Model Abstractions for Diagnosing Hybrid Systems ⋆

Gregory Provan *

* *Department of Computer Science, University College Cork, Cork, Ireland (Tel: 353-21-490-1911; e-mail: g.provan@ cs.ucc.ie).*

Abstract: Hybrid systems models are a powerful tool for representing systems with both discrete and continuous dynamics. However, computationally these models are challenging to perform most classes of inference on. In this article we focus on diagnosing hybrid systems. Rather than work on the full hybrid model, for which diagnosis is undecidable in the general case, we abstract the model to a propositional model-based diagnosis (MBD) model. We describe how we can translate a reference hybrid systems model into a propositional diagnosis model, which involves translating the model itself, as well as a sequence of observed events, or trace. We provide an illustrative example of the process, and outline how this process guarantees that any diagnoses in the hybrid systems model will be preserved in the MBD model.

1. INTRODUCTION

The theory of hybrid systems can model systems that exhibit both discrete and continuous behaviours, such as photocopy machines, automobiles, aircraft, etc. Although much work has been done for modeling and verification, there is little work on efficient methods for diagnosis and safety analysis. Hybrid systems diagnosis is inherently difficult due to the continuous dynamics and mode switching of such models. Continuous-valued diagnostics methods can be used for a single mode (Blanke et al. [2003]), but mode-switching can cause instability in observer-based diagnostics inference even with known mode changes (Böker and Lunze [2002]). Another key impediment to performing such analysis is the complexity of the entailed inference: checking reachability for even very simple hybrid systems is undecidable (Henzinger et al. [1995]). Although decidable classes have been identified (Alur et al. [2000]), there are no computational tools that can efficiently reason with real-world models.

In this work we are interested in diagnosing a hybrid system Φ_H, which is undecidable in general, since it entails a form of reachability analysis. To diagnose such systems we abstract the system into a representationally and computationally simpler model, a discrete propositional logic model Φ_D. Φ_D is based on standard model-based diagnosis (MBD) theory (Reiter [1987]), has many mature inference tools, and has diagnosis inference complexity ranging from poly-time to Σ_2^p-complete, depending on the method used for representing the fault behaviours and formulae (Eiter and Gottlob [1995]).

We use abstraction to simplify the hybrid systems model, while preserving key relevant behaviours. Abstraction transforms the inherently infinite-state system of Φ_H into a finite-state model (Alur et al. [2000]). We adopt two abstraction methods: (1) abstract interpretation, which approximates the values of variables; and (2) predicate abstraction, which approximates relationships between variables. The resulting qualitative model will have a finite number of states, so it becomes feasible to perform a number of inference tasks, such as computing the reachable states, the diagnoses, etc.

Abstracting a hybrid system Φ_H to a propositional diagnosis model Φ_D makes sense computationally, but a key issue is whether such an abstraction preserves the diagnoses of Φ_H. One main contribution is showing that using an abstraction operator that over-approximates the state transitions guarantees that all the diagnoses of Φ_H are preserved in the abstract model, at the cost of increasing the space of diagnosis candidates. In addition to this general result, we show some specific conditions on Φ_H that can guarantee that Φ_D will preserve the set of diagnoses computable from Φ_H given a suitably transformed set of observed transitions (an observable trace) input to Φ_D. The proposed conditions describe a hybrid system Φ_H, which includes both normal and failure states, in which all reachable states will be reachable within the continuous portion of Φ_H, including the discrete failure states. These conditions cover a wide range of real-world applications, subject to defining the failures in terms of the underlying continuous dynamics, such as is done in Zhao et al. [2005].

Our contributions are as follows:

(1) We show that we can guarantee that the diagnosis space of Φ_H is preserved in Φ_D by using an abstraction operator that over-approximates the state transitions;
(2) We extend a hybrid systems abstraction methodology (Tiwari [2008]) to enable the generation of propositional MBD models;
(3) We show how we can untime the event sequence (trace) of a hybrid system Φ_H to create an observation suitable for an MBD model;
(4) We describe particular conditions on the abstraction operator that guarantee that the diagnosis space of Φ_H is preserved in Φ_D;
(5) We illustrate our approach with a detailed example.

⋆ Supported by SFI grants 04/IN3/I524 and 06/SRC/I1091.

2. RELATED WORK

This section reviews prior work in related areas.

There has been a lot of work on abstraction in model-based diagnosis (MBD), such as (Saitta et al. [2007], Maier and Sachenbacher [2008]), which have focused on abstracting either propositional models (Saitta et al. [2007]) or qualitative models (Maier and Sachenbacher [2008]). In contrast, in this article we start with hybrid systems models, in which we aim to abstract *both* continuous and (possibly infinite) discrete spaces, which is significantly more difficult than these discrete abstraction approaches.

We adopt the use of over-approximation of the transition relation of Φ_H, which is a standard abstraction technique. Several related over-approximation approaches have been published, based on techniques such as hyper-rectangles, polyhedra and their projections, or ellipsoids (Clarke et al. [2003]). Most of these approaches attempt to obtain conservative but tight approximations to sets of reachable states for hybrid systems. Abstracting transition relations for hybrid systems is inherently complicated, because these relations, as a general rule, do not have analytical solutions, and even when analytical solutions exist, creating tight over-approximations is challenging. We adopt the approach of Tiwari [2008], which converts the transition relations into polynomials and subsequently into propositional equations.

More efficient abstractions have been developed for specific classes of hybrid systems. For example, for piecewise affine systems, Hofbaur and Rienmuller [2008] have developed 2D grid abstractions. Such techniques have been applied to the abstraction of gene regulatory networks (Batt et al. [2008]), where the relative order of threshold parameters and ratios of parameters are used for phase space partitioning in abstract regions. Such model-specific approaches can complement the generic abstraction methodology studied in this article in appropriate applications.

