Kalpa Publications in Computing

Volume 4, 2018, Pages 37–51

28th International Workshop on
Principles of Diagnosis (DX'17)

An Algebraic Approach for Diagnosing Discrete-Time Hybrid Systems

Gregory Provan[1]

Department of Computer Science,
University College Cork,
Ireland
g.provan@cs.ucc.ie

Abstract

A broad range of real-world systems can be defined using discrete-time hybrid systems, e.g., chemical process plants and manufacturing systems. We characterize this application domain using a class of discrete-event systems, max-plus linear discrete-event systems, which captures synchronization without concurrency or selection. The model framework of these hybrid systems is non-linear in a conventional algebra, but linear in the max-plus algebra, thereby enabling linear-time inference. We use an observer-based framework for monitoring and diagnosing max-plus diagnostics models, and further improve computational efficiency by searching over only the most-likely space of behaviours. We illustrate our approach using a chemical process-control example.

1 Introduction

Diagnosing hybrid systems is known to be challenging, since tasks such as tracking and state estimation require different underlying mathematics, algorithms, and often inference tools for the continuous and discrete aspects. Faults might occur within the continuous aspects (e.g., valve stuck partially shut) or the discrete aspect (on/off actuator fails in the *on* state) [1]; these faults may evidence themselves as gradual or abrupt events. Some approaches (e.g., [2]) combine the two aspects, while other approaches map the aspects into a single framework, e.g., a probabilistic framework for which we can use particle filters or dynamic Bayesian networks to compute diagnoses [3].

This article uses a max-plus algebraic framework for diagnosing a class of hybrid systems that allow synchronization without concurrency or selection. We define a max-plus algebra over the max-plus semi-ring $\langle \mathbb{R}_{max}, \oplus, \otimes \rangle$, which is the set $\mathbb{R}_{max} = \mathbb{R} \cup \{-\infty\}$ together with operations $x \oplus y = max(x, y)$ and $x \otimes y = x + y$. The additive and multiplicative identities are taken to be $\epsilon = -\infty$ and $e = 0$, respectively.

We adopt the max-plus algebra, one of many idempotent semi-rings used for computational inference, not only because its operations are associative, commutative and distributive (as in conventional algebra), but also because it transforms inference on system timed dynamics (that are non-linear in a conventional algebra) to be linear in the max-plus algebra [4]. Max-plus methods have been widely used for modelling and inference, e.g., [5, 6], but have never been used for HS diagnostics inference.

We model the discrete transitions of a hybrid system (HS) with a discrete-event system (DES). The DES captures the event-driven state evolution depends entirely on the occurrence of discrete events over time. We assume a DES model whose state-space is described by a discrete set with transitions that are observed at discrete instants in time. Models describing a DES are non-linear in the conventional algebra. We model this class of DES with a max-plus linear discrete-event system in which inference is linear within the max-plus algebra [4, 7], and hence is computationally more efficient than traditional approaches [8].[1]

This article proposes a semi-ring algebraic framework for modeling and diagnostic inference of a hybrid system (HS). We describe a discrete-time hybrid system based on a $(max, +)$-linear (MPL) algebra defined over a set $\mathbb{Z}^+$ or $\mathbb{R}_{max} = \mathbb{R} \cup \{-\infty\}$ [9, 10]. We extend this model to a switching MPL (SMPL) framework [5], which introduces modes that the system switches between. We further extend SMPL systems with stochastic switching behaviors to capture the stochastic nature of faults occurring. We introduce stochastic fault occurrence through a probability distribution over mode transitions. The stochastic SMPL framework provides a rich theoretical basis for describing a set of real-world systems, e.g., piece-wise-affine (PWA) systems in the time-driven domain [11].

We employ a computationally efficient observer-based diagnostics approach to monitor the system and isolate faults. Our diagnostics approach uses the max-plus model for efficient inference, and also restricts the space of diagnoses considered during fault isolation by computing only the most-likely system behaviours (rather than using the space of all possible behaviours).

Finally, we show the generalizability of this algebraic approach, namely that just by changing the underlying algebraic operations we can define a range of stochastic hybrid systems, such as Markov switching systems or even non-linear systems whose dynamics typically are defined using particle filters [3]. We show how all the above approaches use the same formulation, and differ only in the underlying algebraic operations.

Our contributions are as follows:

- We model a HS using a switching $(max, +)$-linear (SMPL) algebra.
- We define a classical observer-based monitoring framework, and extend this for fault isolation.
- We show that an approximation technique can compute diagnoses in time polynomial in the problem size, even though the general diagnostic inference task is NP-hard.
- We show how we can modify our HS computations to fit several other stochastic frameworks without changing the state-space description, but only by modifying the underlying algebraic operations.

2 Problem statement

This section summarizes the underlying diagnostic task. Because it is computationally prohibitive to pre-define all possible fault dynamics within an observer-based FDI system (i.e., have an observer for every fault combination). As a consequence, we use an approximation approach that we describe below.

