QCBFS: Leveraging Qualitative Knowledge in Simulation-Based Diagnosis

T. K. Satish Kumar
Knowledge Systems Laboratory
Stanford University
tksk@ksl.stanford.edu

Abstract

In continuously evolving systems (including hybrid systems), trajectory selection (diagnosis) based on limited observations is usually a difficult thing to do. Probabilistic approaches to this problem try to select hypotheses based on their posterior probabilities conditioned on the observations made. These approaches, however, try to relate parameters of a hypothesis directly to the observations made (usually under unwarranted assumptions) without respecting the complexity of the equation models according to which they may be related - hence leading to their inaccuracy. Computationally also, the number of competing hypotheses may be too large to gain tractability over a reasonably big physical system. There is also no elegant way of leveraging qualitative knowledge of the system to obtain computational gains. In this paper, we remove all these drawbacks by first formulating the diagnosis problem as a CSP (requiring simulation to perform consistency checks). We then define a cost model and describe an algorithm called QCBFS which not only avoids having to deal with too many competing hypotheses, but also provides a unifying framework to leverage any qualitative knowledge or probability estimates from previous approaches. Probability estimates imported from other sources can affect only the focusing power of QCBFS but not its accuracy.

Introduction

In continuously evolving systems (including hybrid systems), trajectory selection (diagnosis) based on observations made across the time line is usually a difficult thing to do. A traditional approach to this problem has been to compute the posterior probabilities of models (trajectories) conditioned on the observations made and then to pick the one with the highest such probability. This process however, is often subject to imprecision resulting from an underestimation of the complexity of the equation models that guide the evolution of the system. Probability models try to directly relate the different modes in which the system could exist to the observations made. Because they neglect the complexity of the true relationships, such models tend to be imprecise. Also, there are usually an extremely large number of candidate hypotheses for which such probability terms have to be estimated and this makes such an approach infeasible.

Another drawback of the traditional approaches is that there is no elegant way of making use of any qualitative knowledge that may be available. Previous attempts have been made to reduce the search space for the candidate solution by leveraging qualitative knowledge. One previous attempt, for example, has been to use qualitative knowledge as an oracle to increase the posterior probabilities of certain candidates as opposed to others (McIlraith 2000). In (Mosterman and Biswas 1999), qualitative knowledge in the form of TCGs (Temporal Causal Graphs) was used to first generate a qualitative candidate diagnosis over which model fitting techniques could be applied for subsequent quantitative refinement.

Consider diagnosing the discrete state in which a system is by making observations on some of its parameters over a time line. One interesting way to perform the diagnosis is to cast it as a CSP (Constraint Satisfaction Problem). We might know that the system is in one of a finite set of possible states, each of which is defined by the combination of modes in which individual components could be in. A state is consistent if upon simulation we get results that match the actual observations made. Such an approach may especially make sense when the equation models according to which a system evolves differ arbitrarily across different discrete modes of the system. We can also extend this approach towards diagnosing a system over a complete time line under a theory of actions and faults. Here, we are also interested in reasoning about possible transitions between discrete modes of the system.

The above process is computationally very expensive because the consistency checks usually involve simulation of complex equation models. Often however, we have some form of qualitative knowledge that we can leverage. This knowledge can be in many forms. It can either be in the form of cause-effect relations between parameters of the system which a TCG tries to capture, or it can be in the form of an induced ordering on the modes of a component with respect to a variable. For example, the flow variable induces an ordering on the modes of a valve (*open* and *close*) for qualitative states 0 and +ve. It now becomes necessary for us to solve CSP formulations of the mode identification problem in a way so that we can leverage qualitative knowledge to avoid performing too many consistency checks. We also have to incorporate the knowledge we have about the prior probabilities of the modes under which system components can behave and make use of any other estimates of posterior probabilities of competing hypotheses to obtain computational gains.

In this paper, we first formulate the diagnosis problem as a CSP for a layered architecture of system dynamics. We concentrate first on diagnosing a discrete mode in which a system is, under the observations made (assuming that this mode does not change). We will show that we can then extend our theory to include reasoning about actions and faults. We will develop a search algorithm called QCBFS (extending the algorithm CBFS (Williams and Nayak 1996)) to incorporate qualitative knowledge and other probability estimates. As we will see, QCBFS poses the interesting question of performing optimal search when one node has the power to eliminate others upon not satisfying the goal conditions. We show that different kinds of probability estimates (from earlier approaches) can be used as heuristics to guide the search. Moreover, any inaccuracy in these estimates affects only the focusing power of QCBFS and its ability to choose the right hypothesis is always retained.

