# A Possibilistic I/O Hidden Semi-Markov Model For Assessing Cyber-Physical Systems Effectiveness

Gérald Rocher[1,2], Jean-Yves Tigli[2], Stéphane Lavirotte[2] and Nhan Le Thanh[2]
[1]*GFI Informatique, Groupe Innovation*, Saint-Ouen, France,
[2]*Université Côte d'Azur, CNRS, Laboratoire I3S, UMR 7271*, France.
Email: gerald.rocher@gfi.fr, jean-yves.tigli@unice.fr, stephane.lavirotte@unice.fr, nhan.le-thanh@unice.fr

***Abstract*—The complexity of the Cyber-Physical Systems (CPS) and their interactions with the physical environment makes them difficult to model completely. Once deployed, these systems are therefore subject to unexpected events that may degrade their behavior. Consequently, it becomes necessary to evaluate, at runtime, the degree of effectiveness of these systems. To address this concern, a new approach is proposed, drawing its foundations from the possibility theory and accounting for temporal constraints. The degree of effectiveness is then given as the degree of possibility. A step further, the Choquet integral operator is used for aggregating multiple evaluations, providing us with a measure of the users' satisfaction w.r.t. their preferences. Finally the proposed approach is validated through a use-case in the smart-home domain.**

***Index Terms*—Cyber-physical Systems, dependability, trustworthiness, effectiveness, Markov models, Possibility theory**

## I. Introduction

Cyber-Physical Systems (CPS), in the broad sense, are computing systems whose purposes are achieved from interactions with the physical world by means of transducers (sensors and actuators). These systems pose new challenges in terms of dependability. Indeed, their behavior can be affected by unanticipated physical processes over which they have no control and which may potentially hamper the achievement of their purposes.

Consequently, due to these *uncertainties*, designers of such systems can no longer lean on comprehensive and reliable models for anticipating and removing faults that may arise at runtime. Thus, uncertainty is now considered a first-class concern in the CPS community [1] and has given raise to several projects attempting to address it. For instance, *RELYonIT*[2], *U-Test* [3] and *Dependable Internet of Things in Adverse Environments* [4]. Most of the approaches address the problem by attempting to model uncertainty. This makes sense within controlled environments where uncertainties can be toroughly identified and modeled.

In this paper however, it is assumed that, in the context of CPS, uncertainties are unlikely to be completely identified [5]. We do start from the premise that the *effects* to be produced by these systems over time are, at the very least, known at design time. Therefore, it is a matter of (1) modeling the expected effects and their evolution (denoted as 'behavior' in the sequel), (2) observing the systems *in vivo* and (3) evaluating them for *effectiveness* (Fig.1).

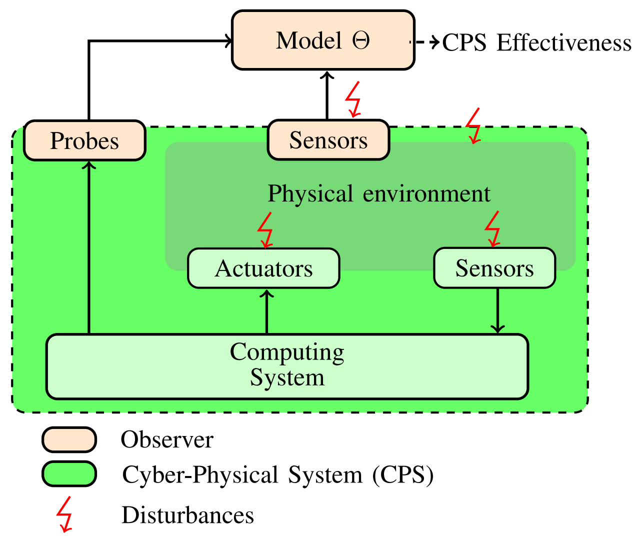


**Fig. 1:** Approach overview. The model $\Theta$ reflects the evolution of the effects to be produced by the CPS, characterized through observable physical properties and computing system internal signals. The effectiveness is computed from the model and the observations.

The solution sought to be provided can thus be stated in these terms: *Given the model* $\Theta$*, compute the possibility that an observation sequence* $\vec{y}_{1:K}$ *has been produced by the CPS*. The possibility provides us with a direct insight on the effectiveness of the CPS as a distance between the desired and the observed behavior.

The contribution of this paper is twofold:

1) A new modeling framework is introduced for describing the evolution of the effects to be produced by the CPS. This model is denoted as *Possibilistic Input/Output Hidden semi-Markov Model* (P-IOHSMM). An algorithm is proposed for computing the *degree of possibility*, taking into account temporal constraints.
2) The case where multiple behaviors (controlled either by one or multiple CPS) are simultaneously assessed within a given environment is investigated. The objective is to obtain a global value for the degree of possibility. To this end, the Choquet integral aggregation operator is used.

## II. Background

### A. Fuzzy sets and possibility theory

Fuzzy sets [6] are sets whose elements have *degrees of membership*. More formally, let $U$ be a universe of discourse. A fuzzy value on $U$ is characterized by a fuzzy set $F \in U$ [7]. A *membership function* $\mu_F : U \to [0, 1]$ is associated with the fuzzy set $F$ and $\mu_F(v)$ , $v \in U$, denotes the degree of membership of $v$ in $F$. When the membership function $\mu_F(v)$ is explained to be a measure of the *possibility* that a variable $V$ has the value $v$, where $V$ takes values in $U$, a *fuzzy value* is described by a *possibility distribution* $\pi_V(.)$. Hence, $\pi_V(v)$ denotes the possibility that $v$ is true. Furthermore, given two variables $X$ and $Y$ with respective possibility distributions $\pi_X(.)$ and $\pi_Y(.)$, the *joint possibility distribution* $\pi_{X,Y}(.)$ is given by [8]:

$$\pi_{X,Y}(u, v) = \min(\pi_X(u), \pi_Y(v)) \tag{II-A.1}$$

Let us consider the context of estimating the effectiveness of a CPS by observing its behavior. In this context, a variable $V$ corresponds to the range of values a particular sensor can take. Thus, $\pi_V(.)$ defines a *viability zone* of the CPS [9] and $\pi_V(v)$ denotes the *degree of possibility* that the behavior of the CPS, characterized by $v$, lies within the viability zone.

