use the value read by the sensor FS applied to the inverted transfer function of F to obtain a value for node 2." This new value is propagated throughout the network (downstream through components D, E, and F). If the predicted values now match the previously observed values, component B is a possible culprit. Otherwise, component B is exonerated. Notice that constraint suspension of component B is achieved by ignoring its transfer function and using an observed discrepant value and inverting it to find a value to replace its transfer function result. This inverting and repropagation could take place several times per suspect component. Also, notice that for global behavior, constraint suspension is difficult or impossible to achieve since the behavior of the system is not localized to individual components. Therefore, inversion or suspension of constraints are not guaranteed to be always accurate or effective.

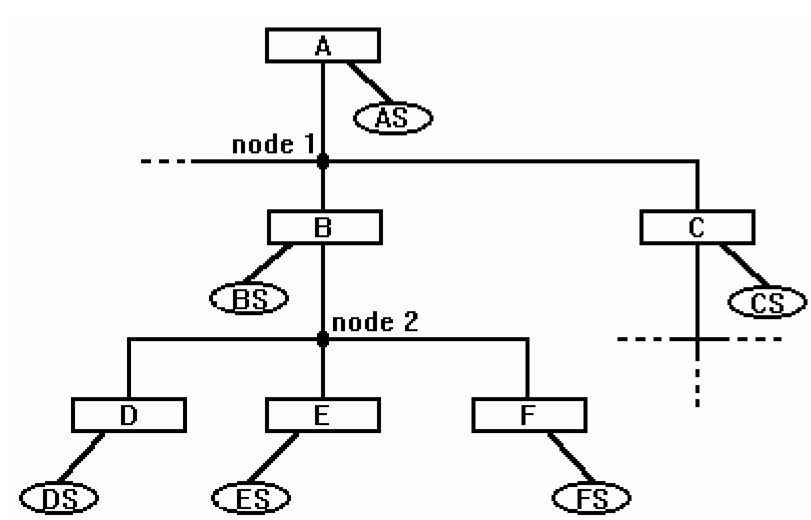

Figure 1 - Constraint Suspension Example

A diagnostic reasoning method that employs a conflict-oriented view can be used to replace the constraint suspension method and eliminate the use of inversion and iterative propagation in the fault isolation step. The combination of robust models and conflict-oriented reasoning, applied within a device-centered diagnostic framework, results in a robust, real-time diagnostic reasoner for power systems.

A methodology that reduces the set of all devices to the set that contains the faulty component(s) by utilizing *conflict sets* (groups of components that are inconsistent with the correct behavior of the system when all are assumed to be normal) solely, and by performing specified set operations on the conflict sets is called the *conflict-oriented approach*. Reiter [1987] outlines a formal approach to the use of conflict sets in first principles methods. The formalisms used to express Reiter's approach will be used to introduce the approach developed for the IPC diagnostic reasoning system.

Reiter describes a **system** as a pair (SD, COMPONENTS) where SD, or system description, is a first principles representation. Reiter uses a set of first-order equations. A system along with its observations is described as (SD, COMPONENTS, OBS). Using AB() to denote abnormality and ¬AB() to denote normality (not abnormal), a conflict is described as $SD \cup \{\neg AB(c1),...,\neg AB(Cn)\} \cup OBS$ being inconsistent. $SD \cup \{\neg AB(c1),...,\neg AB(Cn)\} \cup OBS$ says that when all the components are normal, the system description will yield the same values as the observed values. When it is inconsistent (i.e., the observed values are different), there is a conflict.

The trivial way to restore consistency is $SD \cup \{AB(c1),...,AB(Cn)\} \cup OBS$. This is the assumption that <u>all</u> the components are abnormal. The **Principle of Parsimony** suggests that a diagnosis be made up of a *minimal set* of abnormal components, that is, only those components that are behaving abnormally. Determining this minimal set is a better way to restore consistency. A diagnosis, then, is a minimal set $\Delta \subseteq$ COMPONENTS such that $SD \cup OBS \cup \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in COMPONENTS - \Delta\}$ is consistent. In other words, a diagnosis is a subset of all components that when assumed to be the only abnormal components in the system, results in the prediction of the system description being consistent with the observations [25].

Reiter further defines a *hitting set* H which is a set that contains one or more elements of each set $S \in C$ (C is a collection of sets). This hitting set is minimal iff there exists no subset of H that is a hitting set for C. The minimal hitting set Δ is the minimal set that Reiter uses to determine his diagnoses. The collection of sets C is a set of conflict sets. A *conflict set* is described by $\Delta' \cup K$ where $\Delta' \subseteq \Delta$ and $K \subseteq$ COMPONENTS - Δ [25]. Therefore, a minimal set Δ that is a minimal hitting set H is a set that contains the Δ' portion of all the conflict sets C. To reiterate, a *conflict set* is a $\Delta' \cup K$ set where SD $\cup$ $\{\neg AB(c_1),\ldots,\neg AB(c_n)\} \cup$ OBS is inconsistent and $\{c_1,\ldots,c_n\} = \Delta' \cup K$. A minimal hitting set Δ is the combination of the Δ' portions of all conflict sets.

A diagnosis for our technique is somewhat different than Reiter's, and more relaxed as well, since its primary goal is rapid on-line fault isolation. Our algorithm does not rely on finding hitting sets. In this approach, therefore, we further define an *isolation set* Ω where the minimal set $\Delta \subseteq \Omega \subseteq$ COMPONENTS such that SD $\cup$ OBS $\cup$ $\{AB(c) \mid c \in \Omega\} \cup \{\neg AB(c) \mid c \in \text{COMPONENTS} - \Omega\}$ is consistent. In other words, an isolation set may include normal components within it, but COMPONENTS - Ω is guaranteed to have no abnormal components.

Reiter's approach does not require the use of minimal conflict sets; but, as demonstrated later, their use can increase the efficiency of obtaining hitting sets. Reiter's expression of this principle of diagnosis is defined in his theorem 4.4 which follows.

