Pervasive Diagnosis: Integration of Active Diagnosis into Production Plans

Lukas Kuhn, Bob Price, Johan de Kleer, Minh Do, Rong Zhou
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
{lkuhn,bprice,dekleer,minhdo,rzhou}@parc.com
Contact: 650-812-4802 FAX: 650-812-4334

Abstract

In model-based control, a planner uses a system description to create a plan that achieves production goals. The same model can be used by model-based diagnosis to indirectly infer the condition of components in a system from partially informative sensors. Existing work has demonstrated that diagnosis can be used to adapt the control of a system to changes in its components, however diagnosis must either make inferences from passive observations of production plans, or production must be halted to take specific diagnostic actions. In this paper, we observe that the declarative nature of model-based control allows the planner to achieve production goals in multiple ways. We show that this flexibility can be exploited by a novel paradigm we call *pervasive diagnosis* which produces *diagnostic production plans* that simultaneously achieve production goals while generating additional information about component conditions. We derive an efficient heuristic search for these diagnostic production plans and show through experiments on a model of an industrial digital printing press that the theoretical increase in information can be realized on practical real-time systems and used to obtain higher long-run productivity than a decoupled combination of planning and diagnosis.

1 Introduction

Artificial intelligence has long been animated by the vision of creating fully autonomous systems that not only act on the larger world, but also maintain and optimize themselves. The increased autonomy, reliability and flexibility of such systems are important in domains ranging from space craft, to manufacturing processes, to computer networks. Autonomy can be seen as the combination of two processes: diagnosis of the current condition of components in a system from weakly informative sensor readings and model-based control of system operation optimized for the current condition of the system. In an aerospace domain, flight dynamics models can be used to diagnose faults in flight control surfaces from noisy observations of flight trajectories. A model-based flight controller could then compensate for the faults by using alternative control surfaces or engine thrust to achieve the pilot's goals.

Diagnosis and model-based control are typically combined in one of two ways: 1) alternation of an *explicit diagnosis* mode with a model-based control mode or 2) parallel execution of a *passive diagnosis* process with a model-based control mode. The alternation of an *explicit diagnosis* mode with a model-based control mode typically results in long periods in which regular operation must be halted. This is particularly true when diagnosing rare intermittent faults. The combination of a *passive diagnosis* process with model-based control is often unsuccessful as regular operation may not sufficiently exercise the underlying system to isolate an underlying fault.

In this paper we introduce a new paradigm, *pervasive diagnosis*, in which certain parameters of model-based control are actively manipulated to maximize diagnostic information. *Active diagnosis* and model-based control can therefore occur *simultaneously* leading to higher overall throughput and reliability than a naive combination of diagnosis and regular operation. In this paper, we show that *pervasive diagnosis* can be efficiently implemented by combining model-based probabilistic inference together with our own unique decomposition of the information gain associated with executing a plan and an efficient heuristic search that exploits this decomposition. Our formulation turns out to be an efficient general solution to the problem of creating plans that produce maximal information gain from a system.

In the next section we examine how Pervasive Diagnosis compares to existing work. In subsequent sections we describe pervasive diagnosis in detail, and explain a specific instantiation of pervasive diagnosis. The implementation of this instantiation is then demonstrated on a commercial scale printing system.

2 Related Work

The general mechanisms for inferring underlying causes of observations have a long history in artificial intelligence and engineering including logic based frameworks [15], continuous non-linear systems [11], probabilistic models for mobile robotics [17], xerographic systems [18], and hybrid logical probabilistic diagnosis [12] .

In active diagnosis, specific inputs or control actions are chosen to maximize diagnostic information obtained from a system. The optimal sequence of tests has been calculated for static circuit domains [3]. This can be extended to sequential circuits with persistent state and dynamics through time-frame expansion methods section 8.2.2 of [2]. Finding explicit diagnosis tests can be done using a SAT formulation [1]. The use of explicit diagnosis jobs has been suggested for model-based control systems [7].

The combination of passive diagnosis to obtain information and model-based control conditioned on the information has appeared in many domains including automatic compensation for faulty flight controls [14], choosing safe plans for planetary rovers [4], maintaining wireless sensor networks [13] and automotive engine control [10].

We are not aware of any other systems which explicitly seek to increase the diagnostic information returned by plans intended to achieve operational goals.

3 Pervasive Diagnosis

Pervasive diagnosis is a new paradigm in which production is actively manipulated to maximize diagnostic information. Active diagnosis and production can therefore occur *simultaneously* leading to higher long run productivity than passive diagnosis or alternating active diagnosis with production.

The integration of diagnostic goals in the production strategy results in *informative production.* The primary objective in informative production is to continue production. Under the assumption that there are various ways to achieve the production goals, informative production simultaneously maximizes diagnostic information. The literature describes different types of production such as simple production, time efficient production, cost efficient production and, robust production. All of those share the primary objective of achieving production but differ in the way they approach the goal. In simple production any strategy that achieves the production goal qualifies. In all other approaches the set of production strategies, those which achieve the production goals, are ranked by a secondary objective function and the best production strategy dominates. For example in time efficient production, strategies are ranked by cost and the most cost efficient production strategy dominates. Similar to other production strategies informative production ranks the set of plans that achieve production goals by their potential information gain and selects the most promising strategy.