For instance, let us consider a state $x$. One can define the possibility distribution $\pi_x(.)$ where $\pi_x(v)$ denotes the degree of possibility that $v$ corresponds to the expected emission level while being in this state (Fig.2). In this paper, *normalized possibility distributions* are considered such that $\pi_V(.)$ takes values in the range $[0, 1]$ and $\exists v$, $\pi_V(v) = 1.0$.

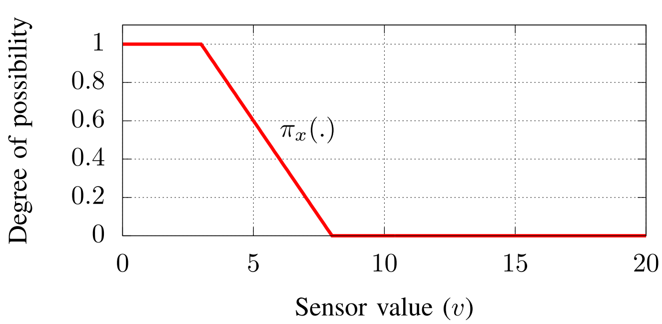


**Fig. 2:** Let us consider a state $x$. One can define the possibility distribution $\pi_x(.)$. Thus, $\pi_x(v)$ denotes the degree of possibility that $v$ corresponds to the expected level while being in this state.

### B. Possibilistic Input/Output Hidden Markov Model

Possibility theory-based Hidden Markov Models have been applied in numerous areas (e.g. speech recognition [10] and intrusion detection [11], just to name a few) and can be found in the literature under the terms *Possibilistic HMMs* or *Fuzzy HMMs*. Their utilization is justified for real life situations unlikely to be described precisely through probability density functions (pdfs) [12]. In this context, possibility distributions allow some flexibility leading, for instance, the false alarm rate of classical probabilistic HMM-based Intrusion Detection Systems (IDS) to be significantly lowered [11].

HMMs rely on computationally efficient algorithms, offering solutions to address the following canonical problems:

- The evaluation problem consists in computing the degree of possibility of an observation sequence to have been produced by the model,
- The state estimation problem (*filtering*) consists in computing the degree of possibility of ending in a particular state given an observation sequence,
- The state sequence decoding problem consists in computing the most possible underlying hidden state sequence that has been ran through to produce an observation sequence,
- The parameter learning problem consists in computing the parameters of the model that maximize the degree of possibility of an observation sequence.

The solutions HMMs provide to address these problems along with their ability to represent dynamical systems and capture their uncertainties [13][14], make them particularly well adapted for assessing CPS effectiveness [15]. Our approach lies on the Possibilistic Input/Output HMM (P-IOHMM) which draws its foundations from [16]. This model, whose possibilistic network is depicted in Fig.3, extends standard possibilistic HMMs by allowing state transition and/or emission possibilities to be conditionally dependent on an input sequence.

Formally, a P-IOHMM is defined by the tuple $< Q, \vec{\pi}, A, \vec{B} >$ where:

- $Q = \{x_1, x_2, \ldots, x_N\}$ is the finite set of hidden states; $x_{(k)}$ denotes a hidden state at time $k$,
- $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)^T$ is the initial state distribution vector. $\pi_i$ denotes the degree of possibility of the state $i$ to be the first state of a state sequence,
- $A$ is the $N \times N$ state-transition matrix, where each cell $A_{ij}$ of the matrix is an input $d_{ij}$-dimensional distribution of possibility ($1 \le i, j \le N$). $A_{ij}(\vec{u}) = p(x_{(k+1)} = j | x_{(k)} = i, \vec{u}_{(k)} = \vec{u})$ denotes the degree of possibility of transitioning to state $x_{(k+1)} = j$ at time $k + 1$, given the current state $x_{(k)} = i$ and the input vector $\vec{u}_{(k)} = \vec{u}$ at time $k$. The sample space of $\vec{u}_{(k)}$ is continuous (i.e. realizations of $\vec{u}_{(k)} \in \mathbb{R}^{d_{ij}}$),
- $\vec{B} = (B_1, B_2, \ldots, B_N)^T$ is the state-emission vector, where each element $B_i$ ($1 \le i \le N$) is an output $b_i$-dimensional distribution of possibility. $B_i(\vec{y}) = p(\vec{y}_{(k)} = \vec{y} | x_{(k)} = i)$ denotes the degree of possibility of observing the output vector $\vec{y}_{(k)} = \vec{y}$ at time $k$ while being in the state $x_{(k)} = i$. The sample space of $\vec{y}_{(k)}$ is continuous (i.e. realizations of $\vec{y}_{(k)} \in \mathbb{R}^{b_i}$).

With this model, the evaluation problem is put in these terms: *Given the parameters $\Theta =< \vec{\pi}, A, \vec{B} >$ of the model and an input sequence $\vec{u}_{1:K}$ of length $K$, compute the degree of possibility of an output sequence $\vec{y}_{1:K}$ to have been produced by the model, i.e. $\mathcal{P}^\star{}_{(\vec{y}_{1:K}|\Theta,\vec{u}_{1:K})}$.* The solution to this problem (as well as the solution to the state sequence decoding

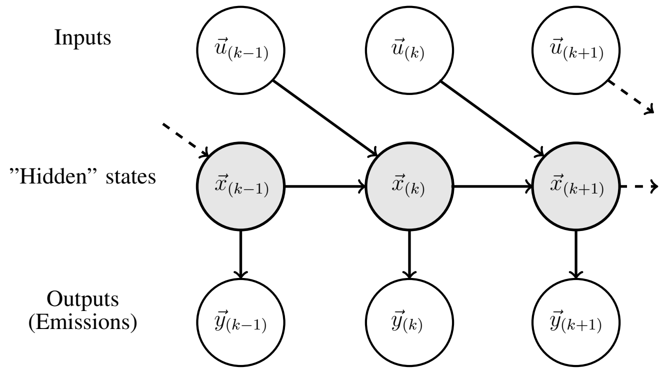


**Fig. 3:** Possibilistic network expressing conditional dependencies for an P-IOHMM. The model is said "hidden" because the states of the processes they model are not directly observable but inferred from inputs $\vec{u}$ and outputs $\vec{y}$.

problem) is given by the following possibilistic version of the *Viterbi* algorithm (which is derived from [12]).

