Hybrid Systems Diagnosis

Sheila McIlraith[1], Gautam Biswas[2], Dan Clancy[3], and Vineet Gupta[3]

[1] Knowledge Systems Lab, Stanford University, Stanford, CA 94305
[2] Computer Science Department, Vanderbilt University, Nashville, TN 37212
[3] Caelum Research Corporation, NASA Ames Research Center, Moffett Field, CA 94035

Abstract. This paper reports on an on-going project to investigate techniques to diagnose complex dynamical systems that are modeled as hybrid systems. In particular, we examine continuous systems with embedded supervisory controllers that experience abrupt, partial or full failure of component devices. We cast the diagnosis problem as a model selection problem. To reduce the space of potential models under consideration, we exploit techniques from qualitative reasoning to conjecture an initial set of qualitative candidate diagnoses, which induce a smaller set of models. We refine these diagnoses using parameter estimation and model fitting techniques. As a motivating case study, we have examined the problem of diagnosing NASA's Sprint AERCam, a small spherical robotic camera unit with 12 thrusters that enable both linear and rotational motion.

1 Introduction

The objective of our project has been to investigate how to diagnose hybrid systems – complex dynamical systems whose behavior is modeled as a hybrid system. Hybrid models comprise both discrete and continuous behavior. They are typically represented as a sequence of piecewise continuous behaviors interleaved with discrete transitions (e.g., [7]). Each period of continuous behavior represents a so-called *mode* of the system. For example, in the case of NASA's Sprint AERCam, modes might include *translate_X-axis*, *rotate_X-axis*, *translate_Y-axis*, etc. [1]. In the case of an Airbus fly-by-wire system, modes might include *take-off*, *landing*, *climbing*, and *cruise*. Mode transitions generally result in changes to the set of equations governing the continuous behavior of the system, as well as to the state vector that initializes that behavior in the new mode. Discrete transitions that dictate mode switching are modeled by finite state automata, temporal logics, switching functions, or some other transition system, while continuous behavior within a mode is modeled by, e.g., ordinary differential equations (ODEs) or differential and algebraic equations (DAEs).

The problem we address in this paper is how to diagnose such hybrid systems. For the purposes of this paper, we consider the class of hybrid systems that are continuous systems with an embedded supervisory controller, but whose hybrid models contain no autonomous jumps. I.e., all nominal transitions between system modes are induced by a controller action, none are induced by the system state and model [7]. The class of systems we consider can be modeled as a composition of a set of component subsystems, each of which is itself a hybrid system. We assume that the system operation is being tracked by a monitoring and observer system (e.g., [19]) that ensures that the system behavior predicted by the model does not deviate significantly from the observed

behavior in normal system operation. When observations occur outside this range, the behavior is deemed to be aberrant and diagnosis is initiated. In this paper, we consider faults whose onset is abrupt, and which result in partial or complete degradation of component behavior. The general problem we wish to address can be stated as follows: *Given a hybrid model of system behavior, a history of executed controller actions, a history of observations, including observations of aberrant behavior relative to the model, isolate the fault that is the cause for the aberrant behavior.* Diagnosis is done online in conjunction with the continued operation of the system. Hence, we divide our diagnosis task into two stages, initial conjecturing of candidate diagnosis and subsequent refinement and tracking to select the most likely diagnoses.

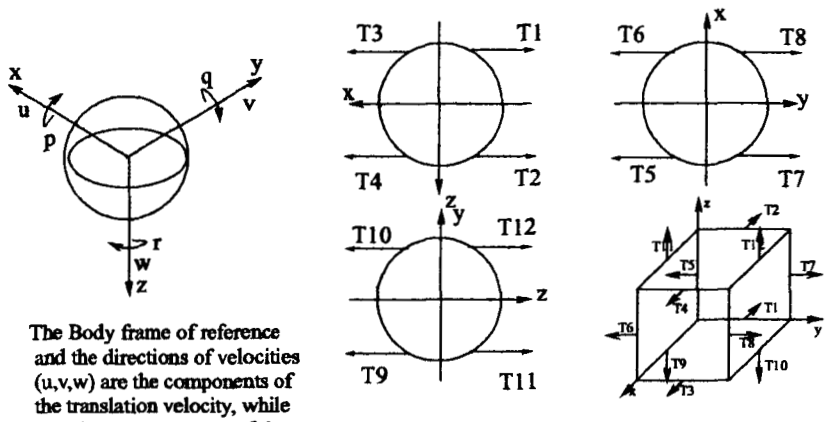
In this paper we conceive the diagnosis problem as a model selection problem. The task is to find a mathematical model and associated parameter values that best fit the system data. These models dictate the components of the system that have malfunctioned, their mode of failure, the estimated time of failure and any additional parameters that further characterize the failure. To address this diagnosis problem, we propose to exploit AI techniques for qualitative diagnosis of continuous systems to generate an initial set of qualitative candidate diagnoses and associated models, thus drastically reducing the number of potential models for our system. This is followed by parameter estimation and model fitting techniques to select the most likely mode and system parameters for candidate models of system behavior, given both past and subsequent observations of system behavior and controller actions. The main contributions of the paper are: 1) formulation of the hybrid diagnosis problem; 2) the exploitation of techniques for qualitative diagnosis of continuous systems to reduce the diagnosis search space; and 3) the use of parameter estimation and data fitting techniques for evaluation and comparison of candidate diagnoses.