There is some relation to abstraction for qualitative simulation (QS), but the goals of this work are very different. QS aims to create a qualitative differential model with properties that are qualitatively equivalent to the initial model; such a model requires custom qualitative algebras and inference techniques. Here, we make a more radical abstraction, generating a propositional diagnosis model with standard propositional logic semantics. In other words, we want a diagnosis model that can be solved by traditional diagnosis algorithms, e.g., GDE, ATMS, or by modified SAT solvers. The properties we aim to preserve from Φ_H are fewer than those preserved by a qualitative model. For example, we are only interested in preserving a subset Q_N of nominal states and a set Q_F of failure states, plus the transitions from Q_N to Q_F; the key is just to distinguish states in Q_N from those in Q_F. In contrast, a QS model aims to preserve *all* the qualitatively significant states of Φ_H, and the transitions among those states.

The qualitative models of (Kuipers [1986]) roughly correspond to the abstract transition systems that we develop. Tiwari [2008] actually translates a hybrid system into a qualitative model; we instead generate a diagnosis model together with an observation necessary to diagnose potential faults. Shults and Kuipers [1997] prove a range of formal properties that are preserved by qualitative models. This work is closely related to several approaches to abstracting hybrid systems models, such as Alur et al. [2000], Tiwari and Khanna [2002], Tiwari [2008]. These papers focus primarily on properties such as the semantics of the hybrid system considered, the class of (a) formulas preserved, (b) hybrid systems analysed, or (c) abstract systems generated, or the type of abstraction (e.g., conservative, accurate, etc.).

Accurate abstractions, or bisimulations, can create decidable systems, for which clear results can be obtained (Alur et al. [2000]). Olivero et al. [1994] abstracts some restricted classes of linear hybrid systems into simpler class of hybrid systems, timed automata. Henzinger et al. [1998a] abstracts a nonlinear hybrid automaton in terms of a linear hybrid automaton. Tiwari [2008] abstracts a hybrid system as a qualitative model; we extend this work by defining a diagnosis model based on the qualitative abstraction. All of the work on hybrid systems abstraction focuses on model abstractions, whereas we also have to define an abstraction for the trace, in order to obtain a system observation for which a diagnosis can be computed.

3. NOTATION AND PRELIMINARIES

This section introduces our notation. We first define the hybrid-systems language we will use. Then we introduce our diagnosis modeling language.

3.1 Hybrid Systems

Hybrid automata (Henzinger et al. [1998a]) are mathematical models for representing hybrid systems. In contrast to discrete transition systems, hybrid automata can make both discrete and continuous transitions and hence, their semantics are given in terms of the states, which are uncountably many, reached over a continuous real time interval. We can also define the theory of hybrid automata in terms of infinite-state transition systems (Henzinger et al. [1998b]) that contain uncountably many states, but are interpreted over discrete time steps.

We adopt an extended version of a hybrid system that has modes for normal and failure states. We also assume that only a subset of events are observable, and in our models we denote these events as those relating to sensors and actuators changing state.

Definition 1. A hybrid system is defined as $\Phi = (Q, X, \Sigma, Q_0, E, f, G)$ where

- Q is the set of discrete states or modes of the system,
- $X \subseteq \Re^n$ is the continuous state space,
- Σ is a finite set of transition labels or events,
- $Q_0 \subseteq Q \times X$ is the set of initial conditions,
- $E \subset Q \times \Sigma \times Q$ is the transition relation, which defines the set of (controlled and autonomous) discrete transitions,
- $f : \Re \times Q \times X$ is the flow condition for every mode defined by a differential equation,
- and $G : E \rightarrow 2^X \times \pi$ is a partial function that associates a guard condition (represented as a subset of X) with each autonomous transition, given a probability π.

The probability π introduces randomness into the transitions, which is important for transitions to failure states, which we assume occur randomly.

A state of a hybrid system is described by the pair (q, x), where $q \in Q$ and $x \in X$. We define $\mathcal{R}(x_0, q_0)$ as the set of reachable states from (x_0, q_0).

We assume that the set of modes of the hybrid system is partitioned such that $Q = Q_N \cup Q_F$, where Q_N and Q_F are the set of normal modes and faulty modes respectively. Similarly, we partition the set of transition labeling events as $\Sigma = \Sigma_N \cup \Sigma_F$. Σ_N is the set of endogenous (controlled) events transitions to normal modes, and Σ_F is the set of (exogenous) failure events labels transitions to faulty modes. Note that if information about the continuous dynamics for the faulty modes is available, then we associate a flow condition with these modes. In the simple model proposed here, we assume that whenever a transition is activated by the underlying continuous dynamics, the actual transition to a normal mode or a faulty mode is determined by the stochastic parameter π, which reflects the stochastic nature of failures occurring.

We partition our events into two subsets: $\Sigma_o \subseteq \Sigma$ is a subset of *observable* events, and $\Sigma_u \subseteq \Sigma$ is a subset of *unobservable* events. We assume that all transitions to fault states are unobservable.

We define a trace as a sequence of events.

Definition 2. (Trace). A trace $\gamma(\Phi_H, (q_0, x_0))$ is a sequence $\{(q_0, x_0), \sigma_1, ..., \sigma_m\}$, where $\sigma_i \in \Sigma$.

The observable subset of a trace $\gamma_o \subseteq \gamma$ is just the sequence of observable events given by the projection $\zeta : \Sigma^* \rightarrow \Sigma_o$ (Sampath et al. [1995]). In other words, ζ "erases" the unobservable events in a trace. We define a system trace, Γ_Φ, as the set of all traces of a system starting from an arbitrary initial state (q, x).