2.1 Objective

Consider a discrete-time affine system whose dynamics obeys one of μ possible models (known and observable), with one model corresponding to a system mode. Our objective, given a measured output $\boldsymbol{y}(k)$ that disagrees with the expected output $\hat{\boldsymbol{y}}(k)$, is to determine the system mode $\gamma(k)$ at time k that

[1] For example, in the max-plus algebra exponentiation reduces to conventional multiplication.

most closely generates the observed dynamics. To achieve this goal, we look for the (shortest) sequence of inputs $\boldsymbol{U}_{0,k} = (\boldsymbol{u}(0), ..., \boldsymbol{u}(k))$ and measurements $\boldsymbol{Y}_{0,k} = (\boldsymbol{y}(0), ..., \boldsymbol{y}(k))$ such that the output at time k is consistent with only one mode $\gamma(k) \in \Gamma$. Since multiple input sequences of minimal length l may satisfy this requirement, we select the mode that minimizes a given cost function. In the following, we assume that only one model is active during each discrete step in $[0, ..., N]$.

2.2 System Architecture

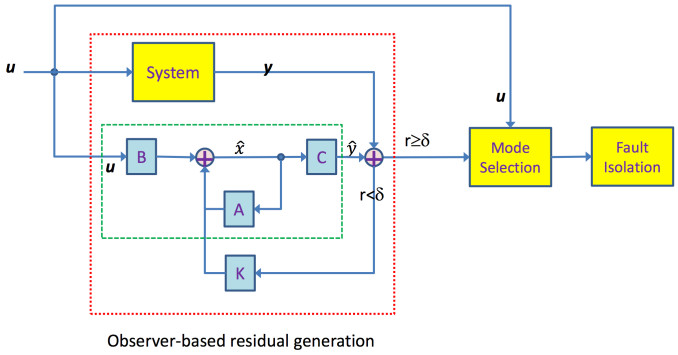

Figure 1: We diagnose hybrid systems using a 3-steps approach, with steps: (1) monitoring and residual generation, (2) mode selection and (3) fault isolation.

Our task consists of the 3-step process shown in Figure 1.

1. *Observer-based monitoring and residual generation*: In this phase we use an observer-based approach to detect anomalies, using a residual r. The residual generation process, and the matrices $A,\ B,\ C$, are described in Definition 3.

2. *Mode selection*: Given an anomaly, this phase computes the set $\Gamma^* \subseteq \Gamma$ of most-likely modes, and runs a simulation (given input $\boldsymbol{x}(0), \boldsymbol{U}_{0,k}$) for each mode $\gamma_i \in \Gamma^*$ to estimate the corresponding residual value $\boldsymbol{r}_i$.

3. *Fault isolation*: In this phase we compute the most likely failure mode for the system based on a cost function $\mathcal{J}(\boldsymbol{y}, \hat{\boldsymbol{y}})$ and the set of $\boldsymbol{r}_i$.

In this figure, we adopt the state-space model (as presented in Definition 4), where $A,\ B$ are the state matrices, C the output matrix, and K the observation matrix.

3 Related Work

This section described prior work related to our approach.

3.1 Diagnosing Hybrid Systems

Researchers have directed considerable attention to the monitoring and fault diagnosis of hybrid systems, e.g., [19, 20, 1].Our approach is the first to employ a (max,+) algebra for this task. The (max,+)

algebraic approach is computationally more efficient than existing approaches, but may suffer from requiring longer time delays for observations for fault isolation, and a limitation to a class of HS that can be diagnosed based on timing anomalies.

3.2 Algebraic Descriptions of Hybrid Systems

The class of switching MPL systems is related to (max,+) automata [10], which can also be characterized as non-stationary autonomous max-plus-linear systems with finitely valued dynamics (i.e. systems of the form $\boldsymbol{x}(k+1) = A(k) \otimes \boldsymbol{x}(k), \quad \boldsymbol{y}(k) = C \otimes \boldsymbol{x}(k)$ where $A(k)$ takes its values in a finite set $\{A(1), \cdots, A(N)\}$. The main differences are that the class of systems considered here have an additional input ($\boldsymbol{u}(k)$), and that we define the switching mechanism completely and explicitly for the switching max-plus-linear systems (which is not true for (max,+) automata).

We represent max-min-plus-scaling (MMPS) systems in state-space form using the operations maximization, minimization, addition and scalar multiplication. MMPS systems are equivalent to a particular class of hybrid systems, continuous piecewise (PWA) systems [12]. PWA systems are defined by partitioning the state space of the system in a finite number of polyhedral regions and associating to each region a different affine dynamic [13]. This relation between PWA and MMPS systems enables the study of certain structural properties of PWA systems, such as observability and controllability but also in designing controller schemes like model predictive control (MPC) [14].

Our work differs from other modeling techniques for discrete-event systems, such as Petri nets, extended state machines, event-graphs, formal languages, generalized semi Markov processes, nonlinear programming, automata, computer simulation models (see, e.g., [15, 16]), in that we employ an algebraic approach with observers.

3.3 Petri net models

Timed Petri nets include as a subclass timed event graphs (TEG), where we represent places as “arcs" and transitions as “nodes" [17]. In this case, all places have a single transition upstream (we remove competition in either consumption or supply of tokens in TEG) and a single one downstream (we resolve all potential conflicts in using tokens in places by some predefined policy). We gain computational advantage, although these limitations restrict some application domains; the limitations can generally be satisfied by making some design and scheduling decisions at an abstract (or hierarchical) level.

Diagnosis of Petri nets (e.g., [18]) has a rich history. What is different in our approach is the use of state-space descriptions together with observer-based monitoring, and computationally efficient methods based on the (max,+) algebra.

4 Max-plus Algebra and Switching Max-plus Linear Systems

This section summarizes the max-plus algebra, max-plus linear (MPL) systems, and switching max-plus linear (SMPL) systems. MPL systems can be generalized to capture a broad range of hybrid systems. By introducing modes we can capture switching behaviours [5]. We can introduce different forms of uncertainty in the model to capture different types of stochastic behaviours, e.g., see [21]. In this article we introduce uncertainty in mode switching, in order to capture uncertainty in the onset of fault modes.

