A Systematic Analysis of Measurement Selection Algorithms for Fault Isolation in Dynamic Systems

Sriram Narasimhan
Department of Computer Science
Box 1679, Station B
Vanderbilt University
Nashville, TN
nsriram@vuse.vanderbilt.edu

Pieter J. Mosterman *
Institute of Robotics and
System Dynamics
DLR Oberpfaffenhofen
D-82230 Wessling
Pieter.J.Mosterman@dlr.de

Gautam Biswas †
Department of Computer Science
Box 1679, Station B
Vanderbilt University
Nashville, TN 37235
biswas@vuse.vanderbilt.edu

Abstract

Fault detection and isolation in dynamic physical systems based on analysis of transient system behavior involves monitoring for deviations and linking observed deviations to changes in system parameters. Components linked to the changed parameter values may be faulty. Functional redundancy is exploited in choosing measurements to uniquely isolate all faults. Measurement selection algorithms determine the minimum set of system variables that can uniquely identify all single faults in the system. This paper analyzes a set of measurement selection algorithms in the context of transient analysis in dynamic systems. Results show that an A^*-search algorithm provides good results while considerably reducing the computational complexity of the measurement selection process.

Introduction

Fault detection and isolation in dynamic systems requires a system model to predict the system behavior under different conditions and a temporal sequence of measurements to derive faults from transient behaviors. Typically when the systems involved are complex, fault isolation is performed using a subset of the possible measurements that can be made on the system. When human operators are involved in the control and supervising loop this is essential to prevent cognitive overload. For more automated systems, such as the ones available on space vehicles, aircraft, and automobiles, the number of measurements processed is limited by sensor and processor cost, and limited communication bandwidth. Therefore, it is critical that system models be carefully analyzed to identify measurement points that provide the most discriminatory information to quickly isolate the true faults. This task of selecting subsets of measurements for successful diagnosis is the *measurement selection* problem.

*Pieter J. Mosterman is supported by a grant from the DFG Schwerpunktprogramm KONDISK.

†Gautam Biswas is supported by grants from PNC, Japan and Hewlett-Packard, Co.

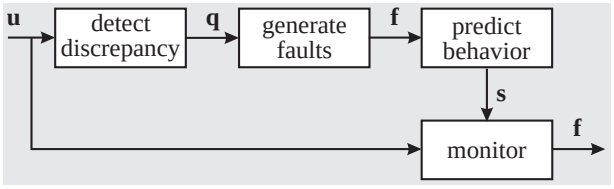

Figure 1: **The diagnosis process.**

In real situations, desired measurement points may be inaccessible or the state of the art of sensor technology is such that certain measurements are expensive and inaccurate, so they are not used. Moreover, regulatory laws may require certain measurements be made. Therefore, the set of measurements available for analysis may differ from the desired set, which makes it important to analyze which sets of faults can actually be discriminated. This is referred to as the *diagnosability* problem.

Early approaches to measurement selection was based on decision-theoretic methods in a sequential diagnosis framework (deKleer & Williams 1987). The measurement with the best goodness measure, defined in information theory terms, was selected at each step. These ideas were further developed by Yu and Biswas (1992) for steady state diagnosis of continuous systems. In other work, Chien, Doyle, and de Mello (1991) developed measurement selection criteria based on sensitivity analysis, cascading alarm analysis, potential damage analysis, and teleological analysis. This paper develops measurement selection algorithms in the framework of a qualitative first-order methodology for fault detection and isolation in dynamic systems.

Diagnosis from Transients

The overall diagnosis methodology (Mosterman & Biswas 1997; 1998), is based on a qualitative reasoning framework, shown in Fig. 1. The monitoring system periodically polls all the sensors to detect deviations