1) Initialization

$$\begin{aligned} &\alpha_{(1)}(i) = \min\left(\pi_i, B_i(\vec{y}_{(1)})\right), 1 \le i \le N \\ &\varphi_{(1)}(i) = 0 \end{aligned} \tag{II-B.1}$$

where $\alpha_{(1)}(i)$ is the joint possibility of starting in state $i$ and observing $\vec{y}_{(1)}$

2) Recursion

$$\alpha_{(k+1)}(i) =$$

$$\min\left\{\left[\max_{1\le j\le N}\left[\min(\alpha_{(k)}(j), A_{ji}(\vec{u}_{(k)}))\right]\right], B_i(\vec{y}_{(k+1)})\right\} \tag{II-B.2}$$

$$\varphi_{(k+1)}(i) = \arg\max_{1\le j\le N}\left[\min(\alpha_{(k)}(j), A_{ji}(\vec{u}_{(k)}))\right]$$

for $1 \le k \le K-1$, $1 \le i \le N$,

3) Termination

$$\mathcal{P}^{\star}{}_{(\vec{y}_{1:K}|\Theta,\vec{u}_{1:K})} = \max_{1\le i\le N}\left[\alpha_{(K)}(i)\right] \tag{II-B.3}$$

$$\mathcal{X}^{\star}{}_{(K)} = \arg\max_{1\le j\le N}\left[\alpha_{(K)}(i)\right] \tag{II-B.4}$$

4) Estimated hidden state sequence backtracking

$$\mathcal{X}^{\star}{}_{(k)} = \varphi_{(k+1)}(\mathcal{X}^{\star}{}_{(k+1)}),\ k = K-2, K-3, ..., 1 \tag{II-B.5}$$

## III. Possibilistic Input/Output Hidden Semi-Markov Model

In this section, the P-IOHMM previously described is extended with *temporal aspects*. In the context of assessing CPS effectiveness, handling temporal aspects allows to cover a broader range of systems, e.g. aiming at controlling processes with intrinsic inertia (e.g. thermal process). This model is denoted as *Possibilistic Input/Output Hidden semi-Markov Model* (P-IOHSMM). Temporal aspects are handled at state-transition and state emission levels. It extends P-IOHMM (Section.II-B) by allowing state transition and emission possibilities to also be conditionally dependent on the amount of time that has elapsed since the last entry into the current state. To this end, the P-IOHMM model is extended with two additional elements:

- $\vec{S_d} = (S_{d_1}, S_{d_2}, \ldots, S_{d_N})^T$ is the state duration vector where each element $S_{d_i}$ $(1 \le i \le N)$ is a one-dimensional distribution of possibility. $S_{d_i}(z)$ $(z \in \mathbb{N})$ is the degree of possibility of being in the state $x_{(k)} = i$ during $z$ consecutive observations[1].
- $T_d$ is the $N \times N$ state-transition duration matrix where each cell $T_{d_{ij}}$ $(1 \le i, j \le N)$ is a one-dimensional distribution of possibility. $T_{d_{ij}}(z)$ $(z \in \mathbb{N})$ is the degree of possibility for the state-transition $A_{ij}$ to last during $z$ consecutive observations. In other words, it is the elapsed time required for the output of the next state $x_{(k+1)} = j$ to get stabilized (settling time) with $T_{d_{ij}} < S_{d_j}$.

Then, the notion of *unknown state* is introduced, i.e. state that is not explicitly defined in the model. This is particularly relevant for state-transitions for which a settling time is allowed and during which the state of the system might be undefined (transient state). Thus, an unknown state $x^?$ is defined as being a state where the possibility of being in any defined state is equal to 0 (II-B.1, II-B.2 and II-B.5):

$$x^?_{(k)} = \begin{cases} \mathcal{X}^{\star}{}_{(k)} & \text{if } \sum_{i=1}^{N} \alpha_{(k)}(i) > 0 \\ -1 & \text{otherwise} \end{cases} \qquad 1 \le k \le K$$

The main idea behind our approach is then to extend the Viterbi algorithm in order to reveal unknown states in the decoded state sequence. Thus, one can leverage the decoded state sequence obtained from this model for computing the state and state-transition durations. Algorithms are detailed in Appendix.A. Furthermore, some explanations are provided in Section.V-D and Fig.10.

To the best of our knowledge, this approach is new. Although different, it is worth noting the work done on pattern recognition using temporal fuzzy automata [17].

## IV. Aggregating Degrees of possibility

In the previous section (Section.III) a new modeling framework has been introduced for assessing, through the degree of possibility, the CPS behavioral effectiveness by taking into account temporal constraints. In this section, the case where *multiple behaviors* (controlled either by one or multiple CPS) are simultaneously assessed within the physical environment is investigated. Our intention here is to address this issue *globally*, i.e. obtain a global behavioral assessment from the local assessments. To this end, the *Choquet Integral* aggregation operator is leveraged. This operator is widely used for multi-criteria decision making [18] and preferences modeling [19].

Let $C = \{c_1, c_2, \ldots, c_n\}$ be the finite set of criteria. A *capacity* [20] is a set function $\mu : 2^C \to [0, 1]$ satisfying:

- $\forall A, B \in 2^C, A \subseteq B \Rightarrow \mu(A) \le \mu(B)$ (monotonicity)

[1]The elapsed time is given as the number of observations and consequently it depends on their sampling rate.

- $\mu(C) = 1$ (normality)
- $\mu(\emptyset) = 0$

Thus, $\mu(\{k \in 2^C\})$ is the *weight* of the criterion $k$.