4.1 Max-plus Algebra

This section outlines the basis for our algebraic frameworks. We first define $\epsilon = -\infty$ and $\mathbb{R}_{max} = \mathbb{R} \cup \{\epsilon\}$.

Definition 1 (max-plus-algebra [4, 22]). *A max-plus-algebra $\langle \mathbb{R}_{max}, \oplus, \otimes \rangle$, for numbers $x, y \in \mathbb{R}_{max}$ defines addition ($\oplus$) and multiplication ($\otimes$) as follows:*

$$x \oplus y = max(x, y) \quad (1)$$
$$x \otimes y = x + y, \quad (2)$$

We extend these to matrix operations as follows:

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = max(a_{ij}, b_{ij}), \quad (3)$$
$$[A \otimes C]_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes c_{kj} = \max_{k=1,\cdots,n} (a_{ik} + c_{kj}) \quad (4)$$

for matrices $A, B \in \mathbb{R}_{max}^{m \times n}$ and $C \in \mathbb{R}_{max}^{n \times p}$.

4.2 Max-plus Linear Systems

Max-plus-linear (MPL) systems are a class of discrete-event system that allow synchronization but no concurrency or choice [4]. We define MPL systems using two operators, *max* and $+$. The *max* function models the synchronization between events: an event occurs once all processes it depends on have finished. The $+$ function models the process times: the moment a process finishes must equal the sum of starting time and the time the process takes to finish. MPL systems are called max-plus-linear systems since the underlying temporal algebra has computational complexity that is "linear" in the max-plus algebra [4].

Definition 2 (Max-plus Linear system).

$$\boldsymbol{x}(k) = A(k) \otimes \boldsymbol{x}(k-1) \oplus B(k) \otimes \boldsymbol{u}(k), \quad (5)$$

with $A \in \mathbb{R}_{max}^{n \times n}$ and $B \in \mathbb{R}_{max}^{n \times m}$, with a number n of states and m of inputs.

The index k denotes the event counter. For MPL systems the state $\boldsymbol{x}(k)$[2] typically contains the time instants at which the internal events occur for the k^{th} time, the input $\boldsymbol{u}(k)$ contains the time instants at which the input events occur for the k^{th} time, the output $\boldsymbol{y}(k)$ contains the time instants at which the output events occur for the k^{th} time.

4.3 Switching Max-plus Linear Systems

We now extend our framework to cover systems that can switch between different modes of operation [5]. We assume that a system operates in some mode $\gamma \in \Gamma$, where $|\Gamma| = \eta$ modes. We partition Γ into a subset Γ_f of η_f fault modes and Γ_n of η_n nominal modes.

Definition 3 (Switching Max-plus-linear (SMPL) system). *A switching max-plus-linear (SMPL) state space model exists in mode $\gamma(k)$ for event step k as governed by*

$$\hat{\boldsymbol{x}}(k+1) = A^{(\gamma(k))} \otimes \hat{\boldsymbol{x}}(k) \oplus B^{(\gamma(k))} \otimes \boldsymbol{u}(k) \quad (6)$$
$$\hat{\boldsymbol{y}}(k) = C^{(\gamma(k))} \otimes \hat{\boldsymbol{x}}(k), \quad (7)$$

in which the matrices $A^{(\gamma(k))}, B^{(\gamma(k))}, C^{(\gamma(k))}$ are the system matrices for mode $\gamma(k)$.

[2] In this article, boldface variables denote vectors.

The switching allows us to model mode changes over both nominal and fault modes. Such mode switches include changes in the structure of the system, such as breaking a synchronization or changing the order of events. Each mode γ corresponds to a set of required synchronizations and an event order schedule, which leads to a model with system matrices $(A^{(\gamma(k))}, B^{(\gamma(k))})$ for the γ^{th} model. The mode $\gamma(k)$ determines which max-plus linear model is valid during the k^{th} event. The moments of switching are determined by a switching mechanism, which may be governed by the previous state $\boldsymbol{x}(k-1)$, the previous mode $\gamma(k-1)$, the input variable $\boldsymbol{u}(k)$ and an (additional) control variable $\boldsymbol{w}(k)$.

We partition $\mathbb{R}_{max}^{n_z}$ into η subsets $Z(i), i = 1, \cdots, \eta$. The mode $\gamma(k)$ is now obtained by determining the set that contains $\gamma(k)$ at event step k. So if $\gamma(k) \in Z(i)$, then $\gamma(k) = i$. The switching mechanism is application-dependent; in some systems it will depend on the state $\boldsymbol{x}(k-1)$ and input $\boldsymbol{u}(k)$, while in other examples $\gamma(k)$ will be governed by $\boldsymbol{w}(k)$.

4.4 Stochastic SMPL Systems

In real-world scenarios, fault transitions are stochastic. We can capture that behaviour in the SMPL framework using the mode transition behaviours (switching mechanism). In our original definition, the functional form of $\gamma(k)$ was left open. We can define stochastic failure-mode transitions, together with deterministic nominal-mode transitions, using a Markov transition matrix [21].