In Section 2 we provide a brief description of NASA's Sprint AERCam, which we have used as a motivating example and which we will use to illustrate certain concepts in this paper. In Section 3 we present a formal characterization of the class of hybrid systems we study and the diagnosis problem they present. In Section 4 we describe our approach to hybrid diagnosis and the algorithms we use to achieve hybrid diagnosis. The generation of initial candidate qualitative diagnoses is described in Section 4.1, and the subsequent quantitative fitting and tracking of candidate diagnoses and their models is described in Section 4.2. In the final two sections, we briefly discuss related work and summarize our contributions.

2 Motivating Example: The AERCam

We are using NASA's Sprint AERCam and a simulation of system dynamics and the controller written in Hybrid CC (HCC) as a testbed for this work. We describe the dynamic model of the AERCam system briefly, a more detailed description of the model and simulation appear in [1].

The AERCam is a small spherical robotic camera unit, with 12 thrusters that allow both linear and rotational motion (Fig. 1). For the purposes of this model, we assume the sphere is uniform, and the fuel that powers the movement is in the center of the sphere. The fuel depletes as the thrusters fire.



The Body frame of reference and the directions of velocities (u,v,w) are the components of the translation velocity, while (p,q,r) are components of the angular velocity.

Three views of the AERCam, showing the thrusters, and showing all the thrusters together in the cube circumscribing the AERCam.

Fig. 1. The AERCam axes and thrusters

The dynamics of the AERCam are described in the AERCam body frame of reference. The translation velocity of this frame with respect to the shuttle inertial frame of reference is 0. However, its orientation is the same as the orientation of the AERCam, thus its orientation with respect to the shuttle reference frame changes as the AERCam rotates (i.e., it is not an inertial frame). The twelve thrusters are aligned so that there are four along each major axis in the AERCam body frame. For modeling purposes, we assume the positions of the thrusters are on the centers of the edges of a cube circumscribing the AERCam. Thus, for example, thrusters T_1, T_2, T_3, T_4 are parallel to the x-axis and are used for translation along the x-axis or rotation around the y-axis. I.e., firing thrusters T_1 and T_2 results in translation along the positive x-axis, and firing thrusters T_1 and T_4 results in a negative rotation around the y-axis. AERCam operations are simplified by limiting them to either translation or rotation. Thrusters are either on or off, therefore, the control actions are discrete. In a normal mode of operation, only two thrusters are on at any time.

2.1 AERCam dynamics

A simplified model of the AERCam dynamics based on Newtonian laws is derived using an inertial frame of reference fixed to the space shuttle. The AERCam position in this frame is defined as the triple (x, y, z). Let $\vec{\mathbf{V}}$ be the velocity in the AERCam body frame, with its vector components given by (u, v, w). The frame rotates with respect to the inertial reference frame with velocity $\omega = (p, q, r)$, the angular velocity of the AERCam. The rotating body frame implies an additional Coriolis force acting upon the AERCam. We assume uniform rotational velocity since in the normal mode of opera-

tion, the AERCam does not translate and rotate at the same time [2, pg. 130]. Similar equations can be derived for the rotational dynamics [1].

$$d(m\,\vec{\mathbf{V}})/dt = \vec{\mathbf{F}} - 2m(\vec{\mathbf{V}} \times \vec{\omega}) \quad \text{Newton's Law}$$

$$\vec{\mathbf{V}}\, dm/dt + m d(\vec{\mathbf{V}})/dt = \vec{\mathbf{F}} - 2m(\vec{\omega} \times \vec{\mathbf{V}})$$

The resultant equation for each coordinate:

$$du/dt = F_x/m - 2(qw - vr) - (u/m) * dm/dt$$
$$dv/dt = F_y/m - 2(ru - pw) - (v/m) * dm/dt$$
$$dw/dt = F_z/m - 2(pv - qu) - (w/m) * dm/dt$$

2.2 Position Control Mode of the AERCam

In the position control mode, the AERCam is directed to go to a specified position and point the camera in a particular direction. Assume the AERCam is at position A and directed to go to position B. In the first phase, the AERCam rotates to get one set of thrusters pointed towards B. These are then fired, and the AERCam cruises towards B. Upon reaching a position close to B, it fires thrusters to converge to B, and then rotates to point the camera in the desired direction.

To facilitate the illustration of the diagnosis problem, we use a simple trapezoidal controller, which we explain in two dimensions. Suppose the task is to travel along the x-axis for some distance, then along the y-axis. Such manoeuvres are needed for navigating in the space shuttle. In order to do this, the AERCam fires its x thrusters for some time. Upon reaching the desired velocity, these are switched off. When the AERCam has reached a position close to the desired x position, the reverse thrusters are switched on, and the AERCam is brought to a halt — the velocity graph is a trapezium. The process is analogous for the y direction.

3 Problem Formulation

In this section we provide our formulation of the hybrid diagnosis problem.