Reiter's [1987] theorem 4.4. $\Delta \subseteq$ COMPONENTS is a diagnosis for (SD, COMPONENTS, OBS) iff Δ is a minimal hitting set for the collection of conflict sets for (SD, COMPONENTS, OBS).

Reiter develops an algorithm to compute hitting sets in support of this theory. See [25] for details. The result of his algorithm is that each hitting set constitutes a diagnosis. This can be contrasted with the results of the our approach, where the final reduced set of sets (i.e. the group of sets) together are the one diagnosis that the algorithm will produce. Note that these reduced sets are not hitting sets. Whereas our approach would yield (A), (B,BM), Reiter's algorithm would yield (A,B), (A,BM) for a multiple fault. Our isolation set, therefore, is (A,B,BM). The diagnosis which is the union of all the elements of the remaining sets called the isolation set is obtained in the following manner.

In the first step, conflict sets are obtained from the ordered dependency records of all discrepant components. Since these conflict sets include sensor components, these are all minimal conflict sets with respect to the other dependency records (i.e. the sensor component would add a unique component to the conflict set such that another conflict set that was a proper subset of the first could not exist). Under the assumption that more than one sensor cannot fail at the same time, all of the conflict sets may not be minimal. This, however, is not relevant in terms of final result but may be used as an optimizing strategy.

The enumeration of conflict sets is another easily identifiable contrast to Reiter's method. Reiter does not start with a collection of conflict sets but rather builds one using a function which returns the necessary conflict sets. Our approach does not use such a function. It does use a Conflict Test (CT) function that is called as CT(SD, α, OBS) where α is a component that is tested to determine if there exists a set $S \in D - C$ such that $\alpha \in S$ where D is the set of all dependency records in the system, C is the collection of conflict sets that were gathered from the list of all discrepant components, and S is one of the sets in D-C. CT(SD, α, OBS) returns TRUE if $\alpha \in S \in D - C$ is consistent or FALSE otherwise.

The specific steps in the IPC algorithm are:

1) Compute D at initialization.

2) Identify discrepancies that cannot be explained (discrepancy detection).

3) Compute the set of ordered dependency records C from the discrepancies which form a modified conflict set.

4) Compute D-C.

5) Stop when only singles are left or no change in conflict sets.

6) Pick first element a of any conflict set that is not a single and call CT(SD, α, OBS).

   If TRUE, remove element from all conflict sets.

   ELSE take intersection of all sets containing element

7) Stop if there are no other sets remaining. Otherwise, Go to (5).

Upon completion of the above algorithm, the union of the resulting set of sets make up the isolation set which is the diagnosis. This diagnosis was achieved by a single observation and required no iterative propagation of component values which is needed for constraint suspension.

As mentioned before, however, the resulting isolation set may not be a minimal diagnosis. In this case, a method of testing may be used to discriminate further. This testing may be carried out by closing successive switches and waiting for faults to reappear. Nevertheless, the objective of isolating a fault as rapidly as possible is achieved while maintaining an accuracy as good as constraint suspension.

An example using the IPC algorithm follows.

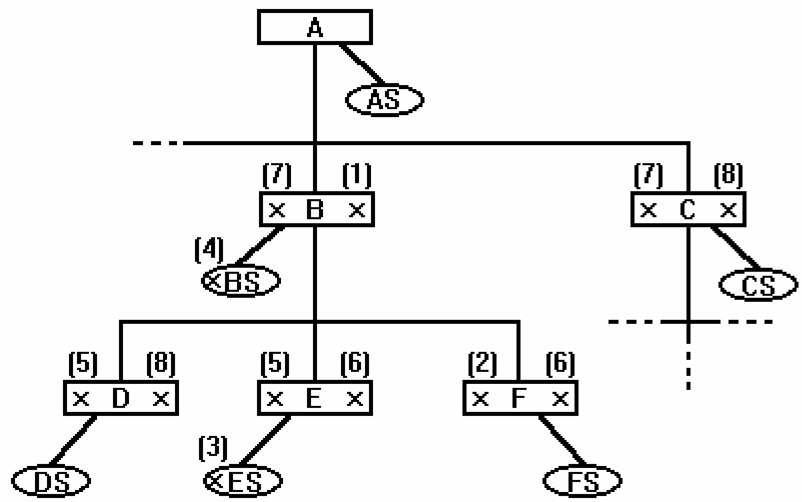

Figure 2 - Example System

Figure 2 is used to demonstrate the IPC conflict-oriented algorithm. The rectangular boxes represent component switches while the circles represent sensors. The component switches we will assume to be remote power controllers (RPCs). RPCs will trip to an open circuit when a zero or very low impedance fault to ground occurs (*called hard fault*s) and do nothing when a high impedance fault to ground occurs (called *soft fault*s). A number of faults can be used in our example. Fault #1 refers to a fault in component B that is equivalent to a 3k switch fault in the SSM-PMAD (to be discussed later). Likewise, fault #8 refers to a multiple fault occurring in a 1k RPC (component D in Figure 2) and a 3k RPC (component C in Figure 2).

If fault #1 were a direct short to ground, our algorithm would first start from step 3 with a collection of conflict sets C = {(A B D) (A B E) (A B F) (A B) (A)} that were generated from the dependency records of the components that are in the path of a discrepancy to the source (D, E, F, B, and A). The set (A C *) is a subset of the group of sets D-C (remember that D is the set of all dependency records in the system that was computed at initialization). Note that the asterisk is used to show that there may be other elements in the set that are not shown in the diagram. Step 5 of the conflict-oriented algorithm does not apply so step 6 is applied to C. The first element of any set happens to be "A" and A is an element of (A C *), therefore, the CT function will return TRUE which means that element A must be removed from all the conflict sets. C is now equal to {(B D) (B E) (B F) (B)}. The next element to choose is "B" which will cause the CT function to return FALSE since "B" is not an element of (A C *). The intersection of all remaining sets yields {(B)}. Since this is a single, step 6 will break out, and the isolation set Ω is the union of all sets within this set (which is (B)).