In this article, we assume that faults occur randomly, inducing random mode switches from a nominal mode to a fault mode. Once a fault occurs, it is persistent. We capture this using a stochastic variable π_{ij}, which defines the probability of switching from mode $\gamma_i(k-1)$ at time $k-1$ to mode γ_j at time k:

$$\pi_{ij} = P[\gamma_j(k)|\gamma_i(k-1)].$$

For example, we may have a stochastic switch from a nominal mode $\gamma_i(k-1)$ to a failure mode $\gamma_j(k)$, where the switching probability is 0.01.

We can define a switching probability matrix for the stochastic variable π_{ij} over η modes, with entries given by $\pi_{ij}, i, j = 1, ..., \eta$ as:

$$P_S = \begin{bmatrix} \pi_{11} & \cdots & \pi_{\eta 1} \\ \vdots & \ddots & \vdots \\ \pi_{1\eta} & \cdots & \pi_{\eta\eta} \end{bmatrix} \tag{8}$$

4.5 Generality of Approach

Our algebraic approach is general and extensible, in that we can maintain the problem structure and obtain a different problem by changing the semi-ring. For example, we can maintain the problem structure of definition 3, and simply by changing to the semi-ring $\langle [0,1], (+, \times) \rangle$, we obtain an HS defined by a dynamic Bayesian network [23]. Furthermore, we can still use the inference architecture of Figure 1 to solve this dynamic Bayesian network.

We make this change by defining $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{u}$ as stochastic variables, matrices $A,\ B,\ C$ as Markov transition matrices. With this modification of the model representation, we can apply the semi-ring operations over the semi-ring $\langle [0,1], (+, \times) \rangle$.

In an analogous fashion, we can substitute several different semi-rings into definition 3 to obtain different HS formulations, with no change in inference tools other than the semi-ring operations.

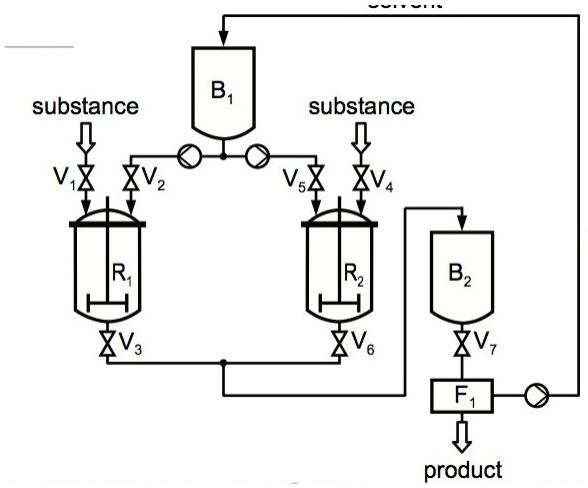

Figure 2: Chemical Process consisting of mixing two chemicals using a solvent. The process involves using solvent (from tank B_1) to dilutes two chemicals in reactors $R_1,\ R_2$, and then mix them in tank B_2.

5 Running Example

As a running example we use a chemical process, as shown in figure 2; this example is take from [24]. The process mixes two chemicals, $C_1,\ C_2$ using a solvent. The main steps are as follows:

1. We dilute $C_1,\ C_2$ with solvent (from tank B_1) in reactors $R_1,\ R_2$, respectively.
2. We mix the diluted chemicals together in tank B_2.
3. We filter the solvent into tank B_1, which it then used for the next cycle; the product is then extracted and packaged.

For this process we measure the times when the levels of reactors $R_1,\ R_2$ and tanks $B_1,\ B_2$ are full/empty, and have as controls the times when we open and shut the valves $V_1, ..., V_7$.

5.1 Discrete Nominal Model

This process follows a fixed sequence of discrete steps.

1. *Dilution*: We release C_1 by opening valve V_1 to dilute it with solvent by opening valve V_2 from tank B_1, storing the mixture in reactor R_1. We release C_2 by opening valve V_4 to dilute it with solvent by opening valve V_5 from tank B_1, storing the mixture in reactor R_2.
2. *Mixing*: We mix the diluted chemicals together in tank B_2 by opening valves $V_3,\ V_6$.
3. *Filtration*: We filter the solvent into tank B_1 by opening valve V_7.

We use $\boldsymbol{x}$ to denote the times when certain events occur, e.g., the reactors R_1/R_2 and tank B_2 are empty or full (see Table 1). We use $\boldsymbol{u}$ to denote the times when we open or close some of the valves (see Table 2).

x_i	Event	Transition
x_1/x_2	R_1/R_2 full (Start reaction 1/2)	q_3/q_7
x_3/x_4	R_1/R_2 empty	q_4/q_8
x_5	Start filter	q_9
x_6	Solvent available	q_{10}

Table 1: Correspondence of events and transitions for $\boldsymbol{x}$

$\boldsymbol{u}$	Event	Transition
u_1/u_2	Open V_1/V_2	q_1/q_2
u_3/u_4	Open V_4/V_5	q_5/q_6

Table 2: Correspondence of events and transitions for $\boldsymbol{u}$

We can represent this example in our state-space equation as

$$\boldsymbol{x}(k+1) = A(k) \otimes \boldsymbol{x}(k) \oplus B(k) \otimes \boldsymbol{u}(k),$$

where $\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6]$, $\boldsymbol{u} = [u_1\ u_2\ u_3\ u_4]$,

$$A(k) = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \tau_3 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \tau_6 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \tau_7 & \epsilon \end{bmatrix}$$

$$B(k) = \begin{bmatrix} \tau_1 & \tau_2 & \epsilon & \epsilon \\ \epsilon & \epsilon & \tau_4 & \tau_5 \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon \end{bmatrix}$$

In our observation equation $\boldsymbol{y} = C\boldsymbol{x}$, we assume a fully observable system, i.e., C is the identity matrix. The residuals are given by $r_i = |\hat{y}_i - y_i|$, for $i = 1, \cdots, 6$.