4 System

As part of our group work on self-aware, planner-driven systems [8] we have designed and built the modular redundant printing engines illustrated in Figure 1. The system is controlled by a model-based planner [5]. The model of the systems describes all the components in the system, the connections between the components and all the actions a component can take. The task of the planner is to find the sequence of actions, called plan, which will move sheets through the system to generate the requested output.

Expanding this work on model-based controlled system we integrate a framework which integrates planning and

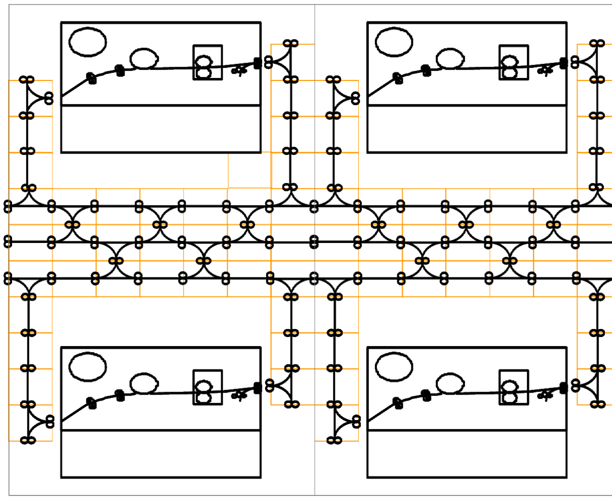

Figure 1: Model of PARC's prototype highly redundant printer. It consists of two towers each containing 2 printers (large rectangles). Sheets enters on the left and exit on the right. Dark black edges with small rollers represents a possible paper path. There are three main paper (horizontal) highways within the fixture. The fixture incorporates 2 types of media handling modules represented by small lighter edge rectangles. The motivation for this design is to continue printing even if some of the print engines fail or some of the paper handling modules fail or jam.

diagnosis to optimize production for long-run productivity called 'Pervasive Diagnosis'.

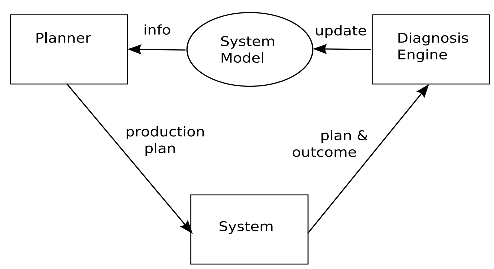

Figure 2: Overall System Architecture

The overall framework of the system is depicted schematically in Figure 2. The planner creates an informative production plan that achieves production goals and generates informative observations. This plan is then executed on the system. For many domains, a plan consisting of numerous actions must be executed before a useful observation can be made. The diagnostic engine updates its beliefs to be consistent with the plan executed by the system and the observations. The diagnostic engine forwards updated beliefs to the planner to inform the planner about the new information needs of the system. The planner and diagnosis engine both run in real time.

5 Representing Information

In order to develop an algorithm to find plans efficiently, we will need to be more precise about what a system is, what the effect of a production plan on the system is, and what makes a plan informative.

We model a system as a state machine with actions A. The system is controlled by plan p consisting of a sequence of actions $a_1, a_2, \ldots, a_n$ drawn from A. Executing an action potentially changes the system state. Part of the system state may be used to represent the state of a product. In this way, execution of actions can also contribute to creating a product. The internal constraints of the system limit the set of feasible plans to a subset of all possible sequences. From the point of view of diagnosis, executing the actions of plan p results in a single observable, the plan outcome or observation O. In keeping with the diagnosis literature, we define two outcomes: the abnormal outcome, denoted $ab(p)$, in which the plan fails to achieve its production goal and the *not* abnormal outcome, denoted $\neg ab(p)$, in which the plan does achieve the production goal.

Information about the system is represented by the diagnosis engine's belief in various possible hypotheses. A hypothesis is an assignment of abnormal or not abnormal to each of the system actions *e.g.*, $h = [ab(a_1), \neg ab(a_2), \ldots, ab(a_n)]$. In the single fault case, exactly one action will be abnormal. Let H_{sys} be the set of all hypotheses. We distinguish one special hypothesis: the *no fault* hypothesis, h_0 under which all actions are not abnormal. Since every hypothesis is a complete assignment of abnormality to each action, they are all unique and mutually exclusive (*e.g.*, $\forall h_i, h_j \in H_{sys}, h_i \neq h_j$.)

The system's beliefs are represented by a probability distribution over the hypothesis space H_{sys}, $\Pr(H)$. The beliefs are updated by a diagnosis engine from past observations using Bayes' rule to get a posterior distribution over the unknown hypothesis H given observation O and plan P:

$$\Pr(H|O,P) = \alpha \Pr(O|H,P) \Pr(H)$$

The probability update is addressed in more detail in a related forthcoming paper. For this paper we simply assume that we have a diagnosis engine that maintains the distribution for us.

A plan is said to be informative, if it contributes information to the diagnosis engine's beliefs. We can measure this formally as the mutual information between the system beliefs $\Pr(H)$ and the plan outcome conditioned on the plan executed, $I(H; O|P = p)$. The mutual information is defined in terms of entropy or uncertainty implied by a probability distribution. A uniform distribution has high uncertainty and a deterministic one low uncertainty. An informative plan reduces the uncertainty of the system's beliefs. Intuitively, plans with outcomes that are hard to predict are the more informative. If we know a plan will succeed with certainty, we learn nothing by executing it. In a forthcoming paper, we explain how to calculate the optimal amount of uncertainty T a diagnosis engine should seek in a diagnosis plan in order to maximize information. In the case of persistent faults, the optimal uncertainty about the outcome would be $T = 0.5$. In the intermittent case, the uncertainty should lie in the range $0.36 \leq T \leq 0.5$. In this paper, we focus on the problem of finding a plan with a given amount of uncertainty T.