Note that if the sensor at component "B" were missing, the conflict set (A B) would not appear in the set C. However, this does not change the result. The IPC algorithm would still diagnose the isolation set Ω = (B).

The following tables show the result of applying the IPC algorithm to the tests indicated with respect to the example system. Table 1 shows the IPC conflict-oriented algorithm results for single point failures, Table 2 for sensor failures, and Table 3 for multiple failures. Set **D** for the example system is {(A) (A *) (A B) (A B D) (A B E) (A B F) (A C) (A C *)} which holds for each table. The set (A *) indicates the portion of the system below component A that is not shown in Figure 2. Likewise, the set (A C *) indicates the portion of the system below component C that is also not shown in the diagram. The number of the fault type in the left most column of the tables are associated with the fault location illustrated in Figure 2 of the example system. The third column of each table shows a subset of **D-C** to the extent of what is needed to carry out the IPC algorithm.

Fault Type	**C**	$\subseteq$ **D-C**	Ω
1) Soft Fault at 3K RPC	(A B) (A)	(A C *) (A B D) (A B E) (A B F)	(B)
1) Hard fault at 3k RPC	(A B D) (A B E) (A B F) (A B) (A)	(A C *)	(B)
2) Soft fault	(A B F) (A B) (A)	(A C *) (A B D)	(F)

at 1k RPC		(A B E)	
2) Hard fault	(A B F) (A B) (A)	(A C *) (A B D)	(F)
at 1k RPC		(A B E)	

Table 1. Single Point Failures

Fault Type	**C**	**⊆ D-C**	**Ω**
3) Sensor fault	(A B E ES)	(A B D) (A B F)	(E ES)
in 1k RPC		(A C *)	
4) Sensor fault	(A B BS)	(A B D) (A B F)	(BS)
in 3k RPC		(A C *)	

Table 2. Sensor Failures

Fault #4 (in Table 2) shows a sensor failure at a 3k RPC. The set of conflict sets C for this fault includes the sensor BS since it is the only discrepant sensor. C is equal to {(A B BS)}. The first two elements are elements of a set in D-C shown in Table 2. Therefore, the CT function will return TRUE and the elements are removed from C. The algorithm ends in step #5 since the only set left has only a single component. Taking the union of the result leaves Ω = (BS).

Fault Type	**C**	**⊆ D-C**	**Ω**
5) Soft faults	(A B D) (A B E)	(A B F) (A C)	(D E)
at 1k RPCs	(A B) (A)		

6)	Hard faults at 1k RPCs	(A B E) (A B F) (A C F)	(A B D) (A C *)	(E F)
7)	Soft faults at 3k RPCs	(A B) (A C) (A)	(A *) (A B D) (A C *)	(B C)
7)	Hard faults at 3k RPCs	(A B) (A C) (A) (A B D) ... (A C *)	(A *)	(B C)
8)	Soft faults at 3K & 1K RPCs	(A B D) (A C) (A) (A B)	(A *) (A B E) (A B F) (A C *)	(D C)
8)	Hard faults at 3K & 1K RPCs	(A B D) (A B) (A) (A C) (A C *)	(A *) (A B E) (A B F)	(D C)

Table 3. Multiple Failures

Fault #8 (in Table 3) shows multiple faults at components "C" and "D." Assuming these faults are hard faults, C is equal to {(A B D) (A B) (A) (A C) (A C *)} and {(A *) (A B E) (A B F)} $\subseteq$ D-C. The first element of any conflict set is "A". The CT function returns TRUE since "A" is a member of a set in D-C. Removing "A" from the conflict sets yield C = {(B D) (B) (C) (C *)}. Choosing "B" as the next element reduces C to {(D) (C) (C *)}. The element "D" cannot be chosen next since the set (D) contains a single component. Therefore, "C" is chosen next. The CT function returns FALSE which indicates that an intersection of all the sets with element "C" must occur. The resulting set is {(D) (C)}. Taking the union of the result leaves Ω = (D C).

A requirement for the constraint suspension method is to determine if the components assumed to be faulty will explain the discrepancy between the observed and predicted behavior within the fault isolation phase of diagnosis. Our conflict-oriented method defers this affirmation until after the fault isolation phase by determining the best guess for the location of the fault as facilitated by using the hierarchical structure of the device-centered model. This eliminates the need for fault models and for iterative propagation (re-prediction) of values during hypothesis checking. The result is the fastest possible fault isolation for a domain where this is essential.

4.0 ROBUST MODELLING

4.1 Global Scope in FDIR

Dynamic changes in the status of loads (i.e., closed or open switches) cannot be properly represented without the inclusion of global behaviors which are difficult to model in device-centered models. These occur in dynamic systems where altering devices in one area of a model indirectly effects another area of the model not directly connected to the devices altered. Orthodox modelling of this behavior in unidirectional paradigms results in a circular network of objects. An upstream (reverse) traversal of the network that would be performed upon realizing a fault would encounter infinite looping. An unorthodox method of providing that feedback information is needed that would not interfere with diagnostic methods.

Davis [1984] alludes to global behaviors in the explanation of *aggregate properties* which he describes as phenomena that occurs in the system that cannot be attributed to any one component. Davis explains that "the more interesting faults are those that are local to a single component but that disable the desired aggregate property. Tracing the fault from the disappearance of the aggregate property back to a single component can be difficult."

HOIST [32] alludes to the global behavior problem when writing about unobservable inner states of machines. In these cases, HOIST "must speculate about inner states based on observable inputs and outputs. In general, the further back in the machine the diagnostician must speculate (because of long chains of observable inner states), the less sure he is of the cause." Compensating for this type of phenomenon creates a huge search space that might be of little help. The authors suggest resorting to some sort of meta rule that utilizes shallow reasoning. This compromises efficiency and robustness in order to obtain piece-meal handling of global behaviors.

