A Hybrid User Model for Virtual Stochastic Sensors

Claudia Krull

Faculty of Computer Science, Otto-von-Guericke-University, Magdeburg, 39106 Germany ; *claudia.krull@ovgu.de*

SNE 33(1), 2023, 35-43, DOI: 10.11128/sne.33.tn.10636
Selected ASIM SST 2022 Postconf. Publication:2023-02-01;
Received Revised Improved:2023-02-27; Accepted:2023-03-10
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Virtual stochastic sensors (VSS) enable the reconstruction of unobserved behavior of discrete or hybrid stochastic systems from observable output. Augmented stochastic Petri nets (ASPN) are user models for VSS and describe discrete stochastic models that produce discrete output symbols based on the transitions or system states. Hybrid ASPN (H-ASPN) can describe hybrid stochastic systems with continuous quantities that are influenced by and interact with the discrete system parts, producing samples of these continuous system quantities as observable output. Real-world systems often contain both of these types of observable output. In household energy models, the total consumption is a continuous quantity that can be sampled regularly. Additionally, the usage behavior of some appliances might be known in advance or can be monitored easily, resulting in observable discrete symbols. Being able to utilize both of these for behavior reconstruction, promises better results than using either one alone. In this paper, we describe an extended H-ASPN paradigm for modeling symbol output as well as sample measurement for partially observable hybrid stochastic systems. We demonstrate the paradigm on a small non-intrusive appliance load monitoring (NIALM) example and test the behavior reconstruction. The extended H-ASPN modeling paradigm enables faster and more reliable behavior reconstruction results, when using both observable symbols and samples. The experimental results indicate that extended H-ASPN could lead the way to practically feasible VSS for hybrid systems.

Introduction

Simulation and modeling are used in a forward manner to observe existing systems, build abstract representations and experiment with these in order to draw conclusions on the original systems behavior. Many systems are however not directly observable, but only through their output or interaction with the environment. To analyze these partially observable systems, a backward approach of behavior reconstruction based on the observed output becomes necessary.

Virtual stochastic sensors (VSS) formalize and solve the inverse problem of determining unobservable likely stochastic system behavior based on observable stochastic output. In [1] VSS for discrete and hybrid systems are formalized and tested on academic and real-world applications. Two VSS user models are introduced: Augmented stochastic Petri nets (ASPN) can describe discrete systems that generate output in the form of discrete observable symbols from an alphabet. This output can be triggered by the firing of a transition or based on the current system state. Hybrid augmented stochastic Petri nets (H-ASPN) describe hybrid stochastic systems with continuous reward measures that are sampled independently of the system behavior to generate output protocols.

In a partially observable real-world system, the symbols or samples that are observable can be very different in nature and generated by a wide range of processes: light barriers can detect the passing of an object or person; physical sensors can measure temperature, air pressure, or light intensity; RFID readers can detect the time and ID of an item passing or residing within their range. Most of these can be categorized as either sample or discrete symbol, but a single system may generate both types of observable output. e.g. smart meters can record household energy consumption, which is a continuous measure, and electrical usage monitors can detect when a specific appliance was switched on and off, which can be interpreted as a specific symbol. Currently, ASPN can model discrete signal emissions, and H-ASPN the samples of continuous measures, but there exists no modeling paradigm, that can incorporate both types of emissions. Restricting the model to only one type of emission would disregard readily available information and therefore likely lead to mediocre results

compared to an approach including both types of emissions. Therefore, we introduce a model type that can represent both samples and symbols.

In Section 1 some background on VSS and other related work is given. The formalization of an extended hybrid augmented stochastic Petri net is shown in Section 2, also addressing necessary changes to the underlying solution algorithm. We demonstrate the paradigm and the behavior reconstruction capability using a NIALM problem in Section 3, showing the benefit of combining symbol and sample output. Section 4 will discuss the results and implications.

1 State of the Art

The concept of virtual stochastic sensors (VSS) was introduced in [2] and has since evolved into a framework to describe and solve backward problems in modeling and simulation ([1]). VSS enable the behavior reconstruction of a broad range of partially observable stochastic systems based on the observable output.