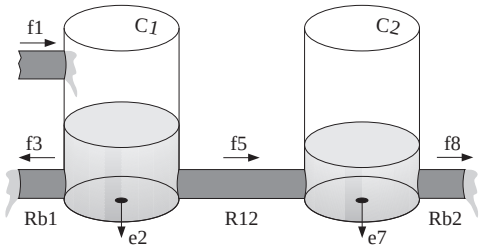

Figure 2: **The bi-tank system**

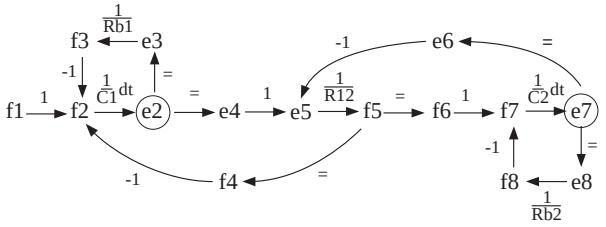

Figure 3: **Temporal causal graph of the bi-tank system.**

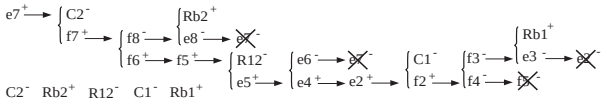

Figure 4: **Backward propagation to find faults.**

from normal value in the observed variables. Initial fault sets derived from the observed discrepancies are used to predict future system behavior. Progressive monitoring is then applied to compare actual observations with the predictions to refine the hypothesized fault set. Our work on diagnosis and measurement selection is based on the assumptions that the only kind of faults in the system are component failures, and any deviation in system behavior is caused only by these component failures. In the current framework, we deal with single fault hypotheses.

The system model for diagnosis analysis is a temporal causal graph, derived from a bond graph model. The temporal causal graph is a directed graph, where the vertices are the system variables (efforts and flows) and the edges represent the causal relation between these variables qualified by the temporal characteristics of the relation. Systematic methods for deriving the temporal causal graph from a bond graph model are presented in Mosterman & Biswas (1998).

Consider the example of the two-tank system in Fig. 2. The two tanks of capacity C_1 and C_2 have two outlet pipes with resistance R_{b_1} and R_{b_2}, respectively and a connecting pipe with resistance R_{12}. The temporal causal graph of system behavior is shown in Fig. 3. The net flow rate into the tank $f_2 = f_1 - f_3 - f_4$. f_1 and f_2 are directly related, which is indicated by the edge with label 1 on the edge from f_1 to f_2, whereas f_2 and f_3 are inversely related, and this is indicated by the label -1 on the edge from f_2 to f_3. The net flow f_2 into C_1 has an integral relation to the amount of stored liquid in the tank, which appears as a time-derivative effect, $\frac{1}{C_1}dt$, between the pressure e_2 and flow f_2. Edges with labels 1, -1, R and $\frac{1}{R}$ imply instantaneous propagations, whereas edges with labels $\frac{1}{C}dt$ and $\frac{1}{I}dt$ imply a time delay in the propagation. Temporal effects play an important role in developing the fault isolation algorithms. The fault isolation and refinement task is a three step process: (i) initial component parameter implication (ii) prediction and, (iii) monitoring and fault isolation.

Initial Component Parameter Implication

The set of component parameters that appear in the temporal causal graph define the list of faults that can occur in the system. Given a set of observations, expressed in qualitative form $(-, 0, +)$, a backward propagation algorithm using a depth first search is invoked to generate the number of faults that are consistent with the initial set of deviant observations. The back propagation starts at a deviant observation and proceeds backwards in the temporal causal graph. The output of the back propagation is a fault hypothesis set which contains all the likely faults. An example is shown in Fig. 4 for a deviating right tank pressure ${e_7}^+$ in the bi-tank system (Fig. 2). ${e_7}^+$ initiates back propagation along $f_7 \stackrel{\frac{1}{C_2}dt}{\rightarrow} e_7^+$ and implicates C_2 below normal (C_2^-) or f_7 above normal (f_7^+). The next step along $f_6 \stackrel{1}{\rightarrow} f_7^+$ implicates f_6^+, and $f_8 \stackrel{-1}{\rightarrow} f_7^+$ implicates f_8^-, and so on. This results in the five fault hypotheses shown in Fig. 4.

Prediction

One of the critical issues in diagnosis of dynamic systems is the ability to predict the behavior of the system when a fault occurs. Predicted behaviors are compared with the actual observations to establish the hypothesized faults in the system. A forward propagation algorithm propagates the effect of faulty parameters along instantaneous and temporal links in the temporal causal graph to predict qualitative deviations in magnitude and derivatives of the observations under the fault conditions. This is called a *signature*

Definition 1 (Signature) *The prediction of* 0^{th}, 1^{st}, *and higher order time-derivative effects of a system variable expressed qualitatively (*low, normal, high*) in response to a fault is called its signature.*

Figure 5: **Forward propagation yields signatures.**

The order of a signature is the highest time derivative that is included in the signature. For each fault in the hypothesis set, forward propagation starts at the fault and proceeds in the direction of the arrows in the temporal causal graph. Initially, all deviation propagations are 0^{th} order magnitude values. When an integrating edge is traversed, the magnitude change becomes a 1^{st}-order (derivative) change, shown by an $\uparrow$ ($\downarrow$) in the temporal causal graph (Fig. 5). Similarly, a first order change propagating across an integrating edge creates a second order (derivative) change ($\uparrow\uparrow$ ($\downarrow\downarrow$) in Fig. 5), and so on. Forward propagation with increasing derivatives is terminated when a signature of sufficient order is generated (Mosterman & Biswas 1998). This is discussed in greater detail in the next section.