Definition 1 (Hybrid System). A hybrid system is a 5-tuple $\langle \mathcal{M}, X, \mathcal{F}, \Sigma, \phi \rangle$, where

- $\mathcal{M}$, finite set of system modes $(\mu_1, \ldots, \mu_k)$.
- $X \subseteq R^n$, continuous state variables. $x(t)$ is the continuous behavior at time t.
- $\mathcal{F}$, finite set of functions $\{f_{\mu_1}, \ldots, f_{\mu_k}\}$, and associated parameter values θ such that for each mode, μ_i, $f_{\mu_i}(t, \theta, x(t)) : R \times R \times X \rightarrow X$ defines the continuous behavior of the system in μ_i.[1]
- Σ, finite set of actions $(\sigma_1, \ldots, \sigma_l)$, which transition the system between modes.
- ϕ, transition function which maps an action, mode and system state vector into a new mode and initial state vector, i.e., $\phi : \Sigma \times \mathcal{M} \times X \rightarrow \mathcal{M} \times X$.

To define the hybrid diagnosis problem, we augment Definition 1 as follows.

[1] Parameter value ranges may be associated with θ.

Definition 2 (Diagnosable Hybrid System). A diagnosable hybrid system, $\langle \mathcal{M}, X, \mathcal{F}, \Sigma, \phi, COMPS \rangle$ is a hybrid system comprised of m potentially malfunctioning components $COMPS = (c_1, \ldots, c_m)$ where

- For each $\mu \in \mathcal{M}$, μ includes a designation of whether each $c_i \in COMPS$ is operating normally, or abnormally, i.e., $(\neg)ab(c_i)$.
- We assume that transitions to fault modes are achieved by exogenous actions. Hence, $\Sigma = \Sigma_c \cup \Sigma_e$, where
  - Σ_c is a finite set of controller actions, and
  - Σ_e is a finite set of exogenous actions.
- $\mathcal{A}$, the controller action history, the sequence of time-indexed controller actions performed.
- $X_{obs} \subseteq X$, continuous state variables that are observable. $x_{obs}(t)$ is the observations at time t.
- $\mathcal{O}$, the observation history, the sequence of time-indexed observations.

For notational convenience, μ_F denotes a faulty mode, i.e., a mode for which at least one $c_i \in COMPS$ is $ab(c_i)$ in μ_F. θ_F denotes the parameters associated with f_{μ_F}.

In the case of the AERCam example, the potentially malfunctioning components are the 12 thrusters, and a mode μ includes the behavior mode (e.g., translate-x, translate-y, rotate-x, etc.) and $(\neg)ab(T_i), i = 1, \ldots .12$, for each thruster. The continuous state vector includes the x, y, z position of the AERCam, velocity and acceleration. The parameter values, θ associated with each f_μ are the percentage degradation of each of the thrusters.

Definition 3 (Model). A model, Mod of a diagnosable hybrid systems is a time-indexed mode sequence and associated parameter values $([\mu_1, \ldots, \mu_m], [\theta_1, \ldots, \theta_m])$

Notice that each model of the system, $(\boldsymbol{\mu}, \boldsymbol{\theta})$ induces a corresponding time-indexed piecewise continuous sequence of functions $[f_{\mu_1}, \ldots, f_{\mu_m}]$ dictating system behavior.

In this paper we make several simplifying assumptions regarding our diagnosis task. In particular, we make a single-time fault assumption. We assume that our systems do not experience multiple sequential faults. Further, we assume that faults are abrupt, resulting in partial or full degradation of component behavior. We cast the hybrid diagnosis task as the problem of finding the most likely model for the observation history, $P(Mod \mid \mathcal{O})$. I.e, the sequence of modes and parameter values $(\boldsymbol{\mu}, \boldsymbol{\theta})$ that best fit the observations over time. Under normal operation, the model of the system Mod_{normal} is fully dictated by the sequence of controller actions $\mathcal{A}$ and the nominal parameter values, θ. Once again, we assume that the system operation is being tracked by a monitoring and observer system (e.g., [19]) that ensures that the system behavior predicted by the model does not deviate significantly from the observed behavior in normal system operation. When observations occur outside this range, the behavior is deemed to be aberrant and diagnosis is initiated. Given a diagnosable hybrid system $\langle \mathcal{M}, X, \mathcal{F}, \Sigma, \phi, COMPS \rangle$, a controller action history, $\mathcal{A}$ and a history of observations, $\mathcal{O}$ which includes observations of aberrant behavior, the **hybrid diagnosis task** is to determine what components are faulty, what fault mode caused the aberrant behavior, when it occurred, and what the values of the parameters associated with the fault mode are. In the AERCam system, a diagnosis might be that thruster T_1 experienced a blockage fault of 50%, at time t_i.

Once Mod_{normal} has been rejected, we must find a new most likely model from among the potentially exponential (in $COMPS$) number of mode sequences, occurring within a large but bounded time range. We propose to exploit previous research on temporal causal graphs for qualitative diagnosis of continuous systems [18], to compute a set of candidate qualitative diagnoses that are consistent with our system, in order to identify a preliminary subset of candidate models, whose likelihood can be estimated.

Definition 4 (D-tuple). A D-tuple is a 4-tuple $\langle C, \mu_F, t_F, \theta_F \rangle$, where μ_F is a fault mode, t_F is the time the fault mode commenced, θ_F is the parameter values associated with the fault mode behavior, and C is the set of failed (abnormal) components in μ_F.