The first task is to be able to predict the uncertainty associated with a given plan $p = [a_1, a_2, \ldots, a_n]$. We denote the set of unique actions in a plan $A_p = \bigcup_i \{a_i \in p\}$. We assume *catastrophic* failures, meaning a plan will be abnormal $ab(p)$ if one or more of its actions are abnormal:

$$ab(a_1) \vee \cdots \vee ab(a_n) \Rightarrow ab(p) \text{ for } a_i \in A_p \quad (1)$$

The predicted probability of a plan action being abnormal will be a function of the probability assigned to all relevant hypotheses. The set of hypotheses that bear on the uncertainty of the outcome of plan p is denoted H_p and is defined as:

$$H_p = \{h | h \in H_{sys} \text{ and } h \Rightarrow ab(a), a \in A_p\}. \quad (2)$$

Given a distribution over hypotheses and the set H_p of explanatory hypotheses for plan p, it is possible to calculate the probability that plan p will fail. Since every hypothesis $h \in H_p$ contains at least one abnormal action that is also in plan p, hypothesis h being true implies $ab(p)$:

$$(h_1 \vee h_2 \vee \cdots \vee h_m) \Leftrightarrow ab(p) \text{ where } h_j \in H_p \quad (3)$$

Since the hypotheses are mutually exclusive by definition, the probability of a plan failure $\Pr(ab(p))$ is defined as the sum of all probabilities of hypotheses which imply the plan to fail:

$$\Pr(ab(p)) = \sum_{h \in H_p} \Pr(h) \quad (4)$$

5.1 Search for Highly Diagnostic Plans

Our goal now is to find a plan which achieves production goals, but is also informative: that is, the plan is observed to have an abnormal outcome with probability T. Intentionally choosing a plan with a positive probability of failure lowers our immediate production, but the information gained allows us to maximize future production over a longer time horizon. A naive approach search would generate all possible sequences and filter this list to get the plans that achieve goals and are informative.

$$p^{opt} = \text{argmin}_{\text{achievesGoal}(p) \in \mathcal{P}} |Pr(ab(p)) - T|. \quad (5)$$

Unfortunately the space of plans $\mathcal{P}$ is typically exponential in plan length. We can reduce this by considering families of plans that share structure. In A* planning, there are a set of partial plans $p_{I \to S_1}, p_{I \to S_2}, \ldots, p_{I \to S_n}$. These plans go from the initial state I to the intermediate states $S_1, S_2, \ldots, S_n$. At each step, A* tries to expand the plan most likely to achieve the goal in the best way. The ideal plan p would start with the prefix $p_{I \to S_n}$ which takes us to state S_n and continue with the suffix plan $p_{S_n \to G}$ leading from the state S_n to the goal state G. A* chooses the partial plan $p_{I \to S_n}$ to expand using a heuristic function $f(S_n)$. The heuristic function $f(S_n)$ estimates the total path quality as the quality of the plan prefix $p_{I \rightarrow S_n}$, traditionally written $g(S_n)$, plus the *predicted* quality of the suffix $p_{S_n \rightarrow G}$, traditionally written $h(S_n)$.

$$f(S_n) = g(S_n) + h(S_n). \quad (6)$$

If $f(S_n)$ never overestimates the true quality of the complete the plan, then $f(S_n)$ is said to be admissible and A* is guaranteed to return an optimal plan. The underestimation causes A* to be optimistic in the face of uncertainty ensuring that uncertain plans are explored before committing to completed plans known to be high in quality. The more accurate the heuristic function, the more A* will focus its search only on the high quality plans.

We want the search to run in real time online. We use an idea from the search community to automatically construct a good heuristic function from the description the system architecture and dynamics. Consider the example in Figure 3. In this example, the graph represents legal action sequences or possible plans that can be executed on the system. The system starts in the start state labeled S and follows the arcs through the graph to reach the goal state G.

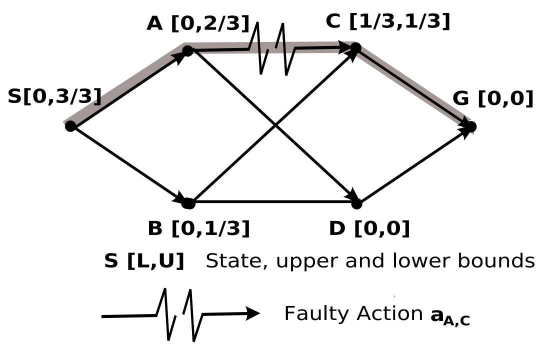

Figure 3: Machine topology places limits on what can be learned in the suffix of a plan

Suppose that we execute a plan that moves the system through the state sequence $[S, A, C, G]$. The sequence is shown as a shaded background on the relevant links in the figure. We assume the plan resulted in an abnormal outcome. Unknown to the diagnosis engine, the abnormal outcome was caused by action $a_{A,C}$. To keep things simple, imagine there is only this single, persistent fault. A diagnosis engine would now suspect all of the actions along the plan path to be faulty, We would have three positive probability hypotheses corresponding to suspected actions: $\{\{a_{S,A}\}, \{a_{A,C}\}, \{a_{C,G}\}\}$. In the absence of additional information, we would assign them equal probability (see Table 1).