A *complete* signature contains derivatives specified to its sufficient order. For example, a complete second order signature for effort variable e_7 corresponding to fault R_{b2}^{+} is $(0, +, -)$. Monitoring should report a non normal value when a complete signature of an observed variable has a deviating value. The output of the forward propagation algorithm is a set of signatures for every selected observation, one for each fault in the fault hypothesis set.

Monitoring and Fault Isolation

The monitoring system samples the chosen subset of observations at a predetermined rate, till transient analysis is suspended. At each time step, a qualitative form of the observations are compared against the signatures reported by the forward propagation algorithm. Depending on the nature and quality of sensors employed and the desired sensitivity, measurements with percentage deviations greater than 2-5% of the nominal value are reported to be above (or below) normal. For dynamic systems, fault transients change over time. For example, a variable magnitude due to a fault may initially be normal but if its first derivative is predicted to be above normal, over time this variable magnitude will deviate above normal.

A hypothesized fault is dropped if the observations and the reported signature for that candidate do not match. The comparison process is executed over a number of time steps and higher order derivative effects are incorporated into the comparison process in the form of *progressive monitoring*. Progressive monitoring is activated when there is a discrepancy between a predicted value and a monitored value that deviates (this applies to 0^{th} and higher order derivatives). At every time point, it is checked whether the next higher derivative could make the prediction consistent with the observation. If this next higher derivative value is normal the next higher derivative value is considered, until there is either a conflict between prediction and observation, a confirmation, or an unknown value is found. When progressive monitoring is invoked it moves the first nonzero derivative to fill in lower derivative value slots. As an example if we apply progressive monitoring with the signature 00+ and a measurement whose value was initially normal, but has now deviated above normal would be consistent with this signature.

In our work, we assume only magnitude changes and slopes can be measured reliably and compared against predicted signature values. Higher order terms in a signature have an effect only through progressive monitoring. Therefore, it is clear that the unique values of magnitude and slope that contribute to differentiating among faults are the values: (i) 00, (ii) $++$, (iii) $--$, (iv) $+-$, (v) $-+$, (vi) $+0$, and (vii) -0. We do not use normal values for fault discrimination. In a dynamic system, a measurement that is currently normal may deviate at some later time because of higher-order effects. It is hard to determine when a variable value becomes steady, therefore, it is also hard to differentiate (vi) and (vii) from (ii)-(v). This implies that every measurement can subdivide a set of faults into at most 4 categories corresponding to the groups (ii)-(v) listed above. Sometimes forward propagation leads to conflicting values ($+$ and $-$) for the same observation, and this results in an unknown value (?) being recorded in the signature. This reduces the sensitivity of the diagnosis algorithm, therefore, such measurements are not likely to be selected. However, this does not affect the measurement selection algorithms discussed in the next section.

Measurement Selection

Given that we would like to use a minimal number of sensors but retain complete diagnosability it is critical that system models be analyzed to identify a subset of measurement points that provides the most discriminatory information to *completely diagnose* the given system.

Definition 2 (Complete diagnosability) *A fault isolation system is completely diagnosable if all hypothesized faults can be uniquely isolated.*

The problem of finding the minimum subset of observations for complete diagnosability is called *measurement selection.*

Definition 3 (Measurement Selection) *Given a diagnosis algorithm, measurement selection identifies a minimal subset of observations that can uniquely detect and isolate every single fault in the system, i.e, the subset of observations makes the system completely diagnosable.*

Our diagnosis framework illustrates that there is a trade off between generating higher order derivatives in signatures that propagate down by progressive monitoring and considering more measurements in diagnostic analysis (Mosterman & Biswas 1998). Higher order signatures consider more extensive propagation effects for each chosen measurement, and, therefore, reduce the number of measurements required for complete diagnosability. However, the disadvantage in using higher order signatures is that it may take longer for higher order effects to manifest. As a result, these higher order effects may be masked by compensating effects within the system, or because of interactions caused by cascading faults. Moreover, the extensive propagation may cause a small eventual deviation that cannot be reliably detected in a noisy environment.