Originally, hidden non-Markovian models (HnMM) were developed as computational models for VSS ([3]). Increasing feasibility, conversive hidden non-Markovian models restrict the modeling power to enable a much more efficient solution ([4]). Hybrid hidden non-Markovian models (HHnMM) expand the paradigm to include continuous measures ([5]). The Proxel method ([6]) is used for the actual behavior reconstruction task. It is a state space-based simulation method, which enables a deterministic analysis of stochastic models with arbitrarily distributed process durations employing the method of supplementary variables. Behavior reconstruction for discrete systems of realistic size is currently feasible. However, the extension to hybrid systems increases the computational complexity drastically, making hybrid VSS only applicable to small scale academic models so far.

The user models defined for VSS are augmented stochastic Petri-nets (ASPN), which were first introduced in [4]. ASPN are based on well known stochastic Petri net paradigms ([7, 8]) and expand these using ideas from hidden Markov models (HMM) ([9]) by emissions of discrete symbols. These emissions can either depend on the current system state or can be triggered by the firing of a transition. In both cases, the emission time is recorded in a protocol along with the emitted symbol. The hybrid components in H-ASPN are modeled using ideas from stochastic reward nets (SRN) ([10]) and fluid stochastic Petri nets (FSPN) ([11, 12]). The observable output of a hybrid ASPN is generated by an independent sampling process, that records the values of one or more of the continuous quantities in a protocol along with the sampling time stamp. [1]

By utilizing both continuous and discrete output for VSS behavior reconstruction, we hope to increase the degree of observability of the systems. The concept of observability is also found in control theory, where Kalman filters are used to estimate the development of a system. What is also similar to VSS is the goal of determining a systems internal state from external measurements. Originally, Kalman Filters are designed to estimate the development of deterministic linear systems in discrete steps. Since then, Kalman Filters have been extended to deal with stochastic and non-linear systems and are widely used to predict unobserved or future states of such systems. The general idea is to predict the system development based on an estimate of the current state, and then correct the prediction based on possibly noisy measurements. [13, 14]

VSS differ from Kalman Filters in that it is not necessary to formalize the exact structure of the state and observation equations, including the noise and error terms. Instead, by mimicking the system development and exploring the expanded model state space step by step, we not only avoid the mathematical complexity of Kalman Filters, but also allow for infrequent measurements, observations stemming from state changes and completely unobservable states. VSS represent an addition to the myriad of tools for estimating unobservable quantities and can be employed when the structure and dynamics of the partially observable system are known and can be described by a discrete or hybrid stochastic model such as the ones presented here.

2 Hybrid Augmented Stochastic Petri Nets

In this section, we describe the combination of ASPN and H-ASPN in detail, including the individual model components. The notation and semantics use the formalization from [1]. Therefore, the individual elements do not differ from the ones described there, but only their combination.

Formally, an H-ASPN is a tuple

$$HnMM = (P, T, I, O, H, M_0, V, b, W, w_0, GF, ir, rr)$$

with the following elements:

- $P = \{p_1 \dots p_{|P|}\}$ is a set of *Places* representing physical locations or system states. The marking of the H-ASPN is given by the distribution of tokens in the places of the net $M \in \hat{M} = \mathbb{N}^{|P|}$.
- $T = \{t_1, \dots t_{|T|}\}$ is a set of transitions, which is partitioned into a set of immediate transitions T_I and a set of timed transitions T_T.
  - An element of T_I is associated with a probability $T_I \to [0,1]$, denoting the firing probability in competition situations. If the probability is not stated, it defaults to 1. Immediate transitions can fire as soon as they are enabled, according to their assigned probability.
  - An element of T_T is associated with an arbitrary continuous distribution function and possibly a memory policy (race-age or race-enable). This distribution describes the firing time, which needs to elapse between enabling and firing of a transition.
- $I, O, H : P \times T \to \mathbb{N}_0$ are incidence functions and specify the connection between places and transitions.
  - If $I(p,t) > 0$, an input arc leads from place p to transition t. The value of $I(p,t)$ determines the number of tokens that need to be present in place p for transition t to be enabled, and likewise the number of tokens destroyed in p, when t fires.
  - If $O(p,t) > 0$, an output arc leads from transition t to place p. The value of $O(p,t)$ determines the number of tokens that are created in place p when t fires.
  - If $H(p,t) > 0$, an inhibitor arc leads from place p to transition t. The value of $H(p,t)$ determines the number of tokens that need to be present in p for transition t to be disabled.
- $M_0 = (m_1 \dots m_{|P|})$ ($M_0 : P \to \mathbb{N}_0$) holds the initial marking of the H-ASPN, where m_i denotes the number of tokens in place p_i at the initialization of the system.
- $V = \{v_1 \dots v_{|V|}\}$ is the set of discrete output symbols of the net.
- $b : V \times P \cup V \times T \to [0,1]$ describes the output behavior of the net, mapping the element generating the output and the output symbol to an output probability. In most cases, outputs are associated either to transitions or to places, a combination is however also conceivable and therefore not ruled out.
  - If the output is generated depending on the current system state, then $b : V \times P \to [0,1]$. If a state can produce output symbols, then the sum of the probabilities of all output symbols of the state must be 1: $\exists b(v_i, p_j) > 0 \Rightarrow \sum_{k=1\dots|V|} b(v_k, p_j) = 1$.
  - If the output is generated depending on the ASPN transitions, then $b : V \times T \to [0,1]$. If a transition can produce output symbols, then the sum of the probabilities of all output symbols of that particular transition must be 1: $\exists b(v_i, t_j) > 0 \Rightarrow \sum_{k=1\dots|V|} b(v_k, t_j) = 1$.
- $W : \{w_1 \dots w_{|W|}\}$ is a set of variables, representing continuous system quantities. The current values of these quantities are given in $w_{|W|} \in \hat{W} = \mathbb{R}^{|W|}$.
- $w_0 = (w_1 \dots w_{|W|})$ contains the initial values of the continuous system quantities.
- $GF : T \times \mathbb{N}^{|P|} \times \mathbb{R}^{|W|} \to 0,1$ describes the marking dependent guard of each transition. If the Boolean expression evaluates to 1, the transition is enabled, and it is disabled otherwise.
- $ir : T \times \hat{M} \times \hat{W} \to \mathbb{R}$ (or $\mathbb{R}^{|W|}$) describes a type of impulse reward, which can change the value of a continuous quantity immediately. The change can be dependent on the current marking of the ASPN.
- $rr : P \times \hat{M} \times \hat{W} \to \mathbb{R}$ (or $\mathbb{R}^{|W|}$) describes a type of rate reward, which can change the value of a continuous quantity continuously. The rate of change can be dependent on the current marking of the net. The rate reward can be associated to a specific place or can be completely independent of the discrete system state and be represented by an ODE.

The marking of the places and the values of the continuous quantities together form the potential state space of the H-ASPN $\hat{M} \times \hat{W}$. The elements P, T, I, O, H, M_0 were taken from Petri nets and are widely known. For more details on the semantics and dynamics of SPN refer to general Petri net literature [7, 8].

Elements V and b hold the augmentation information. The specific output semantics are the following:

when a transition with associated output symbols fires, one of these symbols is emitted, sampled according to their probabilities. When symbols are associated with places or specific system states, observations are made through an independent process. The symbol is sampled depending on the current system state and associated output symbols. Observed outputs are collected in a protocol with their respective emission or sampling time stamps. Samples of the continuous measures can be taken at any time, and will be collected in a protocol associated with the time when they were drawn. The protocol can contain values of different continuous quantities, but not every continuous system quantity must be represented.

2.1 Modifications to Computational Model and Solution Algorithm

In order to enable behavior reconstruction for this modified H-ASPN model type, we also need to adapt the computational model and solution algorithm. Analogously to the combination of the user models, the computational model was created by combining the output processes of hidden non-Markovian models (HnMM) and hybrid hidden non-Markovian models (HHnMM) [1] to incorporate discrete symbol emissions as well as samples of continuous model variables.

As solution method, we extended the Proxel-based analysis algorithm for hybrid models (see [1, 5]) to perform behavior reconstruction for extended H-ASPN. The trace can now contain samples of the continuous model quantities as well as discrete output symbols, and both will be handled accordingly. As described in [5], the parametrization process for the Proxel method for hybrid systems is tedious, since the inclusion of continuous variables considerably increases the complexity of the models. Countering the inherent problem of state-space explosion of the Proxel method becomes more difficult with more method parameters. Instead of having a binary decision on the validity of a Proxel, as with discrete outputs, the estimate of the continuous measure in the Proxel can be within ε of the observed sample. The size of ε and the cutoff probability or number for pruning the Proxel tree both need to be balanced, in order to achieve a useful reconstruction result in a computationally feasible manner.