HOIST uses rules to map inputs to outputs. Other techniques to represent causal relationships include Petri-nets and "unrestricted code" [7]. However the methodology, two of the main problems in accounting for near real-time global behavior in MBR remain: 1) avoiding computationally expensive procedures such as those that involve the enumeration of fault modes and component candidates and 2) avoiding the restrictions that occur with assigning directionality.

The determination of a minimal diagnosis is not often needed for fault isolation and, therefore, the focus of diagnostic reasoning should be on the sections of the system in which the fault is more likely to be located [8]. This allows some room for strategies that speed up an implementation.

4.2 Robust Models

Modelling systems exist that can quite accurately simulate most real world problems. An integration of such robust models with a model-based diagnostic tool can be used to significantly enhance the local behavior description in knowledge base device models.

Fishwick & Zeigler [1991] discuss qualitative physics in terms of reasoning about physical systems. This reasoning can be in diagnostics, simulation, training, etc. The authors main concern is that

AI researchers are modelling systems themselves without first consulting the research in the field of systems simulation.

This investigation starts from the premise that, indeed, there is a significant body of work already done in accurately modelling systems and it would be beneficial to use this to solve the problem of global behaviors in device-centered, model-based diagnostics. The approach described here to solving the global behavior problem, therefore, is based on using the quantitative robust models built in the systems area and integrating them in an AI-based inferencing mechanism.

Although such models are robust, they might not be in suitable forms for use in MBR systems. The task, therefore, is to determine a proper integration of these models so that they can be used in monitoring and diagnosis, within the over-all framework of device-centered MBR.

The robust model provides the correct values of the system by naturally keeping track of global values (i.e., voltages and currents) throughout the system as is normally done in the true system. This robust model provides simulated sensor "readings" to the diagnostic program dynamically (i.e., even with sudden changes to the structure of the model by way of the addition or reduction of loads). The diagnostic program accepts the sensor values calculated by the robust model and uses them to detect discrepancies with the actual system. Upon the detection of a discrepancy, the diagnostic program uses the conflict-oriented methodology to identify the fault. The upstream command that controls the faulted object is turned off (isolation) and recovery is performed if necessary.

Recovery takes place after a fault only when a maintained load is affected. In this case, the switches that affect unfailed upstream components are simply turned on. The new switch configuration is then fed to the robust system so that new sensor measurement values can be calculated.

The model-based diagnostic system is able to detect insidious faults that are below typical instrument sensitivity thresholds because detection of abnormal conditions is based on a comparison of values, rather than on absolute values, as has traditionally been done. This allows for very high sensitivity to low-current faults that would normally not be detected.

Of course, the sensitivity of the system is only as good as the complexity of the model. If the model cannot represent, or at least incorporate the possibility of, some transient disturbance, then there is the danger that it may confuse a normal transient with faulty behavior. An example of this would be a transient voltage wave due to the normal closing of a circuit breaker. The robust model must take this into account by allowing the possibility of such an event, and provide the ability to register such closings as commands (planned and desired action) so that the voltage transient is expected.

This approach solves the issue of global behaviors since the causal relationships that determine values throughout the system are represented by the robust model. Coupled with good suspect selection and fault isolation strategies provided by the conflict-oriented reasoner, the result is an integrated paradigm for rapid FDIR in complex systems. The use of robust models is best explained through a description of the prototype IPC and the testbed. This is done in section 5 below.

5.0 PROTOTYPE DEVELOPMENT AND EVOLUTION

5.1 Testbed System

The system to be monitored is a subsystem of the distribution network in the Space Station Freedom. This was represented by the *Space Station Module Power Management And Distribution* (*SSM-*

PMAD, or *PMAD* for short). It consists of a variety of buses and switches connected to two power sources that supply power to various loads throughout the space station. The hardware testbed is located at the NASA Marshall Space Flight Center (MSFC) in Huntsville, Alabama.

The PMAD is composed of two power sources, the *port* and *starboard* supplies, and three load centers, LC-1, LC-2 and LC-3. Each load center can be supplied from either power source. Figure 3 depicts the overall schematic of the PMAD.

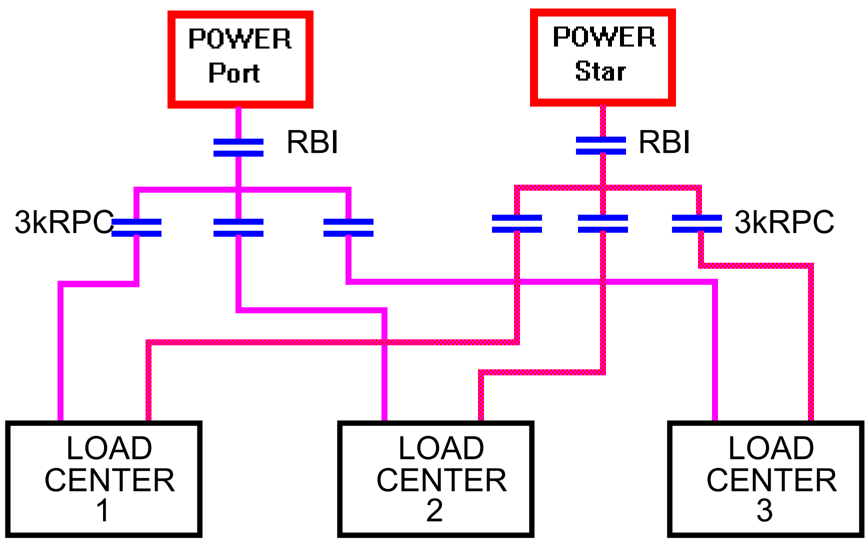

Figure 3 - SSM-PMAD Main Schematic

An example of a load center is shown in Figure 4 below.