If $f = (q_f, x_f)$ is a failure state, we can define Γ_f as the set of all traces starting from failure state f: $\Gamma(f) = \{\gamma(\Phi_H, (q_f, x_f))\}$. We now introduce a consistency-based notion of diagnosis for hybrid systems. Intuitively, a fault can be isolated if a fault-free system cannot generate a trace γ_f including a failure state (called an anomalous trace), i.e., the fault-free system and trace γ_f are inconsistent. More generally, a measured trace γ' is said to be consistent with a model Φ_H and corresponding system trace Γ_{Φ_H} if $\gamma' \in \Gamma_{\Phi_H}$.

Given these definitions, we can now describe what a hybrid systems diagnosis is.

Definition 3. (Candidate HS-Diagnosis). Given a hybrid system $\Phi_H = (Q, X, \Sigma, Q_0, E, f, G)$, initial conditions (q_0, x_0), and anomalous trace γ, a candidate diagnosis δ is a failure state consistent with γ, i.e., where $\gamma \in \Gamma(\delta)$.

More generally, given an abnormal trace γ, the set of all candidate diagnoses is given by $\Delta = \{\delta | \gamma \in \Gamma(\delta)\}$.

3.2 Propositional Diagnosis Model

In this article we adopt a temporally extended version of the diagnosis framework introduced in Reiter [1987].

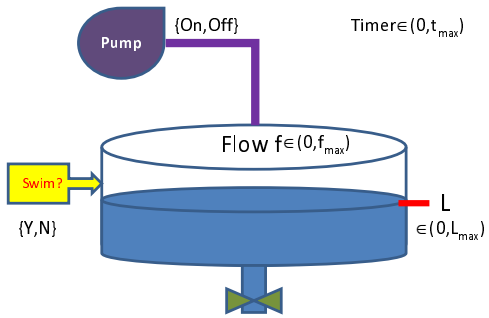

Fig. 1. Schematic for Swimming Pool Example

Definition 4. (Propositional Diagnosis Model). A discrete diagnosis model is specified by a tuple $\Phi_D = \{I, V, \mathcal{E}, \Pi\}$, where

- $I \subset \mathbb{N}$ is a temporal index;
- V is a set of discrete-valued variables indexed by I, such that $V_f \subset V$ is the set of failure mode variables, and $V_o \subset V$ is the set of observable variables;
- $\mathcal{E} \subseteq V_f \times \mathcal{L}_n$ consists of propositional equations (where $\mathcal{L}_n$ is a propositional wff over $(V \setminus V_f)$; and
- Π is a discrete probability distribution over the equations and/or variables.

This *temporal* diagnosis model obeys standard logical semantics, and differs from a classical MBD model only in terms of temporal indexing of variables. Further, this definition of diagnosis model is an instance of a transition system (Stark [1989]).

We also need to specify an observation in order to define a diagnosis within this framework.

Definition 5. (Observation). An observation α is an instantiation of V_o.

Definition 6. (Diagnosis). Given a diagnostic system $\Phi_D = \{I, V, \mathcal{E}, \Pi\}$, an observation α over some variables in V_o, a diagnosis δ is an assignment to all variables in V_f such that $\Phi_D \wedge \alpha \wedge \delta \not\models \perp$.

We can define diagnosis minimality with respect to several criteria, such as subset- or cardinality-minimality, or the probabilistically most-likely diagnoses using Π, the discrete probability distribution over Φ_D.

Note that there are several differences between the hybrid-systems and propositional diagnosis models. Whereas the hybrid-systems model has both continuous and discrete variables, the diagnosis model has only discrete. In addition, the diagnosis model requires the specification of failure-mode and observable variables, which the hybrid-systems model specifies implicitly in its trace. A diagnosis model of this form has only discrete dynamics.

4. ILLUSTRATIVE EXAMPLE

We use an illustrative example taken from Sokolsky and Hong [2001], in order to provide an intuitive notion of the issues we address.

4.1 Hybrid Systems Model

Consider a swimming pool equipped with a pump that controls the water level L, and a switch that indicates if

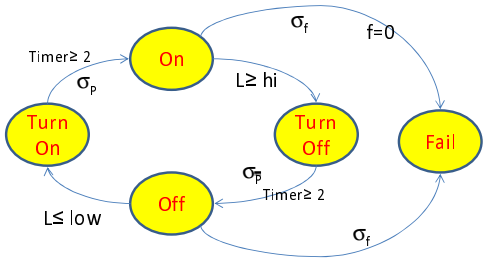

Fig. 2. Full Automaton for Pump

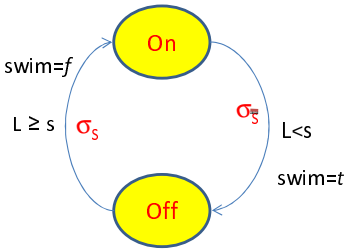

Fig. 3. Automaton for *Swim?* indicator

the water is deep enough to swim in. The pump, when on, fills the pool at a constant overall flowrate f, and when off, allows water to drain out of the pool. [1] The pool's level is governed by the equation $\frac{\partial L}{\partial t} = f$. Changing state in the pump actuator takes 2 time units (as measured by a *Timer*), during which it is in state *TurnOn* or *TurnOff*. The swimming level indicator switch *Swim?* is *on* when the level $L \geq s$, where s is a swimmable level, and is *off* otherwise.

Figure 1 shows the hybrid system schematic for the swimming pool example. The possible values for *Pump*,*Swim?*, L and f are shown. Note that L, f and *Timer* are continuous-valued variables.

