ABSTRACT

A methodology for the design of observers is proposed for a special class of hybrid dynamical systems, which are motivated by traffic and manufacturing applications. In particular, the hybrid systems under study are based on switched system models with constant drift and constant output, rendering all subsystems unobservable by themselves. However, an observer can be derived due to the fixed switching pattern, even though the switching times may be unknown. A main step of the observer design methodology is the usage of a discrete-time linear observer based on the discretized hybrid dynamics at the event times that are visible. Based on this step a continuous-time observer is built that requires additional modes compared to the original hybrid system. This continuous-time observer is shown to asymptotically reconstruct the state of the original system under suitable assumptions. A manufacturing system is used to illustrate the proposed observer design methodology.

Keywords
Observer design, Piecewise affine hybrid systems, Manufacturing systems, Traffic applications

1. INTRODUCTION

Nowadays, logic decision making and control actions are combined with continuous (physical) processes in many technological (cyber-physical) systems. Such systems are labeled hybrid as they have interacting continuous and discrete dynamics. Not only in such man-made systems, but hybrid models are also important to describe behavior of many mechanical, biological, electrical and economical systems. Therefore, in the past decades, the structural properties of hybrid systems have been investigated by many researchers. This led to various techniques for controller synthesis, see e.g. [13] for a recent overview. However, many of these controller synthesis methods are based on the assumption that the full state variable of the hybrid system is available, which is hardly ever the case in practice. This renders the design of observers for hybrid systems, providing good estimates of both continuous and discrete states, of crucial importance. Despite this high practical relevance, surprisingly, results on hybrid observer design are still rather limited, cf. [2, 4, 6, 11, 16, 18, 21]. Theoretical results related to the fundamental question of the existence of an observer are related to notions such as final state observability, reconstructability and final state determinability on which also some work in the area of hybrid systems appeared, see, e.g. [3, 5, 7, 8, 10, 20]

In this paper we are interested in designing observers for a special class of piecewise affine hybrid systems (PWAHS), see [10], motivated by switching servers in manufacturing systems serving multiple products consecutively, or traffic applications such as an intersection emptying lanes. The considered switched system is autonomous with the mode dynamics consisting of constant drift and the output within a mode is constant. Hence, the dynamics and output are piecewise constant functions. In particular this implies that all subsystems are unobservable, eliminating many currently available solutions for synthesizing hybrid observers proposed in the literature. Furthermore, the switching signal depends on the underlying dynamics and is therefore unknown.

For this class of PWAHS we propose a methodology for designing continuous-time observers. First, the system is sampled (with varying sampling periods) at so-called visible event times, i.e. times at which the output changes during a mode switch, resulting in a linear time-varying periodic system. Based on the resulting sampled system a periodic discrete-time observer is derived with the guarantee that the observer state converges asymptotically to the original system state’s. Next, this observer is used as a stepping stone for designing an observer in continuous time. This requires the inclusion of additional modes in the observer structure and additional reset laws at visible event times to ensure the asymptotic recovery of the original system’s state. A formal proof of the asymptotic recovery of the systems state is provided. Via an example of a manufacturing system we demonstrate the effectiveness of the proposed observer.

The remainder of this paper is organized as follows. Section 2 introduces the considered system and presents a two-
buffer switching server as an introductory example. Section 3 presents sampling the system at event times. In Section 4 a method for observer design is presented. First, a time-sampled observer is presented. Next, a continuous-time observer is presented. In Section 5, a switching with three buffers is presented for which an observer is derived. Conclusions are provided in Section 6.

Nomenclature
In this paper we use \{e_1, e_2, \ldots, e_M\} as the standard orthogonal normal basis in \( \mathbb{R}^M \) in which \( e_i \) is the vector which contains a 1 at the \( i \)-th entry, and zeros elsewhere. By \( \mathbb{R}_+ \) we denote the set of non-negative reals, i.e. \( \mathbb{R}_+ := [0, \infty) \). Furthermore, the product of matrices is a left multiplication, i.e. \( \prod_{i=1}^{3} A_i = A_3 A_2 A_1 \).

2. CLASS OF PIECEWISE AFFINE HYBRID SYSTEMS
In this section, we present the dynamics of the class of piecewise affine hybrid systems (PWAHS) studied in this paper. Before doing so, we first present an illustrative example of a manufacturing system to motivate the structure of the class. In fact, this manufacturing system is used as a running example throughout the paper.

2.1 Illustrative example
Consider a single server that serves two different job types denoted by \( n = 1, 2 \), see Figure 1. Each job type \( n \) has a separate buffer in which \( x_n \) jobs are stored. Jobs arrive at buffer 1 with a constant arrival rate denoted by \( \lambda_1 > 0 \). The server can only serve one job type at a time and operates based on a clearing policy, i.e., it completely empties the buffer of one job type before it switches to the next job type. The processing speed of job type \( n \) is denoted by \( \mu_n > 0 \). Switching to job type \( n \) requires a setup time with duration \( \gamma_n \geq 0 \) and at least one setup time is non-zero, i.e. \( \gamma_1 + \gamma_2 > 0 \). The only (measurement) information we get from the server is when the server is processing job type 2.