In the following experiment section, we will show how the behavior reconstruction works for the extended paradigm.

3 Example and Experiments

For our proof of concept, we will use a small example constructed from the UMass Smart* Data Set for Sustainability [15] (2013 release). The data was gathered from several private homes over the period of several months, and includes aggregate power readings, circuit and appliance level consumption as well as environmental data. The goal of non-intrusive appliance load monitoring (NIALM) is to dis-aggregate a cumulative household energy consumption into the contributions of individual appliances or circuits. For demonstration purposes, we have chosen a subset of three circuits with different characteristics from House A, namely the *Dryer*, *MasterLights* and *CounterOutlets1*. For a detailed description of the model construction, please refer to [16]. The data set uses UNIX time stamps, and thus seconds as basic time unit. The goal for the experiments is to determine the unobserved system behavior in terms of appliance state changes from the cumulative power consumption, with and without information on individual appliances state changes.

3.1 NIALM Example System

We parameterized the model using the circuit level data of five consecutive days (May 6th to 10th), and performed behavior reconstruction using the aggregate power consumption of the following day (May 11th). The graphical representation of the H-ASPN can be found in Figure 1. Well known Petri-net elements are represented as follows, places as circles, timed transitions as open rectangles, immediate transitions as vertical bars associated with a probability, and arcs as solid line arrows. The places represent tangible appliance usage states (*Dr1*, *Dr2*, *Dr3*, *ML1*, *ML2*, *CO1*, *CO2*). The timed transitions associated with the state changes of the *MasterLights* (*MLon*, *MLoff*) and the *CounterOutlets* (*COon*, *COoff*) emit a symbol when firing, denoted by a dotted arrow. The different rate rewards in the different appliance states are depicted by dashed arrows annotated with the average consumption rate in that state. Since the appliances have separate state spaces, the rewards of the currently active states are added to form the overall current consumption (collected in *Reward r*). For clarity reasons, not all elements of the H-ASPN are named.

In the formal description of the H-ASPN in Figure 2, H, ir and GF are omitted, since the system does not

