A Framework for Inference and Identification of Hybrid-System Models: Mixed Event-/Time-driven Systems (METS)

Mohammad Hossein Basiri * **John G. Thistle** * **Sebastian Fischmeister** *

* *Electrical and Computer Engineering Department, University of Waterloo, ON, Canada*
email: {*mh.basiri, jthistle, sfischme*}*@uwaterloo.ca*

Abstract: This paper proposes a simplified framework for the experimental inference and identification of models of hybrid systems. A problematic feature of such system identification is the presence of different "modes" of evolution of continuous state variables: this may necessitate not only the identification of the dynamics of the different modes, but also the identification of changes of mode. Inspired by the idea that the physics underlying the system is often invariant, this paper proposes a simplified framework that models hybrid systems in the form of separate untimed, "event-driven" and dynamical, "time-driven" components that are coupled only through input and output signals. Signals from the event-driven component are assumed to affect the dynamics of the time-driven component only in a relatively limited manner – either as direct inputs, or by modulating output feedback within the time-driven component; the "intrinsic," "open-loop" dynamics of the time-driven component do not change. These assumptions largely decouple the estimation of the event- and time-driven components, avoiding the problem of distinguishing separate modes, and permitting the leveraging of standard system-identification methods. Two well known examples serve to illustrate the approach.

Keywords: Hybrid systems, system identification, automaton inference

1. INTRODUCTION

Practical hybrid systems are often developed in the absence of a comprehensive mathematical system model. It may be feasible to model the physical "environment" of the system, but factors such as complexity, time pressure, collaborative development, and lack of well established methodologies generally mean that other components such as computer hardware and software are not described with formal precision. Yet a formal model may be needed for development, maintenance, documentation, diagnostics, regulation, and other purposes. Development costs continue to grow as systems become more complex (Hailpern and Santhanam, 2002). A formal model can facilitate testing, debugging, integration, and documentation. It can also demonstrate to clients and regulators that reasonable care has been taken in system development. One possible manner of arriving at such a model is to estimate or infer it from experimental data after implementation.

Indeed, hybrid-system models capture a vast range of realistic applications such as supervisory control (Antsaklis et al., 1993), robotics (Vasudevan, 2017) and vehicle automation (Horowitz and Varaiya, 2000). Formal hybrid-system models may vary considerably in the complexity that they admit in the state-machine or dynamical components, ranging from potentially complex automata with simple time-driven dynamics, such as constant time derivatives, to complex dynamical systems with relatively simple mode-switching (Antsaklis et al., 1993; Bemporad and Morari, 1999; Branicky et al., 1998). Useful models for hybrid systems should admit complexity in all of their components. But the problem of inferring or identifying general hybrid-system models from experimental data is a difficult one, encompassing not only state-machine inference and dynamic-system identification but also the inference and estimation of mechanisms of coupling of distinct components (Ferrari-Trecate et al., 2003). The possible presence of different "modes" of evolution of continuous state variables, and uncertainty as to how and when such modes may change, is particularly problematic, and typically requires the orders of appropriate dynamic models to be known (Breschi et al., 2016). Similar problems arise in other approaches such as that of Pillonetto (2016), unless the mechanism of mode-switching is assumed to be known *a priori* (Uyanık et al., 2016).

This paper proposes a simplified framework for the experimental identification of hybrid systems. It models the system as comprised of separate "event-driven" and "time-driven" components, each of which may exhibit complex behavior, but which are coupled relatively simply, through the exchange of input and output signals. For example, the time-driven evolution of continuous variables is precluded from changing on the basis of a mere change in state of an event-driven component; the dynamics of the continuous variables may only be influenced by a corresponding change in input signals received from event-driven components. Similar models of interaction through signal "interfaces" between the respective components have been applied in modeling, simulation and control, but do not appear to have been exploited for the purposes of inference/identification (Sarjoughian and Cellier, 2013; Liberzon and Nesic, 2006; Lunze, 2003).