Let $X = \{x_1, \dots, x_n\} \in \mathbb{R}^n_+$ the scores obtained for each criterion. The Choquet integral of $x$ for criteria $C$ w.r.t. a capacity $\mu$ is given by:

$$\oint_\mu (X) = \sum_{i=1}^{n} (x_{\sigma(i)} - x_{\sigma(i-1)})\mu(C_{\sigma(i)}) \qquad \text{(IV-.1)}$$

where $\sigma$ is a permutation on $X$ such that $x_{\sigma(1)} \leq \dots \leq x_{\sigma(n)}$ and $C_{\sigma(i)} = \{c_{\sigma(i)}, \dots, c_{\sigma(n)}\}$.

Let us now consider an example where three behaviors are assessed through the degrees of possibility $p_1$, $p_2$ and $p_3$. Thus $|C| = 3$ and $X = \{p_1, p_2, p_3\}$. Now, if one consider $X_1 = \{0.43, 0.98, 0.7\}$ and $X_2 = \{0.84, 0.28, 0.91\}$ with $\mu(C) = 1$, $\mu(\emptyset) = 0$, $\mu(\{1\}) = 0.3$, $\mu(\{2\}) = 0.2$, $\mu(\{3\}) = 0.3$, $\mu(\{1,2\}) = 0.5$, $\mu(\{1,3\}) = 0.87$ and $\mu(\{2,3\}) = 0.45$, then:

$$\begin{aligned}\oint_\mu (X_1) &= 0.43 + (0.7 - 0.43)\mu\{2,3\} + (0.98 - 0.7)\mu(\{2\}) \\ &= 0.607.\end{aligned} \qquad \text{(IV-.2)}$$

$$\begin{aligned}\oint_\mu (X_2) &= 0.28 + (0.84 - 0.28)\mu\{1,3\} + (0.91 - 0.84)\mu(\{3\}) \\ &= 0.788.\end{aligned}$$

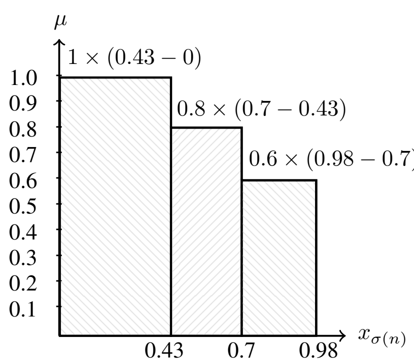


**Fig. 4:** Choquet integral graphical representation. The dashed surface is equal to the Choquet integral $\oint_\mu (X_1)$ where $X_1 = \{0.43, 0.98, 0.7\}$ (see IV-.2)

Choquet integral has interesting properties [21]. In particular it doesn't assume that criteria are independent to each other. For instance [22], in the previous example:

- $\mu(\{1\}) + \mu(\{2\}) = \mu(\{1,2\}) \Rightarrow$ additivity, i.e. $c_1$ and $c_2$ are independent,
- $\mu(\{1\}) + \mu(\{3\}) < \mu(\{1,3\}) \Rightarrow$ super-additivity, i.e. coalition of $c_1$ and $c_3$ has a synergistic effect,
- $\mu(\{2\}) + \mu(\{3\}) > \mu(\{2,3\}) \Rightarrow$ sub-additivity, i.e. coalition of $c_1$ and $c_3$ are redundant to each other.

Thus, capacity function, beyond representing the importance of each criteria, also represents the importance of all their possible coalitions. [23]. The main difficulty here is to determine the $2^C$ weight values where the complexity of identifying these values increases with the number of criteria. In the context of this paper, a criteria corresponds to a behavior. The weight values for each criteria $\in 2^C$ define the preferences of the users w.r.t. the effectiveness of the behaviors ($X = x_1, \dots, x_n$).

## V. Experiments and Results

### A. Environment Setup

The HOME I/O 3D simulation environment [24] is used for validating the proposed model. This environment allows to create and monitor a real-time smart home simulation (see Fig.5). It makes available more than 400 I/O points for interacting with lighting, heating and other smart home devices. Additionally, it manages real-time simulation of power consumption as well as brightness and thermal behaviors by taking into account weather conditions, location and properties of the building. Thus, as a basis for our experiments, a simple home automation controller using Home I/O together with *Scratch2* [25] has been developed. This controller is based

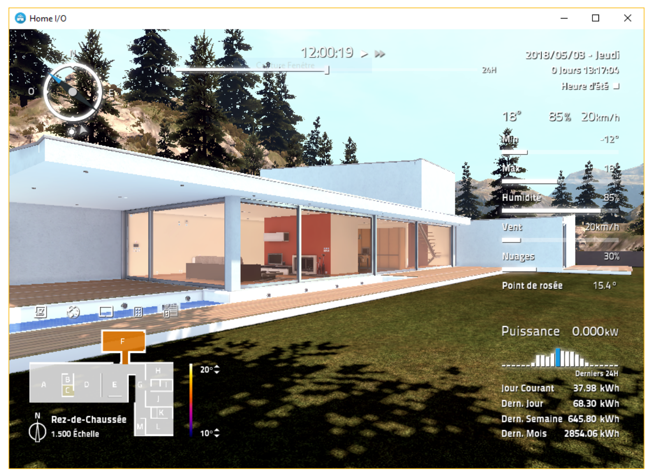


**Fig. 5:** HOME I/O - A 3D interactive Smart Home Simulation


**Fig. 6:** Home I/O makes available more than 400 I/O points for interacting with lighting, heating and other smart home devices

on simple Event Condition Action (ECA) rules allowing to control room lighting and temperature (Fig.7).


**Fig. 7:** ECA rule example

### B. Behavioral models

P-IOHSMM-based behavioral models are associated with each room, covering expected behaviors for luminosity, temperature and power consumption. An example for luminosity is given in Fig.8. This model can be read as follows: while a presence is detected in the room (movement sensor), its luminosity must be higher than 2.5[2] (state $x_2$). Otherwise it could be any value (state $x_1$). The model expects some settling times to get the luminosity stabilized ($T_{d_{1,2}}$, $T_{d_{2,1}}$). Although the models described here are relatively simple, the proposed approach does not limit their complexity. Actually, the only limit is relative to the amount of sensors available within the environment.