Definition 5 (Candidate Qualitative Diagnosis). Given a diagnosable hybrid system with model $Mod = (\mu, \theta)$ an action history $\mathcal{A}$, and a history of observations, $\mathcal{O}$ which includes observations of aberrant behavior, D-tuple $\langle C, \mu_F, t_F, \theta_F \rangle$ is a candidate qualitative diagnosis iff there exists a range of parameter values $\theta_F = [\theta_l, \theta_u]$, and time range $t_F = [t_l, t_u]$ such that the occurrence of fault mode μ_F with parameter values θ_F in time range t_F is consistent with $\mathcal{O}$, $\mathcal{A}$ and Mod.

Hence, a candidate qualitative diagnosis stipulates a fault mode, including one or more faulty components. It also stipulates a lower and upper bound, $[t_l, t_u]$, on the time the fault mode occurred. This range generally corresponds to the start times of the controller induced modes preceding and following the fault, or up to the point the fault was detected. This candidate diagnosis induces an associated *candidate model*, $Mod_C = ([\mu_1, \ldots, \mu_i, \mu_F, \mu'_{i+1}, \ldots, \mu'_m], [\theta_1, \ldots, \theta_i, \theta_F, \theta'_{i+1}, \ldots, \theta'_m])$ corresponding to Mod with the fault mode μ_F and θ_F inserted at t_F. Every subsequent mode, $\mu_{i+1}, \ldots, \mu_m$, has $ab(c_i), c_i \in C$ enforced, and every subsequent set of parameters has the parameters associated with faulty components C enforced. Computing candidate qualitative diagnoses is discussed in Section 4.1.

Since each candidate qualitative diagnosis only conjectured ranges for the time of the fault mode, t_F and parameter values associated with the fault mode, θ_F, the associated candidate models are underconstrained. In Section 4.2, we discuss methods for estimating unique values for t_F and θ_F and for estimating a posterior probability for each of the candidate models, Mod_C, given $\mathcal{O}$.

Definition 6 (Candidate Diagnosis). Given a diagnosable hybrid system, a history of controller actions $\mathcal{A}$, and a history of observations $\mathcal{O}$, D-tuple $\langle C, \mu_F, t_F, \theta_F \rangle$ with associated model Mod_C is a candidate diagnosis for the hybrid system, iff $P(Mod_C \mid \mathcal{O}) > \alpha$, for defined threshold value $\alpha \in [0.1]$.

4 Diagnosing Hybrid Systems

In this section we discuss one method for computing hybrid diagnoses. In Section 4.1 we discuss a technique for generating candidate qualitative diagnoses, and their associated candidate models. In Section 4.2 we discuss techniques for model fitting and for model (and hence diagnosis) comparison. In particular we discuss techniques for estimating the parameters of the candidate models, and the likelihood of the models, and for

continued monitoring and refinement of the candidate models as the system continues to operate and observations continue to be made.

We illustrate these techniques with the following simple AERCam example. Consider the scenario depicted in Fig. 2. In the first accelerate phase, the AERCam is being powered by thrusters $T1$ and $T2$. Assume that at some point in this phase, a sudden leak in the $T2$ thruster causes an abrupt change in its output. As a consequence, the AERCam starts veering to the right of the desired trajectory, as illustrated by the left-most dotted lines in Fig. 2. (The other dotted lines represent other potential candidate diagnoses consistent with the point of detection of the failure.) Soon after this occurs, the supervisory controller commands the AERCam to turn off Thrusters $T1$ and $T2$ with the objective of getting the AERCam to cruise in a straight line. In the faulty situation, the AERCam has some residual angular velocity about the z-axis, so it continues to rotate in the cruise mode. Then the controller turns on thrusters $T3$ and $T4$, to decelerate the AERCam with the objective of bringing it to a halt. Again, this objective is not entirely achieved in the the faulty situation. Next, thrusters $T5$ and $T6$ are switched on, to move the AERCam in the y direction. However, since the AERCam is not in the desired orientation after the failure, the position error due to faulty thruster $T2$ accumulates causing a greater and greater deviation from the desired trajectory of the system. The position of the AERCam is being continuously sensed, filtered for noise and monitored. At some point within the y translation the trajectory exceeds the error bound, i.e., $P(Mod_{normal} < \alpha)$ and is flagged by the monitoring system as aberrant relative to Mod_{normal}. At this point, the diagnosis task begins.

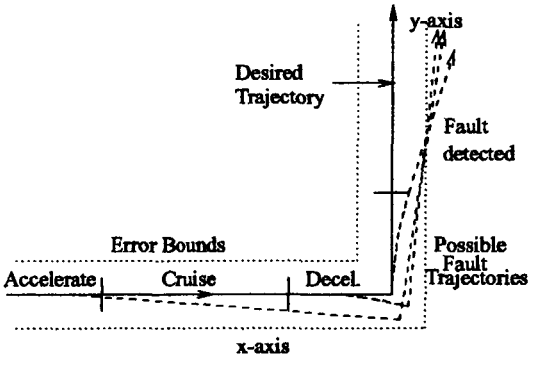


Fig. 2. Possible fault trajectories of AERCam (simplified for illustration purposes).