In this work we focus on a quick analysis of transients to avoid these problems, and, therefore, consider only second order signatures for the measurement selection and fault isolation tasks. The rest of the section discusses the measurement selection algorithms that we study in this work.

Notation and Terminology

Given the system under consideration for diagnostic analysis we define the following: $OSET = \{O_i | i \in (1 \cdots m)\}$ $FSET = \{F_j | j \in (1 \cdots n)\}$, where O_i represents a possible measurement point and F_j a possible fault in the system. In our framework a given component can fail in two different ways, i.e, if f_j is a component parameter, $F_{2j} = f_j^+$ and $F_{2j+1} = f_j^-$. Therefore, n, the number of faults in the system is twice the number of component parameters that can fail. m represents the total number of observations in the system. s[i,j] is defined as the signature of observation O_i for fault F_j, $1 \leq i \leq m, 1 \leq j \leq n$.

We use two different techniques in developing algorithms for our measurement selection algorithms. In the first, we look upon measurement selection as a partitioning problem : Given an initial partition $P_{initial} = p_1 = FSET$, find a subset of observations $(O_1, O_2 \cdots O_i)$ that refine $P_{initial}$ to yield $P_{goal} = \{p_1, p_2 \cdots p_n\}$ such that each group p_k $(k \in \{1 \cdots n\})$ includes exactly one fault F_j. We define an operator $SPLIT(O_i, P_{current})$, which splits all groups $p_1, p_2 \cdots p_p$ in the current partition $P_{current}$ using the observation O_i.

Definition 4 *An observation O_i is said to split a group p_j into two partitions p_{j_1} and p_{j_2} if and only if the following 3 conditions hold :*

(i) $\forall F_p, F_q \in p_{j_1}, s[i,p] = s[i,q]$

(ii) $\forall F_p, F_q \in p_{j_2}, s[i,p] = s[i,q]$

(iii) $\forall F_p \in p_{j_1} \vee \notin p_{j_2} \wedge F_q \in p_{j_2} \vee \notin p_{j_1}, s[i,p] \neq s[i,q]$

Two measurement selection algorithms, *weighted split* and *$A^\star$ search* use the $SPLIT$ operator.

The second technique uses a discriminatory matrix to solve the measurement selection problem.

Definition 5 *For any observation O_i, a discriminatory matrix DM_i is defined as a matrix with 0 and 1 values such that the following conditions hold:*
$\forall F_j, F_k \in FSET,$ *such that* $j \neq k$

$$DM_i[j,k] = \begin{cases} 1 & , if \quad s[i,j] \neq s[i,k] \\ 0 & , if \quad s[i,j] = s[i,k] \end{cases}$$

The 1 entries in the discriminatory matrix indicate the ability of measurement O_i to discriminate between the corresponding pairs of faults.

The discriminating ability of two observations O_p and O_q, can be combined as follows : $\forall F_p, F_q \in FSET, p \neq q \quad DM_{p,q}[j,k] = DM_p[j,k] \vee DM_q[j,k]$ This procedure can be extended to compute the discrimination matrix for any subset of measurements in $OSET$. Using this technique our goal is to come up with a minimum subset of observations $O_1, O_2 \cdots O_i$ such that $DM_{O_1,O_2 \cdots O_i}[j,k]$ is a matrix where all non diagonal elements = 1. As an abbreviation we call this the $ND1$ matrix. The *exhaustive search* and *matrix method* use discriminatory matrices.

Measurement Selection Algorithms

The set cover problem, which is NP-complete (Aho, Hopcroft & Ullman 1983), can be transformed to the measurement selection problem, thereby making it NP-complete too. Hence for any solution that guarantees optimality we cannot guarantee time efficiency, i.e., there is a trade off between optimality of the solution and speed of computation.

Exhaustive search (Mosterman, Biswas & Narasimhan 1997) looks at all possible combinations of observations to determine the minimum observation set but it takes time proportional to $m!$. If speed of computation is of primary importance, *greedy search methods* with goodness measures can be applied to generate quick solutions, with no guarantee for optimality. A more systematic approach is the *$A*$ algorithm* (Nilsson 1980).

Exhaustive Search Method

The exhaustive search presented in Mosterman, Biswas & Narasimhan (1997) uses a breadth first control structure. The execution time of this algorithm is exponential in the number of observations.