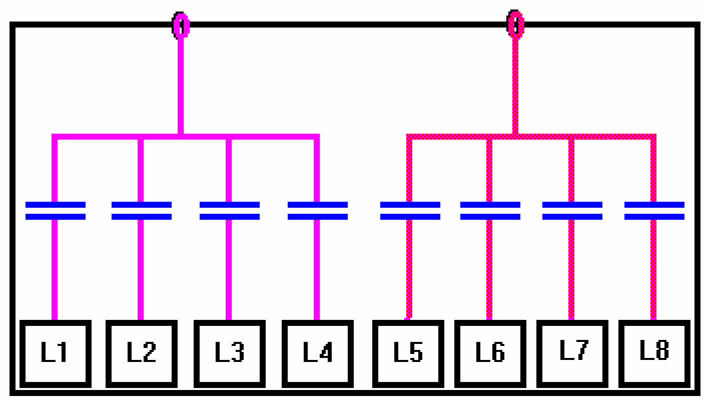

Figure 4 - Load Center Example

The system is comprised of several switches called *remote power controllers* (RPCs) that are rated at either 3 kilowatts (3kW) or 1 kilowatt (1kW). The system also contains two switches called *remote bus isolators* (RBI's) which are closed at initialization and not controlled thereafter. Each load center is powered by either of the two power sources. Load centers 1 & 2 service several independently-supplied loads while load center 3 has two redundantly-supplied and six independently-supplied loads. The redundantly-supplied loads are the only ones that have a capability to recover from faults. For a more detailed description of the PMAD, refer to Lollar [1988] and Gonzalez [1996].

5.2 Robust Model Development

There are typically three subsystems in an electrical power system. These are the source circuit, the power converter circuit, and the load circuit. The system is controlled by the power converter circuit which is used to convey the required power from the source circuit to the load. Figure 5 shows the circuit representation used to develop the robust model for both the Port and Starboard side of the PMAD. The source voltage is represented as a current source and the loads are represented as resistances. The values for resistances used were obtained from NASA-MSFC. A 0.01 ohm resistance was used as the switch resistance.

There are several different robust techniques for modelling circuits. These involve using either *Kirchhoff's voltage law (KVL)* or *Kirchhoff's current Law (KCL)*. KVL says that the sum of the voltage drops around a loop in a circuit is equal to zero. KCL says that the sum of the currents entering a node is equal to zero. A loop is a closed path in a circuit while a node is a connection point between the branches of a circuit that make up a loop. A branch is a section of a circuit that is made up components and carries the same current. In Figure 5, node 1 is the ground node which is used as a reference for other nodes. Therefore, this node does not appear in any calculations as it is considered to be zero. Nodal analysis is used in applying KCL to solve the circuit. As it is preferable in nodal analysis to convert all voltage sources to current sources, this explains the current source in Figure 5. The first step in nodal analysis is to apply KCL to each node except the ground node. The current equations are then substituted into the nodal equations which are then transformed into a matrix that can be solved by a routine matrix solver. The end product of nodal analysis on the Direct Current (DC) circuit in Figure 5 is a matrix that can be used to accurately predict the current and voltage values throughout the circuit. For Alternating Current (AC) circuits, the end product is also the same except that complex values would be involved due to capacitive and inductive elements.

Nodal analysis of this circuit is follows.

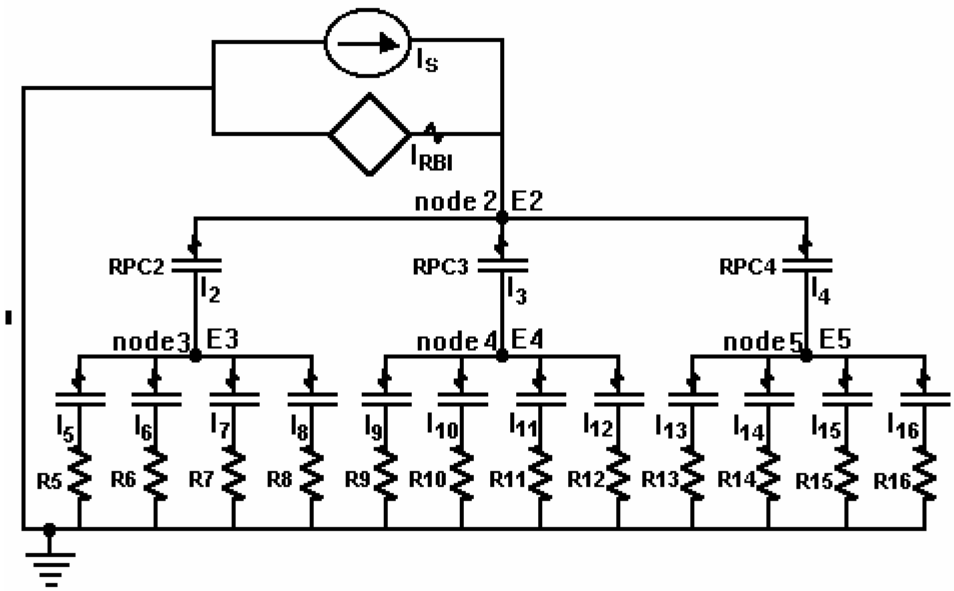

Figure 5 - Robust Model Circuit

Noting that the sum of currents entering a node is equal to zero, the following is obtained.

Node 2:	$0 = -I_s + I_{RBI} + I_2 + I_3 + I_4$
Node 3:	$0 = -I_2 + I_5 + I_6 + I_7 + I_8$
Node 4:	$0 = -I_3 + I_9 + I_{10} + I_{11} + I_{12}$
Node 5:	$0 = -I_4 + I_{13} + I_{14} + I_{15} + I_{16}$

Table 4: Nodal Equations for the Robust Model

Table 5 shows the current equations using Ohm's Law to calculate the current values where E2, E3, E4, and E5 are the voltages at Nodes 2, 3, 4, and 5 respectively.

