Detecting Unforeseen Behaviors During Health Monitoring of Hybrid Systems*

Hatte Leonie⋆ Hamadi Rayen⋆ Chanthery Elodie⋆ Ribot Pauline⋆

***Abstract*— Cyber-physical systems (CPS) are intrinsically hybrid and evolving over time, making their monitoring and diagnosing inherently complex. They require adaptive models capturing unexpected degradations and unforeseen behaviors. Such systems can be modeled by hybrid automata that combine continuous and discrete dynamics. This paper presents HyMED (Hybrid Model Enrichment for Diagnosis), a method that maintains and updates such models from multivariate time series. HyMED detects unforeseen behaviors and embeds a diagnoser estimating health modes. Focusing on the model consistency evaluation step, this work shows how a comparison step triggers updates, ensuring reliable models, as demonstrated through a case study.**

Introduction

Cyber–physical systems (CPS) modeled as hybrid systems are inherently complex, combining physical and digital layers. Their dynamics involve both continuous evolutions (physical processes) and discrete events driven by the outside environment or operators. These systems exhibit multivariate and multiscale behavior and are exposed to noise, delays, and various sources of uncertainty. In practice, hybrid systems are commonly represented using formal hybrid modeling frameworks, such as Hybrid Automata (HA [1]). HA provides rigorous yet intuitive formalism for capturing both discrete and continuous dynamics, and have been successfully applied across various domains including biology and medicine, robotics and industrial processes, cyber-security, and automotive systems ([2], [3], [4], [5]). Thanks to their expressiveness and formal properties, HA are now standard tools for diagnosis and control in hybrid systems.

Let us distinguish between the system (the real CPS under observation) and its model (the HA that approximates its behavior). While the system continuously evolves due to the emergence of new behaviors, the addition of actuators and sensors, or unforeseen faults and degradations, the model remains fixed after the design stage. As a result, discrepancies appear between the model and the system. These evolutions pose major challenges for health monitoring, i.e. diagnosis (estimating the current health state of the system) and prognosis (estimating its future evolutions), as well as maintenance; tasks which rely on the availability of an accurate and reliable model of the system. Re-identifying the full model at every deviation is computationally expensive, time-consuming, and requires significant expert knowledge.

To address this limitation, we propose a methodology that allows the model to be updated online as the system evolves, without complete re-identification or expert knowledge. The proposed approach, named HyMED (Hybrid Model Enrichment for Diagnosis), updates the HA dynamically by refining its structure and continuous dynamics whenever new behaviors are observed. This update can involve the addition of new states, or transitions, extending beyond simple parameter tuning. The objective is to build resilient models able to maintain diagnostic performance even when unforeseen events or environmental changes occur. Such adaptability is essential for accurate diagnosis and reliable prognosis.

This paper focuses on the comparison step of the HyMED methodology, strongly interleaved with a diagnosis step, which determines whether the current model remains consistent with the observed data, or needs structural updating. The comparison step quantifies the similarity between observed and expected dynamics across all system variables and uses these indicators to guide the model update process. To our knowledge, HyMED is the first updating framework specifically designed for health monitoring, integrating a diagnoser: HeMU (Heterogeneous system Monitoring under Uncertainty).

The paper is organized as follows. Section I presents the background on Hybrid Automata and introduces the HyMED methodology together with its embedded HeMU diagnoser. Section II details the proposed comparison step, discusses related work, and highlights the contribution of this paper. Section III motivates the choice of a calibrated similarity threshold for the comparison step, and Section IV illustrates the method on an evolving two-room thermal system, showing how HyMED detects unforeseen behaviors and updates the model accordingly. Finally, Conclusion summarizes the main results and outlines future research directions.

I. Background and methodology

A. Hybrid Automata

The Hybrid Automaton formalism formally describes hybrid systems that combine continuous dynamics with discrete event driven transitions. An HA is typically defined as a tuple $(Q, X, E, F, T, Guard, q_0, x_0)$, where [1]:

- Q is the set of discrete states (called modes)
- $X \in \mathbb{R}^n$ is the continuous state space with $n \in \mathbb{N}$ representing the number of continuous variables
- E is the set of discrete events
- $F : Q \times X \to X$ describes the flows or laws of continuous variables within each state
- $T : Q \times X \times E \to Q$ defines the set of transitions

*This work was not supported by any organization

⋆LAAS-CNRS, Université de Toulouse, INSA, UPS, Toulouse, France
lhatte@laas.fr, echanthe@laas.fr, pribot@laas.fr