Casting Diagnosis as a CSP

Diagnosing a hybrid system is usually a very complex task. It requires us to track not only what discrete jumps the system has undergone, but also the continuous path it has followed in each of these discrete modes, under the presence of process and observation noise. In our view of the problem, it is very important to track the discrete modes first and then use standard filtering approaches (like particle filtering) to track the particular path it has followed inside each of them.

Most discrete modes are characterized by equation models (which could differ arbitrarily from those for other discrete modes) along which continuous evolution of the system takes place - albeit that there may be associated noise. Compositional modeling (Falkenhainer and Forbus 1991) is a technique which relates discrete modes of components to the equations they contribute towards system behavior through the notion of *model fragments*. Here, references to discrete modes of a component occur only in the *preconditions* of *model fragments* and this helps us abstract the behavior of a system to an assignment of the modes of individual components.

In Figure 1, we illustrate the basic problem that we are solving as a first step. Each combination of modes of the individual components (called a model) induces a different equation model according to which the system behaves over a time line. Upon simulation of the corresponding equation models, therefore, predictions can be made for the observation variables at any time point. We would then select that combination of modes for individual components (a model) which makes the right predictions for the true values of the observation variables. Allowing for the presence of process and observation noise, however, we would like to retain models which give a *reasonable* value for $P(O_1, O_2, ...O_t/M)$ (i.e. the probability of observations $O_1...O_t$ given model M).

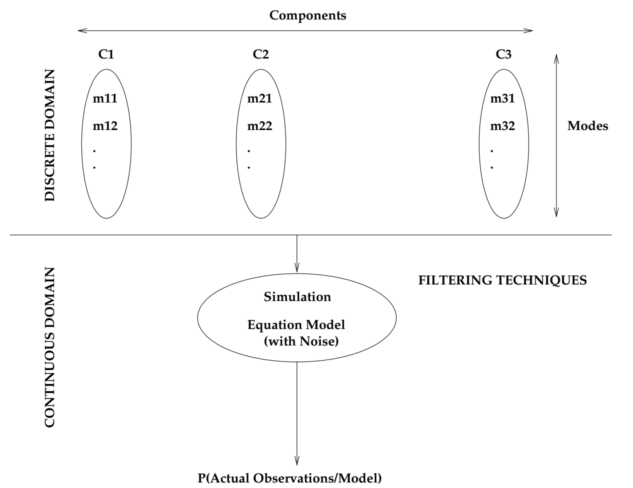


Figure 1: A diagrammatic illustration of the *layered model* for diagnosis.

Meaning of Simulation

The word ”simulation” is used here in a generic sense and can mean any procedure which takes a given model and makes predictions about its behavior over a time line with a high level of accuracy. Because we insist on accuracy, we must be willing to associate a high computational budget with it. A lot of reasoning systems and methodologies like DME (Device Modeling Environment) (Iwasaki and Low 1993), HCC (Hybrid CC) (Carlson and Gupta 1998), or even bayesian inference procedures (including sufficient detail of hidden nodes) would be considered as *simulation* procedures. The underlying theme that we want to capture is that although these reasoning mechanisms are well suited for specific problem domains and meet different requirements, they tend to be computationally expensive, and we should therefore seek to minimize the number of times we use them in our diagnosis engine.

Rank Probabilities

The idea is to make use of the fact that $P(M)$s differ from each other to a much greater degree than $P(O_1, O_2, ...O_t/M)$s for different models. Let us call these probabilities ($P(M)$ and $P(O_1, O_2, ...O_t/M)$) the *rank* and *adaptation* respectively [1]. Therefore, hypothesis A having a higher *rank* than B will also be considered to have a higher posterior probability (conditioned on the observations) unless its *adaptation* is too low.