$I_S = 120/RBI$	$I_9 = E_4/R_9$
$I_{RBI} = E_2/RBI$	$I_{10} = E_4/R_{10}$
$I_2 = E_2/RPC_2 - E_3/RPC_2$	$I_{11} = E_4/R_{11}$
$I_3 = E_2/RPC_3 - E_4/RPC_3$	$I_{12} = E_4/R_{12}$
$I_4 = E_2/RPC_4 - E_5/RPC_4$	$I_{13} = E_5/R_{13}$
$I_5 = E_3/R_5$	$I_{14} = E_5/R_{14}$
$I_6 = E_3/R_6$	$I_{15} = E_5/R_{15}$
$I_7 = E_3/R_7$	$I_{16} = E_5/R_{16}$
$I_8 = E_3/R_8$	

Table 5: Current Equations for the Robust Model

Substitution into the nodal equations yield.

$$\text{Node 2:} \quad 120/RBI = E_2(1/RBI + 1/RPC_2 + 1/RPC_3 + 1/RPC_4) - E_3/RPC_2 - E_4/RPC_3 - E_5/RPC_4$$

$$\text{Node 3:} \quad 0 = -E_2/RPC_2 + E_3(1/RPC_2 + 1/R_5 + 1/R_6 + 1/R_7 + 1/R_8)$$

$$\text{Node 4:} \quad 0 = -E_2/RPC_3 + E_4(1/RPC_3 + 1/R_9 + 1/R_{10} + 1/R_{11} + 1/R_{12})$$

$$\text{Node 5:} \quad 0 = -E_2/RPC_4 + E_5(1/RPC_4 + 1/R_{13} + 1/R_{14} + 1/R_{15} + 1/R_{16})$$

Table 6: Substituted Nodal Equations for the Robust Model

Let:

$R_A = 1/RBI + 1/RPC_2 + 1/RPC_3 + 1/RPC_4$

$R_B = 1/RPC_2 + 1/R_5 + 1/R_6 + 1/R_7 + 1/R_8$

$R_C = 1/RPC_3 + 1/R_9 + 1/R_{10} + 1/R_{11} + 1/R_{12}$

$R_D = 1/RPC_4 + 1/R_{13} + 1/R_{14} + 1/R_{15} + 1/R_{16}$

Figure 6 shows the general matrix equation obtained.

$$\begin{bmatrix} R_A & -1/_{RPC2} & -1/_{RPC3} & -1/_{RPC4} \\ -1/_{RPC2} & R_B & 0 & 0 \\ -1/_{RPC3} & 0 & R_C & 0 \\ -1/_{RPC4} & 0 & 0 & R_D \end{bmatrix} \begin{bmatrix} E2 \\ E3 \\ E4 \\ E5 \end{bmatrix} = \begin{bmatrix} 120/_{RBI} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figure 6 - General Matrix Equation

The load resistances for the SSM-PMAD are as follows. The values for these resistances were obtained from NASA-MSFC.

PORT		STARBOARD	
R5 = 20	R11 = 30	R5 = 16.2	R11 = 30
R6 = 60	R12 = 15	R6 = 15.6	R12 = 60
R7 = 15	R13 = 38	R7 = 16.5	R13 = 31
R8 = 30	R14 = 16	R8 = 13.8	R14 = 30
R9 = 30	R15 = 20	R9 = 30	R15 = 20
R10 = 20	R16 = 40	R10 = 30	R16 = 40

Table 7: SSM-PMAD Load Resistance Values

Substituting these values into the general matrix yields the matrices used as the robust model for the SSM-PMAD. These are shown in Figures 7 and 8.

$$\begin{bmatrix} 400 & -100 & -100 & -100 \\ -100 & 100.167 & 0 & 0 \\ -100 & 0 & 100.183 & 0 \\ -100 & 0 & 0 & 100.1638 \end{bmatrix} \begin{bmatrix} E2 \\ E3 \\ E4 \\ E5 \end{bmatrix} = \begin{bmatrix} 12500 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figure 7 - Complete Port Matrix

$$\begin{bmatrix} 400 & -100 & -100 & -100 \\ -100 & 100.26 & 0 & 0 \\ -100 & 0 & 100.117 & 0 \\ -100 & 0 & 0 & 100.1406 \end{bmatrix} \begin{bmatrix} E2 \\ E3 \\ E4 \\ E5 \end{bmatrix} = \begin{bmatrix} 12500 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figure 8 - Complete Starboard Matrix

These matrices are used to model the functionality of the SSM-PMAD. They are integrated along with a matrix solver into the robust model portion of the IPC.

5.3 Enhanced Complexity Models

One problem arose during the testing and evaluation of the IPC prototype. In cases where low (or zero) impedance faults to ground occur, the assumption of constant voltage made in the above model no longer applies. The switches on the PMAD are set to automatically interrupt whenever a large current is sensed. This local approach is used to protect the PMAD loads in case of controller failure. Therefore, in instances of low impedance faults, the switches are locally tripped before the current value is read and sent to the IPC for analysis. The consequence is that for these type of faults, the IPC simply sees an open-circuit (the tripped-switch) rather than a short circuit to ground.

In such a context, the assumption of constant voltage is adequate. In cases of high impedance faults, where local control of the switch is not exerted due to low current levels, the voltage does remain nearly constant. In low-impedance faults, however, the really sees an open-circuit where the voltage levels remain constant upstream of the tripped switch.

In order to address this deficiency, not only of the model but also of the testbed, the following steps were taken.

The situation was reproduced using a power system simulator called Network Fault Analysis Program (NETFAULT) from the Philadelphia Electric Company (PECO) in lieu of the PMAD. This allowed the IPC to "see" uninterrupted low-impedance faults that cause the voltage levels to collapse. Under such conditions and, using a constant voltage model, the IPC was fooled into thinking that there were several false discrepancies, and erroneously diagnosed numerous components as being faulty.

These problem of false discrepancies was addressed by enhancing the model used by the IPC. The resulting enhanced IPC, or EIPC [24], uses a model that does not assume constant voltage, but rather, constant impedance. This assumption is a robust one as only the faulted part of the circuit changes impedance as a result of a fault. The results of using the EIPC model show that it permits the IPC to correctly diagnose low-impedance uninterrupted faults, while at the same time not affecting the other proven capabilities of the IPC.