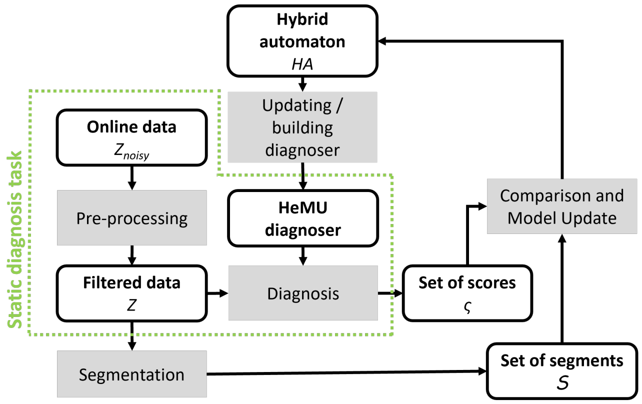

Fig. 1: HyMED overall diagram

- $Guard \subseteq Q \times Q \times X$ is the set of guard conditions for each transition
- $q_0 \in Q$ and $x_0 \in X$ are the initial discrete and continuous state.

A state denotes the pair $(q, x) \in Q \times X$, combining the discrete mode q with its continuous valuation x. The system evolves according to the flow $f(q, x)$ associated with the current mode until a guard or external event enables a transition $t(q, q')$, leading to the new state (q', x').

B. *HyMED methodology*

HyMED is an approach designed to keep HA models consistent with real observed system behavior, even under unforeseen degradations and faults. It is outlined in Fig. 1, while this paper specifically focuses on the *Comparison* step.

HyMED starts with an observed system and its model HA, previously identified. The goal is to monitor the health of the system while enriching HA using multivariate time series (TS) $Z_{noisy} = \{\tilde{z}_1, \ldots, \tilde{z}_n\}$, where each observation $\tilde{z}_i \in \mathbb{R}^m$ represents a vector of m variables.

The first step of HyMED is the *Pre-processing* phase, where online noisy observations of the system Z_{noisy} are filtered online to obtain a denoised TS Z that improves segmentation quality. These filtered data are then processed by HeMU [6], a health-monitoring framework designed to handle noise, missing data, and partial observability. Given the model HA and the Z, the HeMU diagnoser continuously estimates the current health state by simulating system evolution and correcting predictions using observations. It computes for each mode q a belief score $\varsigma(q)$, $\varsigma : Q \to \mathbb{R}^+$ indicating how likely mode q is to be the current mode.

HyMED extends this static diagnosis task (Fig. 1, green box) with enrichment capabilities. The *Segmentation* step [7] partitions Z into segments S, each representing a coherent system behavior. Segmentation is performed automatically using an online change-point detection algorithm [7] that monitors the multivariate signal and triggers a new segment whenever a statistically significant shift in the dynamics of at least one variable is detected, without requiring prior knowledge of when or how the system evolves. A segment is any subsequence $S = \{z_{init}, \ldots, z_{end}\} \subset Z$ containing $|S|$ observations of the same m variables.

Each segment S, together with the diagnoser's scores $\varsigma(q)$ for each mode $q \in Q$, feeds the *Comparison* step, which checks whether HA still reflects the system behavior. This step constitutes the main contribution of this article and is detailed in Section II. If the observed and expected dynamics are consistent, the model is left unchanged; otherwise, the *Model Update* procedure creates new modes and transitions, and identifies the new continuous dynamics. Since the HeMU diagnoser is derived from *HA*, it is updated accordingly.

HyMED thus iteratively maintains both HA and its HeMU diagnoser, ensuring minimal yet sufficient modifications to preserve model validity. For each new segment S, HeMU continues from the previously computed scores and from the last identified (or newly created) mode. To our knowledge, it is the first diagnosis framework capable of automatically updating an HA online from real system data to support reliable health monitoring.

II. Detection of unforeseen behaviors

A. *Related work*

Validating a model, either to confirm identification accuracy or to trigger a model update, requires comparing system observations with the model (or at least simulated data). In the literature, the task is referred to as similarity detection, dissimilarity detection, model validation, membership query or accuracy verification. The main approaches include elastic alignment methods, distance indicators, correlation scores, and envelope-based consistency tests.

Concerning elastic methods for similarity detection, Dynamic Time Warping used in [5], [8], compares misaligned or stretched time series. It is strongly efficient but computationally expensive and lacks a universal similarity threshold. Cross-correlation, employed in [9] is another elastic method that measures similarity under time shifts. It is useful for identifying temporal dependencies when TS are misaligned.