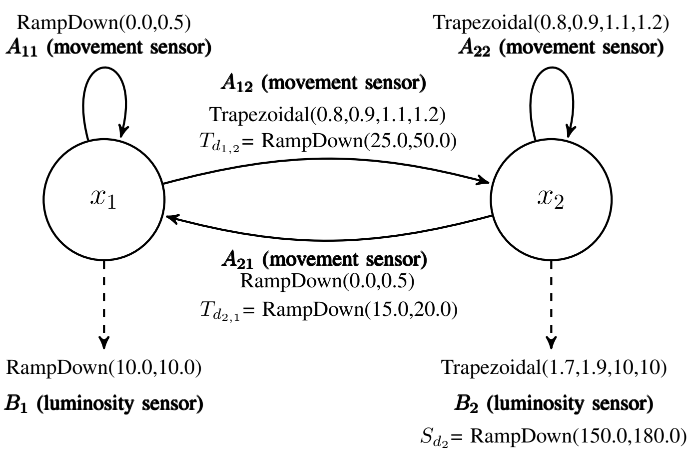


**Fig. 8:** Expected behavioral model for luminosity (assuming 100ms for the observation sampling rate).

Models are implemented as *nodes* and instantiated in a Node-Red[3] flow. Access to the 400 Home I/O points (sampled every second) is achieved from a dedicated node.

### C. Users' profiles

Thus, for each room is obtained, through the possibility measurement, the controller effectiveness for luminosity, temperature and power consumption. However, one is interested in obtaining a global score. To this end, the Choquet integral aggregation operator is leveraged (Section.IV) and four different user preference profiles are defined, modeled through capacity functions with three criteria (luminosity, temperature and power consumption). An example is given in Fig.9. In this example, the user is more interested in having the expected luminosity and power consumption ($\mu(13) = 0.9$), then the expected temperature and power consumption ($\mu(23) = 0.75$), etc.

Criteria $\boldsymbol{C}$={Luminosity*(1)*, Temperature*(2)*, Power consumption*(3)*}

$\boldsymbol{\mu(1)} = 0.4$ $\boldsymbol{\mu(12)} = 0.65$ $\boldsymbol{\mu(23)} = 0.75$ $\boldsymbol{\mu(C)} = 1$

$\boldsymbol{\mu(2)} = 0.2$ $\boldsymbol{\mu(13)} = 0.9$ $\boldsymbol{\mu(\emptyset)} = 0$

$\boldsymbol{\mu(3)} = 0.4$

**Fig. 9:** Preference profile for user 1.

### D. Results

Given the previously described environment, one can inject some disturbances over time. For instance, lowering the shutters, leaving windows open while the heating is on, etc. An example is given in Fig.10:

1. A presence is detected in the living-room. At that point, a state-transition occurs from state 1 to state 2 (see Fig.8). The model allows a certain amount of time for getting the required luminosity level ($T_{d_{1,2}}$= RampDown(25.0,50.0)). However, the luminosity level in the room ($\vec{y}_{1:\mathsf{K}}$) does not reach the required level after the time allocated and the degree of possibility decreases.
2. This case is similar to the previous one except that the state in progress (stateInProgress in Algorithm 2) is an unknown state (the shutter is down, lowering down the luminosity level below 1.7, and a movement is detected. This situation leads to reach an unknown state). Thus, the state-transition temporal constraint is computed from to last valid state (state 1, lastValidState in Algorithm 2) and the most possible next state (state 2, nextState in Algorithm 2). Thus the state-transition timing constraint applied is $T_{d_{1,2}}$= RampDown(25.0,50.0).
3. Here, the expected luminosity level occurs within the expected time frame. Thus, the degree of possibility is re-evaluated to 1.0.
4. Finally, this case exhibits the fact that the state duration goes beyond the expected one ($S_{d_2}$= RampDown(150.0,180.0)). Thus the degree of possibility decreases accordingly.

An application of our approach might be a dashboard exhibiting the effectiveness of each monitored behavior along with inhabitants satisfaction computed from Choquet integral (Fig.11).