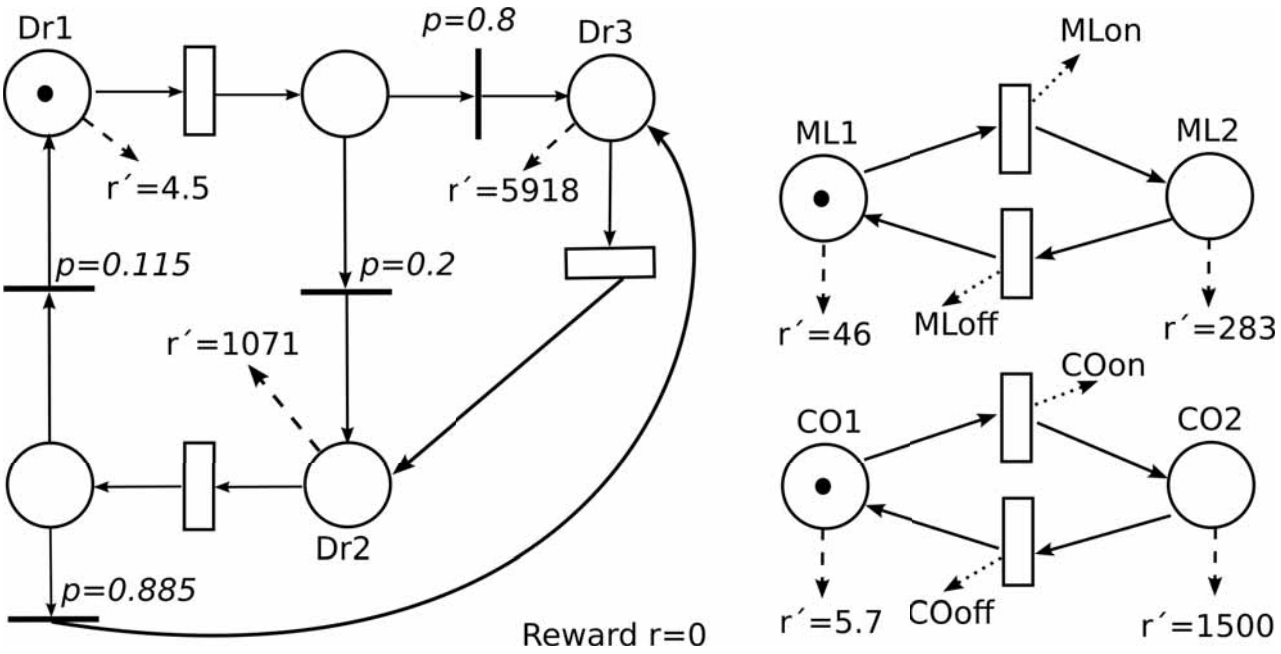

Figure 1: H-ASPN of the NIALM Example Process

include inhibitor arcs, impulse rewards or guard functions.

The timed transitions governing the appliance state changes were fitted with Weibull distributions and Exponential distributions with the same expected values. The Weibull distribution was used, since it mimics the actual switching behavior of the appliances best [16], and the Exponential distribution as comparison, because it is least restrictive, being memoryless. In contrast to the models used in [16], where histograms are used to represent a small number of different consumption levels per state, we only used one average power consumption value per state.

3.2 Validation Experiment

The cumulative consumption trace of May 11th was extracted from the Smart* Data Set, as were the state change points of the *MasterLights* and the *CounterOutlets1*. The time stamps were normalized to start at 0 with the first item of the trace. An excerpt of the trace is depicted in Table 1. The ground truth, in terms of state changes in all three appliances is also known from the data set and depicted in Table 2, where one can see, that on the test day, the *Dryer* was not in use at all, but stayed in state *Dr1* with the lowest consumption.

In the experiments, we want to test, how including some symbols in the trace affects the accuracy of the reconstruction, compared to a reconstruction based only on samples. Therefore, we constructed the model in different variations: (0) no symbol emissions at all, (1) state changes of the *MasterLights* and *CounterOutlets* are detectable, however cannot be distinguished through the symbol emitted, (2) the state changes of *MasterLights* and *CounterOutlets* are detectable and can be distinguished from their symbols, which corresponds to the model formalized in the previous section.

Trace	
Time Stamp	Sample/Symbol
...	...
5169	287789
6175	466894
6175	*MLon*
6178	467708
6287	498265
6358	518035
6362	519155
6554	529992
6554	*MLoff*
6714	538616
...	...

Table 1: Excerpt of the Cumulative Consumption Trace with Symbols

$$
\begin{aligned}
Circuits &= (P,T,I,O,M_0,V,b,W,\vec{v},rr)\\
P &= \{Dr1,Dr2,Dr3,Dr1_{Done},Dr2_{Done},\\
&\quad ML1,ML2,CO1,CO2\}\\
T_I &= \{Dr1to2,Dr1to3,Dr2to1,Dr2to3\}\\
T_T &= \{Dr1_T,Dr2_T,Dr3_T,ML1_T,ML2_T,\\
&\quad CO1_T,CO2_T\}\\
I(p,t) &= \begin{cases}
1: & (p=Dr1 \wedge t=Dr1_T)\\
& \vee(p=Dr3 \wedge t=Dr2_T)\\
& \vee(p=Dr2 \wedge t=Dr3_T($\\
& \vee(p=Dr1_{Done} \wedge$\\
& (t=Dr1to2 \vee t=Dr1to3))\\
& \vee(p=Dr2_{Done} \wedge$\\
& (t=Dr2to1 \vee t=Dr2to3))\\
& \vee(p=ML1 \wedge t=ML1_T)\\
& \vee(p=ML2 \wedge t=ML2_T)\\
& \vee(p=CO1 \wedge t=CO1_T)\\
& \vee(p=CO2 \wedge t=CO2_T)\\
0: & else
\end{cases}\\
O(p,t) &= \begin{cases}
1: & p=Dr1_{Done} \wedge t=Dr1_T\\
& \vee p=Dr2_{Done} \wedge t=Dr2_T\\
& \vee p=Dr1 \wedge t=Dr2to1\\
& \vee p=Dr2 \wedge t=Dr1to2\\
& \vee p=Dr2 \wedge t=Dr3to2\\
& \vee p=Dr3 \wedge t=Dr1to3\\
& \vee p=Dr3 \wedge t=Dr2to3\\
& \vee p=ML1 \wedge t=ML2_T\\
& \vee p=ML2 \wedge t=ML1_T\\
& \vee p=CO1 \wedge t=CO2_T\\
& \vee p=CO2 \wedge t=CO1_T\\
0: & else
\end{cases}\\
M_0 &= (1,0,0,0,0,1,0,1,0)\\
V &= \{MLon,MLoff,COon,COoff\}\\
& b(MLon,ML1_T)=1\\
& b(MLoff,ML2_T)=1\\
& b(COon,CO1_T)=1\\
& b(COoff,CO2_T)=1\\
& else\ b(v,t)=0\\
W &= r\\
\vec{w_0} &= (0)\\
rr(Dr1) &= 4.5\\
rr(Dr2) &= 1071\\
rr(Dr3) &= 5918\\
rr(ML1) &= 46\\
rr(ML2) &= 283\\
rr(CO1) &= 5.7\\
rr(CO2) &= 1500
\end{aligned}
$$