4.1 Qualitative Candidate Generation

Given the current system model $Mod = (\mu, \theta)$ (commonly Mod_{normal}), a history of controller actions $\mathcal{A}$, and a history of observations $\mathcal{O}$ including one or more observa-

tions of aberrant behavior, we wish to generate a set of *candidate qualitative diagnoses* $\langle C, \mu_F, t_F, \theta_F \rangle$, and associated *candidate models* as described in Definition 5. To do so, we extend techniques for generating qualitative diagnoses of continuous dynamic systems to deal with hybrid systems with multiple modes. The model and propagation mechanism, as applied to continuous systems diagnosis, is described in [18].

In the case of our AERCam example, the action history $\mathcal{A}$ is [(on($T1$), on($T2$)), (off($T1$), off($T2$)), (on($T3$), on($T4$)), (off($T3$), off($T4$), on($T5$), on($T6$)), (off($T5$), off($T6$))]; the model, Mod_{normal} is the time-indexed sequence [($accelerate\_x$, $\neg ab(T1-T12), \theta$), ($cruise\_x$, $\neg ab(T1-T12), \theta$),($decelerate\_x$, $\neg ab(T1-T12), \theta$), ($accelerate\_y$, $\neg ab(T1-T12), \theta$),($cruise\_y$, $\neg ab(T1-T12), \theta$)], where θ is a vector of length 12 all of whose entries are 0 (percent degradation in thrusters).

To generate candidate qualitative diagnoses we construct an abstract model of the dynamic system behavior, Mod_{normal} as a temporal causal graph. A part of the temporal causal graph for the AERCam dynamics is shown in Fig. 3. The graph expresses directed cause-effect relations between component parameters and the system state variables. Links between variables are labeled as: (i) $+1$, implying direct proportionality, (ii) -1, implying inverse proportionality, and (iii) $\int$, implying an integrating relation. An integrating relation introduces a temporal delay in that a change on the cause side of the relation affects the derivative of the variable on the effect side. This adds temporal characteristics to the relations between variables. Some edges are labeled by variables, implying the sign of the variable in the particular situation defines the nature of the relationship. The candidate generation algorithm is invoked for every initial instance of an

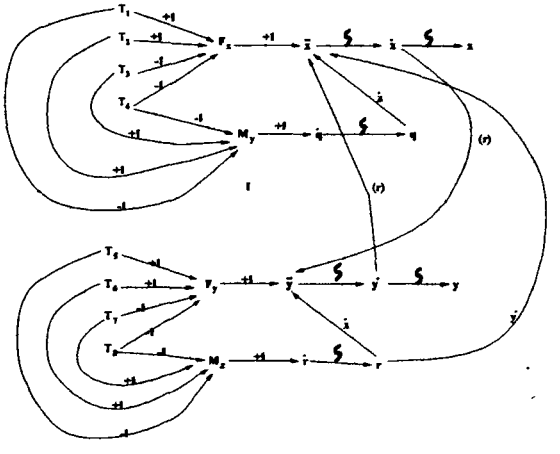


Fig. 3. A subset of the temporal causal graph showing the relations between Thrusters $T1 - T8$ and the x and y positions of the AERCam.

aberrant observation. The aberrant observation plus the controller action history $\mathcal{A}$ are input to a backward propagation algorithm that operates on the temporal causal graph.

The algorithm operates backwards from the last mode in the mode sequence of Mod:

Step 1 For the current mode, extract the corresponding temporal causal graph model, and apply the *Identify Possible Faults* algorithm. Details of this algorithm are presented in [18], but the key aspect of this algorithm is to propagate the aberrant observation expressed as a $\pm$ value, backward depth-first through the graph. For example, given that the $y-$position of the AERCam has deviated $-$ (i.e., below normal), backward propagation implies $d(y)/dt$ is $-$, and so on, till we get $T5^-$ and $T6^-$, implying thrusters $T5$ and $T6$ are possibly faulty with decreased thrust performance. Propagation along a path can terminate if conflicting assignments are made to a node. The goal is to systematically propagate observed discrepancies backward to identify all possible candidate hypotheses that are consistent with the observations. In our example, the component parameters, $COMPS = \{T1, \ldots, T12\}$ form the space of candidate faults.

Step 2 Repeat Step 1 for every mode in the mode sequence, to μ_1. The system model needs to be substituted as the algorithm traverses the mode sequence backwards. Therefore, back propagation will be performed on a different temporal causal graph for each mode in the controller history[2].