![Figure 1: Two-product switching server.](image)

To model this server system, we use a continuous state that consists next to the buffer contents \( x_n \), also of the remaining setup time at the server, denoted by \( x_0 \). Therefore, \( x = [x_0 \ x_1 \ x_2]^T \in \mathbb{R}^{n+1}_+ \) with \( N = 2 \) being the number of job types. For each job type \( n \) the system has two modes, one for setting up to serve the job type and the other for serving the job type. The modes (discrete states) are denoted by \( q \in \mathcal{Q} := \{1, 2, \ldots, Q\} \) with \( Q = 4 \) in this case. The modes 1, 2, 3, and 4 represent setting up the server to serve job type 1, serving job type 1, setting up the server to serve job type 2, and serving job type 2, respectively. Note that the order in which the modes are traversed is fixed. The system evolves from mode 1 after which the cycle is repeated. In each mode \( q \in \mathcal{Q} \) the continuous state \( x \) has a constant drift vector, denoted by \( f_q \), i.e., \( \dot{x} = f_q \). For the example system, these drift vectors are given by

\[
 f_1 = f_3 = \begin{bmatrix} -1 \\ \lambda_1 \\ 0 \end{bmatrix}, \quad f_2 = \begin{bmatrix} 0 \\ \alpha_1 - \mu_1 \\ 0 \end{bmatrix}, \quad f_4 = \begin{bmatrix} 0 \\ \lambda_1 \\ -\mu_2 \end{bmatrix}.
\]

Furthermore, a transition occurs in modes 1 and 3 to the next modes 2 and 4, respectively, when \( x_1 = 0 \) indicating that the setup time has elapsed. A transition from mode 2 to mode 3 occurs when \( x_1 = 0 \) (buffer of job type 1 is empty) and from mode 4 to mode 1 when \( x_2 = 0 \) (buffer of job type 2 is empty).

Due to the cyclic behavior in the way the nodes are traversed, it holds at the event time \( t_k, k \in \mathbb{N} \), that the system switches from mode \( q = ((k-1) \mod Q) + 1 \) to the next mode \( \sigma(q) \) given by

\[
 \sigma(q) := 1 + (q \mod Q) \quad (1)
\]

In addition, at the event time \( t_k \) the setup time \( x_0 \) increases with constant \( \alpha_q, q \in \mathcal{Q} \) given by

\[
 \alpha_1 = \alpha_3 = 0, \quad \alpha_2 = \gamma_2, \quad \alpha_4 = \gamma_1
\]

i.e., we have a reset of the continuous state variable given by

\[
 x(t_0^+) = \begin{bmatrix} x_0(t_k^+) \\ x_1(t_k^+) \\ x_2(t_k^+) \end{bmatrix}^T =
\begin{bmatrix} \alpha_q \\ x_1(t_k^+) \\ x_2(t_k^+) - \gamma_2 \end{bmatrix}^T = x(t_k^-) + \alpha_q e_1 \quad (2)
\]

(as \( x_0(t_k^+) = 0 \), see also Lemma 2.6 below). Note that \( e_1 \) is the unit vector with a 1 at the first entry and zeros elsewhere. This reset law shows that discontinuities only appear in \( x_n \), as the remaining \( \gamma_2 \) continuously in time. Hence, the overall dynamics can be compactly written as

\[
 \begin{aligned}
 \dot{x} &= f_q \\
 q &= 0 \quad \text{if} \ e_k^T x \geq 0, \\
 x^+ &= x + \alpha_q e_1 \\
 q^+ &= \sigma(q) \quad \text{if} \ e_k^T x = 0
\end{aligned}
\]

with \( k_1 = k_3 = 1, k_2 = 2 \) and \( k_4 = 3 \) selecting the flow and jump sets (recall that \( e_k \) is the \( k \)-th unit vector in \{\( e_1, e_2, \ldots, e_{N+1}\)\}). Note that we used here the modeling framework of jump-flow systems as advocated in [12]. In fact, solutions/executions of this system can be interpreted in the sense of [12]. Initial conditions for this system are given by \( x(0) \in \mathbb{R}^{n+1}_+ \) with \( x_0(0) = \alpha_1 \), and \( q(0) = 1 \). Besides we use the convention that \( t_0 = 0 \).

To include the measurement information regarding the knowledge of when the server is processing job type 2, we append the model (3) by the output

\[
 y = h q \quad (4)
\]

with

\[
 h_1 = h_2 = h_3 = 0, \quad h_4 = 4.
\]

Hence, as long as the system is in mode 4, this is directly seen in the output \( y \). When the system is in one of the other modes \( q \in \mathcal{Q} \setminus \{4\} \) the output \( y \) is equal to 0 and no information is available from the server.
2.2 General dynamics

In this section we provide the general description of the class of PWASHS under study, which includes the single server system discussed in the previous section as a particular case.