Greedy Approaches

A simple approach to greedy search is to select one measurement at a time based on a goodness measure till a solution is derived. The goodness measure reflects the ability of the measurements to distinguish between faults. At each step, the measurement that can distinguish the largest subset of faults is selected. We discuss two goodness measures (i) create equal size groups, and (ii) maximize faults isolated at each step to derive the measurement selection algorithm.

Equal Size Groups The motivation behind this heuristic is that binary (equal) splits will lead to a minimal observation set for complete diagnosability. We look at the measurement selection problem as a partitioning problem. At each step, we select an observation O_i, not already selected, that will further split partition $P_{current}$ using the operator $SPLIT(O_i, P_{current})$ into a new partition P_{next} with as many equal size groups as possible. This involves two aspects, (i) create as many groups as possible and (ii) make the difference in size of the new groups as small as possible.

A simple weighted measure that captures these aspects is : $GM_{O_i} = (w_1 * 1/N) + (w_2 * D)$, where N is the number of groups in $P_{current}$ and D the difference in size of groups in $P_{current}$, i.e, $D = \mid P_{max} \mid - \mid P_{min} \mid$, where $\mid P_{max} \mid$ = size of biggest group in $P_{current}$ and $\mid P_{min} \mid$ = size of smallest group in $P_{current}$. w_1 and w_2 are externally determined weights.

At each step, we select the measurement that minimizes GM_{O_i}. This heuristic is implemented by Algorithm 1. The FOR loop in Algorithm 1 is repeated m times. The WHILE loop is repeated at most m times, since at each iteration of the WHILE loop we select one observation to add to $O_{solution}$ and we exit the WHILE loop if $size(OSET_{solution}) > m$. Hence the time complexity of this algorithm is O(m^2). Since this approach selects observations greedily, we are likely to make big leaps towards the goal early on since observations are likely to split $P_{current}$ well but later on when we have more groups, not all groups may split well.

Maximize Isolated Faults The goal here is to select the observations that best discriminate among faults first. We calculate the discriminatory matrix $DM_i[j,k]$ for all observations O_i and all pairs of faults $(\{F_j, F_k\})$. We keep track of the solution set of observations $OSET_{solution}$ to which we add one observation at a time. As observations are added, we keep track of the discriminatory matrix for the set of chosen observations $DM_{solution}[j,k] = DM_{OSET_{solution}}[j,k]$. At each step we select the observation O_i that produces the best $DM_{solution \bigcup O_i}[j,k]$ matrix. The best matrix is the one that has maximum number of 1 entries, which indicates that maximum number of pairs of faults can be distinguished. The goal is reached when $DM_{solution}[j,k] = ND1$. We define $DM_p[j,k] <, (=, >) DM_q[j,k]$ if $DM_p[j,k]$ has lesser, (equal, greater) number of 1 entries than $DM_q[j,k]$. The implementation of this scheme is presented as Algorithm 2.

Algorithm 1 Measurement selection using equal size partition measure.

```
n ← number of faults
m ← number of measurement variables
OSET_solution ← empty
Initialize P_initial to have one group with all faults
Initialize P_goal to have n groups with one fault each
P_current ← P_initial
while P_current ≠ P_goal and size(OSET_solution) ≤ m do
  for all i = 1 to m do
    if O_i ∉ OSET_solution then
      calculate the value GM_Oi
    end if
  end for
  select O_minimum which has minimum value of GM_Oi
  P_next ← SPLIT(O_minimum, P_current)
  OSET_solution ← OSET_solution ⋃ O_minimum
end while
if P_current = P_goal then
  Output OSET_solution as solution
else
  Output No solution
end if
```

It is clear that the nested FOR loops take $m \times n^2$ time. The WHILE loop executes m times in the worst case, since we select one observation to $OSET_{solution}$ at each iteration of the loop and we exit the loop when $\mid OSET_{solution} \mid > m$. Inside the WHILE loop, the FOR loop executes m times. Inside the FOR loop two matrices of size n^2 are compared. Hence each iteration of the FOR loop takes $m \times n^2$ time. Since the WHILE loop executes m times, the time complexity of this algorithm is O($m^2 \times n^2$) time. Although this algorithm will output a solution in a short amount of time we cannot place an upper bound on the size of the solution.