The output of this step is a set of qualitative diagnoses $\langle C, \mu_F, t_F, \theta_F \rangle$, each with an associated candidate model, as described in Section 3. Returning to our AERCam example, three qualitative candidate diagnoses are generated. The first candidate diagnosis is that $T2$ failed in the x acceleration phase. The time of the fault mode transition is $[t_1, t_2]$, and the parameters associated with the failure – the percentage degradation of the component is in the range $[0, 100]$. So the first candidate qualitative diagnosis is $\langle T2, (accelerate\_x, ab(T2), \neg ab(T1, T3 - T12), \theta_F), [t_1, t_2], [0, 100] \rangle$. The candidate model simply has $(accelerate\_x, ab(T2), \neg ab(T1), \neg ab(T3 - T12))$ inserted after the mode $(accelerate\_x, \neg ab(T1 - T12))$, and $ab(T2)$ enforced in every subsequent mode. The second candidate qualitative diagnosis is that $T4$ failed in the deceleration phase of x translation, i.e., $\langle T4, (decelerate\_x, ab(T4), \neg ab(T1 - T3, T5 - T12), \theta_F), [t_3, t_4], [0, 100] \rangle$. The third candidate is that $T6$ failed during y acceleration, i.e., $\langle T6, (accelerate\_y, ab(T6), \neg ab(T1 - T5, T7 - T12), \theta_F), [t_4, t_D], [0, 100] \rangle$, where t_D is the time of detection of the aberrant behavior. In each case θ_F is a vector of length 12 with every entry equal to 0 (percentage degradation), except the entries corresponding to the faulty thrusters, C which will have the range $[0, 100]$.

4.2 Model Fitting and Comparison

Given the candidate qualitative diagnoses and their associated candidate models, the next phase of the diagnosis process is quantitative refinement of the qualitative candidate diagnoses and their associated models through parameter estimation and data fitting, followed by tracking of the fit of subsequent observations to the candidate models. The goal is to at least provide a probabilistic ranking of the plausible candidates, if not a unique model (and hence diagnosis).

[2] We may cut off back-propagation along the mode sequence beyond a time limit.

As observed in the previous section, the model associated with the candidate qualitative diagnosis, Mod_C is underconstrained. Both the time of the fault mode occurrence, t_F and the parameters associated with the faulty behavior θ_F are represented as ranges and must be estimated. Further, the candidate qualitative diagnoses were generated from initial observations of aberrant behavior, and their consistency can be further evaluated by monitoring the qualitative transients associated with each candidate. The refinement process is performed by a set of *trackers* [21], one for each candidate diagnosis and associated model. Each tracker comprises both a *qualitative transient analysis* component and a quantitative *model estimation*, component. The two components operate in parallel as described below.

Qualitative Transient Analysis
The qualitative transient analysis component performs a further qualitative analysis of the consistency of candidate qualitative diagnoses based on monitoring of higher-order transients whose manifestation is seen over a longer period of time. If the transients of a candidate qualitative diagnosis do not remain consistent with subsequent observations, the candidate diagnosis will be eliminated and the *model estimation* component informed. The technique we employ is derived from techniques for qualitative monitoring of continuous systems. Details of the algorithm appear in [18].

Model Estimation
The purpose of the model estimation component is to perform quantitative model fitting, i.e., to provide a quantitative estimate of the parameters of the models and to assign a probability to each of the candidate models (and hence candidate diagnoses), given the noisy observed data. In particular, given a candidate model, Mod_C the model estimation component uses parameter estimation techniques to estimate both the time at which the failure occurred, t_F, and the value for the parameters, θ_F, associated with the conjectured failure mode. In this paper we discuss two alternate approaches to our time and parameter estimation problem. The first approach is based on Expectation Maximization (EM) (e.g., [8]), an iterative technique that converges to an optimal value for t_F and θ_F simultaneously. The second approach we consider employs General Likelihood Ratio (GLR) techniques (e.g., [5]) to estimate the time of failure t_F, and then uses the observations obtained after the failure to estimate the fault parameters, θ_F, by a least squares method. As described in Section 3, the outcome of both approaches is a unique value for t_F and θ_F and a measure of the likelihood of Mod_C given the observations. The proposed approaches to model fitting have trade-offs and we are currently assessing the efficacy of these and other alternative approaches through experimentation.

EM-Based Approach The Expectation Maximization (EM) algorithm (e.g., [8]) provides a technique for finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given set of data, when that data is incomplete or has missing values. The parameter estimation problem we address in this paper is a variant of the motion segmentation problem described in [24]. Here, we define the basic algorithm and the intuition behind our approach. (See [8] for more details.)

The time of failure, $t_F = [t_l, t_u]$ of our candidate qualitative diagnosisdictates the mode in which the failure is conjectured to have occurred. Let us call this mode μ_i. The behavior of our hybrid system in mode μ_i is described by the continuous function

f_{μ_i}, with *known* parameters θ_i. At some (to be estimated) time point t_F within the predicted time period of μ_i, we have conjectured that the system experienced a fault which transitions it into mode μ_F. The behavior of our hybrid system in mode μ_F is described by the continuous function f_{μ_F}, with *unknown* parameters, θ_F. We also have a set of data points $\mathcal{O}' = [x_{obs}(t_l), \ldots, x_{obs}(t_u)] \subseteq \mathcal{O}$, which either reflect the behavior of the system under f_{μ_i} or under f_{μ_F}.

Given all this information, our task is to find 1) values for parameters θ_F, and 2) an assignment of the data points $\mathcal{O}'$ to either μ_i or μ_F so that we maximize the fit of the data to the two functions. The assignment of data points will in turn tell us the value of t_F. EM provides an iterative algorithm which converges to provide a maximum-likelihood estimate for θ_F given $\mathcal{O}'$, i.e., roughly we are calculating the likelihood of θ, $L(\theta) = P(\mathcal{O}' \mid \theta_F, Model)$.