The graph structure and probability estimates can be used to construct heuristic bounds on the uncertainty that can be contributed to a plan by any plan suffix. We build up the heuristic from the end of the plan (right side of figure). Consider action $a_{D,G}$ leading from state

Hypothesis	$\{a_{S,A}\}$	$\{a_{A,C}\}$	$\{a_{C,G}\}$
Probability	1/3	1/3	1/3

Table 1: Probability of Hypotheses (single fault)

D to the goal state G in Figure 3. Action $a_{D,G}$ was not part of the plan that was observed to fail, so it is not a candidate hypothesis. Under the single fault hypothesis it has probability zero of being faulted. If we extend any prefix plan ending in state D with action $a_{D,G}$, we don't increase the failure probability of the extended plan, because the action $a_{D,G}$ has probability zero of being abnormal. There are no other possible plans from D to G so both the upper and lower bound for any plan ending in state D is zero.

Similarly, we determine that state B also has a lower bound of zero since it can be completed by an action $a_{B,D}$ which does not use a suspected action and ends in state D which has lower bound zero. State B has an upper bound of 1/3 since it can be completed by an unsuspected action $a_{B,C}$ to state C which has both an upper and lower bound of 1/3 probability of being abnormal.

Once we have recursively built up bounds on the probability of a suffix being abnormal, we can use these bounds with a forward search for a plan that achieves the target probability T. Consider a plan starting with action $a_{S,A}$. Action $a_{S,A}$ was part of the plan that was observed to be abnormal. If we add $a_{S,A}$ to a partial plan, it must add 1/3 probability to the chance of failure as it is a candidate itself. After $a_{S,A}$ the system would be in state A. A plan could be completed through D. Action $a_{A,D}$ itself, has zero probability of being abnormal, and using our heuristic bound, we know a completion through D must add exactly zero probability of being abnormal. Alternatively, from node A, a plan could also be completed through node C. Action $a_{A,C}$ immediately adds 1/3 probability of failure to our plan and using our heuristic bound, we know the completion through C must add exactly 1/3 probability of being abnormal to a plan. The precomputed heuristic therefore allows us to predict total plan abnormality probability. The lower bound of the total plan is 1/3. This comes from 1/3 from $a_{S,A}$ plus 0 from the completion $a_{A,D}, a_{D,G}$, The upper bound is 3/3 equal to the sum of 1/3 from $a_{S,A}$ plus 1/3 from $a_{A,C}$ and 1/3 from $a_{C,G}$. If we complete this plan through $[a_{A,C}, a_{C,G}]$ the total plan will fail with probability 1. Note that this path is the same as the path originally taken. We already know it fails, so it adds no new knowledge under the persistent faults assumption. If we complete this plan through the suffix $[a_{A,D}, a_{D,G}]$ the failure probability of the total plan will be 1/3 which is closer to $T = 0.5$. The plan may or may not succeed indicating that there is something to be learned. Indeed, if it fails, we will have learned that node $a_{S,A}$ was the failed action. Note, that there is no guarantee that a plan exists for any value between the bounds.

The heuristic bounds are calculated recursively starting from all goal states. A goal state has an empty set of suffix plans $P_{G \to G} = \emptyset$ therefore we set the lower bound $L_G = 0$ and upper bound $U_G = 0$. Each new state S_m calculates its bounds based on the bounds of all possible successor states $\mathcal{SUC}(S_m)$ and on the failure probability of the connecting action a_{S_m,S_n} between S_m and S_n. A successor state S_n of a state S_m is any state that can be reached in a single step starting from node S_m. In the single fault case it is true that the failure probability added to a plan $p_{I \to S_n}$ by concatenating an action a_{S_n,S_m} is independent from the plan $p_{I \to S_n}$ if $H_{p_{I \to S_n}} \cap H_{a_{S_n,S_m}} = \emptyset$.

The lower bound S_m will be determined by the action probabilities linking S_m's to its immediate successors and the lower bounds on these successors. The upper bounds are analogous.

$$L_{S_m} = \min_{S_n \in \mathcal{SUC}(S_m)} [\Pr(ab(a_{S_m,S_n})) + L_{S_n}]$$
$$U_{S_m} = \max_{S_n \in \mathcal{SUC}(S_m)} [\Pr(ab(a_{S_m,S_n})) + U_{S_n}]$$

In contrast to the computation of the heuristic, the search for an informative production plan starts from the initial state S and works recursively forward. The abnormality probability of the empty plan starting at state S is zero plus the best completion. In general, the abnormality probability will be the abnormality probability of the plan up to the current state plus the abnormality probability of the best completion. Since we are uncertain about the completion, its probability of abnormality will be an interval composed of a lower and upper bound. This makes the total abnormality probability an interval as well:

$$I(p_{I \to S_n}) = [\ \Pr(ab(p_{I \to S_n})) + L_{S_n},\ \Pr(ab(p_{I \to S_n})) + U_{S_n})] \quad (7)$$

Recall, the best plan is the one whose *total* failure probability is T. In the persistent fault case, $T = 0.5$. Given an interval describing bounds on the total abnormality probability of a plan, $I(p_{I \to S_n})$ we can construct an interval describing on how close the abnormality probabilities will be to T:

$$|T - I(p_{I \to S_n})| \quad (8)$$

The absolute value folds the range around T. If the estimated total abnormality probability of the plan straddles target probability T, then the interval $|T - I(p_{I \to S_n})|$ straddles zero and the interval will range from zero to the absolute max of $I(p_{I \to S_n})$.