In the proposed framework, inputs from the event-driven component alter the dynamics of the time-driven component only in a relatively limited manner. This reflects an assumption that the "intrinsic," "open-loop" dynamics of the time-driven component are dictated by physical laws, and the event-driven component can only affect inputs to those dynamics, or output-feedback laws applied to them. Consequently, instead of many distinct "modes," there is only a single time-driven component to identify. Datasets do not have to be partitioned owing to mode changes; and all pertinent data can be used to identify a single time-driven component. It is furthermore assumed that, through suitable instrumentation, the logical signals that mediate the components' interaction are logged in the experimental traces from which the system model is to be inferred. So are the continuous-variable outputs of the time-driven component (which is modeled as a discrete-time system for the purposes of the paper). These assumptions largely decouple the estimation of the state-machine and differential/difference equation components, and avoid the problem of distinguishing and estimating different "modes" of the time-driven system. The framework of this paper readily admits standard, recursive approaches to dynamical system identification.

The paper is organized as follows. In Section 2 we provide a general overview of our proposed approach. Section 3 explains the essential preliminaries used in the rest of the paper. In Section 4 we describe our class of models and give a problem statement. Two well known examples in hybrid systems are studied in Section 5 to show the effectiveness of the method. Discussion and conclusions are summarized in Sections 6 and 7 respectively.

2. OVERVIEW

In this section we give a general overview of our proposed framework. We are interested in considering *mixed event-/time-driven systems* (METS), aimed at hybrid model mining and identification purposes. These systems are comprised of components, some of which are purely "event-driven" and others purely "time-driven," which interact only through the exchange of signals. This covers a variety of interesting examples, in which the time-driven part represents the dynamics of physical systems (or clocks), and those dynamics *per se* do not change with the state of the event-driven, "logical" part. The idea is to separate the time-driven dynamics from event-driven dynamics clearly and cleanly. This approach allows us to deal with each part of the system separately and apply well established methods of (untimed) automaton inference and of (continuous- or discrete-time) system identification to derive a model for the system as a whole. Fig. 1 depicts the framework. It is convenient to label the respective blocks as "event-driven component", "time-driven component", "discretizer" and "signal generator". The inputs

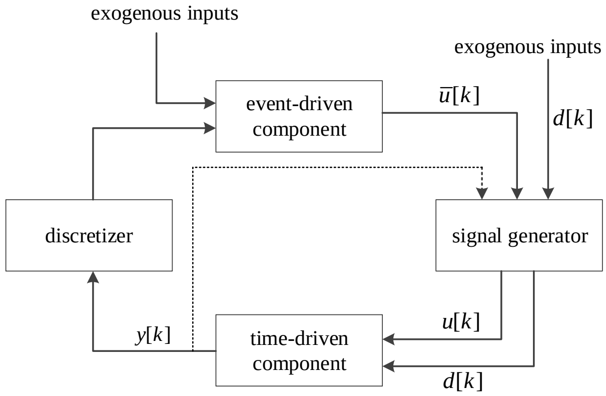

Fig. 1. General view of the proposed framework of a mixed event-/time-driven system

to the time-driven part of the system come from the signal generator component, which is a known system. The value of the signals coming from the event-driven part reflects the state of the event-driven component and is assumed to be piecewise-constant. The time-driven part could also be fed other exogenous inputs $d[k]$. For instance, in automotive applications these could include the throttle position, steering wheel angle, and other commands from the driver (Mathworks®, 2017a). Furthermore, the time-driven part could also consist of internal feedback loops or any time-driven components of controllers, such as a PID feedback law to implement its own control strategies, e.g. a cruise control. The inputs to the event-driven part of the system are in fact the outputs from the discretizer that takes values from a finite set of event symbols (see Fig. 1). These symbols are generated by feeding outputs of the time-driven component to the discretizer. On the basis of the real-valued signals from the time-driven part, some prespecified conditions, which are assumed to be logged in the input/output traces, become true and consequently cause generation of events which are fed to the event-driven component. The event-driven part has a finite set of states and its outputs form inputs to the signal generator. It could also be postulated that the event-driven component has a set of exogenous inputs, e.g. in automotive applications these could model the interactions of a vehicle with other nearby vehicles in order to set the desired path preventing any collisions, alternatively, they could represent other software components of the system, sensors, etc.