The basic EM algorithm comprises two steps: an Expectation Step (E Step), and a Maximization Step (M Step) [24]:

- Select an initial (random) value for θ_F.
- Iterate until convergence:
  - E Step: assign data points to either $f_{\mu_i}(\theta_i)$ or $f_{\mu_F}(\theta_F)$, which ever fits it best.
  - M Step: re-estimate θ_F using the data points assigned to $f_{\mu_F}(\theta_F)$.

The assignment of data points to μ_i and μ_F provides an estimate for t_F. We may exploit the fact that the assignment of data points is temporally correlated with all points before t_f belonging to μ_i, and all points after t_f belonging to μ_f. We may also exploit the fact that data points at the beginning of the interval will belong to μ_i, while those at the end will belong to μ_F. These task-specific qualities help our algorithm converge more quickly.

EM provides a rich algorithm for maximum-likelihood parameter estimation when we don't know the value of t_F. In some hybrid diagnosis applications, depending upon the sensors in our system, and the level of noise in the sensors, we may be able to develop monitoring techniques that will help isolate a reasonable value for t_F, minimizing the need for iteration in EM. In such cases, an alternative to the EM-based approach is to first estimate t_F using the Generalized Likelihood Ratio (GLR) method [5], followed by parameter estimation of θ_F.

GLR + Least Squares Approach Here, we divide the parameter estimation problem into two parts: (i) estimate the time of failure, t_F, using the Generalized Likelihood Ratio (GLR) method, and (ii) apply a standard least squares method for parameter estimation. The intuition is that solving the problem in two parts simplifies the estimation process, and very likely mitigates the numerical convergence problems that arise in dealing with complex higher-order models.

The GLR method for detecting abrupt changes in continuous signals is described in [5]. We have applied it to fault transients analysis in complex fluid thermal systems [16]. Here we provide an overview of the method for the single parameter case. Assume that the signal under scrutiny is a time-indexed sequence of random variables $y(k)$, with probability density function, $p_{\theta_i}(y)$ in desired mode μ_i, and $p_{\theta_F}(y)$ in fault mode μ_F. y is either contained in x_{obs} or computed from x_{obs}. We assume that a fault causes an abrupt change in $y(k)$. In the case of the AERCam, y captures the difference between the observed and expected values of the, e.g., acceleration, as predicted by the model.

The central quantity in the change detection algorithm is the cumulative sum of the log-likelihood ratio for a window of observations between times m and n,

$$S_m^n(\theta_F) = \sum_{k=m}^{n} ln \frac{p_{\theta_F}(y(k))}{p_{\theta_i}(y(k))}.$$

Again, this ratio is a function of two unknowns: t_F and θ_F. The common statistical solution is to use maximum likelihood estimates for these two parameters, resulting in a double maximization:

$$g_n = \max_{1 \leq m \leq n} \sup_{\theta_F} S_m^n(\theta_F).$$

If we assume that probability density functions, $p_{\theta_i}(y)$ and $p_{\theta_F}(y)$ are Gaussian, then g_n reduces to:

$$g_n = \frac{1}{2\sigma_i^2} \max_{1 \leq m \leq n} \frac{1}{n-m+1} \left[\sum_{k=m}^{n} (y(k) - \omega_i) \right]^2,$$

where ω_i and σ_i^2 are the mean and variance for $p_{\theta_i}(y)$, respectively.

When processing a sequence of samples, the point of abrupt change, t_F, is computed from $min\{n : g_n \geq h\}$, where h is an appropriately defined threshold. Hence, the smaller the value of h, the more sensitive the function to change, and unfortunately to false alarms, so h must be set carefully.

Once t_F is estimated, data points observed after t_F, are used to estimate the parameter, θ_F for a hypothesized fault using regression techniques. In the case of the AERCam, the position vector of the AERCam is modeled as a set of quadratic functions in terms of the thruster force. These functions contain one unknown, θ_F, the parameter that corresponds to the degree of degradation in the faulty thruster. The least squares estimate for θ_F is computed, and the the measure of fit of the candidate model to the observed data used to estimated the probability of the candidate model (and hence, diagnosis).

Model Comparison

From the model estimation component, each tracker computes the likelihood of its model Mod_C, and hence of the associated candidate diagnosis $\langle C, \mu_F, t_F, \theta_F \rangle$, as a measure of fit of the observations to the model. As new data $x_{obs}(t)$ are observed, θ_F and t_F, are adjusted and $P(Mod_C \mid x_{obs}(t))$ computed. If the likelihood of Mod_C falls below a predefined acceptable likelihood threshold, α, then its tracker is terminated, and the associated candidate diagnosis $\langle C, \mu_F, t_F, \theta_F \rangle$ removed from the list of candidate diagnoses. Tracking terminates when a unique diagnosis is obtained, or when the diagnoses are sufficiently discriminated to determine suitable controller actions.