Let us define our search heuristic $F(p_{I \to S_n}) = min(|T - I(p_{I \to S_n})|)$. The function F has some interesting properties: Whenever the predicted total plan abnormality probability lies between L and U, F is zero. Intuitively, it is possible that there exists whose abnormality probability exactly achieves probability T. In all cases $F(p_{I \to S_n})$ represents the closest any plan going through state S_n can come to the target abnormality probability exactly T.

In our search we have a whole set of partial plans $\mathcal{P} = \{p_{I \to S_1}, p_{I \to S_2}, \ldots, p_{I \to S_n}\}$. For each partial plan, we evaluate $F(p_{I \to S_n})$ and expand the plan with the lowest value. Since $F(p_{I \to S_n})$ is an underestimate, A* search using this estimate will return the most informative plan that achieves production goals.

5.2 Improving Efficiency by Search Space Pruning

In the previous section we have described an A* like search algorithm to find highly informative diagnostic plans. In many cases the search heuristic returns the same value, namely zero. This leads to little guidance. In this section we introduce several methods to focus the search.

The first method prunes out dominated parts of the search space. Consider an abnormality probability interval for a partial plan $I(p_{I \to S_n})$ that does *not* straddle the target value T. The best possible plan in this interval will be on one of the two boundaries of the interval, whichever one lies closest to the target value T. Let $L_{I(p_{I \to S_n})}$ and $U_{I(p_{I \to S_n})}$ be the lower and upper bound of the abnormality probability interval $I(p_{I \to S_n})$. Then

$$V_{p_{I \to S_n}} = \min(|L_{I(p_{I \to S_n})} - T|, |U_{I(p_{I \to S_n})} - T|) \quad (9)$$

will be the value of the best plan in the interval. Plan $p_{I \to S_i}$ will dominate every plan $p_{I \to S_j}$ where $V_{p_{I \to S_i}} < V_{p_{I \to S_j}} \ \wedge \ T \notin I(p_{I \to S_j})$. We can prune out all dominated plans from the A* search frontier.

The next method is used to intelligently break ties in the heuristic value. The heuristic value determines which node will be expanded next. It is possible that two or more nodes will receive the same value so that we might need a tie-breaking rule to decide which node should be expanded first. The simplest tie-breaking rule would be to pick a node randomly, but we can do better than that.

$V_{p_{I \to S_n}}$ represents a guaranteed lower bound on a total plan $p_{I \to G}$ starting with the partial plan $p_{I \to S_n}$ as prefix. Recall, the upper and lower bounds are realizable, but none of the interior points of the interval are guaranteed to exist. Therefore comparing the V's enables us to decide which of two partial plans has the closest realizable solution. If two partial plans are also identical in this parameter, the information gain is the same, then we prefer the partial plan with less likelihood to fail. We put these two ideas together in a sequential decision procedure. Let $p_{I \to S_1}$ and $p_{I \to S_2}$ be the two partial plans with the same minimum value, *i.e.* $F(p_{I \to S_1}) = F(p_{I \to S_2})$. Then we break the tie by choosing the first rule that applies from the following ordered list:

1. If $V_{p_{I \to S_1}} < V_{p_{I \to S_2}}$ then expand $p_{I \to S_1}$ first
2. If $V_{p_{I \to S_1}} > V_{p_{I \to S_2}}$ then expand $p_{I \to S_2}$ first
3. If $U_{I(p_{I \to S_1})} < U_{I(p_{I \to S_2})}$ then expand $p_{I \to S_1}$ first
4. If $U_{I(p_{I \to S_1})} > U_{I(p_{I \to S_2})}$ then expand $p_{I \to S_2}$ first
5. If $L_{I(p_{I \to S_1})} < L_{I(p_{I \to S_2})}$ then expand $p_{I \to S_2}$ first
6. If $L_{I(p_{I \to S_1})} > L_{I(p_{I \to S_2})}$ then expand $p_{I \to S_1}$ first
7. otherwise pick randomly

6 Experiments

To evaluate the practical benefits of pervasive diagnosis, we implemented the heuristic search. We combined

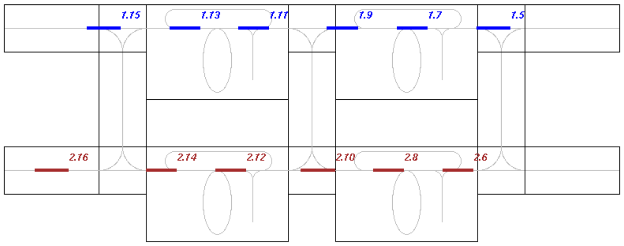

Figure 4: A schematic of a modular printing press used in the experiments. The press consists of 4 print engines (large rectangles) connected by controllable paper handling modules. Sheets enter on the left and exit on the right. There are 48 actions controlling feeders, paper paths, print engines and finisher trays.