It is important to note that the parameter determining whether the *adaptation* of a hypothesis is too low or not, is dependent on the particular physical domain and may incorporate an estimate of the distribution of the prior probabilities of different hypotheses. This is the reason why we require our simulation procedure to be accurate. More than just generically reporting whether the *adaptation* is good

[1] Rank probabilities were first made use of in (Kurien and Nayak 2000)

enough or not, it should be able to make decisions for any fine-tuning of the parameter.

CSP Formulation and Model Selection

Diagnosis is the model selection problem of maximizing $P(M/O_1, O_2, ...O_t)$. These probability terms are non-intuitive to compute because the order of the cause (observations $O_1, O_2....O_t$) and the effect (model M) is reversed. One way to get around this is to maximize $P(M)P(O_1, O_2....O_t/M)$ instead. Unfortunately however, $P(O_1, O_2, ...O_t/M)$ is very hard to compute since it often requires simulation of some kind to achieve any reasonable level of accuracy. It is also infeasible to precompute the simulation results of all possible models and use them later. This is because there are usually an extremely large number of possible models and in general, faults have the property that they can be of arbitrarily low prior probabilities. This restricts us from precomputing and storing the results of simulation for only those hypotheses with high prior probabilities.

Casting diagnosis as a CSP allows us to maximize $P(M/O_1, O_2, ...O_t)$ in a different way (as opposed to previous approaches). The idea is to enumerate models in the order of their *ranks* and reject a hypothesis only if we feel that its *adaptation* is too low - otherwise we approve of it. Note that by doing this, we are making it similar to a *consistency-based* approach. We can employ standard search space pruning techniques like CBFS (Conflict-directed Best First Search) (Williams and Nayak 1996) to do the diagnosis - except now, a different cost model and our desire to leverage other forms of knowledge calls for a more efficient algorithm (QCBFS) which we will describe later in the paper.

CBFS

CBFS (Conflict-directed Best First Search) (Williams and Nayak 1996) is a technique to do simultaneous constraint satisfaction and maximization of prior probabilities of competing hypotheses under a compositional cost model of the modes in which individual components can behave. It works in conjunction with a TMS (Truth Maintenance System) to catch minimal size conflicts. It uses a successor lattice as shown in Figure 3 to do search upon. At any time, it maintains a frontier of nodes (models) under the assurance that it always contains the best model among the ones remaining to be considered. It picks the best model from the frontier and checks it for consistency. If there is a conflict, then it is replaced by its successors that are not covered by the conflict and the next iteration is taken. If there is no inconsistency, then that model is chosen as the required diagnosis. Figure 2 shows a diagrammatic representation of the working of CBFS.

Cost Model

In CBFS, examining the consistency of an assignment to the mode variables was assumed to be not too costly. This was especially so if one maintained an ITMS (Nayak and Williams 1997), since there are only incremental changes in the mode assignments between successors. However, in our

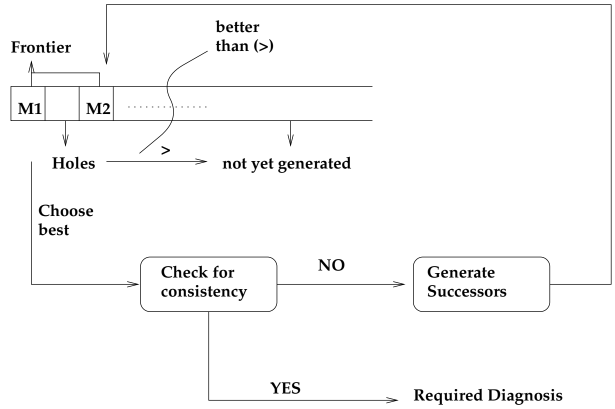


Figure 2: Illustrates how CBFS works.

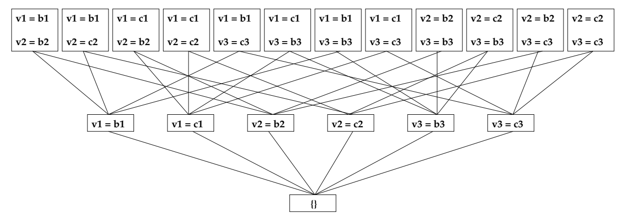


Figure 3: Shows the *successor lattice* used by CBFS. All nodes in level L of the lattice indicate L-way faults in the system. The bottom-most node is the level-0 node indicating *nominal* behavior.

case, a consistency check is an extremely costly affair. This is because it requires simulation over an extended time line. Also, incremental changes in mode assignments cannot be exploited because they can arbitrarily change the equation model according to which the system evolves. All costs related to any search algorithm will therefore be dominated by the number of *simulations* (consistency checks) we do. All other computations, like any processing of the frontier nodes, will be considered as being of negligible cost.