3. PRELIMINARIES

3.1 Modeling

Modeling of the Time-driven Part We state the problem of identifying the time-driven component as that of estimating a model of this form:

$$\begin{cases} x[k+1] = Ax[k] + B\begin{bmatrix} u[k] \\ d[k] \end{bmatrix} \\ \begin{bmatrix} u[k] \\ d[k] \end{bmatrix} = f(y[k], \bar{u}[k], d[k], k) \\ y[k] = Cx[k] \end{cases} \quad (1)$$

where $f(\cdot,\cdot,\cdot,\cdot)$ is a known function (that need not be linear), $d[k] \in \mathbb{R}^d$ is the known vector of exogenous inputs and $\bar{u}[k]$ is the output of the event-driven component of the system. The signal $d[k]$ is simply passed directly through the signal generator and fed to the time-driven component (as is shown in Fig. 1), however; it could in principle be modified by the signal generator component. In the state-space representation of the time-driven part of the system we assume $D = 0$. This assumption will mean that our feedback system is well-posed in the sense that the output of the time-driven part at time $k+1$, $y[k+1]$, only depends on former inputs, $u[k']$, $k' \leq k$. Note that by employing a discrete-time representation we have ruled out Zeno phenomena (Abate et al., 2005). The system identification can be reduced to that of a linear model ($x[k+1] = Ax[k] + Bu[k]$). It should be noted that the above model – in principle – captures systems with different LTI "modes" associated with each of the finitely-many values of $u[k]$:

$$x[k+1] = A_i x[k] + B_i u[k] \tag{2}$$

provided that in each of these modes we have

$$\begin{aligned} x[k+1] &= Ax[k] + K_i y[k] + B_i u[k] \\ &= [A + K_i C]x[k] + B_i u[k] \end{aligned} \tag{3}$$

In other words, the A_i matrices must all be of the form $A_i = A + K_i C$, where i denotes the mode associated with the value $u[k]$. (If $y[k]$ in fact represented the full state of the time-driven system, this would effectively allow for completely different A_i matrices in each mode.)

Modeling of the Event-driven Part In Fig. 1, the event-driven part of the system is considered as a Moore machine that is a sextuple $(Q, I, O, \delta, \lambda, q_0)$ where (Moore, 1956):

Q: The finite set of states

$q_0 \in Q$: The initial state

I: The finite set of input symbols called the input alphabet

O: The finite set of output symbols called the output alphabet

$\delta : Q \times I \rightarrow Q$: The input transition function

$\lambda : Q \rightarrow O$: The output transition function

With the event-driven system in a state $q[k] = q_* \in Q$ when event $\sigma \in I$ is generated at time $k+1$ the state of the event-driven system changes to $q^* := \delta(q_*, \sigma)$ and the output changes to $\lambda(q^*)$.

4. PROBLEM STATEMENT

Given the input/output traces, $\bar{u}[k]$ and $y[k]$, exogenous inputs, $d[k]$, signal generator and discretizer components, the problem is to identify the following:

- a model of the time-driven part of the system from the data $f(y[k], \bar{u}[k], d[k], k)$ and $y[k]$; and
- a Moore machine model for the event-driven component of the system from the data $\bar{u}[k]$ and the corresponding events generated by the discretizer block.

In fact, we assume that the outputs of the time-driven components of the system are known (but not necessarily the states – they generally need to be inferred) and real-valued. Such outputs generate "events" through the discretizer block that may lead to changes in the discrete state. As was mentioned in sections 1 and 2, the conditions on the outputs of the time-driven part that give rise to the events are also assumed to be included in the trace of logged experimental data.