5.2 Continuous Model

We can define a continuous model for each of the process steps as follows. Consider filling reactor R_1 with chemical C_1 (flowing at rate ρ_1) and solvent (flowing at rate ρ_s). If the fluid level of reactor i, $i = 1, 2$ is denoted by h_i, and outflow through a valve with opening parameter ν_i, then we have equation

$$\dot{h_i} = \kappa(\rho_i + \rho_s - \rho_{oi}\nu_i), \tag{9}$$

where κ is a constant that incorporates tank diameter, material properties, etc.

5.3 Discrete Switching Model with Failure Modes

We now show how we model switching from the nominal mode to failure modes. We assume that a switch to a fault mode occurs stochastically, and is not observable except indirectly. Second, we assume that we have models for each failure scenario, where we represent a model in terms of matrix pair $\langle A^{\gamma(k)},\ B^{\gamma(k)}\rangle$ for mode $\gamma(k)$.

We model the impact of faults on the system using different processing times. For example, a leak fault in reactor R_1 leads to a reduction in the value of τ_3, and a partial blockage of valve V_6 leads to an increase in τ_6, the time taken to empty reactor R_2. If we allow component failures, then we can define a different matrix pair $\langle A^{\gamma(k)},\ B^{\gamma(k)}\rangle$ for each failure scenario.[3] For example, if we have a leak fault in reactor R_1, this will cause the reactor to empty more quickly than anticipated, i.e., reduce τ_3 to τ_{3f}. In this case we can simply replace τ_3 with τ_{3f} in matrix $A^{\gamma(k)}$. In an analogous manner, we can define specific matrices for every failure mode of the system.

6 Diagnosing Hybrid Systems using SMPL Automata

This section introduces an observer framework for monitoring SMPL Automata; we then extend this to isolate faults in these automata.

6.1 Observers

We now extend a max-plus-linear state space model that is in mode $\gamma(k) \in 1, \cdots, \eta$ for event step k into a monitored system using a residual vector $\boldsymbol{r}(k)$. We define our state specification with observer as follows:

Definition 4 (Observer State Space Model)**.**

$$\begin{aligned} \hat{\boldsymbol{x}}(k+1) &= A^{(\gamma(k))} \otimes \hat{\boldsymbol{x}}(k) \oplus B^{(\gamma(k))} \otimes \boldsymbol{u}(k) \\ &\quad \oplus K^{(\gamma(k))} \otimes \boldsymbol{r}(k) \qquad (10) \\ \hat{\boldsymbol{y}}(k) &= C^{(\gamma(k))} \otimes \hat{\boldsymbol{x}}(k) \qquad (11) \\ \boldsymbol{r}(k) &= |\hat{\boldsymbol{y}}(k) - \boldsymbol{y}(k)|, \qquad (12) \end{aligned}$$

where variables with a hat ($\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$) correspond to model predictions, and variable $\boldsymbol{y}$ (without a hat) corresponds to the measured output.

In this description, $K^{(\gamma(k))}$ is the observer gain matrix that we must tune. Further details of observer-based control can be found in [26].

Given an observer, we can monitor our system and identify anomalous behaviour using the residual as follows:

Definition 5 (Fault detection)**.** *Given a non-negative threshold $\delta \in \mathbb{R}$, an anomaly (corresponding to a fault) exists if $|\boldsymbol{r}(k)| > \delta$.*

6.2 Isolating Faults

This section describes a method for fault isolation given an anomaly. In general, we can isolate faults in this framework using a range of approaches, e.g., a bank of residual generators, ARRs, etc. A classical approach for fault isolation is to use a bank of residual generators, one for each fault to be diagnosed

[3] We assume that the observation matrix remains the same.

[8]. This approach can be easily accommodated in our framework; however, it does not scale well to the large number of possible multiple-fault combinations.[4] As a consequence, it is computationally prohibitive to search the entire space, and and using a bank of residual generators typically is limited to single-fault scenarios.

In the following we address the most-likely multiple-fault scenarios. In particular, we use the fault-transition probabilities to focus inference on the most-likely fault trajectories.

We need to introduce a few definitions to clarify our fault isolation procedure. We are interested in multiple-fault diagnoses, where we allow each failure mode $\gamma \in \Gamma_f$ to take on a discrete set of values.

Definition 6 (Trajectory)**.** *A trajectory is a sequence of events and states.*

Definition 7 (Observation sequence)**.** *An observation sequence is a sequence of observable events.*

For a stochastic transition, we compute the probability of a future state.

Definition 8 (Stochastic State Estimation Task)**.** *Given a sequence $\boldsymbol{Y}$ of k observable events and initial state $\boldsymbol{x}(0)$, the stochastic state estimation task is to identify $P(\boldsymbol{x}(k))$, the probability of state $\boldsymbol{x}$ at time k.*

We can use the switching probability matrix P_S to compute $P(\gamma(k)|P(\gamma(0)))$, where $\gamma(0)$ is the initial mode. P_S^k denotes the probability distribution over arriving at any mode after k steps. More precisely, the ij^{th} entry denotes the probability of moving from mode i to mode j after k steps. We can use this matrix to compute the probability $P(\gamma(k)|P(\gamma(0)))$ for any mode $\gamma \in \Gamma$.