Encoding Qualitative Knowledge

Qualitative knowledge in system dynamics is usually available in many forms. The most popular way of capturing such knowledge has been through the use of TCGs (Mosterman and Biswas 1999). We also introduce the idea of *induced orderings* on the modes of a component as another way of capturing qualitative information. We combine this with TCGs to yield the notion of ELs (Elimination Lattices), which we will use directly in performing effective search.

Temporal Causal Graph

A TCG is an abstract model of the dynamic behavior of a system derived from a bond graph. TCGs have been made use of in several ways to perform diagnosis. In (Lerner et. al. 2000), they were used to provide the skeleton structure for bayesian networks (to do probabilistic reasoning over the system). In (Mosterman and Biswas 1999), they were used to provide candidate qualitative diagnosis so that one could later apply quantitative refinements through parameter estimation and data fitting in the hope that this would reduce the search space. Further, in (McIlraith 2000), the TCGs were used to provide oracles for affecting the posterior probabilities of competing hypotheses.

Mode Orderings

System dynamics is usually determined by integrating the description of individual components. The variables connected with the modes of components in their individual descriptions often induce an ordering on them. For example, the state of a valve (open/close) is ordered with respect to the amount of liquid flowing through it (+ve/zero). But the flow variable may not be directly observable in the system to allow for the use of this qualitative knowledge. Often however, the temporal causal graph captures other causality relations by which the flow variable may be connected to other observable variables. In general, therefore, although in many cases the modes of a particular component are ordered only in relation to some directly connected variables, other observable variables in the system are proportionally related to these so that one can still retain notions related to the ordering on the modes of individual components.

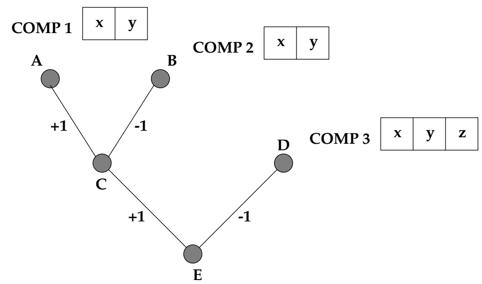
Elimination Lattice

Consider the CSP formulation of the diagnosis problem. Suppose a particular assignment of modes to the components of the system produces simulation results which do not match with the actual observations; then not only can we record this assignment as being a *nogood*, we can also infer (only from qualitative arguments) that some other assignments are *nogoods* too. For example, if we are keeping the bouncing of a ball under observation and we observe that the maximum heights reached fall short of the simulation results for the *nominally blown* case of the ball, then we can also conclude that the ball cannot be *overblown* and restrict our search space only to the *underblown* area. It can be noticed that this way of inferring additional *nogoods* every time inconsistencies are come across can greatly reduce our search space if we follow an *information-theoretically* optimum order of trying out the various assignments.

An *EL (Elimination Lattice)* is an attempt to merge the ideas of the TCG and the ordering induced on the modes of individual components. In its explicit form, an EL should be able to provide all combinations of mode assignments that can be inferred to be *nogoods* (from qualitative arguments) for every assignment recorded as a *nogood* upon simulation (depending upon how the observations and predications compare with each other). We can however have a more implicit and compact representation as follows (see Figure 4).

Definition A *conflict tuple* is an assignment of modes to the individual components of the system such that the predicted behavior (simulation results) does not match the actual observations within the level of accuracy required.

Definition A *conflict tag* for an observation variable E being *greater* than its predicted value for a conflict tuple is the ordered set $[t_1, ...t_n]$ (where $t_i \in \{<, >, -\}$ and n is the number of components) such that: (1) $t_i =' <'$ if the



	Conflict Tuple	Tag
E: Actual Value > Predicted Value	<c1, c2, c3>	(<, >, >)
E: Actual Value < Predicted Value	<c1, c2, c3>	(>, <, <)

Figure 4: Shows the *ConflictDB* entry for observation variable E with respect to the components COMP 1,2&3.

variable inducing an ordering on the modes of $component_i$ is in direct proportionality with E; (2) $t_i =' >'$ if there is an inverse proportionality and (3) $t_i =' -'$ otherwise. (Interchange $' <'$ and $' >'$ for actual value of E being *lesser* than its predicted value).