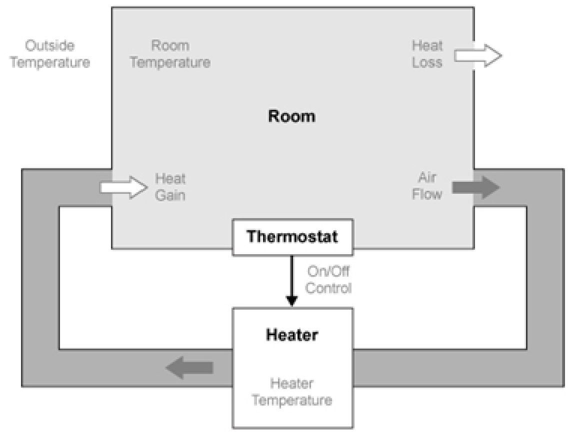

Fig. 2. Temperature regulation system of a house (Mathworks®, 2017b)

5. CASE STUDY

In this section we present two well known examples in hybrid systems, i.e., a temperature regulation system (Fig. 2) and a system of interconnected fluid tanks (Fig. 7) (Arogeti et al., 2010; Wu et al., 2005; Ravi and Thyagarajan, 2013; Joseph et al., 2011; Ábrahám and Schupp, 2012; Johansson et al., 2004; Balbis et al., 2007; Bak et al., 2015). For these two systems, we contrast METS with methods requiring the identification of different system modes.

5.1 Temperature Regulation System

Consider a temperature regulating system containing one thermostat and two heaters so that the thermostat could turn either of the heaters on/1 or off/0 to regulate the room temperature on a desired value. For this example, we explain the modeling procedure with details together with the correspondence of METS to the system under study. Here, the time-driven part of the system consists of the dynamics of the temperature of the room which should be regulated to a setpoint value based on the thermodynamics principles outlined in (Mathworks®, 2017b). This part has outside temperature changes and heat gain from a heating system as its inputs. Here, we postulate the outside temperature changes as a disturbance input of the time-driven part and the heat gain as the control input. On the basis of the dynamics stated in (Mathworks®, 2017b), calculation of the heat gain control-input signal from the thermostat output (on/off), heater and outside temperature can be done as follows which in principle describes the signal generator component of METS:

$$\frac{dQ_{\text{gain}}}{dt} = \underbrace{\text{Thermostat Output (on/off)}}_{\bar{u}[k]} \times K(T_{\text{heater}} - T_{\text{room}}) \tag{4}$$

where $\frac{dQ_{\text{gain}}}{dt}$ is the rate of heat gain and K is a coefficient in $J/h \cdot {}^\circ C$. T_{heater} and T_{room} are the heater temperature and room temperature respectively. In addition, in this example the signal generator block simply generates piecewise-constant signals using values given by the outputs of the event-driven system. The model of the dynamics of room temperature changes based on the rate of heat loss and rate of heat gain has the following structure which in principle describes the time-driven component of the proposed framework,

- Rate of temperature changes in the room:

$$\begin{aligned}\frac{dT_{\text{room}}}{dt} &= \frac{1}{m_{\text{roomair}}c_{\text{air}}}\left(\frac{dQ_{\text{gain}}}{dt} - \frac{dQ_{\text{loss}}}{dt}\right)\\ &= \frac{1}{m_{\text{roomair}}c_{\text{air}}}\Bigg(\underbrace{M_{\text{heaterair}}c_{\text{air}}}_{K}(T_{\text{heater}} - T_{\text{room}})\\ &\qquad - \frac{T_{\text{room}} - T_{\text{outside}}}{R}\Bigg)\end{aligned} \tag{5}$$

where m_{roomair}, c_{air}, $M_{\text{heaterair}}$ and R are mass of air in the room, specific heat capacity, constant rate of air mass passing through the heater and thermal resistance respectively. The terms $\frac{dQ_{\text{loss}}}{dt}$ and T_{outside} represent the rate of heat loss and the outside temperature respectively.