A* Search

The $A^\star$ algorithm guarantees an optimal solution, and, in general requires less time than an exhaustive search. It requires the definition of two functions, g and h. For any node g defines the distance between the starting point and the current node, and h is an estimate of the

Algorithm 2 Measurement selection using matrix manipulation.

```
n ← number of faults
m ← number of measurement variables
Initialize DM_solution[j,k] to all 0 entries
Initialize OSET_solution to empty set
for all i = 1 to m do
  for all Faults F_j do
    for all Faults F_k do
      calculate DM_i[j,k]
    end for
  end for
end for
while DM_solution[j,k] ≠ Identitymatrix and
        size(OSET_solution) ≤ m do
  DM_minimum[j,k] ← DM_solution[j,k]
  for all i = 1 to m do
    if O_i ∉ OSET_solution then
      if DM_i[j,k] > DM_minimum[j,k] then
        DM_minimum[j,k] ← DM_i[j,k]
        O_next ← O_i
      end if
    end if
  end for
  OSET_solution ← OSET_solution ⋃ O_next
  DM_solution[j,k] ← DM_solution,O_next[j,k]
end while
if DM_solution[j,k] = Identitymatrix then
  Output OSET_solution as solution
else
  Output NO SOLUTION POSSIBLE
end if
```

distance from the current node to the goal. If h is an underestimate of the distance to the goal, the solution obtained from the $A^\star$ search will be optimal (Nilsson 1980).

In our algorithm, we keep track of a list of nodes (OPEN), where each node, N_i consists of a partition P_i and a list of observations $OSET_i$ that were used to derive P_i from the initial partition $P_{initial}$. For each node N_i we can calculate the g and h values as follows. Since g represents the actual distance from the $P_{initial}$ to P_i, $g = \mid OSET_i \mid$. Suppose there are t groups in P_i and the size of the largest group, $p_{largest}$ is $\mid p_{largest} \mid = t_l$. We derive the h function in two ways.

(1) By selecting one observation O_j we can split P_i which has t groups to a new partition $P_{new} \leftarrow SPLIT(O_j, P_i)$. P_{new} can have at most $4 \times t$ groups. Extending this, if we select h observations then we can get at most $4^h \times t$ groups. Our goal is to have atleast n groups, $4^h \times t \geq n$ for the goal state. Therefore, $h_1 = log_4(t)$. This h value is always an underestimate of the actual number of measurements required to achieve the goal state.

(2) By selecting one observation we can split $p_{largest}$ into at most 4 groups $p_{l_1} \cdots p_{l_4}$. At best $\mid p_{l_1} \mid = \mid p_{l_2} \mid = \mid p_{l_2} \mid = \frac{t_l}{4}$. Extending this argument, if we select h observations then we have a group of size $\frac{t_l}{4^h}$ or bigger. Our goal is to get groups of size 1, therefore $t_l/(4^h) = 1$. Therefore, $h_2 = log_4(t_l)$ Again it is easy to see why h_2 is an underestimate.

Since the size of the largest partition and the number of partitions are not necessarily the same, we get two different estimates of the distance to the goal. We calculate both estimates and then select the greater of the two values. Hence,

$$h = \begin{cases} h_1 & \text{, if } \quad h_2 \leq h_1 \\ h_2 & \text{, if } \quad otherwise \end{cases}$$

The $A^\star$ algorithm is presented as Algorithm 3. We initialize node N_1 to $P_{initial}$ and $OSET_1$ = NULL. At each step we select the node N_i that has the lowest value for $f_i = g_i + h_i$. In case of ties we select the node that has the lowest value for h_i among the nodes that have the lowest value for f_i. If ties remain, we make a random selection from among the tied nodes. For the selected node N_i, if $P_i = P_{goal}$ then $OSET_i$ is the required solution. If $P_i \neq P_{goal}$, we create $m - \mid OSET_i \mid$ new nodes in the following manner: $\forall$ *observations* $O_j \notin OSET_i$, we create a new node N_{new} with values $P_{new} = SPLIT(O_j, P_i)$ and $OSET_{new} = OSET_i \bigcup O_j$. We add all the newly created nodes to OPEN, delete N_i from OPEN and repeat the process.

The algorithm provides an optimal solution, because the selected h function is an underestimate of the goal. The number of sets selected from $OPEN$ for expansion before the solution is reached, is a good indicator of the time complexity of the algorithm. In the exhaustive search case this number will be $m \times (m-1) \times (m-2) \ldots \times (m - \mid OSET_{solution} \mid)$. In the worst case, $A^\star$ will expand the same number of nodes as the exhaustive search. In the best case, the algorithm performs like a greedy algorithm picking only the required observations in turn. In this case, the number of nodes expanded is $\mid OSET_{solution} \mid$. In practice, neither case is likely to occur and the actual time complexity lies somewhere in between.