Figure 2: Formal description of H-ASPN of the NIALM Example Process

Generating Path	
Time Stamp	State Change
6175	$ML1_T$
6554	$ML2_T$
6982	$ML1_T$
7381	$ML2_T$
7659	$CO1_T$
7874	$CO2_T$

Table 2: Ground Truth for Appliance State Changes

In case (0), the trace only contains samples of the continuous quantity, and the times when they were measured. In cases (1) and (2) the trace also contains symbols with the time stamps of when they were emitted.

For the experiments, we used a maximum simulation time of 10,000, encompassing all symbols and state changes in the ground truth, and a Proxel discretization time step of 60 seconds. For all experiments shown here, we varied the ε cutoff threshold and Proxel cutoff number to result in a feasible run, with smallest possible ε. Both parameters affect the number of paths discovered and the runtime [5, 1], and therefore will not be compared here directly. The ε parameter had to be chosen at 10,000 or more, since the time step of 60 could result in large gaps between actual and reconstructed state change times, resulting in larger deviations between the actual and the reconstructed consumption.

As a first test, we ran the Proxel analysis algorithm with the version (0) models and traces without symbol emissions. The model with Weibull distributions resulted in paths with 166-170 state changes, which corresponds to one state change in almost every time step of the analysis, most of these of *MasterLights* or *CounterOutlet*. Thus the reconstructed paths using the model with Weibull distributions bear no resemblance to the ground truth path, and are therefore not practically useful. The most likely path reconstructed using the model with Exponential distributions is shown in Table 3. The algorithm failed to reconstruct the state changes in the *MasterLights*, mis-matching two of them as *CounterOutlet* switches, but at least matching the switch timings. Considering the ambiguity and large time step size of the reconstruction, this is a mediocre result.

As a next step, we used model variant (2), where *MasterLight* and *CounterOutlets1* state changes are detectable and can be distinguished by the symbols emitted. Unfortunately, the paths reconstructed using the model with Weibull distributions again contained one state change in almost every time step, and were therefore not useful here. The most likely path reconstructed using the model with Exponential distributions is shown in Table 4. The path resembles the ground truth very closely, only the *Dryer* state change from $Dr1$ to $Dr2$ at 6120 and back again 2 minutes later does not correspond to the actual system behavior. This shows that, even if only some state changes are detectable, the reconstruction resembles the ground truth much better than without detectable state changes.

Reconstructed Path	
Time Stamp	State Change
6120	$CO1_T$
6240	$CO2_T$
7620	$CO1_T$
7920	$CO2_T$

Table 3: Reconstructed Path Without Detectable State Changes

Reconstructed Path	
Time Stamp	State Change
6120	$DR1_T$
6120	$DR1to2$
6180	$ML1_T$
6240	$DR2_T$
6240	$DR2to1$
6600	$ML2_T$
7020	$ML1_T$
7440	$ML2_T$
7680	$CO1_T$
7920	$CO2_T$

Table 4: Reconstructed Path With Distinguishable State Changes

In model variant (1) all detectable state changes emit the same symbol, which decreases the systems degree of observability by making the state changes indistinguishable. The reconstructed most likely path still corresponds to the one with distinguishable state changes depicted in Table 3, correctly matching *MasterLight* and *CounterOutlets1* state changes.