Essentially, the general dynamics of the class of systems is given by (3) and (4) with continuous state

\[ x = [x_0, x_1, \ldots, x_N]^T \in \mathbb{R}^{N+1} \]

with \( N \in \mathbb{N}_{\geq 1} \), discrete state \( q \in \mathcal{Q} = \{1, 2, \ldots, Q\} \) with \( Q \in \mathbb{N}_{\geq 1} \) and output \( y \in \mathcal{Q}_0 := \mathcal{Q} \cup \{0\} \). The data of the system are given by the drift vectors \( f_q \in \mathbb{R}^{N+1} \), the outputs \( h_q \in \mathcal{Q}_0 \), and \( k_q \in \{1, 2, \ldots, N+1\} \) for each mode \( q \in \mathcal{Q} \) together with the reset parameters \( \alpha_q, q \in \mathcal{Q} \). In addition, we have for outputs \( h_q, q \in \mathcal{Q} \) that \( h_q \in \{0, q\} \) for all \( q \in \mathcal{Q} \). Note that, as in the single server system example, the general dynamics exhibit a cycle, i.e. a sequence of \( \mathcal{Q} \) consecutive modes being repeated over time. Some other special characteristics in the data being inherited from server-like systems in manufacturing and traffic applications are summarized below.

Assumption 2.1. For all \( q \in \mathcal{Q} \) it holds that \( e_k^T f_q < 0 \) and \( e_i^T f_q \geq 0 \) when \( i \in \{1, \ldots, N+1\} \setminus \{k_q\} \).

This assumption guarantees that only one continuous state component decreases (being the one that also triggers the mode transition).

Assumption 2.2. For all \( q \in \mathcal{Q} \)

\[
\begin{align*}
   e_k^T f_q &= -1, & \text{if } k_q = 1, \\
   e_i^T f_q &= 0, & \text{if } k_q \neq 1.
\end{align*}
\]

This assumption expresses that \( x_0 \) is indeed a timer-related variable with only \( 0 \) and \( -1 \) as slopes. In case \( x_0 \) acts as a timer that triggers the next event (i.e. \( k_q = 1 \)) then \( e_k^T f_q = -1 \), otherwise it is 0.

Assumption 2.3.

\[
\sum_{q=1}^{Q} \alpha_q > 0,
\]

If transformed to the server system example, this assumption states that during a cycle of modes at least one setup of non-zero duration is required.

Assumption 2.4. For all \( q \in \mathcal{Q} \) it holds that

\[
\alpha_q > 0 \iff k_{\sigma(q)} = 1.
\]

This assumption states that if a mode transition in mode \( q \) governs a jump in the state \( x_0 \), this state \( x_0 \) decreases in mode \( \sigma(q) \) and triggers the next mode transition (and vice versa).

Assumption 2.5. There is at least one \( q \in \mathcal{Q} \) such that \( h_q = q \).

This assumption states that we get at least some information from the system.

Throughout the paper we assume that all the mentioned assumptions are true (without further reference).

2.3 Basic results

In this section we derive some basic results for the class of PWASHS under study. To do so, let us denote, as before, by \( t_k \) the \( k \)-th time occurrence of a transition. Then we can prove the following lemma.

Lemma 2.6. \( x_0(t_k^+) = 0 \) for all \( k \in \mathbb{N}_{\geq 1} \).

Proof. Due to Assumption 2.2 it is clear that state \( x_0 \) only increases by means of a jump, see (3). In case of a jump in \( x_0 \) at time \( t_k \) when going from mode \( q \) to mode \( \sigma(q) \), state \( x_0 \) decreases in mode \( \sigma(q) \) until \( x_0 = 0 \), i.e. \( x_0(t_{k+1}^-) = 0 \) see (8) and (3a). In case of no jump (\( \alpha_q = 0 \) at \( t_k \), we have due to Assumption 2.2 that \( x_0 = 0 \) in mode \( \sigma(q) \) and due to (3b) \( x_0(t_k^+) = 0 \). Hence, then also \( x_0(t_{k+1}^-) = 0 \).

Lemma 2.7. Consider \( q \in \mathcal{Q} \) such that \( \alpha_q > 0 \). The dwell time in mode \( \sigma(q) \) is equal to \( \alpha_q \).

Proof. From (5) and (8) we know that if \( \alpha_q > 0 \) we have \( e_k^T f_{\sigma(q)} = -1 \). Due to (3b) it holds that \( x_0(t_k^+) = \alpha_q \) for event time \( t_k \) where the system switches from mode \( q \) to mode \( \sigma(q) \). In mode \( \sigma(q) \) state \( x_0 \) is the only state that decreases until \( x_0(t_{k+1}^-) = 0 \), see (8) and (3a). Hence, the dwell time in mode \( \sigma(q) \) is therefore given by

\[
\frac{\alpha_q}{-e_k^T f_{\sigma(q)}} = \alpha_q.
\]

For the class of PWASHS the following statements can be made regarding Zeno behavior and fixed points:

Lemma 2.8. Zeno behavior is not present in system (3).

Proof. Equations (7) and (8) imply that there exists at least one mode in the cycle with \( \alpha_q > 0 \), \( q \in \mathcal{Q} \). Lemma 2.7 shows that the dwell time in mode \( \sigma(q) \) is \( \alpha_q \). Therefore, the dwell time of a cycle is bounded away from zero and no Zeno behavior occurs.

Lemma 2.9. System (3) does not contain any fixed points.