Experimental Results

All the measurement selection algorithms were implemented on a Java platform and tested on a number of systems. Four systems were used to test the measurement selection algorithms. The two tank system and its temporal causal graph are illustrated in Figs. 2 and 3. It is a second order system with five component parameters, $\{C_1, C_2, R_{b1}, R_{12}, R_{b2}\}$. The secondary sodium cooling loop[1] and its temporal causal graph

[1] We use a simplified version of the system, and assume

Table 1: Measurement selection results of the four algorithms.

System	n	m	Exhaustive Search	A* Search	Matrix method	Weighted split
Two Tank	10	16	e1,f8	e5,f3	e5,f3	e5,f3
Secondary sodium cooling loop	22	50	e2,e11,f1,f18	e2,e11,f1,f18	e2,e11,e12,f1,f19	e2,e11,e12,f1,f19

n = number of faults in the system, m = number of measurement variables in the system, e_k,f_k = effort and flow variables on bond number k

Algorithm 3 Measurement selection using A* algorithm.

n ← number of faults
m ← number of measurement variables
$N_{new} = (P_{initial}, emptyset)$
$OPEN = OPEN \bigcup N_{new}$
Select from OPEN , the node N_{lowest} with lowest value for f
while $P_{lowest} \neq P_{goal} and OPEN \neq empty$ **do**
  for all i = 1 to m **do**
    if $O_i \notin OSET_{lowest}$ **then**
      $N_{new} \leftarrow (SPLIT(O_i, P_{lowest}), OSET_{lowest} \bigcup O_i)$
      $OPEN \leftarrow OPEN \bigcup N_{new}$
    end if
  end for
  $OPEN \leftarrow OPEN - N_{lowest}$
end while
if $P_{lowest} = P_{goal}$ **then**
  Output $OSET_{lowest}$ as solution
else
  Output No Solution
end if

are illustrated in Figs. 6 and 7, respectively. Heat from the reactor core is transported to the turbine by a primary and secondary cooling system. Liquid sodium is pumped through an intermediate heat exchanger to transport heat from the primary cooling loop to the feed water loop by means of a superheater and evaporator vessel. This model contains eleven parameters, $\{R_1, R_2, R_3, R_4, R_5, C_{OFC}, C_{EV}, C_{SH}, I_{IHX}, m_1, m_2\}$. A three tank example with three tanks and two connecting pipes. There is only one outlet pipe from tank 3. This is a third order system with six component parameters, $\{C_1, C_2, C_3, R_{12}, R_{23}, R_{b3}\}$. A cooling fin which is designed to conduct heat away from a hot fluid along the fin and ultimately to the surrounding air. This is a fifth order system with 12 parameters.

Discussion of Results

Table 1 summarizes the results of running the measurement selection algorithms on the two tank and sodium cooling loop system. Table 2 shows the number of nodes expanded when an exhaustive search and an A* search is applied to the systems discussed.

that only the R, C, and I components can fail.

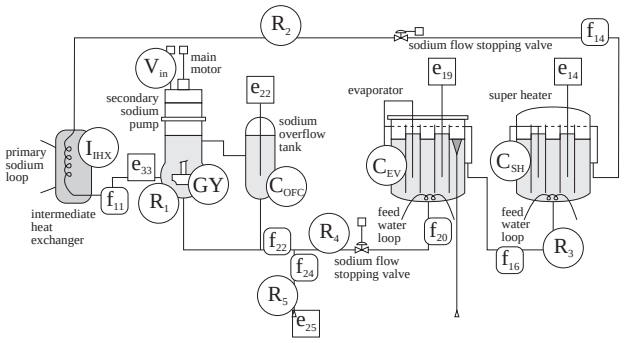

Figure 6: **Secondary sodium cooling loop.**

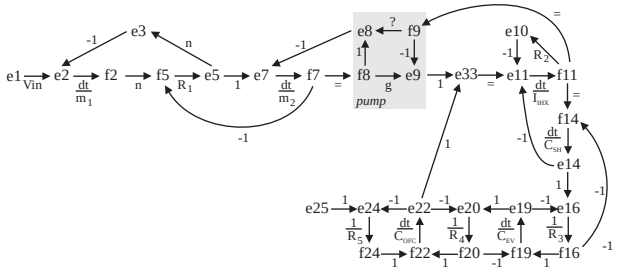

Figure 7: **Temporal causal graph for Secondary sodium cooling loop.**

Table 2: Number of nodes expanded for Exhaustive search and A* search.