3.3 Performance Analysis

We also want to compare the algorithm performance when including or excluding symbols. We will only use the two models using Exponential distributions, since only these resulted in useful reconstructions. When comparing the runtime for both algorithms with the same ε and cutoff threshold parameters, the analysis of

the model without symbol emissions needs 20 seconds, whereas the analysis when including symbols needs only about 4 seconds.

Figure 3 shows the development of the number of valid paths over time. The upper graph shows the number oscillating around 1500 but staying above 500 after an initial period. The lower graph shows more pronounced drops in this number, some of which can be associated to observed symbols. This is due to the higher degree of observability of the model including detectable state changes, and thus a larger number of paths becoming invalid throughout the analysis.

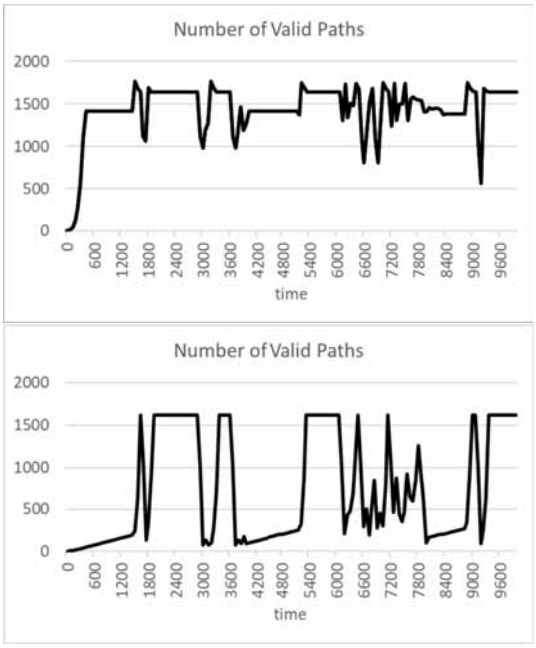

Figure 3: Valid Paths Over Simulation Time Without (Top) And With (Bottom) Detectable State Changes

Figure 4 shows the development of the trace probability in log scaling over time. In both cases, the probability decreases gradually, until the first detectable state change, also in the model without state change detection. In the lower graph, each detectable state change leads to a more noticeable drop in trace probability, since invalidating paths decrease the remaining probability in the analysis.

3.4 Result Discussion

Since the application example presented here is intended only as a proof of concept for the usefulness of

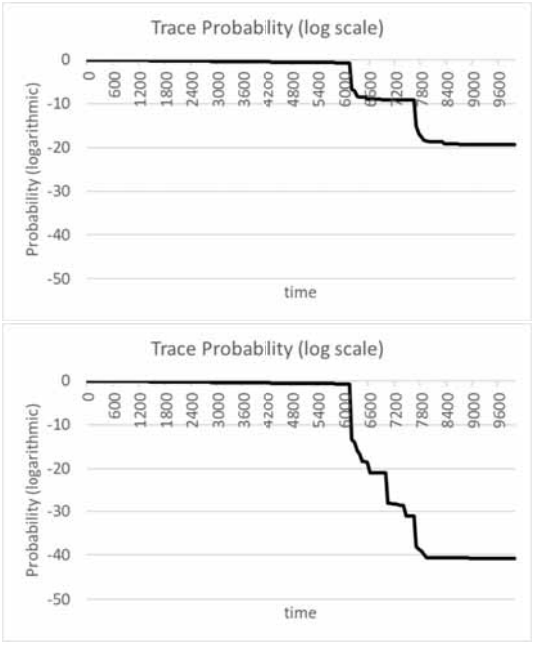

Figure 4: Trace Probability Over Simulation Time Without (Top) And With (Bottom) Detectable State Changes

extended Hybrid-ASPN, improving the results further is future work, and needs more extensive research. Based on these experiments, we can conclude, that behavior reconstruction based on a combination of discrete symbol emissions and samples of continuous measures is possible using the Proxel-based method. Furthermore, is the reconstruction accuracy of the method considerably improved by using the combination of symbols and samples. In the example tested here, the model was to coarse to allow accurate reconstruction only based on samples of the continuous quantity. Only the inclusion of samples resulted in reconstructed paths close to the ground truth. Increasing the models degree of observability through the inclusion of symbols, results in a 5 times faster runtime for the example investigated here. This is due to a smaller number of possible paths that need to be tracked during the analysis, thus resulting in a reduction in state space explosion.