The EIPC uses the basic robust model employed in the IPC for steady-state simulation of the power system. It differs in that it uses a preprocessor to calculate all the impedances in the circuit as seen from the interrupting devices by the voltage and current measurements therein. Rather than comparing the currents and assuming constant voltage as in the IPC, it compares the measured impedances to the simulated ones. See Pettersson [1995] for a description of the impedance model used and detailed test results. Pettersson's enhanced approach did not appreciably affect the speed of execution of the IPC.

6.0 SUMMARY AND CONCLUSION

The techniques which form the basis for the research reported here - that of applying a conflict-oriented diagnostic approach and a robust quantitative modeling technique - were incorporated in a prototype system called the *Intelligent Power Controller (IPC)* [20]. The IPC had earlier served as the prototype for testing other reasoning techniques such as constraint suspension as well as modeling the power system using another technique developed by the authors called *meta-objects* [20] which addressed the global behavior problem.

The combination of a conflict-oriented diagnostic reasoning technique and the robust models were implemented in the IPC and it was subjected to a number of tests on the PMAD as well as on a power system fault analysis program (NETFAULT). The results of these tests illustrated that rapid isolation was attainable (< 0.5 sec.) [14]. Some of the measurements obtained are indicated below.

6.1 Algorithm Scalability

Reasoning from first principles is a widely accepted format for diagnosis. One of its criticisms is that it is difficult to accurately reflect this methodology in real time. For this reason, first order representations have been the traditional means of modeling using first principles. These methods, when incorporated with device-centered techniques that use unidirectional assumptions, have yielded an efficient approach to fault isolation. First order methods allow rapid and flexible quantitative manipulations; but, these methods are often not accurate in their predictions and are impossible to use for global phenomena in systems. The introduction of meta objects into the first order method compensates somewhat for the difficulty in modeling this global behavior. However, these models become increasingly complex with minor added complexity to the actual system. This suggests that for moderately complex systems, the meta-object enhanced first-order system would become unmanageable which translates into slower, less reliable models. Because meta-objects are a simplistic solution to modeling global behavior, robust models were used to accurately model more complex systems.

A robust model approach was developed along with a needed conflict-oriented diagnostic algorithm. These contributions were integrated into what is now called the Intelligent Power Controller (IPC) system. The resulting work outlines a methodology that accurately reflects the system being diagnosed, that eliminates problems with global behaviors, and that avoids costly inversion and iterative propagation based strategies. The test results obtained indicate a successful strategy for real-time on-line fault isolation of a power distribution system. See Gonzalez [1996] for a detailed description of test results.

While the SSM-PMAD is comparable to real-world systems, in terms of size and structure, systems may exist that contain many more components. The cost associated with the diagnosis of a system of any size is dependent upon the CT function returning a true or false, the cost in taking an intersection of the conflict sets, and the cost in removing an element from each conflict set that contains that element. Each of these costs are associated with step #6 of the IPC conflict-oriented diagnostic algorithm shown earlier. This step is accessed a maximum number of times equal to the height of the hierarchical tree structure (number of elements in a conflict set) multiplied by the number of discrepant sensors (number of conflict sets). Thus, the worst case complexity of the IPC algorithm may be expressed by the following equation.

Worst Case # of Loops = maximum # of conflict sets * max # of elements in a conflict set

An increase in the number of components in the system would increase the height of the hierarchical structure. At worst, this hierarchical structure would be in the form of a binary tree, making the height of the tree $\log_2(n)$. The same increase would, in the worst case, increase the number of conflict sets linearly provided all components have a sensor. However, if this were ever to occur, only one pass through the loop would be required since all the sensors would be discrepant and, therefore, D - C would be an empty set. The culprit would certainly be the root component.

This insight indicates that the worst performance is likely to be a fault occurring midway [$\log_2(n)/2$] in the hierarchical structure. In the worst case scenario, approximately 2 to the power of $\log_2(n)/2$ conflict sets would be generated. Note that since the height H of a binary tree is $\log_2(n)$, the number of elements under a particular node at H is $n = 2^H$ and the number of elements under a particular node at half that height is $2^{H/2}$. From this, we can expect the complexity (based on the number of loops) to be

Worst Case # of Loops = maximum # of conflict sets * max # of elements in a conflict set

$$= 2^{\left(\frac{\log_2 n}{2}\right)} \bullet \frac{\log_2 n}{2}$$

$$\text{since } 2^{(\log_2 n)} = n$$

we have

$$= \left[2^{(\log_2 n)}\right]^{\frac{1}{2}} \bullet \frac{\log_2 n}{2}$$

$$= \frac{1}{2}\sqrt{n} \cdot \log_2 n$$

Therefore, the complexity of the IPC algorithm is $O(\sqrt{n} \cdot \log_2 n)$.

In conclusion, the scalability of the IPC system is excellent. The system performs at $O(\sqrt{n} \cdot \log_2 n)$ worst case for a binary tree and remains the same for other hierarchical structures since there is a trade off between the number of conflict sets generated and the height of the structure.

Currently, the system is limited to power distribution systems and may produce non-minimal diagnoses if the target system is not sufficiently populated with sensors. A future research interest would be to apply this methodology to other systems that exhibit global behavior. It is hoped that this system may be used in diagnosing any system that has been robustly modelled and tested and that lends itself well to a device-centered representation. Another topic of interest would be to apply further probes and tests for hypothesis discrimination after the isolation of a non-minimal set of components.

7.0 BIBLIOGRAPHY

[1] Adamovits, P., B. Pagurek. 1993. *Simulation (model) based fault detection and diagnosis of a spacecraft electrical power system.* Proceedings of the Ninth Conference on Artificial Intelligence for Applications, Orlando, Florida. March 1-5. IEEE Computer Society Press. Los Alamitos, CA. p 422-428.

[2] Adams, T. L. 1986. *Model-based Reasoning for Automated Fault Diagnosis and Recovery Planning in Space Power Systems*. Proceedings of the Intersociety Energy Conversion Engineering Conference, 1986. pp. 1763-1769.

[3] Ashworth, B. 1989. *An architecture for automated fault diagnosis.* Proceedings of the 24th IECEC. August 1989. pp 10-15.