Proof. The condition \( e_k^T f_q < 0 \) guarantees that \( e_k^T x \) decreases at constant rate, until it reaches the jump criterion \( e_k^T f_q = 0 \). Therefore, every mode is left in finite time. Since Zeno behavior is excluded, the system has no fixed points.
LEMMA 2.10. The system (3) is a positive system in the sense that if \( x(0) \in \mathbb{R}^{n+1}_+ \) with \( x_0(0) = a_0 \), and \( q(0) = 1 \), then \( x(t) \in \mathbb{R}^{n+1}_+ \) for all \( t \in \mathbb{R}_+ \).

Proof. The proof follows similar reasoning as the proof of Lemma 2.6 using Assumptions 2.1 and 2.2. □

2.4 Visible and invisible modes and event times

In the remainder of this paper we use the following notations in which we make a distinction between visible and invisible modes, and visible and invisible events.

A mode \( q \in \mathcal{Q} \) is called visible if \( h(q) = q \) and otherwise it is called invisible. The set of visible modes is denoted by \( \mathcal{Q}_v \), i.e.,

\[
\mathcal{Q}_v = \{ q \in \mathcal{Q} \mid h(q) = q \}.
\]

Transitions to and from visible modes are called visible events and transitions from invisible modes to invisible modes are called invisible events. To describe the visible events we define the set

\[
\mathcal{V} = \mathcal{Q}_v \cup \{ q \in \mathcal{Q} \mid \sigma(q) \in \mathcal{Q}_v \}
\]

which is enumerated as \( \{ v_1, v_2, \ldots, v_M \} \subseteq \mathcal{Q} \) with \( 1 \leq v_1 < v_2 < \ldots < v_m < v_{m+1} < \ldots < v_M \leq Q \).

Using the above notation, if at time \( t_k \) the \( k \)-th event occurs jumping from mode \( q = ((k-1) \mod Q) + 1 \) to mode \( \sigma(q) \), this event is visible if and only if \( q = ((k-1) \mod Q) + 1 \in \mathcal{V} \). Hence, loosely speaking, we know when the system enters or leaves visible modes and the corresponding events are visible.

In the remainder we will use \( j \) as the visible event counter and denote visible event times by \( t_j = t_{k(j)} \), \( j \in \mathbb{N} \), where \( k(j) \) translates the visible event \( j \) into the corresponding ordinary event \( k \), i.e.,

\[
k(j) = v_{(j-1) \mod M} + 1 + (j-1) \cdot \frac{M}{Q},
\]

where \( \lfloor r \rfloor \) denotes the largest integer smaller than \( r \in \mathbb{R} \). Hence, at event time \( t_j \) the transition takes place from the visible mode \( q = v_{(j-1) \mod M} + 1 \) to \( \sigma(q) \). Note that the mode \( \sigma(q) \) is not necessarily visible, although it might be.

3. Sampling the hybrid system

One of the main ideas in the observer design methodology is to derive the desired continuous-time observer for system (3) from a discrete-time observer. To that end, we sample the system at the visible event times \( t_j \), \( j = 1, 2, \ldots \). To easily derive the sampled dynamics, we split it into three parts. First the system is sampled at all events \( t_k \), \( k = 1, 2, \ldots \). Second, the state dimension is reduced by removing \( x_0 \). In the third step the dynamics only at the visible events is constructed.

3.1 Sampling at all event times \( t_k \)

Let \( x(t_k) \) denote the state at time \( t_k \) just before the jump from mode \( q = q(k) = ((k-1) \mod Q) + 1 \) to mode \( \sigma(q(k)) \). Sampling at all event times \( t_k \), \( k \in \mathbb{N} \), results in the system

\[
x(t_{k+1}) = \tilde{A}_k x(t_k) + \tilde{a}_k,
\]

\[
t_{k+1} = t_k + \tilde{C}_k x(t_k) + \tilde{c}_k.
\]

Remark 3.1. Notice that from Assumptions 2.1 and 2.2 we obtain that the \( \tilde{A}_k, \tilde{a}_k, \tilde{C}_k, \text{ and } \tilde{c}_k \) only contain non-negative elements, resulting in a positive system, which is to be expected based on Lemma 2.10.

Example

For the illustrative system in Section 2.1, we obtain a 4-periodic system for which these matrices are as follows:

\[
\tilde{A}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \tilde{a}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},
\]

\[
\tilde{C}_1 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad \tilde{c}_1 = 0,
\]

\[
\tilde{A}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \tilde{a}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},
\]

\[
\tilde{C}_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \tilde{c}_2 = 0,
\]

\[
\tilde{A}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \tilde{a}_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},
\]

\[
\tilde{C}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \quad \tilde{c}_3 = 0,
\]

\[
\tilde{A}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad \tilde{a}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},
\]

\[
\tilde{C}_4 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad \tilde{c}_4 = 0.
\]

3.2 State reduction

From Lemma 2.6 we have \( x_0(t_k) = 0 \) for all \( k \in \mathbb{N} \). Therefore, we can consider the reduced state \( \tilde{x} = [x_1 \ldots x_N]^T = [0 \ I] x \in \mathbb{R}^N_+ \), for which the dynamics at event times becomes

\[
\tilde{x}(t_{k+1}) = \tilde{A}_k \tilde{x}(t_k) + \tilde{a}_k,
\]

\[
t_{k+1} = t_k + \tilde{C}_k \tilde{x}(t_k) + \tilde{c}_k.
\]