[2]Values are obtained from the simulation.
[3]https://nodered.org/

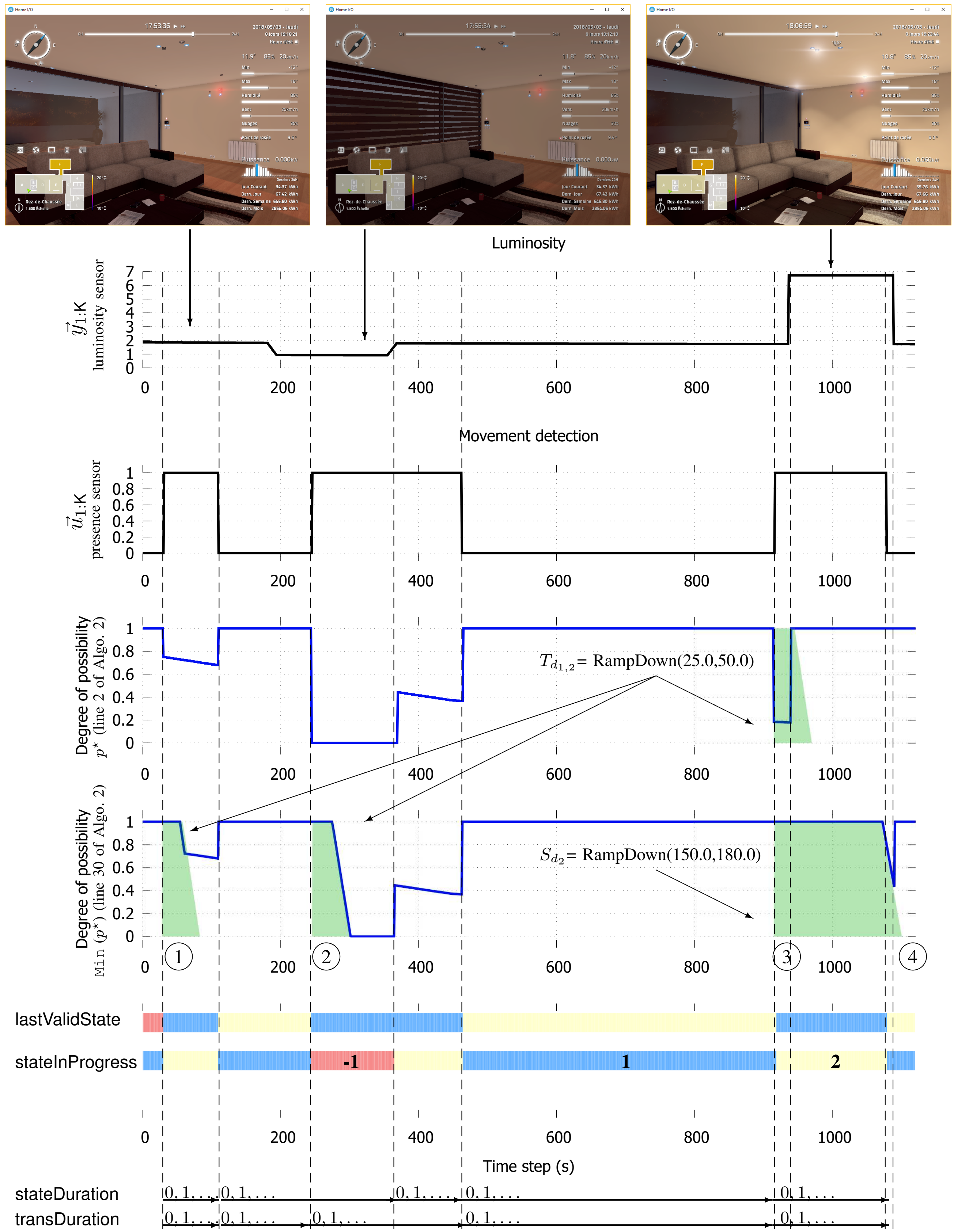


**Fig. 10:** Simulation results. The length of the input and output sequences is set to 5 and the observation sampling rate is set to 100 ms

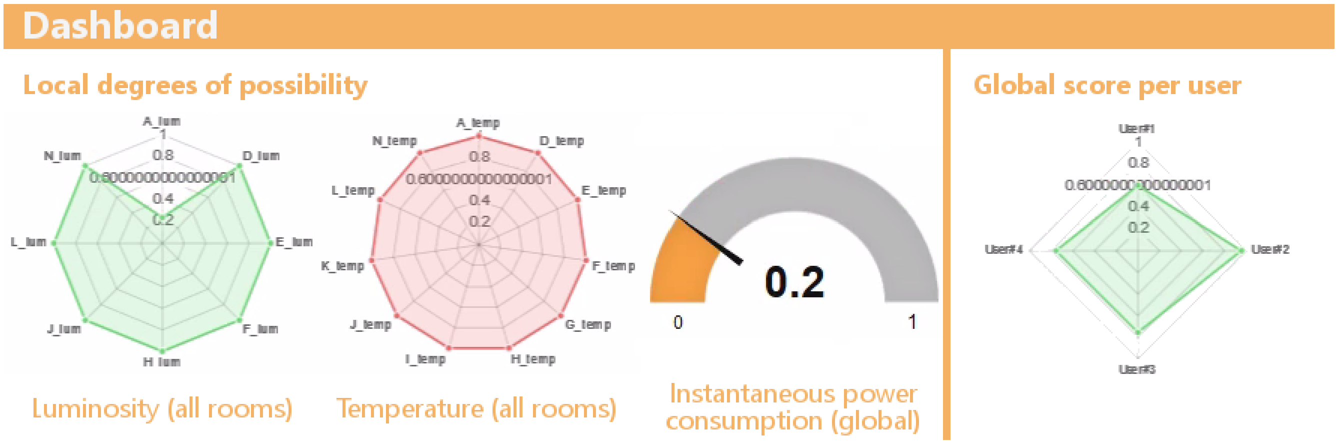


**Fig. 11:** An application of our approach might be a dashboard exhibiting the compliance of each monitored behavior (here, the overall instantaneous power consumption along with the luminosity and temperature behaviors which are observed for each room) along with inhabitants satisfaction obtained from Choquet integral operator.

## VI. Conclusion

Cyber-Physical Systems (CPS) are computing systems whose purposes are achieved from interactions with the physical world by means of transducers (sensors and actuators). These systems pose new challenges in terms of dependability, their behavior being affected by unanticipated physical processes over which they have no control and which may potentially hamper the achievement of their purposes (i.e. uncertainties). It is now recognized that designers of such systems can no longer lean, at design time, on comprehensive and reliable models for anticipating and removing faults that may arise once these systems are deployed. Instead, these systems have to be monitored *in vivo* and evaluated for effectiveness throughout their life cycle.

In this context, a new modeling framework, denoted as *Possibilistic Input/Output Hidden semi-Markov Model* (P-IOHSMM), has been introduced for describing the evolution of the effects to be produced by the CPS. Moreover, an algorithm for computing the likelihood of an observation sequence, here given as a degree of possibility, has been defined, taking into account temporal constraints. The degree of possibility provides us with a direct insight into the system effectiveness w.r.t. the effects it is supposed to produce over time. This modeling framework allows to cover a broad range of systems and applications including those aiming at controlling processes with intrinsic inertia (e.g. thermal process).

The case where multiple systems and applications are simultaneously assessed within a given environment has been investigated. So as to obtain a global effectiveness score, the Choquet integral aggregation operator has been leveraged. This operator allows to account for users' preferences, enriching the effectiveness score with a notion of satisfaction.

Finally the proposed approach has been validated through a use-case in the smart-home domain. The results demonstrate the soundness and efficiency of the proposed approach for estimating the CPS effectiveness at runtime.

In the future, we plan to facilitate the description of the effects to be produced by the CPS. Indeed, developing such model might be a complex task for non expert people. To this end, we will investigate end-user programming approaches allowing to derive a model from simple behavioral rules.

## VII. Acknowledgment

This work was supported by GFI Informatique, Innovation group, Saint-Ouen, France.