We adopt as our baseline diagnostic approach the use of multiple observers, where we compute with each observer a residual tuned to a particular fault. We assume that we compute just the single-fault diagnoses with each residual. We will then compare this approach with a multiple-fault approach.

In this article we adopt an approximation-based approach for multiple-fault fault isolation, where we investigate the most-likely sub-space of the diagnosis space. To do this, we must compute the most-likely trajectories. Fortunately, these are easily computed using algebraic techniques. Assume that we identify an anomaly at k steps. Using a $(max, *)$-algebra where $*$ is standard multiplication, we can compute the probability of a fault occurring at k steps using P_S^k using the $(max, *)$-algebra, which computes the probability of paths of length k [27]. The entry π_{ij} in P_S^k denotes the maximum probability of the k-step path from mode i to mode j. We use a threshold δ_π such that we consider fault occurrence for fault j only if $\pi_{ij} \geq \delta_\pi$. We consider the set Γ_f of faults.

We identify the fault that minimizes the loss function $\mathcal{J}(\hat{\boldsymbol{y}}, \boldsymbol{y})$:

$$\gamma_f^* \;=\; \arg\min_{\gamma_f \in \Gamma_f} \mathcal{J}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \tag{13}$$

In this article, we use a probabilistic loss function, so we compute the highest-probability fault.

7 Computational Complexity

The complexity of diagnostic inference in a switching max-plus system (definition 3) depends on two factors:

Fault detection To identify an anomaly, we must solve our system to compute $\boldsymbol{r}$, which requires solving a matrix relation of the form given by equations 6 and 7.

Fault Isolation This phase of inference requires us to identify the failure mode that "explains" the anomaly, i.e., that minimizes our diagnostics cost function.

We now define the complexity of each factor in turn.

[4] The diagnosis space is exponentially-growing in Γ_f, the number of fault modes.

7.1 Fault Detection

Given an observation $\boldsymbol{y}(k)$ at time k, computing a residual $\boldsymbol{r}(k) = |\boldsymbol{y}(k) - \hat{\boldsymbol{y}}(k)|$ involves estimating the output using $\hat{\boldsymbol{y}}(k)$, which we can calculate from equations 6 and 7 as

$$\begin{aligned} \hat{\boldsymbol{y}}(k) &= C \otimes \hat{\boldsymbol{x}}(k) \quad \text{for } k = 1, 2, ... \\ &= C \otimes \left[A^{\otimes^k} \otimes \boldsymbol{x}(0) \oplus \bigoplus_{i=1}^{k} A^{\otimes^{k-i}} \otimes B \otimes \boldsymbol{u}(i) \right] \end{aligned}$$

Computing the k^{th} power of a matrix, for $k \in \mathbb{N}_0$, takes the form

$$(A^{\otimes^k})_{ij} = \max_{i_1, i_2, ..., i_{k-1}} (a_{ii_1} + a_{i_1 i_2} + ... + a_{i_{k-1} j}) \ \ \forall i, j.$$

This is clearly linear in the size of the matrix. From this, we can see that computing the residual is linear in the size of the matrices involved. This contrasts with traditional matrix operations, which are $O(n^3)$ for $n \times n$ matrices.

7.2 Fault Isolation

Isolating faults is the computationally taxing part of the problem. Below, we outline the worst-case complexity of this problem, and then show the approximation technique that we adopt.

We can define our diagnostic problem as follows:

Definition 9 (SMPL diagnosis)**.** *Given an SMPL system with initial condition $\boldsymbol{x}(0)$ and anomalous observation $\boldsymbol{y}$, compute a switching sequence ending with a persistent fault that generates an output $\hat{\boldsymbol{y}}$ such that $\mathcal{J}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is minimized over all permutations of switching sequences.*

We can use this problem formulation to prove a decision version of our diagnostics task.

Proposition 1 (SMPL diagnosis complexity)**.** *Given an integer SMPL system with initial condition $\boldsymbol{x}(0)$ and anomalous observation $\boldsymbol{y}(k)$ at time k, it is NP-complete to compute if there exists a switching sequence ending with a persistent fault that generates an output $\hat{\boldsymbol{y}}(k) = \boldsymbol{y}(k)$ at time k.*

We prove this result in [28]. The full diagnosis problem (definition 9) is an optimization version of the decision problem, so is NP-hard. The problem is the exponential number of switching sequences that must be analyzed.

7.3 Approximation Algorithm

We use an approximation technique to explore a polynomial number of switching sequences (trajectories), rather than the (worst-case) exponential number of switching sequences. We use the stochastic function governing fault transitions to assign probabilities to the trajectories, and explore only the trajectories whose probability is higher than a threshold φ. By controlling the value of φ we can limit the number of trajectories to be polynomial in $|\boldsymbol{x}|$. This gives us a principle way to trade off inference speed with fault isolation accuracy.

Alternatively, we could solve this problem as a mixed-integer linear programming problem [29], for which a number of efficient solvers exist.