Figure 2 shows the full automaton for the *Pump*. In the figure we show some transitions and guards for transitions. For example, σ_P denotes the event of the pump turning on, and $\sigma_{\bar{P}}$ denotes the event of the pump turning off. Note that we assume that the pump can fail (state *Fail*), in which case it produces no flow.

Figure 3 shows the automaton for the *Swim?* indicator. In the figure we show the transitions, and the guards for the transitions.

4.2 State Space Abstraction

To create the abstract (qualitative) model, we transform the continuous-valued variables into discrete-valued counterparts, and specify qualitative relations for all equations in the hybrid systems model. In this model, we introduce a discrete variable $\dot{L}$ for $\frac{\partial L}{\partial t}$, with domain $\{In, 0, Out\}$, and define discrete domains for L and f of $\{0, low, swim, hi, Ovflow\}$ and $\{In, 0, Out\}$ respectively.

4.3 MBD Model

We represent our MBD model using a set of variables corresponding to variables in Φ_H. For example, we have a variable *Swim?* with domain $\{Y, N\}$, and a variable *Act* denoting the actuator for the pump, with domain {On,Off}. In this example we have one failure-mode variable, that for the pump, with domain {OK,fail}.

We use a simple form of discrete temporal indexing: we denote a variable V at time t using V_t, and a variable W at the preceding time using $W_{t_<}$.

Some normal-mode equations include:

$$(Pump = OK) \Leftrightarrow [(Swim? = N)_{t_<} \wedge (Act = On)_{t_<} \Rightarrow (Swim? = Y)_t]$$
$$(Pump = OK) \Leftrightarrow [(Swim? = Y)_{t_<} \wedge (Act = Off)_{t_<} \Rightarrow (Swim? = N)_t]$$

If we have a weak fault model (which defines only normal behaviour—see de Kleer et al. [1992]), then we need to include only normal-mode equations. However, if we define a strong fault model, then we must include some failure-mode equations, such as:

$$(Pump = fail) \Leftrightarrow [(Swim? = N)_{t_<} \wedge (Act = On)_{t_<} \Rightarrow (Swim? = N)_t]$$
$$(Pump = fail) \Leftrightarrow [(Swim? = Y)_{t_<} \wedge (Act = Off)_{t_<} \Rightarrow (Swim? = Y)_t]$$

We assume that our equations have a first-order Markov structure, i.e., any equation only includes variables covering two different time steps. [2]

5. DIAGNOSIS PROPERTIES OF MODEL ABSTRACTIONS

This section describes soundness and completeness properties of the diagnoses that can be computed from abstract models that over-approximate the reference model. We will present properties that hold for an arbitrary abstraction ξ of a model.

The results we will show bear some resemblance to the soundness/completeness properties that qualitative models preserve from hybrid (or dynamical) systems models. A qualitative simulation algorithm $\mathcal{F}$, given an ODE model Φ and initial state that matches the input to $\mathcal{F}$, is *sound* if there exists a behavior that matches Φ's solution. Kuipers [1986] proved the existence of a sound qualitative simulator, QSIM. Shults and Kuipers [1997] formalized this result using temporal logic, proving that if a CTL* wff β is true for the behaviours produced by QSIM, then a corresponding temporal wff, β', holds for the solution of any ODE consistent with the qualitative differential equation that QSIM used to generate the behaviours.

However, Yilmaz and Say [2006] show that a sound and complete qualitative simulator does not exist: even with restrictions on operating regions and qualitative constraint operators, a sound algorithm $\mathcal{F}$ is inherently incomplete. If we impose a coverage guarantee, then one can specify some input model that causes a simulator $\mathcal{F}$ to generate spurious predictions in its output.

[1] The flow f is the difference between the inflow from the pipe and the outflow through the valve in the pool.

[2] Higher-order Markov structure can be converted to first-order Markov structure using well-known rewrite rules.

Our diagnosis results have some parallels to these results. We ensure soundness and completeness of the abstract diagnoses through using over-approximations, but at the expense of the abstract model generating more diagnoses than the reference model. This is analogous to the spurious behaviours of qualitative simulation.

In the following, we call a diagnosis instance the triple $Z = (\Phi, \Gamma, \Delta)$, where Φ is the model, Γ is the trace, and Δ is the set of candidate diagnoses. Given an abstraction operator ξ, we can show that any abstraction which is an over-approximation is guaranteed to be complete with respect to the diagnostics of the reference model. We define an over-approximation as follows.

Definition 7. (Over-approximation). Given a hybrid systems model Φ_H with corresponding system trace Γ_H, an abstraction Φ', with corresponding system trace Γ', is an over-approximation of Φ_H with respect to abstraction operator ξ if $\Gamma' \subseteq \xi(\Gamma_H)$.

Lemma 1. Given a hybrid systems diagnosis instance Z_H, an abstract diagnosis instance Z_D which is an over-approximation is guaranteed to be complete with respect to Z_H (under abstraction operator ξ). [3]

The following corollary outlines the diagnostics properties in more detail.

Corollary 1. Given a hybrid systems triple Z_H, if an abstraction triple Z' is a sound over-approximation for Z_H,

- If a diagnosis $\delta \in \Delta_H$ exists in Z_H, then a corresponding diagnosis $\delta' \in \Delta'$ will exist in Z'.
- If no diagnosis $\delta' \in \Delta'$ exists in Z', then no corresponding diagnosis $\delta \in \Delta_H$ will exist in Z_H.

The second part of Corollary 1 notes that if a diagnosis δ' is excluded in the abstract model, then it will be excluded in the reference model (and in any less-abstract model).