[4] Ashworth, B., B. Walls. 1990. *Autonomous operation of a Space Station Freedom type power testbed.* Proceedings from the NASA/JSC Fault Diagnosis Workshop in June 1990. p 24-31.

[5] Bau, D., P. Brezillon. 1992. *Model-based diagnosis of power-station control systems.* IEEE Expert. v 7, n 1. Feb. 1992. pp. 36-44.

[6] Blasdel, A. N. 1987. *Automated Fault Handling of a Satellite Electrical Power Subsystem Using a Model-based Expert System*. Proceedings of the Intersociety Energy Conversion Engineering Conference, 1987. pp. 601-606.

[7] Davis, Randall. 1984. *Diagnostic reasoning based on structure and behavior.* Artificial Intelligence. 24(1): p 347-410.

[8] de Kleer, J., A. Mackworth, R. Reiter. 1992. *Characterizing diagnosis and systems.* Artificial Intelligence v56. Elsevier Science Publishers. p 197-222.

[9] Dvorak, D., B. Kuipers. 1991. *Process monitoring and diagnosis: A model-based approach.* IEEE Expert. v 6, n 3. June 1991. pp. 67-74.

[10] Fesq, L.M., A. Stephan, and L. McNamee, 1992. *Modeling Power Systems for Diagnosis: How Good is Good Enough?*.

[11] Fishwick, P. A., B. P. Zeigler. 1991. *Qualitative physics: towards the automation of systems problem solving.* J. Expt. Theor. Artif. Intell. v3 1991 p 219-246.

[12] Gholdston, E. W., D. F. Janik, and G. Lane. 1988. *A Diagnostic Expert System for Space-based Electrical Power Networks*. Proceedings of the Intersociety Energy Conversion Engineering Conference, 1988.

[13] Gonzalez, A. J., R. L. Osborne, C. Kemper, and S. Lowenfeld. 1986. *On-line diagnosis of turbine generators using artificial intelligence.* IEEE Transactions on Energy Conversions. vol. EC-1, no. 2. June 1986. pp. 68-74.

[14] Gonzalez A., R. A. Morris, F. McKenzie, D. Carreira, B. Gann. 1996. *Model-based real-time control of electrical power systems.* IEEE Transactions on Systems, Man, and Cybernetics. v 26, n 4. July 1996.

[15] Hamscher, W., R. Davis. 1987. *Issues in model-based troubleshooting.* A.I. Memo 893, Artificial Intelligence Lab, Massachusetts Institute of Technology.

[16] Hester, T. 1986. *FIES-II: A Real Time Fault Isolation Expert System*. Proceedings of the Intersociety Energy Conversion Engineering Conference, 1986.

[17] Lee, S. C. 1993. *A Fault Detectability Analysis Based on Sensor Relationships for Space Power Systems*. Proceedings of the Intersociety Energy Conversion Engineering Conference, 1993.

[18] Lloyd, B., W. Park, J. White, and M. Divakaruni. 1989. *A Generator Expert Monitoring System*. Proceedings of the Conference on Expert System Applications for the Electric Power Industry, Orlando, FL. June 1989.

[19] Lollar, L.; Weeks, D. J. 1988. *The autonomously managed power systems laboratory.* Proceedings of the 23rd IECEC, vol. 3. 1988.

[20] McKenzie, F. D. 1994. Using robust models to represent global behaviors in model based reasoning. Doctoral dissertation, College of Engineering, University of Central Florida. 1994.

[21] Ng, Hwee Tou. 1991. *Model-based, multiple-fault, diagnosis of dynamic continuous physical devices.* IEEE Expert. v 6, n 6. Dec. 1991. pp. 38-43.

[22] Paasch, R., A. Agogino. 1993. *A structural and behavioral reasoning system for diagnosing large-scale systems.* IEEE Expert. v 8, n 4. August 1993. pp. 31-36.

[23] Padalkar, S., G. Karsai, C. Biegl, J. Sztipanovits, K. Okuda, N. Miyasaka. 1991. *Real-time fault diagnostics.* IEEE Expert. v 6, n 3. June 1991. pp. 75-85.

[24] Pettersson, Goran L. 1995. Impedance-driven model-based diagnosis of electric power distribution system faults. Masters Thesis, Department of Computer Engineering, University of Central Florida.

[25] Reiter, R. 1987. *A theory of diagnosis from first principles.* Artificial Intelligence, v32. Elsevier Science Publishers. p 57-95.

[26] Risdesel, J. 1989. *A survey of fault diagnosis technology.* IECEC August 1989. p 33-47.

[27] Risdesel, J., C. Myers, B. Ashworth. 1989. *Intelligent space power automation.* IEEE International Symposium on Intelligent Control. September 1989, p 49-59.

[28] Scarl, Ethan A., J. Jamieson, & E. New. 1988. *Model-based reasoning for diagnosis and control.* Proceedings of the 1st Florida Artificial Intelligence Research Symposium. p 192-196.

[29] Spier, R. J. and M. E. Liffring. 1989. *Real-Time Expert Systems for Advanced Power Control (A status update)*. Proceedings of the Intersociety Energy Conversion Engineering Conference, 1989.

[30] Sueda, N., M. Iwamasa. 1995. *A pilot system for plant control using model-based reasoning.* IEEE Expert. v 10, n 4. August 1995. pp. 24-31.

[31] Watson, K., B. D. Russell, and I. Hackler. 1988. *Expert System Structures for Fault Detection in Spaceborne Power Systems*. Proceedings of the Intersociety Energy Conversion Engineering Conference, 1988.

[32] Whitehead, J. D., J. W. Roach. 1987. Expert systems without an expert: fault diagnosis based on causal reasoning. Texas Instruments Technical Journal. Winter 1987. p 19-29.

[33] Xiang, Zhigang; Srihari, Sargur N.; 1986. *Diagnosis using multi-level reasoning.* Expert Systems in Government Symposium. McLean, VA, USA 1986 Oct 22-24; Publ by IEEE, p 151-158.