## References


[1] T. Bures, D. Weyns, C. Berger, S. Biffl, M. Daun, T. Gabor, D. Garlan, I. Gerostathopoulos, C. Julien, F. Krikava, R. Mordinyi, and N. Pronios, "Software engineering for smart cyber-physical systems – towards a research agenda: Report on the first international workshop on software engineering for smart cps," *SIGSOFT Softw. Eng. Notes*, vol. 40, no. 6, pp. 28–32, Nov. 2015. [Online]. Available: http://doi.acm.org/10.1145/2830719.2830736

[2] "Relyonit, dependability for the internet of things (european project id 317826)," 2012-2015, http://www.relyonit.eu/ [Online; accessed 5-May-2017].

[3] "Testing cyber-physical systems under uncertainty: Systematic, extensible, and configurable model-based and search-based testing methodologies (european project id 645463)," 2015-2017, http://www.u-test.eu/ [Online; accessed 5-May-2017].

[4] "Graz university of technology, dependable internet of things in adverse environments," 2016-2021, https://www.spsc.tugraz.at/research/programs/dependable-things [Online; accessed 5-May-2017].

[5] D. Garlan, "Software engineering in an uncertain world," in *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, ser. FoSER '10. New York, NY, USA: ACM, 2010, pp. 125–128. [Online]. Available: http://doi.acm.org/10.1145/1882362.1882389

[6] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 100, no. Supplement 1, pp. 9 – 34, 1999.

[7] Z. Ma, F. Zhang, L. Yan, and J. Cheng, *Fuzzy Knowledge Management for the Semantic Web*. Springer Publishing Company, Incorporated, 2013.

[8] D. Dubois, H. T. Nguyen, and H. Prade, *Possibility Theory, Probability and Fuzzy Sets Misunderstandings, Bridges and Gaps*, D. Dubois and H. Prade, Eds. Boston, MA: Springer US, 2000.

[9] J.-P. Aubin, A. Bayen, and P. Saint-Pierre, *Viability Theory: New Directions*. Cham, Switzerland: Springer, 2011. [Online]. Available: https://hal.inria.fr/inria-00636570

[10] D. Tran and M. Wagner, "Fuzzy hidden markov models for speech and speaker recognition," in *18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.99TH8397)*, 1999, pp. 426–430.
[11] Y. Li, Y. Ge, X. Jing, and Z. Bo, "A new intrusion detection method based on fuzzy hmm," in *2008 3rd IEEE Conference on Industrial Electronics and Applications*, June 2008, pp. 36–39.
[12] R. Sujatha and T. Rajalaxmi, "Hierarchical fuzzy hidden markov chain for web applications," *International Journal of Information Technology; Decision Making*, vol. 15, no. 01, pp. 83–118, 2016.
[13] A. M. Fraser, *Hidden Markov Models and Dynamical Systems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.
[14] T. Al-ani, "Hidden markov models in dynamic system modelling and diagnosis," *Hidden Markov models, theory and applications*, pp. 27–50, 2011.
[15] G. Rocher, J. Y. Tigli, and S. Lavirotte, "Probabilistic models towards controlling smart-*; environments," *IEEE Access*, vol. 5, pp. 12 338–12 352, 2017.
[16] Y. Bengio and P. Frasconi, "Input-output hmms for sequence processing," *IEEE Transactions on Neural Networks*, vol. 7, no. 5, pp. 1231–1249, 1996.
[17] G. Bailador and G. Trivio, "Pattern recognition using temporal fuzzy automata," *Fuzzy Sets and Systems*, vol. 161, no. 1, pp. 37 – 55, 2010, special section: New Trends on Pattern Recognition with Fuzzy Models. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165011409003820
[18] M. Grabisch and C. Labreuche, "A decade of application of the choquet and sugeno integrals in multi-criteria decision aid," *Annals of Operations Research*, vol. 175, no. 1, pp. 247–286, Mar 2010. [Online]. Available: https://doi.org/10.1007/s10479-009-0655-8
[19] C. Domshlak, E. Hllermeier, S. Kaci, and H. Prade, "Preferences in ai: An overview," *Artificial Intelligence*, vol. 175, no. 7, pp. 1037 – 1052, 2011, representing, Processing, and Learning Preferences: Theoretical and Practical Challenges. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000437021100049X
[20] G. Choquet, "Theory of capacities," in *Annales de l'institut Fourier*, vol. 5, 1954, pp. 131–295.
[21] M. Detyniecki, "Fundamentals on aggregation operators," *This manuscript is based on Detyniecki?s doctoral thesis and can be downloaded from*, 2001.
[22] A. R. Krishnan, M. M. Kasim, and E. M. N. E. A. Bakar, "A short survey on the usage of choquet integral and its associated fuzzy measure in multiple attribute analysis," *Procedia Computer Science*, vol. 59, pp. 427 – 434, 2015, international Conference on Computer Science and Computational Intelligence (ICCSCI 2015). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187705091502089X
[23] S. Angilella, S. Greco, F. Lamantia, and B. Matarazzo, "Assessing non-additive utility for multicriteria decision aid," *European Journal of Operational Research*, vol. 158, no. 3, pp. 734 – 744, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221703003886
[24] Realgames. Home I/O Smart Home Simulation. [Online]. Available: https://realgames.co/home-io/
[25] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: Programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1592761.1592779

## Appendix A
## Detailed algorithms

**Algorithm 1:** Computation of the most possible next state

```
Function NextState(Θ, u⃗, stateFrom): int
    nextState ← -1
    weight ← 0
    for i ← 1 to N do
        if Θ.A_(stateFrom,i)(u⃗) > weight then
            weight ← Θ.A_(stateFrom,i)(u⃗)
            nextState ← i
    return nextState
```

**Algorithm 2:** Evaluation of the Degree of possibility

**Member:** stateInProgress
**Member:** lastValidState, last state whith value $\geq 0$
**Member:** synchronized, indicates whether the algorithm is synchronized with the sequences
**Member:** transDuration, total elapsed time since the last valid state transition
**Member:** stateDuration, total elapsed time since entry in the stateInProgress

**Input:** $\vec{u}_{1:\mathsf{K}}$, the input sequence
**Input:** $\vec{y}_{1:\mathsf{K}}$, the output sequence
**Input:** $\Theta$, the P-IOHSMM model