Definition Two modes m_1 and m_2 of a component satisfy $m_1 < m_2$ if m_1 occurs before m_2 in the ordering induced on the modes of the component by a connected variable [2]. $m_1 < m_2$ iff $m_2 > m_1$ and $m_1 - m_2$ is always true. Also, $\bar{>} =<$; $\bar{<} =>$; and $\bar{-} = -$.

Definition An assignment of modes $[m_1, ...m_n]$ to components $[comp_1, ...comp_n]$ *passes* through a *conflict tag* $[t_1, ...t_n]$ and its corresponding *conflict tuple* $[c_1, ...c_n]$ if for no i, $m_i \bar{t_i} c_i$ is true and $[m_1, ...m_n] \neq [c_1, ...c_n]$

Definition An assignment of modes to components $[m_1, ...m_n]$ *passes* through the *conflictDB*, if it *passes* through all the *conflict tags* with respect to their *conflict tuples*.

QCBFS

Reconsider the CBFS approach. The algorithm has dual goals - that of checking for a consistent model and that of maximizing its prior probability. Rightfully enough, one can abstract the working of CBFS as producing hypotheses in the order of their prior probabilities and sequentially checking them for consistency. The first one which turns out to be consistent is chosen as the required diagnosis.

Now suppose that we had a third goal of minimizing the number of consistency checks - which under our cost model corresponds to the number of simulations we perform. Notice that some models which become *conflict tuples* also acquire the power to eliminate others by virtue of the corresponding *conflict tags*. This is a reflection of the way in which we make use of qualitative knowledge. The interesting question that arises out of this is whether we should

[2]For the sake of simplicity, we assume that the modes of a component cannot be ordered in more than 1 way by different connected variables

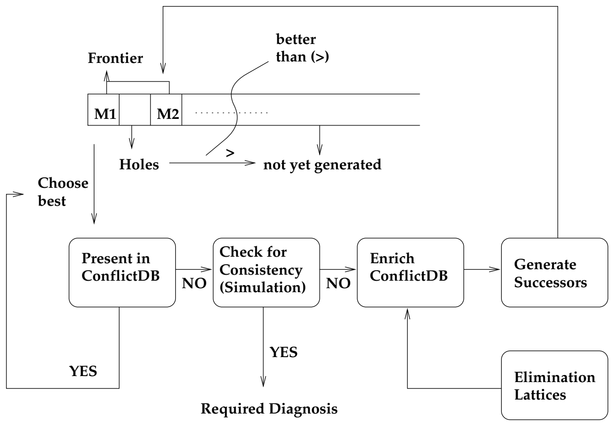


Figure 5: Illustrates how QCBFS works.

always check for the consistency of models in order of their *ranks* and hope to find the required diagnosis early on, or do we check the consistency of models in order of how much information they carry about the elimination of others to be able to have lesser models to perform search over in the future.

The solution to the above question is in the analysis of the recurrence relation for the cost associated in solving a problem of size n (n is the number of models to search over). For the sake of analysis, consider that the models are arranged in order of their *ranks* (which we can produce anyway). The following recurrence relation holds true:

$$C(n) = min_i\{1 + p_i C(i-1) + (1-p_i)C(n-e_i)\}$$

This is because if model H_i in the sequence is chosen, then a cost of 1 is always associated with doing its simulation. If it turns out to be consistent, the best solution could only be in the range of models $H_1 ... H_{i-1}$. Here, p_i is the probability that the model H_i will have a sufficiently high *adaptation* value. If on the other hand, the model fails, then we can also eliminate $e_i - 1$ other models by virtue of the *conflict tags* generated. Note that e_i measures only the number of models that can be eliminated (including H_i itself) among the ones that have not yet been thrown out of consideration. The algorithm is now simply to choose at each step that H_{i^*} which forms the argument of the minimization.