4 Conclusion and Outlook

The paper introduced a user model for virtual stochastic sensors to perform behavior reconstruction of partially observable hybrid systems based on observable output. In contrast to the user models so far, the extended

Hybrid-ASPN enable the modeling of two different types of system output, samples of continuous measures as well as discrete symbols. The new paradigm is demonstrated using a NIALM example problem. The reconstruction experiment shows that being able to utilize both symbols and samples for behavior reconstruction considerably increases the method performance and result accuracy. The increase in performance when including symbols along with sample output shows a way of making virtual stochastic sensors for hybrid systems feasible. Future work must include further investigation of the presented application in NIALM. Complete household models should be tested, as well as a more detailed representation of the state output. This and further real world application scenarios can lead to practically feasible VSS for partially observable hybrid stochastic systems.

References

[1] Krull C. Virtual Stochastic Sensors: Formal Background and Example Applications Reconstructing the Behavior of Partially Observable Discrete and Hybrid Stochastic Systems. Habilitation thesis, Otto-von-Guericke-University Magdeburg. 2021.

[2] Krull C, Buchholz R, Horton G. Virtual Stochastic Sensors: How to gain Insight into Partially Observable Discrete Stochastic Systems. In: *The 30th IASTED International Conference on Modelling, Identification and Control, 14th-16th February 2011, Innsbruck. Austria*. 2011; .

[3] Krull C, Horton G. Hidden Non-Markovian Models: Formalization and Solution Approaches. In: *Proceedings of 6th Vienna Conference on Mathematical Modelling, Vienna, Austria*. 2009; pp. 682–693.

[4] Buchholz R. Conversive Hidden Non-Markovian Models. Ph.D. thesis, submitted to Otto-von-Guericke-University Magdeburg. 2012.

[5] Krull C, Horton G. Virtual Stochastic Sensors for Hybrid Systems: Mutual - Influence between Continuous and Discrete System Parts. In: *Proceedings of ASIM 2014 - 22. Symposium Simulationstechnik, Berlin*. 2014; .

[6] Lazarova-Molnar S. The Proxel-Based Method: Formalisation, Analysis and Applications. Ph.D. thesis, Otto-von-Guericke-University Magdeburg. 2005.

[7] Bobbio A, Puliafito A, Telek M, Trivedi KS. Recent Developments in Non-Markovian Stochastic Petri Nets. *Journal of Systems Circuits and Computers*. 1998;8(1):119–158.

[8] Bause F, Kritzinger PS. *Stochastic Petri Nets*. Vieweg. 2002.

[9] Fink GA. *Markov Models for Pattern Recognition*. Berlin, Heidelberg: Springer. 2008.

[10] Ciardo G, Blakemore A, Jr PFC, Muppala JK, Trivedi KS. Automated generation and analysis of Markov reward models using stochastic reward nets. In: *Linear Algebra, Markov Chains, and Queueing Models*, pp. 145–191. Springer. 1993; .

[11] Wolter K, Horton G, German R. Non-Markovian Fluid Sochastic Petri Nets. *Tech. rep.*, TU Berlin. 1996.

[12] Ciardo G, Nicol D, Trivedi K. Discrete-event simulation of fluid stochastic Petri nets. *IEEE transactions on software engineering*. 1999;25(2).

[13] Ma H, Yan L, Xia Y, Fu M. *Kalman Filtering and Information Fusion*. Springer, Science Press Beijing. 2020.

[14] Triantafyllopoulos K. *Bayesian Inference of State Space Models: Kalman Filtering and Beyond*. Springer. 2021.

[15] Baker S, Mishra A, Irwin D, Cecchet E, Shenoy P, Albrecht J. Smart*: An Open Data Set and Tools for Enabling Reearch in Sustainable Homes. In: *Proceedings of SustKDD'12, Beijing, China*. 2012; .

[16] Krull C, Thiel M, Horton G. Testing Applicability of Virtual Stochastic Sensors for Non-Intrusive Appliance Load Monitoring. In: *Proceeding of the Ninth International Workshop on Practical Applications of Stochastic Modelling*. 2017; .