Simpler approaches for similarity detection compute direct distances between TS like in [10]. The Mean Squared Error and Root Mean Squared Error (RMSE) are widely used as in [2]. RMSE is especially interpretable, as it shares the same units as the data. However, it is sensitive to noise and outliers. The Mean Absolute Error (MAE), in contrast, is more robust to outliers, since each deviation contributes proportionally. All these indicators require thresholds, usually provided by expert knowledge, to determine acceptable similarity.

Correlation scores such as Pearson's coefficient (linear correlation), Spearman's rank correlation (monotonic relationships), and the coefficient of determination (R^2) can be used to assess similarity [11]. R^2 has the advantage of being scale-independent and does not require standardization, but it is primarily effective for regression-type models. It captures both the alignment and the shape of the dynamics, making it suitable for model validation.

In [12] is constructed a tolerance tube of width ε around simulated trajectories, that are considered consistent when they remain within this envelope.

Since HyMED operates online, computational efficiency is essential: each comparison must complete before the next transition to avoid missing system behaviors. The method must also handle multivariate and multiscale time series without letting high-magnitude variables to dominate the similarity measure. These requirements motivate the use of lightweight, interpretable, and scale-robust metrics rather than complex learning-based approaches.

We select the normalized MAE (Eq. (1)–(3)) as the similarity indicator: (i) MAE is more robust to outliers than MSE or RMSE, since each deviation contributes proportionally rather than quadratically; (ii) the per-variable normalization constrains residuals to $[0, 1]$, making the indicator scale-independent and preventing high-magnitude variables from dominating the comparison; (iii) this boundedness is the key hypothesis that enables the distribution-free threshold derivation in Section III.

B. Comparison step

The HyMED *Comparison Step* (Alg. 1), aims to assess whether the behavior of current observations is already represented within the hybrid automaton HA. The algorithm takes as inputs the current observed segment $S = \{z_{init}, \dots, z_{end}\} \subset Z$, the hybrid automaton HA, the belief scores $\varsigma(q)$ associated with each mode $q \in Q$, a calibrated similarity threshold $\tau_{\text{MAE}}(S) \in \mathbb{R}^+$, and the previous mode q_{prev} determined from $HeMU$. The set of modes Q is partitioned into two subsets (line 2): $Q = Q^+ \cup Q^0$, where Q^+ contains modes with nonzero belief scores, and Q^0 contains modes with zero belief scores.

Algorithm 1 HyMED comparison step

Input: S, HA, ς, α, m, q_{prev}

1: $\tau_{\text{MAE}}(S) \leftarrow \sqrt{\ln(m/\alpha)/2|S|}$ ▷ Sec. III
2: $Q^+ = \{q \in Q \mid \varsigma(q) > 0\}$; $Q^0 = \{q \in Q \mid \varsigma(q) = 0\}$
3: **for all** $q \in Q^+$ in descending $\varsigma(q)$ **do**
4: $\quad\hat{S} \leftarrow \text{simulate\_data}\,(q, S, f(q, z_{init}))$
5: $\quad$Compute $\text{MAE}_{\max}(S, \hat{S})$ ▷ Eq. (3)
6: $\quad$**if** $MAE_{\max} \le \tau_{\text{MAE}}(S)$ **then**
7: $\quad\quad$**return** (NoUpdate)
8: $\quad$**end if**
9: **end for** ▷ Model needs updating
10: **for all** $q \in Q^0$ **do**
11: $\quad\hat{S} \leftarrow \text{simulate\_data}\,(q, S, f(q, z_{init}))$
12: $\quad$Compute $\text{MAE}_{\max}(S, \hat{S})$ ▷ Eq. (3)
13: $\quad$**if** $\text{MAE}_{\max} \le \tau_{\text{MAE}}(S)$ **then**
14: $\quad\quad$**return** (AddTransition(HA, q_{prev}, q))
15: $\quad$**end if**
16: **end for**
17: **return** (AddState(HA, S)),
18: $\quad\quad$(AddTransition(HA, q_{prev}, q))

For every mode $q \in Q^+$, we simulate the dynamics defined by the flow $f(q, z_{init})$ to obtain a synthetic segment $\hat{S}$ (simulate_data(), line 4). The simulation is initialized using the initial condition z_{init} (first observation of segment S), and produces the same number of data points as S. This guarantees temporal and dimensional alignment between the observed and simulated trajectories.