System Identification of Time-driven Part In the literature, methods of identification in closed loop are categorized into three main classes: direct, indirect and joint input-output methods (Forssell and Ljung, 1999). In our framework, the input signal $u[k]$ is determined by the signal generator block. Since we identify the time-driven component using this input and the known output, $y[k]$, our identification method is most closely related to the direct method. For this example, $u[k]$ is the heat gain of the room. It is not directly measured, so it is calculated from the raw data via equation (4). There is no need to distinguish or to identify the different modes of the system; whether the thermostat is on or off, the heat gain is calculated and the entire dataset is used for single identification of the time-driven component. This illustrates the benefits of the METS framework: while the system has distinct modes, they are simply modeled by a single time-driven component under different input or feedback signals; there is no need to distinguish the modes in order to infer that time-driven component. To identify a state-space model for the time-driven part (room) we use the given data for $y[k]$ and calculated values (using (4)) for $u[k]$. The simulations are performed for 24 hours with a sampling time of $T_s = 6sec.$ for a simplified version of the system consisting of one thermostat and one heater with two events b and c (which will be described in the next subsection). Here, we assume that the setpoint value for the room temperature is 22° and use previous thermal model dynamics with the following numerical values of the parameters,

$m_{\text{roomair}} = 1470\ (kg)$ $\quad$ $c_{\text{air}} = 1005.4\ (J/kg \cdot^\circ C)$
$M_{\text{heaterair}} = 3600\ (kg/h)$ $\quad$ $K = 3600 \times 1005.4\ (J/h \cdot^\circ C)$
$R = 4.329 \times 10^{-7}\ (h \cdot^\circ C/J)$

As was briefly mentioned before, one of the advantages of our method is that we use the whole trace to identify a single time-driven component rather than using partial traces to identify different models for continuous dynamics for each mode of the logic part of the system. This potentially leads to more accurate system identification.

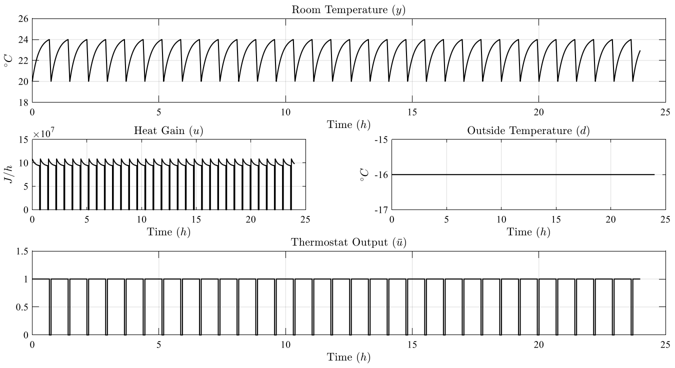

Fig. 3. Input/output signals for outside temperature of -16°

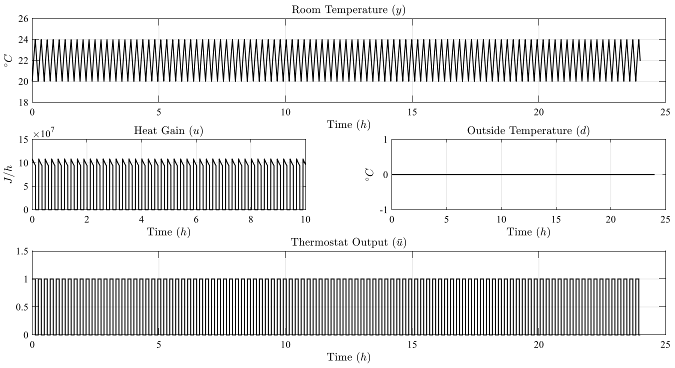

Fig. 4. Input/output signals for outside temperature of 0°

Indeed, in some cases, the partial traces corresponding to a specific mode may contain insufficient data for system identification. Fig. 3 and 4 show different scenarios along with the input and output signals in addition to thermostat output for different outside temperatures. For the scenario in which outside temperature is a constant value -16°, using the System Identification Matlab Toolbox results in the following first-order linear model:

$T_s = 6sec.$:

$$\begin{cases} x[k+1] = 0.9974x[k] + \begin{bmatrix} 8.402 \times 10^{-12} \\ 1.929 \times 10^{-5} \end{bmatrix}^\top \begin{bmatrix} u[k] \\ d[k] \end{bmatrix} \\ y[k] = 133.9x[k] \end{cases} \tag{6}$$