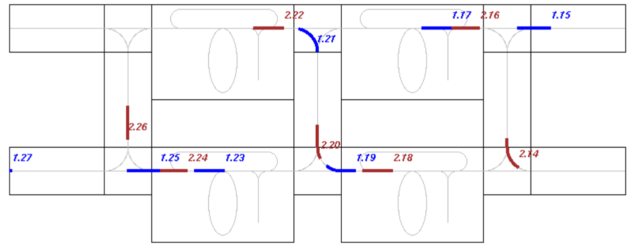

Figure 5: The flexibility of the architecture can be exploited to choose paper paths that use different subsets of components. A sequence of these paths can be used to isolate the fault.

it with an existing model-based planner and diagnosis engine and tested the combined system on a model of a modular digital printing press [16] or [6]. Multiple pathways allow the system to parallelize production, use specialized print engines for specific sheets (spot color) and reroute around failed modules. A schematic diagram showing the paper paths in the machine appears in Figure 4.

We do a test run for each possible abnormal action. The planner then receives a job from the queue. It then sends the plan to a simulation of the printing press. The simulation models the physical dynamics of the paper moving through the system. Plans that execute on this simulation will execute unmodified on our physical prototype machines in the laboratory. The simulation determines the outcome of the job. If the job is completed without any dog ears (bent corners) or wrinkles and deposited in the requested finisher tray, we say the plan succeeded, or in the language of diagnosis, the plan was not abnormal, otherwise the plan was abnormal.

The original plan and the outcome of executing the plan are sent to the diagnosis engine. The engine updates the over hypothesis probabilities. When a fault occurs, the planner greedily searches for the most informative plan. Since there is a delay between submitting a plan and receiving the outcome, we plan production jobs from the job queue without optimizing for information gain until the outcome is returned. This keeps productivity high.

We evaluate performance of passive diagnosis (only normal operation), explicit diagnosis (alternates diagnosis and regular operation) and pervasive diagnosis (regular operation modified to obtain additional diagnostic information). In the case of explicit diagnosis, the planner solely focuses on the needs of the diagnosis engine and thus creates plans hat maximize information gain in regards to the fault hypotheses. In general these plans need not be production plans. In case of passive diagnosis, the planner is not influenced by the diagnosis engine at all, that is, plans will be solely optimized in regards to production. In pervasive diagnosis, the plan is biased to have an outcome probability closest to the target T. As shown in Figure 5, the bias can create paths capable of isolating faults in specific actions.

In our model we assume, that under normal conditions the machine produces output at its nominal rate r_{nom} and that diagnosis efforts begin once some abnormal outcome is observed (i.e. a paper jam, dog ear, etc.). This outcome is the result of some single faulty module within the system. If this module contributes to some plan, it will exhibit aberrant behavior with some probability q, the so called *intermittency rate*, resulting in plan failure. In short we assume single, intermittent, catastrophic faults. Here we use a simple cost model of opportunity costs in terms of unrealized production due to efforts of isolating the faulty component (diagnosis costs) and exchanging this component (repair costs). The cost in this model represent the expected total amount of lost production due to the fault.

The repair cost model is rather simple: The technician receives a list of the printer modules in decreasing order of their fault probability. He or she will then follow a very simple procedure: He or she will step through the list, exchange the next module and test the machine until this test shows the machine working properly. We assume each step takes a constant amount of time and thus a constant cost. Using this model, repair costs can easily be estimated by Equation 10, with exchange cost $c_{exc} = r_{nom} * t_{exc}$.

$$c^t_{rep} = c_{exc} \sum_{i=1}^{|\Pr^t|} i * p^t_i \qquad (10)$$

The cost model for the response cost depends on the diagnosis cost model and the repair cost model as shown in Equation 11.

$$c^{t,policy}_{response} = c^t_{rep} + c^{t,policy}_{diag} \qquad (11)$$

At the moment the unexpected behavior is first observed the cost is 1 (lost unit of production) and the belief state is an uniform distribution over all fault hypothesis for all approaches. With the passive approach, the machine will continue to produce at rate r_{nom}, so diagnostic cost consists solely of the faulty output as shown in Equation 12.

$$c^{t,pass}_{diag} = 1 + n_{faulty} \qquad (12)$$

For explicit diagnosis, we assume no production during diagnosis (reasonable as chances of producing the right output are negligible), thus diagnosis cost is calculated by Equation 13.

$$c_{diag}^{t,exp} = 1 + r_{nom} * t_{diag} \tag{13}$$

Finally for pervasive diagnosis we assume that the machine produces at a reduced rate $r_{perv} \leq r_{diag}$ during diagnosis. Furthermore products can be faulty during production, thus the cost is calculated by Equation 14. Note that in our model the response cost represent the production lost due to the fault and therefore the goal is to minimize response cost.