The similarity between the simulated $\hat{S}$ and observed segment S is quantified for each variable $j \in \{1, \dots, m\}$ at time t via the normalized absolute residual:

$$r_j(t) = \frac{\left|S_j(t) - \hat{S}_j(t)\right|}{\max_t S_j(t) - \min_t S_j(t)}. \tag{1}$$

The per-variable MAE is then simply the mean residual:

$$\text{MAE}_j\big(S, \hat{S}\big) = \frac{1}{|S|}\sum_{t=1}^{|S|} r_j(t), \tag{2}$$

and the global similarity indicator (line 5) is the worst-case MAE across all m variables, ensuring that a discrepancy in any dimension is enough to trigger a model update when needed:

$$\text{MAE}_{\max}\big(S, \hat{S}\big) = \max_{j \in \{1,\dots,m\}} \text{MAE}_j\big(S, \hat{S}\big). \tag{3}$$

The comparison step leads to three mutually exclusive cases (Alg. 1, lines 7,14,18), formally defined as follows.

(i)NoUpdate: if the diagnoser recognizes the current behavior and the similarity indicator confirms it,

$$\exists\, q \in Q^+ \text{ s.t. } \text{MAE}_{\max}(S, \hat{S}) < \tau_{\text{MAE}}(S), \tag{4}$$

the model is left unchanged and the diagnoser resumes from the accepted mode q.

(ii)AddTransition(HA, q_{prev}, q)**:** if a matching mode q in HA exists but is unreachable according to the previous mode q_{prev} according to the diagnoser,

$$\begin{aligned}&(\nexists\, q \in Q^+ \text{ s.t. } \text{MAE}_{\max}(S, \hat{S}) < \tau_{\text{MAE}}(S)) \wedge \\ &(\exists\, q \in Q^0 \text{ s.t. } \text{MAE}_{\max}(S, \hat{S}) < \tau_{\text{MAE}}(S)),\end{aligned} \tag{5}$$

a new transition t_{new} is added from the last detected mode q_{prev} to q and its guard is learned. Various methods can be used to learn the guards of hybrid systems, which is a complex problem [13] that we are currently investigating. For now, the guards are set to "true", which, we acknowledge, alters the semantics of HA and makes it a non-deterministic switched system, which has a significant impact on the diagnosability and verifiability of the model.

(iii)AddState(HA, S)**:** if no existing mode in HA matches the observations S,

$$\nexists\, q \in Q \text{ s.t. } \text{MAE}_{\max}(S, \hat{S}) < \tau_{\text{MAE}}(S), \tag{6}$$

a new mode q_{new} is created, its continuous flow $f(q_{\text{new}})$ is identified via symbolic regression (SR) (PySR [14]), and a transition t_{new} from q_{prev} to q_{new} is added to the automaton.

SR is applied to each variable independently, searching for analytical expressions restricted to polynomial and rational functions of bounded degree, given by expert knowledge: additions, multiplications, and standard elementary functions. This constraint imposed on the model class limits the space of hypotheses; it should reduce the risk of overfitting on short segments and improving interpretability. To further mitigate overfitting, PySR minimizes a complexity-penalized loss, trading expression length against fit quality via a Pareto front. Identifiability on short segments is not guaranteed in general: a segment of length $n < p$ (where p is the number of free parameters of the learned expression) is underdetermined. Regarding stability, the learned flow $f(q_{\text{new}})$ is not constrained to be stable by construction; convergence of the enrichment process therefore relies on the assumption that the underlying system is itself stable within each mode.

Alg. 1 ensures that HyMED only updates the model when necessary, preserving computational efficiency while maintaining consistency between the evolving system and its hybrid representation in health monitoring. To the best of our knowledge, HyMED is the first approach to integrate a diagnoser and selectively compare the observed segment with the most probable modes of the model, rather than exhaustively testing all modes. In contrast, methods such as [5] perform comparisons against every mode of the model, which quickly becomes computationally prohibitive for large systems. By focusing on the most likely modes identified by HeMU, HyMED achieves an efficient comparison process tailored to diagnosis of evolving systems.