Example

For the illustrative system in Section 2.1, these reduced ma-
trices are as follows:
\[
\begin{align*}
\bar{A}_1 &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \\
\bar{a}_1 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
\bar{C}_1 &= \begin{bmatrix} \frac{1}{\mu_1} - \lambda_1 \\ 0 \end{bmatrix}, \\
\bar{c}_1 &= 0,
\end{align*}
\]
\[
\begin{align*}
\bar{A}_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
\bar{a}_2 &= \begin{bmatrix} \gamma_2 \mu_1 \\ 0 \end{bmatrix}, \\
\bar{C}_2 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\
\bar{c}_2 &= \gamma_2.
\end{align*}
\]
\[
\begin{align*}
\bar{A}_3 &= \begin{bmatrix} \frac{1}{\mu_2} - \lambda_1 \\ 0 & 0 \end{bmatrix}, \\
\bar{a}_3 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
\bar{C}_3 &= \begin{bmatrix} 0 & \frac{1}{\mu_2} \\ 0 & 0 \end{bmatrix}, \\
\bar{c}_3 &= 0,
\end{align*}
\]
\[
\begin{align*}
\bar{A}_4 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
\bar{a}_4 &= \begin{bmatrix} \gamma_1 \lambda_1 \\ 0 \end{bmatrix}, \\
\bar{C}_4 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \\
\bar{c}_4 &= \gamma_1.
\end{align*}
\]

3.3 Sampling at visible event times \(t_j\)
Let Maurice: I changed this into \(\bar{x}\) - reduced state - to avoid confusing with \(x\) full state \(\bar{x}_j = \bar{x}(t_{k(j)+1})\) denote the reduced state vector at time \(t_{k(j)+1} = t_j\) just before the \(j\)th visible event, where \(k(j)\) has been defined in (10). Furthermore, let \(y_j = t_j + t_j^\prime\), \(j \in \mathbb{N}\), denote the duration until the next successive visible event. From (13) it is clear that sampling at the visible events results in the system
\[
\begin{align*}
\bar{x}_{j+1} &= \bar{A}_j \bar{x}_j + \bar{a}_j, \quad (14a) \\
\bar{y}_j &= \bar{C}_j \bar{x}_j + \bar{c}_j. \quad (14b)
\end{align*}
\]
Since we have \(M\) different visible events, the system (14) is a periodic linear system with period \(M\). The system matrices in (14) for all \(j = 1, \ldots, M\) are determined by
\[
\begin{align*}
A_j &= \prod_{k=k(j)}^{k(j)+1-1} \bar{A}_k, \quad (15a) \\
\bar{a}_j &= \sum_{k=k(j)}^{k(j)+1-1} \left( \prod_{l=k+1}^{k(j)+1-1} \bar{A}_l \right) \bar{a}_k, \quad (15b) \\
C_j &= \sum_{k=k(j)}^{k(j)+1-1} \bar{C}_k \prod_{l=k(j)}^{k(j)+1-1} \bar{A}_k, \quad (15c) \\
c_j &= \sum_{k=k(j)}^{k(j)+1-1} \bar{c}_k + \sum_{m=k(j)+1}^{k(j)+1-1} \bar{C}_m \sum_{l=k(j)}^{m-1} \left( \prod_{k=l+1}^{m-1} \bar{A}_k \right) \bar{a}_l \quad (15d)
\end{align*}
\]
with \(A_{j+M} = A_j, \ j \in \mathbb{N}\), and similar expressions for \(\bar{a}_j, C_j\) and \(c_j, j \in \mathbb{N}\).

Example
For the illustrative system in Section 2.1, we obtain a 2-periodic system (14) for which the matrices above are given by
\[
\begin{align*}
A_1 &= \begin{bmatrix} 1 & \frac{1}{\mu_2} \\ 0 & 0 \end{bmatrix}, \\
\bar{a}_1 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_1 &= \begin{bmatrix} 0 & \frac{1}{\mu_2} \\ 0 & 0 \end{bmatrix}, \\
\bar{c}_1 &= 0,
\end{align*}
\]
\[
\begin{align*}
A_2 &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \\
\bar{a}_2 &= \begin{bmatrix} \gamma_2 \lambda_1 \\ 0 \end{bmatrix}, \\
C_2 &= \begin{bmatrix} \frac{1}{\mu_1} - \lambda_1 \\ 0 \end{bmatrix}, \\
\bar{c}_2 &= \gamma_1 \lambda_1 + \gamma_1 + \gamma_2.
\end{align*}
\]

4. OBSERVER DESIGN
Notice that we only receive information about the system’s state when visible events occur. Therefore, from the occurrence of visible events we need to reconstruct the system state.

We therefore build an observer in two steps. First, starting from the dynamics (14) we build a linear time-varying (periodic) observer to reconstruct the system’s state at visible event times. Next, we use the dynamics (3) to make an open-loop prediction of the system’s state, which is corrected (if necessary) at the next visible event time.