$$c_{diag}^{t,per} = 1 + n_{faulty} + (r_{nom} - r_{perv}) * t_{diag} \tag{14}$$

In our experiments we set the exchange time for a single module to 10 minutes. The exchange of a single module causes us to halt only one print engine at the time. Due to the fact that the underlying system operates with four print engines we set $t_{exc} = 150 sec$. The nominal rate of the system is $r_{nom} = 3.1 sheets/sec$. Our experiments have shown that the reduced rate of pervasive diagnosis is $r_{perv} = 1.9 sheets/sec$.
Based on the introduced model we compare the performance of passive diagnosis, explicit diagnosis and pervasive diagnosis for three levels of fault intermittency represented by the probability q. When $q = 1$, a faulty action always causes the plan to fail. When $q = 0.01$ a faulty action only causes the plan to fail with a statistical mean of 1/100. The summary of the experimental results are in Table 2. A more detailed visualization of the results is presented in Figure 6 for the intermittency rate $q = 0.01$, in Figure 7 for the intermittency rate $q = 0.1$, and in Figure 8 for the intermittency rate $q = 1$. We averaged experiments over 100 runs to reduce statistical variation.

q=0.01	min. response cost	Time	# of exch. mod.
Passive	1010.22	214.1	2.01
Explicit	947.25	76.6	1.41
Pervasive	768.80	204.6	1.01
q=0.1	min. response cost	Time	# of exch. mod.
Passive	1055.25	35.0	2.10
Explicit	591.31	29.1	1
Pervasive	547.77	37.2	1
q=1.0	min. response cost	Time	# of exch. mod.
Passive	1012.00	7.78	2.01
Explicit	515.43	3.1	1
Pervasive	509.76	3.78	1

Table 2: Pervasive diagnosis has the lowest rate of lost production. (# of exch. mod.: expected # of exchanged modues at the minimal response cost)

Figure 6, 7, and 8) show the expected costs of repair, the cost of diagnosis and their sum over time in terms of lost production in relation to a healthy machine. We plot these costs for fault intermittency rates of 1, 0.1 and 0.01. Cost of repair is computed by estimating the repair time based on the current probability distribution over the fault hypothesis and pricing this downtime according to the nominal machine production rate. Cost

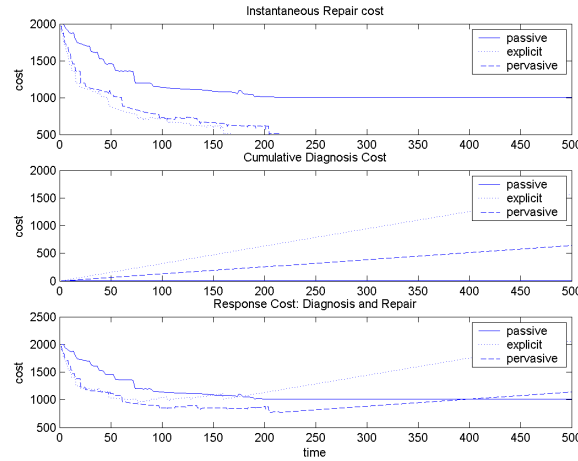

Figure 6: Experimental results for the intermittent rate $q = 0.01$ averaged over 100 runs.

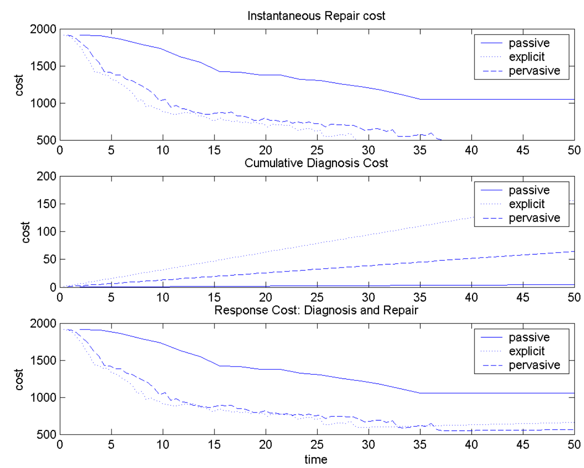

Figure 7: Experimental results for the intermittent rate $q = 0.1$ averaged over 100 runs.

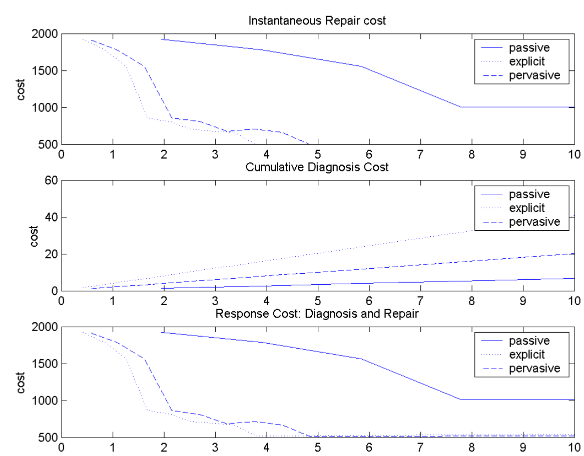

Figure 8: Experimental results for the intermittent rate $q = 1$ averaged over 100 runs.

of diagnosis at time t is the accumulated production deficit in relation to a healthy machine producing at its nominal rate r_{nom}. The x-axis is the amount of time (relative to the first occurrence of the fault) after which one chooses to stop diagnosis and start repairing the machine. The minimum of the sum these costs denotes the optimal point in time (relative to first occurrence of the fault) to switch from diagnosis to repair and gives the minimal expected total loss of production due to the fault.