III. Choice of a calibrated similarity threshold

Industrial CPS are subject to multiple, simultaneous, and heterogeneous noise sources: *Gaussian* noise, *bias* from sensor miscalibration, *drift*, *salt-and-pepper* (sporadic outliers) from communication errors, and *heteroscedastic* noise when sensor noise depends on operating conditions. Deriving a noise-adapted threshold would require identifying the noise type and characteristics from nominal residuals via statistical tests, each carrying minimum segment-size requirements to be determined and their own false-positive rates. When several noise types happen at the same time, the order of detection and correction order impacts the computed threshold value. These difficulties motivate a distribution-free approach: we derive a formal lower bound on the calibrated similarity threshold $\tau_{\text{MAE}}(S)$ used in Alg. 1 holds regardless of the noise distribution, under a weak structural assumption on the residuals.

A. Setup and hypothesis

The goal is to choose $\tau_{\text{MAE}}(S)$ such that, when the model is correct, the probability of a false structural update is at most α, a user-defined upper bound on the false update rate per segment i.e $P(\text{MAE}_{\max} > \tau_{\text{MAE}}(S)) \leq \alpha$.

The Hoeffding inequality would provide such a bound for independent and identically distributed (i.i.d.) residuals, but consecutive residuals in HyMED share information through the system dynamics, making the i.i.d. assumption too strong. We instead rely on the weaker *Martingale Difference Sequence* (MDS) property: a sequence is an MDS if its past does not help predict its future mean, while past values may still influence future variance. Formally, $(\tilde{r}_j(t))_{t=1}^n$ is an MDS if:

$$\mathbb{E}\big[\tilde{r}_j(t) \mid r_j(1), \ldots, r_j(t-1)\big] = 0, \tag{7}$$

almost surely for all $t \geq 1$, where $\tilde{r}_j(t) = r_j(t) - \mu_j$ and $\mu_j = \mathbb{E}[\text{MAE}_j]$. This condition tolerates temporal dependence in variance. Its practical validity in HyMED is discussed in the conclusion.

The MDS condition enables the Azuma–Hoeffding inequality [15], a generalization of Hoeffding's inequality to dependent sequences, under two hypotheses:

Hypothesis 1 (Boundedness): The normalization in Eq. (1) guarantees $r_j(t) \in [0,1]$ for all j and t by construction, regardless of the noise distribution.

Hypothesis 2 (Martingale Difference Sequence): The centered residual sequence $(\tilde{r}_j(t))_{t=1}^n$ satisfies Eq. (7) for all j.

B. Calibrated threshold derivation

Under these two hypotheses, the Azuma–Hoeffding inequality [15] states that for any bounded MDS (D_t) with $|D_t| \leq c$ almost surely:

$$P\left(\frac{1}{n}\sum_{t=1}^{n} D_t \geq u\right) \leq \exp\left(\frac{-2nu^2}{c^2}\right), \quad \forall\, u > 0. \tag{8}$$

Proposition 1 (Distribution-free bound on $\text{MAE}_{\max}$): Under Hypotheses 1 and 2, for any $\tau > 0$, $n = |S|$ samples, m variables, and $\mu_{\max} = \max_j \mu_j$:

$$P(\text{MAE}_{\max} > \mu_{\max} + \tau) \leq m \cdot \exp(-2n\tau^2). \tag{9}$$

The probability that the worst-case MAE exceeds its expected value by more than τ decreases exponentially with n and grows at most linearly with m. The proof applies Eq. (8) to each centered residual sequence $(\tilde{r}_j(t))$ with $c = 1$ (Hypothesis 1), yielding $P(\text{MAE}_j > \mu_j + \tau) \leq \exp(-2n\tau^2)$ for each j. Boole's union bound over m variables gives Eq. (9).

Under H_0 (correct model), the normalized residuals satisfy $\forall j, \mu_j = 0$, so $\mu_{\max} = 0$ and Eq. (9) simplifies to $P(\text{MAE}_{\max} > \tau) \leq m \cdot \exp(-2n\tau^2)$. Setting this bound equal to α and solving for τ yields Eq. (10).

$$\tau_{\text{MAE}}(S) \geq \sqrt{\frac{\ln(m/\alpha)}{2|S|}}, \tag{10}$$

depending only on m, α, and $|S|$, all known at comparison time. This is a lower bound: values below carry no formal guarantee on the false update rate and values above conservatively ensure at most α false updates per segment. Unlike a fixed threshold, it adapts to segment length: short segments yield larger τ_{MAE}, tolerating more variability; long segments yield smaller τ_{MAE}, enforcing stricter consistency.