4.1 Discrete observer at visible event times
Our first goal is to build an observer which reconstructs the state at visible event times \(t_j\), as described by the dynamics (14). To that end, we can use a Luenberger observer
\[
\begin{align*}
\dot{x}_{j+1} &= A_j \hat{x}_j + a_j + L_j (y_j - \hat{y}_j), \quad (16a) \\
\hat{y}_j &= C_j \hat{x}_j + c_j, \quad (16b)
\end{align*}
\]
where \(L_j\) are the observer gains.

Assumption 4.1. M-periodic gains \(L_j, j \in \mathbb{N}\) are available such that the observer (16) guarantees
\[
\lim_{j \to \infty} \|\bar{x}_j - \hat{x}_j\| = 0,
\]
where \(\bar{x}_j, j \in \mathbb{N}\) describe solutions to (14).

Finding observer gains for this periodic system satisfying (4.1) is a known observer design problem, cf. [15, 17, 19], and, in fact, can also be solved using Kalman filters using periodic Riccati equations. For the example presented in the next section, we exploit the periodicity of this system and use a simple sequential algorithm presented in [14] for determining the time-varying observer gains to guarantee deadbeat convergence to zero at visible event times.

4.2 Continuous-time observer
Starting from the reduced state estimates at visible event times, as generated by the observer (16), we will now provide a state estimate \(\hat{x} \in \mathbb{R}^{N+1}\) of the full state \(x\) of (3) (i.e. including \(x_0\)). In fact, we will guarantee that the estimated states \(\hat{x}(t_j^\prime +)\) just after visible events satisfy
\[
\begin{align*}
[0 \ 1] \hat{x}(t_j^\prime +) = \bar{x}_j, \quad (17)
\end{align*}
\]
\(j \in \mathbb{N}\), indicating that just after the visible events the estimates states (without timer \(x_0\)) and the estimates \(x_2\) of the discrete-time observer (16) coincide. In addition, we use the dynamics (3) to make an open-loop prediction of the system’s state after visible events. This open-loop prediction is updated using the observer (16) as soon as a new visible event happens.

However, notice that due to observation errors, the predicted occurrence of the next visible event can be either sooner or later than the actual occurrence of the next visible event. In the latter case we can simply update the observer state according to (16), but in the former case we cannot (yet) use (16) as \(y_j\) (the duration to the next visible event) is not known yet. Hence in the latter case, we have to determine the continuous-time observer dynamics for the period from predicted to actual occurrence of the next visible event. This introduces additional modes (called waiting modes) for the continuous-time observer. We therefore extend the set of modes \(Q\), by defining the set of observer modes:
\[
Q = Q \cup \left\{ v_m + \frac{1}{2} | m = 1, 2, \ldots, M \right\},
\]
where \( v_m + \frac{1}{2}, m = 1, 2, \ldots, M \) are labels to denote the waiting modes. Furthermore, we define the mode transition map

\[
\hat{\sigma}(\hat{q}) := \begin{cases} 
\hat{q} + \frac{1}{2} & \text{if } \hat{q} \in V \\
\sigma(\hat{q}) & \text{if } \hat{q} \in Q \setminus V \\
\sigma(\hat{q} - \frac{1}{2}) & \text{if } \hat{q} \in Q \setminus Q
\end{cases}
\]

which will only be used as long as the next visible event has not yet happened. For these additional modes of the observer, we have to determine a drift vector for the state estimate. Notice from the observer (16) that at the visible event times we update the state estimate according to (16a). That is, we add \( L_j \) times the amount of time that the actual visible event is later than predicted. From this we can derive a continuous evolution of the state estimate keeping \( \hat{x}_0 \) constant (at zero) and using a drift vector of \( L_j \) for the remaining state, i.e., \( \hat{x} = [0 \ L_j]^T \).

Using the above reasoning, we can deduce a continuous-time observer of the form of a jump-flow system [12]. The discrete-time observer (16) will be embedded in the observer by including the state variable \( \hat{x} \in \mathbb{R}^n \) in the continuous observer, which will satisfy \( \hat{x}(t^+_j) = \hat{x}_j \) with \( \hat{x}_j, j \in \mathbb{N} \), solutions to (16). In between visible events \( \hat{x} \) will be constant. We will also include the state variable \( \gamma \in \mathbb{N} \) that keeps track of the visible event counter. This leads to the following flow expressions for the continuous-time observer.

\[
\begin{align*}
\dot{\hat{x}} &= f(\hat{q}) \\
\dot{\hat{x}} &= 0 \\
\dot{\hat{q}} &= 0 \\
\dot{\gamma} &= 0
\end{align*}
\]

\( \text{if } e_k^T \hat{x} \geq 0 \wedge \hat{q} \in Q \vee t \leq t^+_j, \) (18a)

\[
\begin{align*}
\dot{\hat{x}} &= 0 \\
\dot{\hat{q}} &= 0 \\
\dot{\gamma} &= 0
\end{align*}
\]

\( \text{if } \hat{q} \in \hat{Q} \setminus Q \vee t \leq t^+_j, \) (18b)

where in (18b) \( j(\hat{q}) \) is such that \( \nu(\hat{q}) = \hat{q} - \frac{1}{2}, \) and \( t \in \mathbb{R}_+ \) denotes the actual time. Note that (18a) describes the normal flow predictions based on model (3), while (18b) correspond to the waiting modes.