In all our experiments we found the optimal response costs of pervasive diagnosis to be below those of the other two approaches. As expected the respective optimal durations of diagnosis processes are in the order (shortest to longest) of explicit, pervasive and passive. This corresponds to the fact that explicit diagnosis focuses solely on the diagnosis task. Therefore explicit diagnosis is able to select plans with maximal diagnosis information gain and can isolate the faulty component in the shortest amount of time. However, due to the high production loss (production is halted), explicit diagnosis does not result in minimal response costs. Passive diagnosis has the lowest rate of lost production, but incurs the highest expected repair costs due to its lower quality diagnosis. This corresponds to the fact that the plans executed during passive diagnosis are optimized for production regardless of diagnosis needs. Pervasive diagnosis intelligently integrates diagnosis goals into production plans by using planning flexibility. This leads to a lower minimal total expected production loss in comparison to passive and explicit diagnosis.

7 Discussion

In the body of this paper we presented an application of pervasive diagnosis to a printing domain. We believe the technique generalizes to a wide class of production manufacturing problems in which it is important to optimize efficiency but the cost of failure for any one job is low compared to stopping the production system to perform explicit diagnosis.

The application that we have developed illustrates a single fault, single appearance, independent fault instantiation of the pervasive diagnosis framework. To generalize the instantiation we can reduce the set of assumptions by incorporating a different representation of the hypotheses space, the belief model and the belief update. In the most general case, multiple intermittent faults with multiple action appearance, the construction of the heuristic directly extends to an on-line forward heuristic computation similar to the one used in the FF planning system [9]. However, the idea of pervasive diagnosis is not limited to a probability based A^* search. Another possibility of instantiation could be an SAT-solver approach in which the clauses represent failed plans and each satisfying assignment is interpreted as a valid diagnosis.

8 Conclusions

The idea of Pervasive Diagnosis opens up new opportunities to efficiently exploit diagnostic information for the optimization of the throughput of model-based systems. Hard to diagnose intermittent faults which would have required expensive production stoppages can now be addressed on line during production. While pervasive diagnosis has interesting theoretical advantages, we have shown that a combination of heuristic planning and classical diagnosis can be used to create practical real time applications as well.

References

[1] M. F. Ali, A. Veneris, S. Safarpour, M. Abadir, R. Drechsler, and A. Smith. Debugging sequential circuits using boolean satisfiability. In *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 204–209, 2004.

[2] M. L. Bushnell and V. D. Agrawal. *Essentials of Electronic Testingfor Digital, Memory and Mixed-Signal VLSI Circuits*. 2000.

[3] J. de Kleer and B. C. Williams. Reasoning about multiple faults. In *Proceedings of the National Conference on Artificial Intelligence (AAAI86)*, pages 132–139, 1986.

[4] R. Dearden and D. Clancy. Particle filters for real-time fault detection in planetary rovers. In *Proceedings of the Thirteenth International Workshop on Principles of Diagnosis*, pages 1–6, 2002.

[5] M. Do, W. Ruml, and R. Zhou. On-line planning and scheduling: An application to controlling modular printers. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, 2008, to appear.

[6] M. B. Do and W. Ruml. Lessons learned in applying domain-independent planning to high-speed manufacturing. In *ICAPS*, pages 370–373, 2006.

[7] M. P. Fromherz. Planning and scheduling reconfigurable systems with regular and diagnostic jobs. US Patent 7233405, 2007.

[8] M. P. Fromherz, D. G. Bobrow, and J. de Kleer. Model-based computing for design and control of reconfigurable systems. *The AI Magazine*, 24(4):120–130, 2003.

[9] P. Haslum and H. Geffner. Heuristic planning with time and resources. In *Proceedings of ECP-01*, 2001.

[10] Y.-W. Kim, G. Rizzoni, and V. Utkin. Automotive engine diagnosis and control via nonlinear estimation. *IEEE Control Systems*, pages 84–98, 1998.

[11] P. P. Lin and X. Li. Fault diagnosis, prognosis and self-reconfigur=ation for nonlinear dynamic systems using soft computing. In *IEEE International Conference on Systems, Man, and Cybernetics*, Taipei,Taiwan, 2006.

[12] D. Poole. Representing diagnostic knowledge for probabilistic horn abduction. In *International Joint Conference on Artificial Intelligence (IJCAI91)*, pages 1129–1135, 1991.

[13] G. Provan and Y.-L. Chen. Model-based diagnosis and control reconfiguration for discrete event systems: An integrated approach. In *Proceedings of the thirty-eighth Conference on Decision and Control*, pages 1762–1768, Phoenix, Arizona, 1999.

[14] H. E. Rauch. Autonomous control reconfiguration. *IEEE Control Systems Magazine*, 15(6):37–48, 1995.

[15] R. Reiter. A theory of diagnosis from first principles. In *Readings in Model-Based Diagnosis*, pages 29–48, 1992.

[16] W. Ruml, M. B. Do, and M. Fromherz. On-line planning and scheduling for high-speed manufacturing. In *Proc. of ICAPS-05*, pages 30–39, 2005.

[17] V. Verma, G. Gordon, R. Simmons, and S. Thrun. Real-time fault diagnosis: Tractable particle filters for robot fault diagnosis. *IEEE Robotics and Automation Magazine*, pages 56–66, 2004.

[18] C. Zhong and P. Li. Bayesian belief network modeling and diagnosis of xerographic systems. In *Proceedings of the ASME Symposium on Controls and Imaging - IMECE*, Orlando, Florida, 2000.