IV. Illustrative example

A. Benchmark

To illustrate the HyMED methodology and validate its comparison step, a simplified two-room thermal system is used. The full scenario is available in the git repertory[1]. The setup consists of two adjacent rooms R1 and R2 that exchange heat through a door and, for R2, through a window

[1] gitlab.laas.fr/lhatte/hymed-comparison

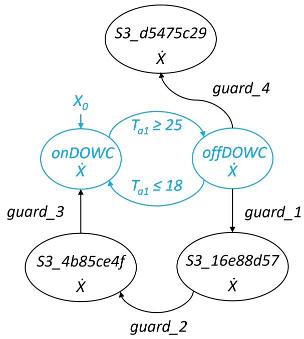

Fig. 2: Nominal HA in blue, modeling a two-room heating system and its enrichment in black

W to the outside environment. Each room (R1, R2) has a heater (H1, H2). Despite its apparent simplicity, this example captures key features of hybrid dynamical systems (modes, exogenous events, added sensors and actuators during the life-cycle) that make it well-suited for health monitoring and model update. In the illustrative example, segment boundaries coincide with known system events for evaluation purposes; in general operation in HyMED, segmentation is performed automatically without a priori knowledge.

Fig. 2 shows the initial HA (in blue), which contains two modes $Q = \{onDOWC, offDOWC\}$ corresponding to H1 being on or off with door open and window closed. Each mode $q \in Q$ is associated with continuous-time dynamics $\dot{x} = f(q, x)$, and transitions are triggered by hysteresis thresholds on the air temperature T_{a_1} of R1. H1 is instrumented, and its core temperature is measured through T_{c_1}, while R2 initially has no active heating; its temperature T_{a_2} is observed but not controlled. Discrete events $E = \{$*window open, window close, door open, door close, H1 on, H1 off, H2 on, H2 off*$\}$ may occur but are not instrumented, and therefore appear as unmodeled exogenous events from the diagnoser's perspective.

Online measurements are segmented into segments $\{S_1, \ldots, S_{13}\}$. From S_1 to S_6, the system follows the nominal behavior encoded in the initial HA, with H1 operating under its hysteresis controller. A first exogenous change occurs in S_7 and S_8 when the window is opened, causing a temperature drop and triggering H1 to compensate. From S_9 to S_{12}, the window is closed again and the system returns to nominal behavior. Finally in S_{13}, H2 is activated, inducing a temperature increase in both rooms and revealing a new behavior not represented in the initial HA.

B. Focus on comparison and model update steps

The HyMED loop operates segment by segment, progressively enriching the HA as new behaviors appear. Table I summarizes, for each segment, the HeMU diagnoser scores, comparison indices, and resulting decisions from Alg. 1. At runtime, the current segment S and current HA are processed by HeMU, which computes the belief scores $\varsigma(q)$ for all modes $q \in Q$. These scores are shown in the fourth column of Tab. I. The segment is then evaluated using Alg. 1 with a calibrated threshold $\tau_{\mathrm{MAE}}(S)$.

Case 1: NoUpdate. Segments S_1 to S_5 constitute the initialization batch. They allow HeMU to converge toward the correct mode, yielding final scores $\varsigma(onDOWC) = 50$ and $\varsigma(offDOWC) = 0$, indicating that the last mode of the batch is *onDOWC*. When S_6 is processed, HeMU identifies the current mode such that $\varsigma(offDOWC) = 50$, $offDOWC \in Q^+$. The comparison step confirms the match: $\mathrm{MAE}_{\max} = 0.002 < \tau_{\mathrm{MAE}}(S_6) = 0.095$, so no structural update is performed. The diagnoser's initial state is updated to *offDOWC*.

Case 2: AddTransition. A transition is added when no Q^+ mode satisfies the similarity constraint and at least one mode in Q^0 shows acceptable similarity, indicating that the current behavior exists in the HA but is unreachable. This occurs for segment S_9. At this stage, the HA contains four modes and HeMU assigns zero belief scores to all of them. The comparison step evaluates modes in Q^0 and detects a strong similarity with *onDOWC* ($\mathrm{MAE}_{\max} = 0.002 < \tau_{\mathrm{MAE}}(S_9) = 0.198$). However, this mode is not reachable from the last detected mode *S3_b6b5bb3e* (introduced at S_8). The algorithm therefore adds a new transition from *s3_b6b5bb3e* to *onDOWC*.