Lunze [2008] shows four examples of abstractions of a hybrid systems model that are over-approximations, and hence display a subset-inclusion property among fault candidates. Based on the methodology in (Lunze [2008]), we extend Lemma 1 to cover diagnoses of a series of abstract models.

Lemma 2. Given a hybrid systems diagnosis instance Z_H and two abstraction diagnosis instances Z' and Z'' with progressively increasing levels of abstraction, the system diagnosis sets will obey the set-inclusion $\Delta_H \subseteq \Delta' \subseteq \Delta''$.

Our abstraction process creates an interesting tradeoff: inference complexity decreases with greater abstraction levels, but the over-approximation used to create the abstract model results in more candidate diagnoses as well. This means that the more abstract models will generate more candidate diagnoses, and most likely will be less able to isolate minimal diagnoses than the more detailed models. In general, the optimal level of abstraction is the one that creates a model with lowest model complexity that can isolate the key faults. It is important to note that more abstract models can result in significant decreases in inference complexity. Even abstracting a propositional strong-fault model to a weak-fault model, as done by Feldman et al. [2009], can reduce inference complexity from NP-hard to polynomial.

At some level of abstraction, the diagnosis model fails to isolate key faults. In abstracting to static MBD models, we can show the following:

Lemma 3. Given a hybrid systems diagnosis instance Z_H and an abstraction diagnosis instance Z', a static MBD model is inadequate for diagnosing even simple hybrid systems abstractions.

For example, for the swimming pool model, we cannot diagnose any faults in the pump, since the control laws of the pump entail comparing previous and current levels of the swimming pool (cf. the model in Section 4.3).

In line with this need for temporal modeling, Behrens et al. [2009] empirically show that a temporal propositional model that encodes just the current and previous states for every variable can diagnose our swimming pool example.

6. BISUMULATION CONDITIONS

Bisimulation is an important tool in the analysis of concurrent systems: roughly speaking, when two concurrent systems are bisimilar, known properties are readily transferred from one system to the other. This section outlines conditions for bisumulation of Φ_H and Φ_D, denoted $\Phi_H \sim \Phi_D$. We adopt (loose) notions of bisumulation to show preservation of diagnoses between Φ_H and Φ_D.

The problem that we solve is as follows:

Problem 1. Given a hybrid system Φ_H and a trace Γ ending in a failure event f, characterise model mapping ξ and trace mapping ς such that, for the abstracted propositional diagnosis system $\Phi_D = \xi(\Phi_H)$ and an observation $\theta = \varsigma(\Gamma)$, fault f can be isolated in $\langle \Phi_H, \Gamma \rangle$ iff fault $\varsigma(f)$ can be isolated in $\langle \Phi_D, \theta \rangle$.

We can simplify the abstraction process by imposing the following conditions. In the following sections we will show how these conditions aid in guaranteeing diagnostics bisumulation.

Condition 1. The abstraction preserves the trace of the set of discrete hybrid system transitions marked as observables (Henzinger et al. [1998b]).

Condition 2. A trace is decomposable into a set of subtraces (which may be recurring).

Condition 3. All discrete transitions are "coherent" with the continuous system evolution, i.e., $\not\exists$ reachable states which are not reachable by the continuous dynamical portion of Φ_H.

7. MODEL ABSTRACTION PROCESS

We now summarise how we solve Problem 1 by abstract interpretation and predicate abstraction. [4] We translate two representations, the model and the observable trace, i.e., the set of observable transition labels outputs by the evolution of the hybrid system. For the model mapping ξ, we extend the qualitative-model abstraction of Tiwari

[3] We omit all proofs due to space limitations.

[4] Please refer to the full paper for technical details omitted here due to space limitations.

[2008] to a propositional diagnosis system mapping. For the trace mapping, we define an algorithm based on trace sub-sequence decomposition that will generate an untimed observation suitable for Φ_D.

7.1 Model Abstraction Process

Our translation algorithm uses the following main steps:

(1) Generate qualitative states for continuous- and discrete-valued states in Φ_H.
(2) Compute abstract transitions for Φ_H.
(3) Create a composite automaton Φ_T from the generated abstract automata $A_1, ..., A_q$ through parallel composition.
(4) Transform Φ_T into Φ_D.

Step 1: Compute the Abstract Set of Discrete States We create an abstracted set of states from the continuous and discrete states in Φ_H, using a finite set $P \subseteq \Re[X]$ of polynomials over the continuous variables X for the continuous-state abstraction. Then, we abstract the initial state $Q_0 \in \Phi_H$. We compute the set P of polynomials in terms of two subsets, P_1 and P_2:

(1) we compute the set P_1 of polynomials from (a) the guards of mode transitions for exiting each mode, and (b) key properties of interest that we want to establish for the given continuous system;
(2) the set P_2 of time derivatives of polynomials in P_1.

We compute P_2 from P_1 as follows: for each $p \in P_1$, add $\dot{p}$, the derivative (with respect to time) of p, to the set P_2 unless $\dot{p}$ is a constant or a constant factor multiple of some existing polynomial in P.

Given Φ_H and the set $P \subseteq \Re[X]$ of polynomials over the set X of variables, the set of abstract states consists of the union of three subsets, i.e., $Q^A = Q_P \cup Q_n \cup Q_f$, where

- $Q_P = \{q_p : p \in P\}$ is the set of states derived from the polynomials P;
- Q_n is the set of states derived from the discrete, normal-mode states in Φ_H; and
- Q_f is the set of states derived from the discrete, failure-mode states in Φ_H.

For the swimming pool example, the steps are as follows.