**Output:** the degree of possibility of the last sequences

**Function** `Evaluate`($\Theta$, $\vec{u}_{1:\mathsf{K}}$, $\vec{y}_{1:\mathsf{K}}$) **: double**

$$\begin{pmatrix} p^\star \\ \mathcal{X}^\star \end{pmatrix} \leftarrow \texttt{Viterbi'}(\Theta, \vec{u}_{1:\mathsf{K}}, \vec{y}_{1:\mathsf{K}})$$

- **if** stateInProgress *is not initialized* **then**
  - stateInProgress = $\mathcal{X}^\star_{(1)}$
- **for** $k \leftarrow 1$ **to** K **do**
  - ***Synchronization***
  - **if** $\mathcal{X}^\star_{(k)}$ != stateInProgress **then**
    - **if** stateInProgress $> 0$ **then**
      - lastValidState = stateInProgress
      - transDuration = 0
    - stateInProgress = $\mathcal{X}^\star_{(k)}$
    - **if** lastValidState $> 0$ **then**
      - synchronized = true
      - **if** stateInProgress $> 0$ *and* stateInProgress != lastValidState **then**
        - stateDuration = 0
  - **if** synchronized **then**
    - transDuration ++
    - stateDuration ++
    - ***State-transition duration constraint***
    - **if** stateInProgress $< 0$ **then**
      - nextState $\leftarrow$ `NextState`($\Theta$,$\vec{u}_{(k)}$,lastValidState)
      - **if** nextState $< 0$ **then** ${p^\star}_{(k)} \leftarrow 0.0$
      - **else if** $\Theta.T_{d_{(\text{lastValidState},\text{nextState})}}$ **then**
        - ${p^\star}_{(k)} \leftarrow$ `Max`(${p^\star}_{(k)}$, $\Theta.T_{d_{(\text{lastValidState},\text{nextState})}}$(transDuration))
    - **else if** $\Theta.T_{d_{(\text{lastValidState},\text{stateInProgress})}}$ **then**
      - ${p^\star}_{(k)} \leftarrow$ `Max`(${p^\star}_{(k)}$, $\Theta.T_{d_{(\text{lastValidState},\text{stateInProgress})}}$(transDuration))
    - ***State duration constraint***
    - **if** stateInProgress $> 0$ **then**
      - **if** $\Theta.\vec{S}_{d_{(\text{stateInProgress})}}$ **then**
        - ${p^\star}_{(k)} \leftarrow$ `Min`(${p^\star}_{(k)}$, $\Theta.\vec{S}_{d_{(\text{stateInProgress})}}$(stateDuration))
- **return** `Min`($p^\star$)

**Algorithm 3:** Computation of $\mathcal{P}^\star$, $\mathcal{X}^\star$ and $x^?$

**Member:** $\mathcal{X}^\star$, contains the estimated state sequence
**Member:** $p^\star$, contains degrees of possibility computed for each $k$, $1 \leq k \leq \mathsf{K}$
**Member:** lastInputVec, is the last vector, i.e. $\vec{u}_{(K)}$, of the previous input sequence

**Function** `Viterbi'`($\Theta$, $\vec{u}_{1:\mathsf{K}}$, $\vec{y}_{1:\mathsf{K}}$)

- from $\leftarrow 1$
- ***Initialization***
- **if** *not* synchronized **then**
  - $\alpha_{(1)}(i) \leftarrow \min\left(\Theta.\pi_i, \Theta.B_i(\vec{y}_{(1)})\right), 1 \leq i \leq N$
  - $\varphi_{(1)}(i) \leftarrow 0$
  - from $\leftarrow 2$
- ***Recursion***
- **for** $k \leftarrow$ from **to** K **do**
  - $x^?_{(k)} \leftarrow$ true
  - **for** $j \leftarrow 1$ **to** $N$ **do**
    - possibility $\leftarrow -\infty$
    - **for** $i \leftarrow 1$ **to** $N$ **do**
      - **if** synchronized *and k = 1* **then** input $\leftarrow$lastInputVec
      - **else** input $\leftarrow \vec{u}(k-1)$
      - **if** *k=1* **then** weight $\leftarrow \Theta.A_{(i,j)}$(input)
      - **else** weight $\leftarrow \min\left[\Theta.A_{(i,j)}(\text{input}), \alpha_{(k-1)}(i)\right]$
      - **if** weight $>$ possibility **then**
        - mostLikelyState $\leftarrow i$
        - possibility $\leftarrow$ weight
    - $\alpha_{(k)}(j) \leftarrow \min\left[\text{possibility}, \Theta.B_{(j)}(\vec{y}_{(k)})\right]$
    - $\varphi_{(k)}(j) \leftarrow$ mostLikelyState
    - **if** $\alpha_{(k)}(j) > 0$ **then** $x^?_{(k)} \leftarrow$ false
  - **if** $x^?_{(k)}$ **then**
    - **for** $i \leftarrow 1$ **to** $N$ **do**
      - **if** *k=1* **then** $\alpha_{(k)}(i) = 1.0$
      - **else** $\alpha_{(k)}(i) = \alpha_{(k-1)}(i)$
- ***Termination***
- possibility $\leftarrow -\infty$
- **for** $i \leftarrow 1$ **to** $N$ **do**
  - **if** $\alpha_{(\mathsf{K})}(i) >$ possibility **then**
    - mostLikelyState $\leftarrow i$
    - **if** $x^?_{(K)}$ **then** possibility $\leftarrow 0.0$
    - **else** possibility $\leftarrow \alpha_{(K)}(i)$
    - ${p^\star}_{(K-1)} \leftarrow$ possibility
- lastInputVec $\leftarrow \vec{u}_{(K)}$
- $\mathcal{X}^\star_{(K)} \leftarrow$ mostLikelyState
- ***Trackback***
- **for** $k \leftarrow \mathsf{K} - 1$ **to** *1* **do**
  - $\mathcal{X}^\star_{(k)} \leftarrow \varphi_{(k+1)}[\mathcal{X}^\star_{(k+1)}]$
  - $p_{(k)} \leftarrow \alpha_{(k)}({\mathcal{X}^\star}_{(k)})$
- **for** $k \leftarrow 1$ **to** K **do**
  - **if** $x^?_{(K)}$ **then** $\mathcal{X}^\star_{(k)} \leftarrow -1$