The jump expressions are given as follows:

\[
\begin{align*}
\hat{x}^+ &= \hat{x} + \alpha q e_1 \\
\hat{x}^+ &= \hat{x} \\
\hat{q}^+ &= \hat{q} \\
\hat{\gamma}^+ &= \hat{\gamma}
\end{align*}
\]

\( \text{if } e_k^T \hat{x} \geq 0 \wedge \hat{q} \in Q \vee t < t^+_j \) (18c)

\[
\begin{align*}
\hat{x}^+ &= \left( A_{\gamma-1} - L_{\gamma-1} C_{\gamma-1} \right) \hat{x} + L_{\gamma-1} (t^+_\gamma - t^+_{\gamma-1}) \\
\hat{q}^+ &= \sigma(\hat{q}(\gamma)) \\
\hat{\gamma}^+ &= \gamma + 1
\end{align*}
\]

(18d)

where \( \hat{q}(j) = ((k(j) - 1) \mod Q) + 1, \) and \( k(j) \) is given by (10), \( j \in \mathbb{N} \). Based on some knowledge of initial conditions of the original system (3), we initialize the observer as \( \hat{x}(0) \in \mathbb{R}^{n+1}_+ \) with \( \hat{x}_0(0) = \alpha_1, \hat{q}(0) = 1, \gamma(0) = 1 \) and \( \hat{x}(0) = \{ 0 \} \hat{\xi}(0) \). Note that (18c) describes the normal jump conditions based on model (3). Besides (18c) describes also that in case the actual visible event occurs later than the forecasted event (determined by \( e_k^T \hat{x} = 0 \)), the discrete state first goes into waiting mode.

**Proposition 4.2.** Suppose Assumption 4.1 holds. Then the continuous-time observer (18) asymptotically reconstructs the state of the system (3), i.e.

\[
\lim_{t \to \infty} \| x(t) - \hat{x}(t) \| = 0.
\]

**Proof.** Since we designed our continuous-time observer such that \( \| x(t^+_j) - \hat{x}(t^+_j) \| = \| x_j - \hat{x}_j \| \), it suffices to show that there exists a constant \( M \) such that \( \| x(t) - \hat{x}(t) \| \leq M \| x_j - \hat{x}_j \| \) for \( t \in [t^+_j, t^+_{\gamma+1}] \). The latter follows from the observation that the observation error only changes when the observer is in a different mode than the observer.

Notice that (11) describes the evolution of the mode changes of the system, i.e., the switching times \( t_k \). Let \( t_k \) denote the switching times as predicted by the continuous-time observer. Then we have from (11b):

\[
t_{k+1} - t_{k+1} = t_k - t_k + \hat{C}_k [x(t_k^-) - \hat{x}(t_k^-)]\]

(19)

Since at time \( t_k^+ \) we have \( \hat{t}_{k(j)} = t_{k(j)} \), it follows from (19) that the duration of the mode difference is a linear function of the initial observer error, i.e., and so is the sum of the durations of mode differences. Finally, since during mode differences the rate of increase is bounded, and the number of invisible modes between two consecutive visible modes is finite, the observation error \( \| x(t) - \hat{x}(t) \| \) on the interval \( t \in [t^+_j, t^+_{j+1}] \) can be upperbounded by \( \| x(t) - \hat{x}(t) \| \leq M \| x_j - \hat{x}_j \| \).

Notice that from (19) we also have that \( \lim_{t \to \infty} | t_k - \hat{t}_k | = 0. \)

**Remark 4.3.** As we observed in Lemma 2.10 and Remark 3.1, we are essentially dealing with a positive system. However, the observer (15) does not necessarily guarantee positivity of the state estimates \( \hat{x}_j \). Though the observer (18) is well defined and asymptotically recovers the state of the original system, even for negative \( \hat{x}_j \), from a physical point of view it would be better to have non-negative state estimates. As we show in the next section by means of an example, it is possible to derive observers that respect the positivity property by generating non-negative state estimates. In fact, designing directly positive observers is one of the questions for future research.
One direction to pursue in this context is considering the dynamics (14) only once every $J$ time-instances and lift the system (see, for instance, [9]) leading to a linear time-invariant positive system. The positive observation problem for linear discrete time-invariant positive systems has been dealt with in [1], where a necessary and sufficient condition for the existence and the design of a positive linear observer of Lueneberger form has been given by means of the feasibility of a linear program (LP). This could form an interesting starting point to obtain positive discrete-time observers of the form (16a) (which is doable under certain assumptions). The step towards a continuous-time observers could follow then mutatis mutandis the line of reasoning as indicated above.

5. THREE-PRODUCT SWITCHING SERVER

To demonstrate the observer design, we introduce a switching server serving three product types, see Figure 2. The buffers are served in order 1, 2, 3 and so on and the server uses a clearing policy, i.e. it completely empties the buffers before switching to serve the next buffer. Products arrive with arrival rates $\lambda_i$ and are served with process rates $\mu_i$, $i=1,2,3$.

$$\lambda_1 \rightarrow x_1 \rightarrow \mu_1 \rightarrow y$$

$$\lambda_2 \rightarrow x_2 \rightarrow \mu_2$$

$$\lambda_3 \rightarrow x_3 \rightarrow \mu_3$$

Figure 2: Three-product switching server with a single measurable output.