Case 3: AddState and AddTransition. The final case occurs when neither Q^+ nor Q^0 provides a satisfactory match. This situation appears at S_7:$\varsigma(offDOWC) = 50$, $offDOWC \in Q^+$, is rejected since $MAE_{\max} = 0.769 > \tau_{\mathrm{MAE}}(S_7) = 0.153$, and $onDOWC \in Q^0$ is also rejected with $\mathrm{MAE}_{\max} = 10.725$. The model is therefore considered incomplete, and a new mode *S3_a403157c* is created. Its continuous dynamics are learned from the segment data using SR. The new mode is connected to the previously visited mode *offDOWC* through a new transition, and becomes the starting point for the next iteration.

For brevity, the description focuses on two representative AddState events; the complete sequence of updates, from S_6 to S_{13}, including the third added state, is reported in Tab. I and illustrated in Fig. 2.

This case study shows that HyMED can detect unforeseen behaviors and integrate new operating conditions by iteratively updating the hybrid model. Structural updates follow a systematic process: (i) adding a mode and/or transition when needed, (ii) learning the corresponding flow equations, and (iii) learning guards. Each update is followed by processing the next incoming segment.

C. Discussion on sensitivity analysis

To assess the robustness of the proposed framework, controlled noise was injected into the signals during experiments. The five noise types introduced in Section III were tested independently, with varying characteristic parameters.

For moderate perturbations (Gaussian noise with a standard deviation of 0.2 and salt-and-pepper noise with outlier probabilities up to 0.08) the comparison step produces an HA structurally identical as in the noise-free case model (same modes and arcs, different flows and guards). The learned HA structure is faulty when the noise intensity increases:

TABLE I: Model validation and structural update for each segment.

S	$\tau_{\mathrm{MAE}}(S)$	q (partition)	$\mathrm{MAE}_{\max}$	Decision
init: S_1–S_5	–	*onDOWC* (Q^+)	–	–
		offDOWC (Q^0)	–	
S_6	0.095	*offDOWC* (Q^+)	0.002	**NoUpdate**
S_7	0.153	*offDOWC* (Q^+)	0.77	**AddState, AddTrans**
		onDOWC (Q^0)	10.7	
S_8	0.038	*onDOWC* (Q^0)	43.7	**AddState, AddTrans**
		offDOWC (Q^0)	10.7	
		s3_a403157c (Q^0)	$\gg 10^{20}$	
S_9	0.198	*onDOWC* (Q^0)	0.002	**AddTrans.**
S_{10}	0.066	*onDOWC* (Q^+)	1.78	**AddTrans.**
		offDOWC (Q^0)	0.001	
S_{11}	0.103	*s3_a403157c* (Q^+)	1854	**NoUpdate**
		onDOWC (Q^+)	0.003	
S_{12}	0.048	*onDOWC* (Q^+)	6.0	**NoUpdate**
		offDOWC (Q^+)	0.001	
S_{13}	0.036	*s3_a403157c* (Q^+)	$\gg 10^{20}$	**AddState, AddTrans**
		offDOWC (Q^+)	0.70	
		onDOWC (Q^0)	4.1	
		s3_b6b5bb3e (Q^0)	25.6	

(up to 0.4 for Gaussian noise and probability of outliers up to 0.16) both produce incorrect structural decisions. Bias noise systematically causes faulty updates, as a constant offset directly shifts the MAE above $\tau_{\mathrm{MAE}}(S)$ regardless of its magnitude. This behavior comes from the nature of the residuals:

$$r_j(t) = \underbrace{\left[x_j^{\text{true}}(t) - x_j^{\text{model}}(t)\right]}_{\text{model error}} + \underbrace{\epsilon_j(t)}_{\text{measurement noise}} . \tag{11}$$

The similarity indicator $\mathrm{MAE}_{\max}$ is blind to the origin of $r_j(t)$: it cannot distinguish a genuine structural deviation from a stochastic perturbation. When noise is sufficiently high, the residual is dominated by $\epsilon_j(t)$ and the calibrated threshold $\tau_{\mathrm{MAE}}(S)$ is crossed even if the model is correct.

However, this sensitivity is not necessarily a limitation in the context of health monitoring, which is the primary application of HyMED. A sensor drift, for instance, produces a residual that grows deterministically over time; if it exceeds $\tau_{\mathrm{MAE}}(S)$, HyMED creates a new mode capturing the drifted behavior. This new mode does not represent a physical operating condition of the system but a degraded observation condition: a sensor fault mode. From a diagnostic perspective, the existence of this mode can serve as a trigger for predictive maintenance. HyMED therefore does not aim to suppress the influence of noise on the model structure, but rather to integrate it faithfully. A model learned under noisy conditions reflects the actual observation conditions under which the system will be monitored, which is a desirable property for health monitoring.