Identify Polynomials: If we consider the guards for the transitions, they are all based on the level L in the pool, and its relation to the swimmable level, s. We can thus represent this polynomial p using $L-s$. The derivative of p is $\dot{L}$, since s is a constant. Hence our full set of polynomials is $P = \{L - s,\ \dot{L}\}$.

Identify Abstracted Discrete States: $P = \{L - s,\ \dot{L}\}$, so we generate the corresponding state variables $\{q_{L-s},\ q_{\dot{L}}\}$, each with domains $\{0, +, -\}$.

Identify Initial Discrete State: Consider an initial state $\langle q_0 = \{Act{=}off, Swim?{=}off\},\ x_0 = \{L = 0, f = 0\}\rangle$. We transform x_0 into the discrete instantiation $\{q_{L-s} = -,\ q_{\dot{L}} = 0\}$. Hence our initial discrete state is given by: $Q^A = \langle Act{=}off, Swim?{=}off, q_{L-s} = -,\ q_{\dot{L}} = 0\rangle$.

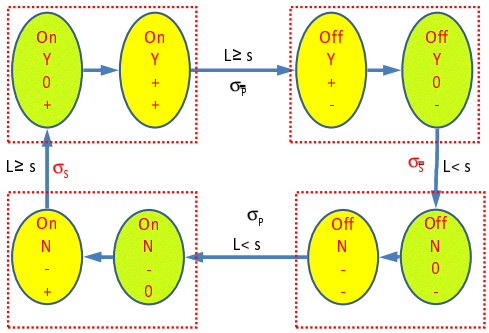

Fig. 4. Complete abstract automaton for Swimming Pool Example. Each state denotes values for $\langle Act, Swim?, q_{L-s},\ q_{\dot{L}}\rangle$.

Step 2: Compute the Abstract Transitions The set $T \subseteq Q^A \times Q^A$ defines the set of transitions in the transition system. The transitions in the abstract system Φ_A from the state q^a are obtained as a union of the discrete and continuous transitions. Condition 3 ensures that these two types of transition are "consistent", simplifying the abstraction process.

(1) *Abstractions of the discrete transitions*: If $(q, \mathcal{S}, q') \in \Sigma$ is a discrete transition of the hybrid automaton Φ_H, where $q, q' \in Q$ are discrete states and $\mathcal{S} \in X$ is a set of continuous states (or the guard) represented by the predicate formula μ over the variables X, then there is an abstract transition $((q, \beta), (q', \beta)) \in T$ if $\Re \models \exists X : (\beta(X) \wedge \mu(X))$.
(2) *Abstractions of the continuous transitions*: Using Condition 3, since the continuous-state transitions are consistent with the discrete transitions, we can assume that the discrete transitions cover the full set of transitions, and consequently no abstraction of the continuous transitions is necessary. Without this condition, we need to to make appropriate abstractions, such as those proposed in Tiwari and Khanna [2002].

Step 3: Create a simplified composite automaton Φ_T We create the automaton based on the abstract states and transitions computed in steps 1 and 2. For our example, we create an abstract automaton by parallel composition of the automata for (abstracted) *Pump* and the *Swim?* indicator, as shown in Figure 4. The abstract transitions are clearly depicted in this automaton.

We delete any states in Figure 4 with a value of zero for either q_{L-s} or $q_{\dot{L}}$, since they have no clear semantical impact on the abstract model. Figure 6 shows the simplified automaton for the swimming pool. The figure shows the transitions and the guards for each transition.

Step 4: Transform Φ_T into Φ_D We generate equation set $\mathcal{E}$ using different algorithms for normal-mode and failure-mode equations. For a normal-mode equation, for every state transition from Q_i to Q_j denoted $Q_i \sigma Q_j$, we generate an equation using the template $(M = OK) \wedge V_i^{t<} \wedge V_\sigma^{t<} \Rightarrow V_j^t$, where Q_i corresponds to $V_i^{t<}$, σ corresponds to $V_\sigma^{t<}$, and V_j corresponds to V_j^t. For a failure-mode equation, for every state transition $Q_i \sigma_f Q_j$ (where fault transition σ_f is unobservable), we generate an equation using the template

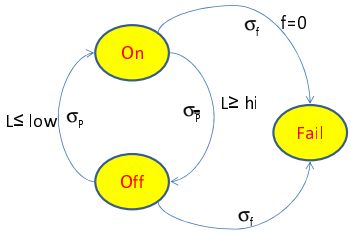

Fig. 5. Simplified Automaton for Swimming Pool Pump

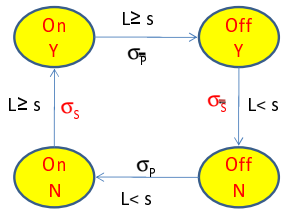

Fig. 6. Composite Automaton for Swimming Pool Example. We exclude the failure state (that is accessible from every other state) for the sake of clarity.

$(M = fail) \wedge V_i^{t<} \Rightarrow V_j^t$, where Q_i corresponds to $V_i^{t<}$, and Q_j corresponds to V_j^t.

7.2 Trace Abstraction Process

In order to transform a trace into an observation that can be used for MBD purposes, we need to extract the observable events corresponding to a particular behaviour of the system. To accomplish this, we need to perform *trace untiming* in order to transform a trace into an MBD observation. Without loss of generality, we assume that a system goes through a set of behaviour-sequences, where a behaviour-sequence is a subsequence of a trace. Hence we can break a trace Γ into $\Gamma = \{q_0, \gamma_1, ..., \gamma_m\}$, where q_0 is the initial state and $\gamma_i \neq \gamma_j,\ i \neq j$.