System	n	m	Nodes expanded		fraction
			Exhaustive	A*	A* / Exhaustive
Two Tank	10	16	31	6	19.35%
Three Tank	12	22	106	8	7.55%
Cooling Fin	24	40	252069	2331	0.92%
Sodium loop	22	50	4178181	30068	0.72%

The results show clearly that the A* method expands only a small fraction of the nodes that the exhaustive search expands while still generating the optimal solution. The effectiveness of the $A^\star$ algorithm is clearly dependent on how close the h function is to the true distance to the goal. Our empirical results show that the h function chosen is a close underestimate of the true value. The difference in nodes expanded becomes more significant as the size of the system increases.

The two non optimal methods produce similar results. For a small system like the two tank system the solution size is optimal but for more complicated systems like the secondary sodium cooling loop the solution size is bigger than optimal (one measurement point more than optimal). One of the main reasons for the non optimal solutions from the greedy methods is that after the first few partitions have been made, measurement points further split only a fraction of the partitions thereby requiring more measurement points to further split all partitions. This approach is similar to breaking a problem into sub problems but in this case each sub problem needs to be solved individually, hence there is a waste of resources.

Conclusions

Our work on measurement selection illustrates the trade-off between the size of the solution obtained and the speed of computation. One of the important factors affecting the size of the solution is the order of the signatures used. Signatures can be predicted to any order derivative and so if we use higher order derivatives we need lesser measurements to completely diagnose the system. However second and higher order derivatives take longer time to manifest in actual measurements. Another factor not taken into account in this paper is the use of discontinuities in measurement values for fault isolation (Mosterman & Biswas 1998). If we can reliably measure discontinuities in the system then we get more discriminating power but discontinuities cannot be detected with any degree of confidence by our current monitoring schemes. Also, we do not consider any cost functions in evaluating observations. The cost function for any observation could be based on several factors like actual cost of sensors, accessibility, and reliability. All of the measurement selection algorithms are general in the sense that if we can define appropriate discriminatory matrices and split operators they are applicable under any diagnosis framework.

In future work, we would like to estimate these cost functions and incorporate them into the search process. We would also like to study techniques to switch between algorithms based on requirements and system configurations. One such strategy would be to perform an $A^\star$ search for a predetermined amount of time and then switch to one of the greedy searches from that point. Another area of interest would be to include domain specific knowledge for commonly used systems to facilitate development of more efficient algorithms without compromising the quality of the solution.

References

Aho, A. V., Hopcroft, J. E., and Ullman, J. D. 1974. *The Design and Analysis of Computer Algorithms.* Addison Wesley.

Chantler, M. J.; Daus, S.; Vikatos, T.; and Coghill, G. M. 1996. The use of quantitative dynamic models and dependency recording for diagnosis. In *Seventh Intl. Workshop on Principles of Diagnosis*, pp. 59–68, Val Morin, Canada.

Chien, S.; Doyle, R.; and de Mello, L. H. 1991. Model-based reasoning for sensor placement. In *Proc. of the AAAI-91 Workshop on Model-Based Reasoning.*

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

Luenberger, D. G. 1979. *Introduction to Dynamic Systems: Theory, Models, & Applications.* New York: John Wiley & Sons.

Mosterman, P. J.; and Biswas, G.; and Narasimhan, S. 1997. Measurement Selection and Diagnosability of Complex Dynamic Systems. In *Eighth Intl. Workshop on Principles of Diagnosis*, pp. 79–86.

Mosterman, P. J., and Biswas, G. 1997. Monitoring, Prediction, and Fault Isolation in Dynamic Physical Systems. In *AAAI-97*, pp.100–105. Rhode Island.

Mosterman, P. J., and Biswas, G. 1998. Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics.* in review.

Mosterman, P. J. 1997. *Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis.* PhD dissertation, Vanderbilt University. http://www.vuse.vanderbilt.edu/~pjm/papers/thesis/p.html.

Nilsson, N. J. 1980. *Principles of Artificial Intelligence.* Palo Alto, CA: Tiogo publishing Co.

Yu, X., and Biswas, G. 1992. A method for diagnosis of continuous-valued systems. In *Proc. of the Third Intl. Workshop on Principles of Diagnosis*, pp.57–65.