Moore-machine Inference for Event-driven Part For model inference of the event-driven part, when the outputs of the time-driven part trigger "events" that may lead to changes in the discrete state, we assume that such events are logged in the trace and can be automatically recognized as such. Furthermore, the conditions on the outputs of the time-driven part that give rise to the events through the discretizer block are also given in the log on the basis of the desired value for the room temperature. Such a condition can simply form the label of the corresponding transition in the event-driven component and serve as the "event symbol". In other words, we simply label the transitions in the event-driven part of the system with conditions on the vector of outputs of the time-driven part; such a transition occurs at time $k+1$ if the event-driven component is in the "source state" of the transition during the interval $[k, k+1)$ and $P(y[k])$ is not true but $P(y[k+1])$ is true where $P(\cdot)$ is

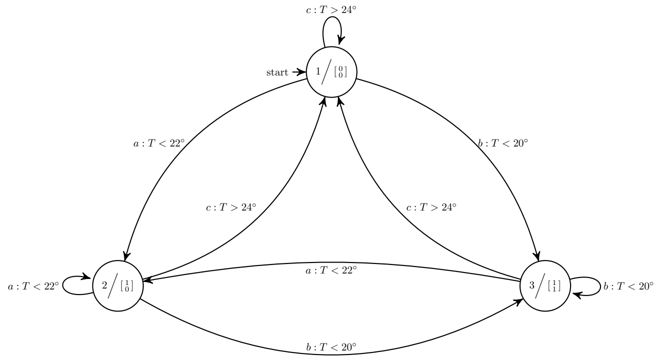

Fig. 5. Moore-machine model of the temperature regulation system

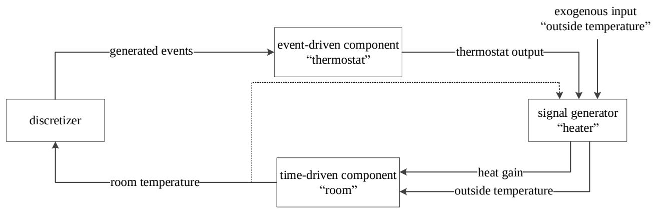

Fig. 6. Framework of the temperature regulation system

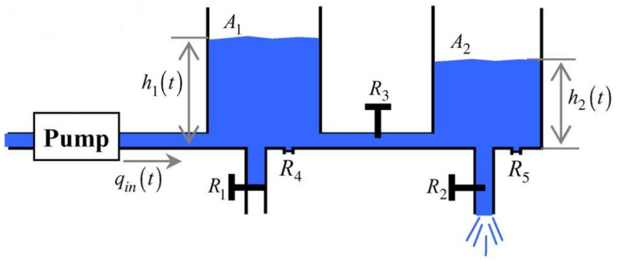

Fig. 7. Tank system (Arogeti et al., 2010)

Table 1. Sample portion of given dataset to model the temperature regulation system

$\bar{\boldsymbol{u}}[\boldsymbol{k}]$	$\cdots$	0	0	1	1	1	1	0	$\cdots$
$\boldsymbol{y}[\boldsymbol{k}]$	$\cdots$	22	21	19.5	20	22	23	24.5	$\cdots$
event	$\cdots$	–	–	$b : T < 20°$	b	b	b	$c : T > 24°$	$\cdots$

the assumed condition. The behavior of the system, which is purely event-driven, could be modeled using a Moore machine. This inference can be performed by applying the software provided by Giantamidis and Tripakis (2016) or the GI Matlab Toolbox (Akram and Xiao, 2011) to the first two rows of Table 1. Fig. 5 shows the inferred Moore-machine for the event-driven part of the original system with one thermostat and two heaters. As was mentioned earlier we have used a portion of this Moore machine consisting of states 1 and 3 for our numerical results. The final framework of the whole system is shown in Fig. 6.