In the MBD model, the set of observable variables is given by V_o. A sub-trace must contain an event label for every $v_i \in V_o$. However, it is important to note that an instantiation of some $v_i \in V_o$ may correspond to multiple observable event labels. For example, in the *Pump* automaton the variable *Act* takes on values *On, Off*, such that *Act=On* corresponds to σ_P and *Act=Off* corresponds to $\sigma_{\bar{P}}$. Hence, given V_o and trace mapping ς, a minimal sub-trace γ^* corresponds to $\varsigma^{-1}(V_o)$. As an example, if we have a trace $\Gamma = \{\sigma_P, \sigma_S, \sigma_{\bar{P}}, \sigma_{\bar{S}}\}$, we will have minimal sub-traces $\{\gamma_1, \gamma_2\}$ where $\gamma_1 = \{\sigma_P, \sigma_S\}$, and $\gamma_2 = \{\sigma_{\bar{P}}, \sigma_{\bar{S}}\}$.

We assume that there is a corresponding observation sequence $\Theta = \{\theta_0, \theta_1, ..., \theta_m\}$, which is obtained by an untiming function $\varsigma : \Sigma^* \rightarrow 2^\Theta$. We also assume that we have a monitoring tool that can extract any sub-sequence. We transform this sub-sequence into an observation $\Theta = \varsigma(\tau_i)$. Because any equation in Φ_D has a first-order Markov structure, i.e., contains variables from at most two time-steps, an observation must contain sub-traces (sets of observable event labels) from two adjacent time steps.

Given a trace Γ_f with a fault event f, we generate an observation by (1) extracting the final two sub-traces $\gamma = \gamma_m,\ \gamma_{m-1}$, and (2) transforming γ in an observation $\theta_m = \varsigma(\gamma)$.

7.3 Trace Abstraction Example: swimming pool

The swimming pool goes through a cycle characterised by the pump turning on, the level filling to a swimmable value (indicated by the switch), the pump turning off, and the level then falling to an un-swimmable value (again indicated by the switch), after which the cycle repeats. The cycle is defined by a transition sequence $\langle\{\sigma_P, \sigma_S, \sigma_{\bar{P}}, \sigma_{\bar{S}}\}\rangle$, which corresponds to two sub-sequences $\langle\gamma_1, \gamma_2\rangle$, where $\gamma_1 = \{\sigma_P, \sigma_S\}$, and $\gamma_2 = \{\sigma_{\bar{P}}, \sigma_{\bar{S}}\}$.

Consider a case where we have 3 cycles, starting at the state given by $\langle q_0 = \{Act{=}off, Swim?{=}N\},\ x_0 = \{L = 0, f = 0\}\rangle$, and ending with a failure event f denoting a failed pump. The trace is given by a set of 7 sub-sequences, $\gamma_1, ..., \gamma_7$: $\gamma(\Phi_H, (q_0, x_0)) = \{\langle\{\sigma_P, \sigma_S\}, \{\sigma_{\bar{P}}, \sigma_{\bar{S}}\}\rangle, \langle\{\sigma_P, \sigma_S\}, \{\sigma_{\bar{P}}, \sigma_{\bar{S}}\}\rangle,\ \langle\{\sigma_P, \sigma_S\}, \{\sigma_{\bar{P}}, \sigma_{\bar{S}}\}\rangle,\ \langle\{\sigma_P, \sigma_{fail}, \sigma_{\bar{S}}\}\rangle\}$. The observable part of this trace omits just the failure event σ_{fail} in the final sub-sequence, giving the final sub-sequence as $\gamma_4 = \{\sigma_P, \sigma_{\bar{S}}\}$. If we "untime" γ_6 and γ_7, we obtain $\varsigma(\gamma_6) = \{Act_{t_<}{=}off, Swim?_{t_<}{=}N\}$ and $\varsigma(\gamma_7) = \{Act_{t_<}{=}on, Swim?_t{=}N\}$. If we generate an observation from γ_6 and γ_7, we step back in reverse chronological order to generate observable variable instantiations for time steps t and $t_<$ such that there is just a single, most recent instantiation for every variable at each of t and $t_<$. In this case, we obtain the observation $\{Swim?_{t_<}{=}N,\ Act_{t_<}{=}on, Swim?_t{=}N\}$. Note that we do not include $Act{=}off_{t_<}$ in this observation, since it would cause a contradiction with $Act{=}on_{t_<}$, which is chronologically more recent.

Using the observation with the MBD model fragment described earlier, we can obtain the diagnosis that *Pump=fail*.

8. PROPERTIES PRESERVED BY ABSTRACTION

This section describes the properties of the hybrid systems model Φ_H that are preserved by the proposed abstraction process. Recall that we adopt a two-step process, in which we first map Φ_H into an automaton (discrete transition system) Φ_T, and then map Φ_T into the diagnosis model Φ_D.

Tiwari [2008] proves that a particular abstraction of a hybrid automaton Φ_H is a discrete transition system Φ_T that bisimulates the discrete system $\xi(\Phi_H)$.

We extend this abstraction with the conditions noted earlier, such that we still maintain the bisimulation property of Φ_T. If we assume that we start with a transition system defined by the abstract composite automaton we described in the article, we now show how our mapping to an MBD model Φ_D allows us to bisimulate Φ_T, and hence also Φ_H, since if $\Phi_D \sim \Phi_T$ and $\Phi_T \sim \Phi_H$, then $\Phi_D \sim \Phi_H$.

Proposition 1. The model transformation from composite automaton Φ_T to MBD model Φ_D is such that $\Phi_D \sim \Phi_T$.

The second key property is guaranteeing that any diagnosis in Φ_H given trace γ exists iff the diagnosis is also valid in the corresponding MBD model given $\varsigma(\gamma)$. We can show this via the following argument.