8 Fault Isolation Example

8.1 Types of Faults

We consider three fault types and evaluate the performance of the diagnosis algorithm:

1. A transition fires earlier than expected. Such an observation occurs when, e.g., the time to empty a tank is decreased by a leak.
2. A transition fires later than expected. This behaviour can be observed when, e.g., the flow through a valve is reduced by a partial clogging.
3. An event no longer occurs. This may be due to a complete breakdown of a particular process component, e.g. a complete clogging of a valve. The firing time $x_i(k+N)$ of this particular transition will equal ∞, i.e. the events generated by the considered component never occur. In addition, all successor transitions of qi can not be enabled any more. Their firing time therefore also equals ∞. Obviously, a measurement of infinite firing times is not feasible. We therefore introduce a threshold T with the convention

$$x_i(k+1) = \infty \text{ if } t - x_i(k) \geq T,$$

where t denotes the absolute time. The threshold T is set based on the process operators expert knowledge. Consider a component corresponding to a predecessor place of transition q_i. A fault in this component, which occurs in the $(k+N)^{th}$ cycle, will disable transition q_i, such that $x_i(k+N) = \infty$ after the passage of time T.

The three scenarios discussed above, early events, delayed events and disabled events, represent the possible behaviour modes a discrete event system can generate when a fault occurs, even for faults that change the event sequence structure.

8.2 Stochastic Filtering

We now show how we can focus diagnostics on a subset of faults using our stochastic switching framework. Consider the space of faults:

Name	Description	Probability
f_1	R_1 leak	0.05
f_2	R_2 leak	0.01
f_3	V_1 clogging	0.01
f_4	V_6 clogging	0.05
f_5	V_4 blocked	0.03

Table 3: Examples of faults with associated probabilities

For each fault we can define a transition matrix. For example, for persistent fault f_1 the matrix is

$$P_{f_1} = \begin{bmatrix} .95 & .05 \\ 0 & 1 \end{bmatrix}$$

By taking powers of this matrix we can calculate path probabilities for failures, and use these for ranking which possible failures to consider. For example, the second power is:

$$P_{f_1}^2 = \begin{bmatrix} .9025 & .0975 \\ 0 & 1 \end{bmatrix}$$

We can generalize this approach to compute multiple-fault conditions, intermittent faults, etc.

We use the trajectory probabilities to avoid searching over all possible trajectories. At each time step, we select the subset S of system trajectories that cumulatively sum to a threshold κ. In our example, trajectories with fault combinations (2 or more faults) containing faults other then f_1 and f_4 are extremely unlikely, and get filtered, leading to the fault scenarios of the single-faults plus $\{f_1,\ f_4\}$.

Given the subset of failure modes we consider, for each trajectory $s_i \in S$ we run a simulation and compute residual r_i. Our diagnosis is the fault mode γ_j that minimizes this residual function.

8.3 Diagnosis of Valve Faults

We now show an example where we induce as a fault the partial clogging of valves $\{f_3,\ f_4\}$. We assume that the process reaction time can be neglected, such that τ_3, τ_6 represent only the outflow time from $R_1,\ R_2$ and the filtering time is determined by the flow through valve V_7. The relevant parameters are therefore the parameters $\tau_i, i = 1, \cdots, 6$, with the corresponding residuals $r_i = \tau_i - \hat{\tau}_i$.

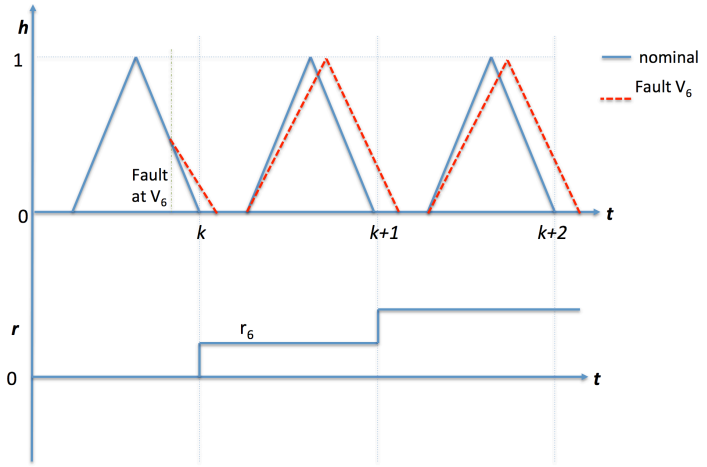

Figure 3: Simulation results for fault in valve V_6, with the nominal simulation shown as a solid line and the fault simulation as a dashed line.

Consider a fault consisting of partial blockage of valve V_6, which causes the reactor R_2 to drain more slowly than expected. Figure 3 shows the nominal simulation (as a solid line) versus the simulation of a fault (dashed line), together with the residual r_6 computed from the fault scenario. In both situations the reactor first fills with chemical and solvent, and then drains when V_6 is opened. The timings $k,\ k+1,\ k+2$ indicate the nominal events when reactor R_2 is empty. At time step k reactor R_2 still have not emptied fully in the faulty case, creating a non-zero residual. The residual increases at the $k+1^{st}$ time step since we have a full cycle of delay, as opposed to the incomplete cycle with less delay at the k^{th} step.

We introduce a probability threshold φ that limits the fault scenarios that we consider. For this example, we can define φ such that we consider only behaviours that result in a single persistent fault occurring.

Given our non-zero residual we run our set of possible fault scenarios, and compute a partial blockage of valve V_6 as the most likely fault at the $k+1^{st}$ time step.

8.4 Discussion

This diagnostics approach provides a computationally tractable method, using only discrete-time observations of event timings. The discrete models are based on timing information, and abstract the continuous dynamics in terms on timing information.

The benefits of this approach include simplicity of modelling and inference. The required level of abstraction may not suit all problems, although further research is necessary to determine that.