Working of QCBFS

Figure 5 shows the working of QCBFS. The algorithm maintains a frontier of nodes over the successor lattice (Figure 2) in the same way as CBFS does. Unlike CBFS however, the choice of the *best* node to expand in the frontier is not according to its *rank*, but according to the argument of minimization for the cost function described earlier. Having selected a hypothesis, it first checks for its presence in the *conflictDB*. The *conflictDB* contains all necessary information for detecting whether a model has been rejected in the past. This may be in the form of *conflict tuples* and *conflict tags* possibly generated for some other model. If a model *passes* through the *conflictDB*, then an actual simulation is done to check for its *adaptation*. If the *adaptation* is not too low, then the hypothesis is the required diagnosis. Otherwise, the EL is used to enrich the *conflictDB*; and the successors of the hypothesis are added into the frontier.

Issues and Observations for QCBFS

The recurrence relation described above raises a lot of issues - all of which do not have the best answers, and approximations may be required at many places. We provide a discussion on each of these below.

Position of frontier hypotheses The position of a hypothesis H (presumably belonging to the frontier) among all hypotheses (including the ones not yet generated) in the sorted order of their *ranks*, is not necessarily the same as its position among the hypotheses that have already been generated (which include those present in the frontier). This is because of the presence of *holes* as shown in Figure 5. However, we assume that it is tolerable to compare the true position of two hypotheses by examining only the current frontier.

Models yet to be generated Note that in the recurrence relation, we are seeking a minimization over all models. It is a good approximation to just sort the frontier nodes in terms of their ranks and then do the minimization only over them. This is a reasonable thing to do because the size of the subproblems resulting from a simulation made on H_i tends to increase with i (thereby increasing the cost). Since we know that most nodes yet to be generated are beyond the frontier, the chances that the best node to be expanded next is among them, is very low. We assume that the computation required in sorting the frontier nodes by their *ranks* is negligible compared to the simulation costs.

Probability estimates In the recurrence relation, p_i was the probability that model H_i would not be rejected upon simulation. This is where the estimates made by other probabilistic approaches can be made use of. Note that any inaccuracy of these estimates does not affect the accuracy of QCBFS, rather it may only affect the heuristic power of the search. Also note that p_i is not the same as the *adaptation* of the hypothesis we get upon simulation. It is the probability that this *adaptation* is not too low. Since it is being used only for its *heuristic power*, its calculation can be made from computationally cheap methods.

The cost function The cost function itself may be arbitrarily complex and one does not know $C(i-1)$ or $C(n-e_i)$ to its fullest accuracy unless the recurrence is solved to complete precision. However a linear estimation for $C(n)$ may serve as a good approximation. Other empirical approaches derived from a characterization of the domain of the problem can be incorporated here.

Number of models eliminated e_i measures the number of models (not yet eliminated) one can eliminate from consideration, if the simulation of H_i does not match the actual observations. Note that this number depends on whether the simulation results turn out to be higher or lower than the actual values for each observation variable. Although for each of these cases one can calculate the required number easily (from the *Elimination Lattice* and *conflictDB*), we still need to give them appropriate weighting factors to get a correct estimate of e_i. These estimations may not be too hard because we just have to surmise in each case whether the model produces higher or lower values for the observation variables. As before, any inaccuracy here only affects the *heuristic power* of the search process.

Observation CBFS can now be considered as a special case of QCBFS. In CBFS, none of the models had the power to eliminate others - making e_i = 1. Also, there was no way in which other estimates of posterior probabilities could be used. This made models differ only in their $C(i-1)$ terms and because this was monotonic with respect to the rank of the model (viz i), the best model to be expanded at any stage was always the first one.

Observation If the required model is present in position i, then all models $H_1...H_{i-1}$ will provably undergo an estimation of their p_is. Although this may seem to pose the same disadvantages as the previous approaches, we should note that this estimation is not done beyond computationally cheap methods. Our primary goal is to remove the inaccuracy of previous approaches by making use of *simulation* (which we assume is a costly process). But the interesting thing is that in doing so, we do not lose computationally also because of our ability to leverage other kinds of knowledge.