The calibrated threshold $\tau_{\mathrm{MAE}}(S)$ controls the sensitivity of this integration: a higher α absorbs more noise without triggering updates, while a lower α captures finer deviations at the cost of more frequent structural updates. For $m = 2$, varying α from 0.01 to 0.1 changes $\tau_{\mathrm{MAE}}(S)$ by a factor of $\sqrt{\ln(200)/\ln(20)} \approx 1.33$, independently of the segment length $|S|$. Overall, the approach tolerates realistic noise while preserving the model structure.

Conclusion and perspectives

This paper presents the comparison step of the HyMED methodology, a model-based method for health monitoring of hybrid systems. HyMED continuously analyses online time-series data to assess the consistency between the real system and its hybrid automaton model, and triggers structural updates when unforeseen behaviors are detected. A central element of HyMED is the HeMU diagnoser, which guides the comparison and update decisions. Unlike existing approaches, HyMED explicitly integrates a diagnosis reasoning process into the model update loop. Experimental results on a two-room heating system illustrate its ability to identify emerging behaviors and modify the HA structure accordingly, highlighting its relevance for predictive maintenance and anomaly detection in evolving systems. Future work will focus on automating guard inference to achieve deterministic models and more reliable adaptation in real-world settings, on evaluating HyMED on larger benchmarks and . The calibrated threshold $\tau_{\mathrm{MAE}}(S)$ lower-bounds the false-update rate but is heuristic since the MDS assumption is only approximate; tighter guarantees accounting for error autocorrelation are also left for future work.

References

[1] T. A. Henzinger, "The theory of hybrid automata," in *IEEE Symp. on Logic in Computer Science*, 1996, pp. 278–292.

[2] I. Lamrani, A. Banerjee, and S. K. S. Gupta, "Hymn: Mining linear hybrid automata from input output traces of cyber-physical systems," in *IEEE Industrial Cyber-Physical Systems*, 2018, pp. 264–269.

[3] C. Lacoquelle, L. Travé-Massuyès, X. Pucel, N. B. Roa, and C. Merle, "Deviation tracking with incomplete and distorted data–application to motion trajectories of industrial robots," in *DX*, 2022.

[4] J. Quevedo, H. Sánchez, D. Rotondo, T. Escobet, and V. Puig, "A two-tank benchmark for detection and isolation of cyber attacks," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 770–775, 2018.

[5] I. Saberi, F. Faghih, and F. S. Bavil, "A passive online technique for learning hybrid automata from input/output traces," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 1, pp. 1–24, 2022.

[6] A. Vignolles, E. Chanthery, and P. Ribot, "A holistic advanced diagnosis approach for systems under uncertainty," in *Int. Workshop on Principle of Diagnosis*, 2021.

[7] L. Hatte, P. Ribot, and E. Chanthery, "Dynamic time series segmentation for health monitoring of hybrid systems," in *IEEE SMC*, 2025.

[8] A. Gurung, M. Waga, and K. Suenaga, "Learning nonlinear hybrid automata from input–output time-series data," in *Int. Symp. on Automated Technology for Verification and Analysis*, 2023, pp. 33–52.

[9] S. G. Douma, X. Bombois, and P. M. J. V. den Hof, "Validity of the standard cross-correlation test for model structure validation," *Automatica*, vol. 44, no. 5, pp. 1285–1294, 2008.

[10] R. Medhat, S. Ramesh, B. Bonakdarpour, and S. Fischmeister, "A framework for mining hybrid automata from input/output traces," in *Int. Conf. on Embedded Software*, 2015, pp. 177–186.

[11] D. R. Legates and G. J. McCabe Jr., "Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation," *Water Resources Research*, vol. 35, no. 1, pp. 233–241, 1999.

[12] M. G. Soto, T. A. Henzinger, and C. Schilling, "Synthesis of hybrid automata with affine dynamics from time-series data," in *Int. Conf. on Hybrid Systems: Computation and Control*, 2021, pp. 1–11.

[13] ——, "Synthesis of parametric hybrid automata from time series," in *Int. Symp. on Automated Technology for Verification and Analysis*, 2022, pp. 337–353.

[14] M. Cranmer, "Interpretable machine learning for science with PySR and symbolic regression," arXiv:2305.01582 [astro-ph.IM], 2023.

[15] K. Azuma, "Weighted sums of certain dependent random variables," *Tohoku Mathematical Journal, Second Series*, vol. 19, no. 3, pp. 357–367, 1967.