Writing this system in the form (14) gives

$$A_1 = \begin{bmatrix} 0 & 0 & 0 \\ \frac{\lambda_1}{\lambda_1 - \lambda_2} & 1 & 0 \\ \frac{\lambda_1}{\lambda_1 - \lambda_3} & 0 & 1 \end{bmatrix}, \quad C_1 = \begin{bmatrix} \frac{1}{\mu_1 - \lambda_1} & 0 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1 - \frac{\lambda_1}{\lambda_2 - \lambda_1} + \frac{\lambda_2}{(\mu_2 - \lambda_2)(\mu_3 - \lambda_3)} & \frac{\lambda_1}{\lambda_1 - \lambda_2} \\ 0 & \frac{1}{\mu_2 - \lambda_2} & \frac{\lambda_1}{\lambda_1 - \lambda_3} \\ 0 & \frac{\lambda_1}{\lambda_1 - \lambda_3} & \frac{1}{\mu_3 - \lambda_3} \end{bmatrix},$$

$$C_2 = \begin{bmatrix} \frac{\lambda_1}{\lambda_2 - \lambda_1} & \frac{\lambda_1}{\lambda_1 - \lambda_3} & \frac{1}{\mu_3 - \lambda_3} \end{bmatrix}.$$

where $a_1$ and $c_1$ have been omitted since they cancel out in the observer error dynamics. Note that though the pairs $(A_1, C_1)$ and $(A_2, C_2)$ are unobservable, we can build a periodic dead-beat observer by using the observer gains

$$L_1 = \begin{bmatrix} 0 & \lambda_2 & \lambda_3 \end{bmatrix}^T, \quad (20a)$$

$$L_2 = \begin{bmatrix} 0 & \frac{\lambda_2 \lambda_3}{\mu_2} & 0 \end{bmatrix}^T. \quad (20b)$$

Using these gains, the matrices $A_1 - L_1 C_1$ and $L_1 C_1$ ($i = 1, 2$) are positive matrices, yielding a positive observer. For

$$\lambda = [1 \ 2 \ 3]^T, \quad \mu = [8 \ 10 \ 12]^T,$$

$$\gamma = [5 \ 10 \ 15]^T, \quad \hat{x}(50) = [70 \ 20 \ 30]^T,$$

a simulation result is presented in Figure 3. The buffer levels of the plant are presented by dashed lines and the estimated buffer levels are presented by solid lines. The observer starts at $t = 50$ in mode 5, i.e. serving products from buffer 2. The observer finishes serving products from buffer 2 and sets up to serve products from buffer 3 until the first visible event occurs at $t = 70.7$. According to the initial estimated state, this event was predicted at $t = 81.7$ with $\hat{x}(81.7) = [101.7 \ 58.3 \ 15]^T$. Using update (16a) with $L_2$ results in $\hat{x}(70.7) = [90.7 \ 52.8 \ 15]^T$, causing jumps in $\tilde{x}_2$ and $\tilde{x}_3$. Also, the estimated mode changes to serving products from buffer 1.

![Simulation results of the 3 buffer system](image)

Figure 3: Simulation results of the 3 buffer system (dashed lines) and observer estimation (solid lines) starting at $t = 50$.

6. CONCLUSIONS

This paper presented a methodology to design observers for a special class of piecewise affine hybrid systems (PWAHS), being highly relevant in the context of manufacturing and traffic applications. Although all subsystems are unobservable and not all events are visible, a continuous-time observer was constructed that guarantees the estimate converges to the state of the plant under suitable conditions. One of the main ideas in the construction of the continuous-time observer was sampling of the system at the visible events leading to a discrete-time periodic linear system, for which an observer can be designed using standard techniques from control theory. If this step is successfully performed, a continuous-time observer that asymptotically recovers the true state of the original hybrid systems can be synthesized. Indeed, the discrete-time observer can then be used as a blueprint for the continuous-time observer, where besides the plant dynamics additional ‘waiting’ modes are assigned.
to the observer. Occurrence of visible events before the time that the event was predicted to occur results in a discrete state switch and an update of the continuous states. The observer switches to a ‘waiting’ mode if an event occurs later than predicted. These principles are formally shown to the result in a successful observer design.

The results in this paper lay down the fundamental ideas for observer design for the class of PWAHS considered. These ideas lead to various other questions that will be considered in future work. First of all, it would be of interest to take the positivity of the state variable into account leading to observers that always create positive state estimates as well (positive observers). Some first hints were already provided in this direction. In addition, it is of interest to formulate necessary and sufficient conditions in terms of the data of the original PWAHS system when the proposed design is indeed successful and how this relates to fundamental observability and detectability properties. Finally, it is of interest to investigate to what extent the observer design principles put forward in this paper can be applied to more general classes of hybrid systems. As such, this paper provides ideas that might be fruitfully exploited into various future research directions.

7. ACKNOWLEDGMENTS

This work is supported by the Netherlands Organization for Scientific Research (NWO-VIDI grant 639.072.072).

8. REFERENCES