Observation Sometimes it might be a good idea to not simulate a hypothesis if its estimated probability p_i itself is too low. We might want to do this when we have found a consistent hypothesis H_k, but are solving $H_1..H_{k-1}$ in search of a possibly better one (which might also be consistent). If the estimated probabilities of all these $(H_1...H_{k-1})$ is very low, then we might just want to call off the further probes. The important thing to note here is that any such knowledge that we want to use should be incorporated in the *conflictDB* without affecting the recurrence relation. The *conflictDB* can now do this additional test of very low p_is (among other tests) before allowing for simulation.

Extending the Theory

The theory developed so far can be extended to diagnose system behavior over an extended time line where the discrete modes themselves change. Under the assumption that a system changes its discrete modes only in association with an action, one can extend the theory by simply introducing mode variables on the transitions an action can cause (Kurien and Nayak 2000). Abrupt faults involving the spontaneous change of a discrete mode of a system, can be dealt with by searching for an appropriate point in the time line to add a NOOP [3] and reduce it to the previous problem. The point where we introduce the NOOP is guided by an examination of the *adaptation curve*. The *adaptation curve* is a plot of the cumulative probability $P(O_1, O_2, ...O_t/M)$ built *online* [4] with each observation made.

[3] A degenerate operation to indicate that, in reality, no action has been taken.

[4] Used in the *algorithmic* sense of the word

Summary and Future Work

In this paper, we presented QCBFS as an algorithm to perform diagnosis on physical systems with components behaving in various modes. QCBFS provides a unifying framework in which we can incorporate and reason about various kinds of knowledge. This includes qualitative knowledge of the system dynamics, probability estimates, rank probabilities, compositional costs (a reflection of independence assumptions for component behaviors) and virtually any other heuristic. The basic philosophy upon which QCBFS differs from other approaches is that it does not look for the little structure present among the different discrete modes of a system (which many previous approaches wrongfully look for). Instead, it exploits the good amount of structure present in the behavior of each individual component under the conjecture that the overall structure among the global discrete modes of the system is *compositional* with respect to these. Future work is directed towards an empirical verification of the performance of the algorithm and a detailed study of the different approximations involved.

References

[1] Carlson, B., and Gupta, V. 1998. Hybrid CC and Interval Constraints. In *Proc. HS'98: Computation and Control*.

[2] Falkenhainer, B., and Forbus, K. 1991. Compositional Modeling: Finding the Right Model for the Job. *Artificial Intelligence, 51:95-143*.

[3] de Kleer, J., and Williams, B. C. 1989. Diagnosis with Behavioral Modes. In *Proc. IJCAI-89, 1324-1330*.

[4] Dressler, O., and Struss, P. 1994. Model-Based Diagnosis with the Default-Based Diagnosis Engine: Effective Control Strategies that Work in Practice. In *Proc. ECAI-94*.

[5] Iwasaki, Y,, and Low, C. M. 1993. Model Generation and Simulation of Device Behavior with Continuous and Discrete Changes. In *Intelligent Systems Engineering, 1(2)*.

[6] Kurien, J., and Nayak, P. P. 2000. Back to the Future for Consistency- Based Trajectory Tracking. In *Proc. Seventeenth National Conference on Artificial Intelligence*.

[7] Lerner, U., Parr, R., Koller, D., and Biswas, G. 2000. Bayesian Fault Detection and Diagnosis in Dynamic Systems. In *Proc. Seventeenth National Conference on Artificial Intelligence*.

[8] McIlraith, S., Biswas, G., Clancy, D., and Gupta, V. 1999. Hybrid Systems Diagnosis. In *Proc. Tenth International Workshop on Principles of Diagnosis - DX'99*.

[9] McIlraith, S. 2000. Diagnosing Hybrid Systems: A Bayesian Model Selection Approach. In *Proc. Eleventh International Workshop on Principles of Diagnosis*.

[10] Mosterman, P., and Biswas, G. 1999. Diagnosis of Continuous Valued Systems in Transient Operating Modes. In *IEEE Transactions on Systems, Man, and Cybernetics, 1999. vol. 29, no. 6, pp. 554-565*.

[11] Nayak, P. P., and Williams B. C. 1997. Fast Context Switching in Real-time Propositional Reasoning. In *Proc. Fourteenth National Conference on Artificial Intelligence*.

[12] Williams B. C., and Nayak, P. P. 1996. A Model-Based Approach to Reactive Self-Configuring Systems. In *Proc. Thirteenth National Conference on Artificial Intelligence*.