5.2 A System of Interconnected Fluid Tanks

Consider the system containing two tanks shown in Fig. 7. To show how METS can be easily applied to this system,

Table 2. Metrics for identification of the time-driven component of the temperature regulation system, using 24 hours of data

	Pole	DC gain	
	(h^{-1})	$u \to y$ $(h \cdot^\circ C/J)$	$d \to y$ (dimensionless)
Actual system	-1.5630	4.3290×10^{-7}	1
$T_s = 1min.$	-1.7781	3.6547×10^{-7}	0.6222
$T_s = 30sec.$	-1.5702	4.2496×10^{-7}	0.9555
$T_s = 6sec.$	-1.5620	4.3113×10^{-7}	0.9899

it is straightforward to consider the discrete values of the valves connecting the two tanks together and those for controlling outflow rates as the outputs of the event-driven part ($\bar{u}[k]$). Having used fluid dynamics, one can compute inflow rates (through the signal generator block) filling the tanks ($u[k]$). These input values together with level of the tanks ($y[k]$) can be used to identify a single discrete-time dynamics equation describing the time-driven part of the system. Based on the desired level for each tank, level values can generate corresponding events through the discretizer block. These events along with the aforementioned values for $\bar{u}[k]$ can be utilized to infer the Moore machine pertaining to the event-driven part. This will result in a single time-driven and a single event-driven part coupling through the signal generator and the discretizer blocks which models the whole system with no need to identify and model different modes of the system.

6. DISCUSSION

In examples such as the previous one the number of the states of the event-driven system potentially grows exponentially with the number of valves. But the same is true of the number of modes in other hybrid system models. Such models can therefore entail a discrimination among a large number of different modes and a separate system identification problem for each distinct mode.

Turning to the first example, we consider the datasets to identify the time-driven component. Table 2 quantifies the accuracy of identification of the time-driven component for different dataset sizes and sampling times. As was mentioned earlier in subsection 3.1.1 the time-driven part may contain nonlinear structure. In our case one of the inputs to the time-driven part was the heat gain, $u[k]$, calculated via (4). As is obvious from (4) because of the multiplication of thermostat output ($\bar{u}[k]$), which exhibits a switching behavior, by the room temperature ($y[k]$) the resulting equation for heat gain input constitutes a nonlinear equation for the state evolution of the system, $x[k+1]$. In fact, one of the advantages of the proposed approach, as was mentioned before, is that the known function f need not be linear. Moreover, the switching leads to numerous mode changes: about 64 in 24 hours when the outside temperature is -16° (Fig. 3), and approximately 3 times as many when the outside temperature is 0° (Fig. 4). But in the METS framework there is no need to distinguish the different modes and the whole dataset can be used for a single, recursive system identification.

7. CONCLUSION

In this paper we proposed a new framework for the modeling and identification of hybrid systems. The basis of the proposed approach is to separate the continuous dynamics from the discrete logic as much as possible. In particular, our model features a single event-driven component and a single time-driven component, as opposed to a multiplicity of different time-driven "modes." In other words, we are distributing different possible existing "modes" of the system into the event-driven and time-driven components of the proposed framework. The different dynamics of the respective modes are modeled simply by changes of inputs, or changes of feedbacks, applied to the time-driven component. Hence, there is no need to partition the available data into separate modes in order to identify the time-driven component. This provides a larger dataset for identification of a single time-driven component which potentially results in a more accurate model. In this way, identification of a given hybrid system can be cleanly decomposed into two separate identification/inference problems, for the respective time- and event-driven components. The system identification problem for the time-driven part can be formulated largely as a standard problem; and inferring a Moore automaton for the event-driven part can be done by the available tools and methods in the literature.

REFERENCES

Abate, A., Ames, A.D., and Sastry, S.S. (2005). Stochastic approximations of hybrid systems. In *American Control Conference, 2005. Proceedings.*, 1557–1562. IEEE.

Ábrahám, E. and Schupp, S. (2012). Modeling and analysis of hybrid systems.

Akram, H.I. and Xiao, H. (2011). *GI Matlab Toolbox*. Technische Universität München, Germany. Retrived May 19, 2017 from `https://code.google.com/archive/p/gitoolbox/`.

Antsaklis, P.J., Stiver, J.A., and Lemmon, M. (1993). Hybrid system modeling and autonomous control systems. In *Hybrid systems*, 366–392. Springer.