5 Related Work

The specific problem of diagnosing hybrid systems has received little attention to date, although there is much related work. Within the AI community, there has been a great

deal of research on diagnosing static systems (e.g., [14]), while much less on diagnosing discrete dynamical systems (e.g., [17,25]), and qualitative representations of continuous systems (e.g.,. [18]). Within the FDI community, the largest proportion of research has focused on diagnosing continuous systems (e.g., [13, 11]). The most common model-based approaches use observer schemes(e.g., [12, 20]), where the goal is to design residual generators based on observed discrepancies, such that individual residuals are sensitive to a particular subset of faults. There is also complementary work by Basseville [4], using model-based statistical processing techniques for early fault detection and residual identification. [18] perform residual generation and analysis task in a qualitative framework to address some of the computational issues that arise in handling the complex dynamics that occur in fault transients, with some preliminary work on building multiple observers for hybrid systems [19]. Diagnosis of discrete-event systems has also been studied within the FDI community (e.g, [22, 15]). Fabre et al. [10] have employed stochastic Petri nets based on a Hidden Markov Model probabilistic scheme for alarm analysis. Unfortunately, it is not clear how to systematically derive such representations from the physical system models that we work with.

6 Summary

In this paper we addressed the problem of diagnosing hybrid systems. The main contributions of the paper are 1) formulation of the hybrid diagnosis problem as model selection; 2) the exploitation of techniques for qualitative diagnosis of continuous systems to reduce the diagnosis search space; and 3) the use of parameter estimation and data fitting techniques for evaluation and comparison of candidate diagnoses. This work continues with experimental analysis of the proposed techniques, and a more formal characterization of our approach in terms of Bayesian model selection.

Acknowledgements

This work was funded in part by NASA grant NAG 21337. The first author would like to thank David Fleet for useful discussion relating to this work.

References

1. L. Alenius and V. Gupta. Modeling an AERCam: A case study in modeling with concurrent constraint languages. In *Proceedings of the CP'97 Workshop on Modeling and Computation in the Concurrent Constraint Languages*, 1998.
2. V. I. Arnold. *Mathematical Methods of Classical Mechanics*. Springer Verlag, 1978.
3. P. Baroni, G. Lamperti, P. Pogliano and M. Zanella Diagnosis of large active systems *Artificial Intelligence*, 110(1):135–183, 1999.
4. M. Basseville. On-board component fault detection and isolation using a statistical local approach. *Automatica*, vol. 34, no. 11, 1998.
5. M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

6. J. A. Blimes. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report TR-97-021, International Computer Science Institute (ICSI) and Computer Science Division, Dept. of Electrical Engineering and Computer Science, U.C. Berkeley, 1998.
7. M. Branicky. *Studies in Hybrid Systems: Modeling, Analysis, and Control.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1995.
8. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society Ser. B*, 39:1–38, 1977.
9. B. Etkin and L. D. Reid. *Dynamics of Flight:Stability and Control.* John Wiley and Sons, 1995.
10. E. Fabre, A. Aghasaryan, A. Benveniste, R. Boubour and C. Jard. Fault detection and diagnosis in distributed systems: an approach by partially stochastic Petri nets. *Journal of Discrete Event Dynamic Systems,* vol. 8, no. 2, pp. 203-231, 1998.
11. P.M. Frank. Fault diagnosis in dynamic systems using analytic and knowledge-based redundancy: a survey and some new results. *Automatica,* vol. 26, pp. 459-474, 1990.
12. E.A. Garcia and P.M. Frank. Deterministic nonlinear observer-based approaches to fault diagnosis: a survey. *Control Engineering Practice*, 5(5):663–670, 1999.
13. J.J. Gertler. *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker, New York, 1988.
14. W. Hamscher, L. Console and J. de Kleer *Readings in Model-based Diagnosis.* Morgan Kaufmann, 1992.
15. J. Lunze. A timed discrete-event abstraction of continuous-variable systems. *Intl. Jour. of Control,* vol. 72, no. 13, pp. 1147-1164, 1999.
16. E.J. Manders, P.J. Mosterman, and G. Biswas. Signal to symbol transformation techniques for robust diagnosis in transcend. In *10th Int. Workshop on Principles of Diagnosis*, pp. 155–165, 1999.
17. S. McIlraith. Explanatory diagnosis: Conjecturing actions to explain observations. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pp. 167–177, 1998.
18. P. Mosterman and G. Biswas. Diagnosis of continuous valued systems in transient operating regions. *IEEE Transactions on Systems, Man, and Cybernetics*, 1999. vol. 29, no. 6, pp. 554-565, 1999.
19. P. Mosterman and G. Biswas. Building hybrid observers for complex dynamic systems using model abstractions. In *International Workshop on Hybrid Systems: Computation and Control*, Nijmegen, Netherlands, March 1999.
20. R.J. Patton and J. Chen. Observer-based fault detection and isolation: robustness and applications. *Control Engineering Practice,* 5(5):671–682, 1997.
21. B. Rinner and B. Kuipers. Monitoring piecewise continuous behavior by refining trackers and models. In *Hybrid Systems and AI: Modeling, Analysis and Control of Discrete + Continuous Systems*, AAAI Technical Report SS-99-05, pp. 164–169, 1999.
22. M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen and D. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Trans. on Control Systems Technology,* vol. 4, no. 2, pp. 105-124, 1996.
23. W. Sweet. The glass cockpit. *IEEE Spectrum*, pages 30–38, September 1995.
24. Y. Weiss. Motion segmentation using EM – a short tutorial. http://www-bcs.mit.edu/people/yweiss/tutorials.html, 1997.
25. B. Williams and P.P. Nayak. A model-based approach to reactive self-configuring systems. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 971–978, 1996.