9 Summary

This article has proposed a max-plus algebraic approach for solving a class of PWA hybrid systems. For this class of system the max-plus algebraic approach is computationally faster than traditional methods. We have described an approximation technique that is of complexity polynomial in the problem size, even though the general diagnostic inference task is NP-hard. We have illustrated our approach on a process-control example.

This approach provides a novel computational framework for diagnosing hybrid systems. We build on a significant base of work on modeling and controlling systems using the max-plus algebra. There are many avenues for future work, including applying this approach to large systems to test scaling properties, studying the impact of switching probabilities on fault isolation accuracy, and comparing our approach to state-of-the-art methods.

References

[1] Janan Zaytoon and Stéphane Lafortune. Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control*, 37(2):308–320, 2013.

[2] Mehdi Bayoudh and Louise Travé-Massuyès. Diagnosability analysis of hybrid systems cast in a discrete-event framework. *Discrete Event Dynamic Systems*, 24(3):309–338, 2014.

[3] Xenofon Koutsoukos, James Kurien, and Feng Zhao. Monitoring and diagnosis of hybrid systems using particle filtering methods. In *International Symposium on Mathematical Theory of Networks and Systems*, 2002.

[4] F. Baccelli, G. Cohen, G.J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event Systems*. John Wiley and Sons Ltd., Chichester, 1992.

[5] Ton JJ van den Boom and Bart De Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Control engineering practice*, 14(10):1199–1211, 2006.

[6] Rabah Boukra, Sébastien Lahaye, and Jean-Louis Boimond. New representations for (max,+) automata with applications to performance evaluation and control of discrete event systems. *Discrete Event Dynamic Systems*, 25(1-2):295–322, 2015.

[7] P. Butkovic. *Max-linear systems: theory and algorithms*. Springer, 2010.

[8] Marcin Witczak. *Modelling and estimation strategies for fault diagnosis of non-linear systems: from analytical to soft computing approaches*, volume 354. Springer Science & Business Media, 2007.

[9] Stéphane Gaubert. Performance evaluation of (max,+) automata. *IEEE transactions on automatic Control*, 40(12):2014–2025, 1995.

[10] Stephane Gaubert. Methods and applications of (max,+) linear algebra. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 261–282. Springer, 1997.

[11] Ton JJ van den Boom and Bart De Schutter. Modeling and control of switching max-plus-linear systems with random and deterministic switching. *Discrete Event Dynamic Systems*, 22(3):293–332, 2012.

[12] E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automat. Contr.*, 26(2):346–358, April 1981.

[13] Domine Leenaerts and Wim MG Van Bokhoven. *Piecewise linear modeling and analysis*. Springer Science & Business Media, 2013.

[14] Alberto Bemporad and Manfred Morari. Robust model predictive control: A survey. *Robustness in identification and control*, pages 207–226, 1999.

[15] H Proth Hillion and J-M Proth. Performance evaluation of job-shop systems using timed event-graphs. *IEEE transactions on automatic control*, 34(1):3–9, 1989.

[16] Robin A Sahner, Kishor Trivedi, and Antonio Puliafito. *Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package*. Springer Science & Business Media, 2012.

[17] Lawrence E Holloway, Bruce H Krogh, and Alessandro Giua. A survey of petri net methods for controlled discrete event systems. *Discrete Event Dynamic Systems*, 7(2):151–190, 1997.

[18] Francesco Basile. Overview of fault diagnosis methods based on Petri net models. In *Control Conference (ECC), 2014 European*, pages 2636–2642. IEEE, 2014.

[19] Shai A Arogeti, Danwei Wang, and Chang Boon Low. Mode tracking and FDI of hybrid systems. In *10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 892–897. IEEE, 2008.

[20] Feng Zhao, Xenofon Koutsoukos, Horst Haussecker, James Reich, and Patrick Cheung. Monitoring and fault diagnosis of hybrid systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1225–1240, 2005.

[21] TJJ Van den Boom and B De Schutter. Model predictive control for perturbed max-plus-linear systems: A stochastic approach. *International Journal of Control*, 77(3):302–309, 2004.

[22] R.A. Cuninghame-Green. *Minimax Algebra*. Number 166 in Lecture Notes in Economics and Mathematical Systems. Springer, 1979.

[23] Uri Lerner, Ronald Parr, Daphne Koller, Gautam Biswas, et al. Bayesian fault detection and diagnosis in dynamic systems. In *AAAI/IAAI*, pages 531–537, 2000.

[24] Hans-Michael Hanisch. Analysis of place/transition nets with timed arcs and its application to batch process control. *Application and Theory of Petri Nets 1993*, pages 282–299, 1993.

[25] Gernot Schullerus and Volker Krebs. Diagnosis of batch processes based on parameter estimation of discrete event models. In *2001 European Control Conference (ECC)*, pages 1612–1617. IEEE, 2001.

[26] Laurent Hardouin, Ying Shang, Carlos Andrey Maia, and Bertrand Cottenceau. Observer-based controllers for max-plus linear systems. *IEEE Transactions on Automatic Control*, 2016.

[27] Robert Manger. A catalogue of useful composite semirings for solving path problems in graphs. In *Proceedings of the 11th International Conference on Operational Research (KOI 2006)*, 2008.

[28] G. Provan. Computing diagnoses in switching max-plus systems. In *submitted*, 2017.

[29] Bart De Schutter, WPMH Heemels, and Alberto Bemporad. On the equivalence of linear complementarity problems. *Operations Research Letters*, 30(4):211–222, 2002.