Arogeti, S.A., Wang, D., and Low, C.B. (2010). Mode identification of hybrid systems in the presence of fault. *IEEE Transactions on Industrial Electronics*, 57(4), 1452–1467.

Bak, S., Bogomolov, S., and Johnson, T.T. (2015). Hyst: a source transformation and translation tool for hybrid automaton models. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, 128–133. ACM.

Balbis, L., Ordys, A.W., Grimble, M.J., and Pang, Y. (2007). Tutorial introduction to the modelling and control of hybrid systems. *International Journal of Modelling, Identification and Control*, 2(4), 259–272.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Branicky, M.S., Borkar, V.S., and Mitter, S.K. (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1), 31–45.

Breschi, V., Bemporad, A., and Piga, D. (2016). Identification of hybrid and linear parameter varying models via recursive piecewise affine regression and discrimination. In *Control Conference (ECC), 2016 European*, 2632–2637. IEEE.

Ferrari-Trecate, G., Muselli, M., Liberati, D., and Morari, M. (2003). A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2), 205–217.

Forssell, U. and Ljung, L. (1999). Closed-loop identification revisited. *Automatica*, 35(7), 1215–1241.

Giantamidis, G. and Tripakis, S. (2016). Learning moore machines from input-output traces. In *FM 2016: Formal Methods: 21st International Symposium, Limassol, Cyprus, November 9-11, 2016, Proceedings 21*, 291–309. Springer.

Hailpern, B. and Santhanam, P. (2002). Software debugging, testing, and verification. *IBM Systems Journal*, 41(1), 4–12.

Horowitz, R. and Varaiya, P. (2000). Control design of an automated highway system. *Proceedings of the IEEE*, 88(7), 913–925.

Johansson, K.H., Lygeros, J., and Sastry, S. (2004). Modeling of hybrid systems.

Joseph, C., George, V., and Venkateswaran, P. (2011). Hybrid modeling and discrete controller design of three-tank benchmark system. In *Instrumentation Control and Automation (ICA), 2011 2nd International Conference on*, 170–174. IEEE.

Liberzon, D. and Nesic, D. (2006). Stability analysis of hybrid systems via small-gain theorems. In *HSCC*, 421–435. Springer.

Lunze, J. (2003). What is a hybrid system? In S. Engell, G. Frehse, and E. Schnieder (eds.), *Modelling, analysis and design of hybrid systems*, volume 279, 3–14. Springer.

Mathworks® (2017a). *Engine Timing Model with Closed Loop Control*. Retrived May 19, 2017 from `https://www.mathworks.com/help/simulink/examples/engine-timing-model-with-closed-loop-control.html?s_tid=gn_loc_drop`.

Mathworks® (2017b). *Thermal Model of a House*. Retrived May 19, 2017 from `https://www.mathworks.com/help/simulink/gs/define-system.html`.

Moore, E.F. (1956). Gedanken-experiments on sequential machines. *Automata studies*, 34, 129–153.

Pillonetto, G. (2016). A new kernel-based approach to hybrid system identification. *Automatica*, 70, 21–31.

Ravi, V. and Thyagarajan, T. (2013). Application of hybrid modeling and control technique to a non linear hybrid system. In *Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference on*, 660–666. IEEE.

Sarjoughian, H.S. and Cellier, F.E. (2013). *Discrete Event Modeling and Simulation Technologies: A Tapestry of Systems and AI-Based Theories and Methodologies.* Springer Science & Business Media.

Uyanık, İ., Saranlı, U., Morgÿl, Ö., and Ankaralı, M.M. (2016). Parametric identification of hybrid linear-time-periodic systems. *IFAC-PapersOnLine*, 49(9), 7–12.

Vasudevan, R. (2017). Hybrid system identification via switched system optimal control for bipedal robotic walking. In *Robotics Research*, 635–650. Springer.

Wu, J., Biswas, G., Abdelwahed, S., and Manders, E. (2005). A hybrid control system design and implementation for a three tank testbed. In *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, 645–650. IEEE.