We first need to prove properties about traces in Φ_H given an initial condition Q_0 and the corresponding observation in $\Phi_D = \xi(\Phi_H)$ given $\Theta_0 = \xi(Q_0)$.

Proposition 2. Given a trace γ of a hybrid system Φ_H based on initial conditions Q_0, and the diagnosis model $\Phi_D = \xi(\Phi_H)$ with corresponding unobservable setting $\Theta_0 = \varsigma(Q_0)$, if $\gamma \in L(\Phi_H, Q_0)$, then $\Phi_D \cup \varsigma(Q_0) \cup \xi(\gamma) \not\models \perp$.

We can use Proposition 2 to directly prove a result about *diagnosis abstraction bisimulation*, which (partially) satisfies Problem 1.

Proposition 3. (Diagnosis abstraction bisimulation). Given a hybrid system Φ_H and a trace Γ ending in a failure event f, $\exists$ a model mapping ξ and trace mapping ς such that, for the abstracted propositional diagnosis system $\Phi_D = \xi(\Phi_H)$ and an observation $\theta = \varsigma(\Gamma)$, fault f exists in $\langle \Phi_H, \Gamma \rangle$ if fault $\xi(f)$ exists in $\langle \Phi_D, \theta \rangle$.

9. SUMMARY AND CONCLUSIONS

This paper has described a methodology for transforming hybrid systems diagnosis models into propositional MBD models, such that we preserve the possible diagnoses of the hybrid systems diagnosis model. This can lead to significant computational gains in hybrid systems diagnosis. To guarantee diagnosis bisimulation, we have described three conditions that must be satisfied.

We described how we can translate a reference hybrid systems model into a propositional diagnosis model, which involves translating the model itself, as well as a sequence of observed events, or trace. We provided an example of the process. The abstraction process imposes a large number of constraints on the hybrid system that can be adopted. We leave it to future work to identify the practical ramifications of these restrictions, and whether we can relax the restrictions.

We have implemented this approach and applied it to building control systems applications (Behrens et al. [2009]). Future work includes refining and extending this approach, and identifying the range of systems for which it is applicable.

REFERENCES

R. Alur, TA Henzinger, G. Lafferriere, and GJ Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

G. Batt, H. de Jong, M. Page, and J. Geiselmann. Symbolic reachability analysis of genetic regulatory networks using discrete abstractions. *Automatica*, 44(4):982–989, 2008.

M. Behrens, G. Provan, M. Boubekeur, and A. Mady. Model-Driven Diagnostics Generation for Industrial Automation. In *IEEE Itnl Conf. on Industrial Informatics*. IEEE, 2009.

M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, 2003.

G. Böker and J. Lunze. Stability and performance of switching Kalman filters. *International Journal of Control*, 75(16):1269–1281, 2002.

E. Clarke, A. Fehnker, Z. Han, B. Krogh, O. Stursberg, and M. Theobald. Verification of hybrid systems based on counterexample-guided abstraction refinement. *Lecture Notes in Computer Science*, pages 192–207, 2003.

Johan de Kleer, Alan Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2-3):197–222, 1992.

T. Eiter and G. Gottlob. The complexity of logic-based abduction. *J. of the ACM (JACM)*, 42(1):3–42, 1995.

A. Feldman, G. Provan, and A. van Gemund. Characterizing Strong-Fault Diagnostic Models. In *Proc. IJCAI'09*, July 2009.

TA Henzinger, P.H. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):540–554, 1998a.

T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *Proc. ACM Symp. on Theory of Comp.*, pages 373–382, 1995.

T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. Whart's Decidable about Hybrid Automata? *Journal of Computer and System Sciences*, 57:94–124, 1998b.

Michael Hofbaur and Theresa Rienmuller. Qualitative Abstraction of Piecewise Affine Systems. In *Proc. QR'08*, June 24-26 2008.

B. Kuipers. Qualitative simulation. *Artificial intelligence*, 29(3):289–338, 1986.

J. Lunze. Fault diagnosis of discretely controlled continuous systems by means of discrete-event models. *Discrete Event Dynamic Systems*, 18(2):181–210, 2008.

P. Maier and M. Sachenbacher. Constraint optimization and abstraction for embedded intelligent systems. *Lecture Notes in Computer Science*, 5015:338, 2008.

A. Olivero, J. Sifakis, and S. Yovine. Using abstractions for the verification of linear hybrid systems. *Lecture Notes in Computer Science*, 818:81–94, 1994.

R. Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32:57–96, 1987.

L. Saitta, P. Torasso, and G. Torta. Formalizing the Abstraction Process in Model-Based Diagnosis. *Lecture Notes in Computer Science*, 4612:314, 2007.

M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.

B. Shults and B.J. Kuipers. Proving properties of continuous systems: qualitative simulation and temporal logic. *Artificial Intelligence*, 92(1-2):91–129, 1997.

O. Sokolsky and H.S. Hong. Qualitative modeling of hybrid systems. In *Proc. Workshop on Formal Models in Software Development*, June 2001.

E.W. Stark. Concurrent transition systems. *Theoretical Computer Science*, 1989.

A. Tiwari. Abstractions for hybrid systems. *Formal Methods in System Design*, 32(1):57–83, 2008.

A. Tiwari and G. Khanna. Series of abstractions for hybrid automata. *Lecture Notes in Computer Science*, pages 465–478, 2002.

O. Yilmaz and A.C.C. Say. Causes of Ineradicable Spurious Predictions in Qualitative Simulation. *Journal of Artificial Intelligence Research*, 27:551–575, 2006.

F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, and P. Cheung. Monitoring and fault diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(6):1225–1240, 2005.