Finding Behavioral Patterns of UAV Operators using Multichannel Hidden Markov Models

Víctor Rodríguez-Fernández
Universidad Autónoma
de Madrid (UAM)
28049, Madrid, Spain.
Email: victor.rodriguez@inv.uam.es

Antonio Gonzalez-Pardo
Universidad Autónoma
de Madrid (UAM)
28049, Madrid, Spain.
Email: antonio.gonzalez@uam.es

David Camacho
Universidad Autónoma
de Madrid (UAM)
28049, Madrid, Spain.
Email: david.camacho@uam.es

***Abstract*—In recent years Unmanned Aerial Vehicles (UAVs) have become a very popular topic in many different research fields and industrial applications. These technologies, and the related industries, are expected to grow dramatically by 2020. Although the systems designed to control UAVs are increasingly autonomous, the role of UAV operators is still a critical aspect that guarantee the mission success, specially when one single operator must supervise multiple UAVs. For this reason, much effort from different areas has been put into the study and analysis of the operator behavior. This work presents a new method to find and model behavioral patterns among UAV operators in a lightweight multi-UAV simulation environment. Our approach is based on MultiChannel (or Multivariate) Hidden Markov Models (MC-HMMs), which allow to gather in the same model parallel data sequences, such as the combination of operator interactions and mission events. The different steps for preprocessing data, creating, selecting and analyzing the model are described, and an experiment with inexperienced operators has been carried out to show how a descriptive model of behaviour can be generated using this modelling technique.**

I. Introduction

In the last decade, the development of systems based on Unmanned Aerial Vehicles (UAVs) has grown considerably, and it is expected to grow even more by 2020 [1]. This growth is produced due to the interest of both the industry and the research community. On the one hand, the different potential applications of this type of systems, such as infrastructure inspection, monitoring coastal zones, traffic and disaster management, agriculture or forestry have attracted the interest of the industry [2]. On the other hand, the research community is also interested in UAVs due to the challenging problems that must be faced from different fields such as Machine Learning, Automated Planning, Multiagent Systems, Simulation, Computer Vision, Robotics, Aeronautics, etc.

The role of a UAV operator is a critical aspect in this type of systems, due to the high costs involved in any real mission. In this sense, UAV operators are commonly trained using interactive environments, such as simulators, where they are asked to face different situations and alerts to get used to them, and thus, to be able to solve the situation successfully in a real scenario.

Monitoring the training programs provide relevant information regarding operators that can be used to understand their underlying cognitive process. This information can be extracted from both the operator interactions and the simulation events, and it allows to exploit the different behavioral patterns among UAV operators in a specific type of mission. These patterns can be later used in a wide range of applications, such as building behavioral models for different type of operators, for creating adaptive experiences in the learning process according to the operator's evolution, or for predicting future interactions and detecting abnormal deviations from the expected behavior.

The analysis of human behavioral patterns has been studied in different research fields and applied in a wide variety of applications. As an example, applied psychologists traditionally use behavioral models to understand the theoretical aspects of human decision making [3]. Another application of the computational models of human cognitive processes is to predict the consequences of high workload an time pressure [4]. Recent approaches for modeling user interactions are exclusively data-driven, and rely on pattern recognition techniques to predict future behaviors from user interface events. Some of the most popular modeling techniques in this field are Tree-based models [5], Bayesian Networks [6], Clustering [7] and Markov-based models [8].

This paper analyzes the applicability of Hidden Markov Models (HMMs) to identify the different cognitive states defining the behavior of UAV operators. Unlike other previous works in this field, in which these models were built based on the operator interactions as the only input variable [9], [10], [11], the contribution of this work is related to the usage of parallel sources of information: the interactions performed by the operators in the simulation environment, and the information that describes the state of the simulation. Since a classical HMM does not allow the modelling of parallel sequences of data, we will study the applicability of Multichannel Hidden Markov Models (MC-HMMs) (or Multivariate) to define the cognitive states.

To test the proposed MC-HMM, we make use of a simple multi-UAV simulation environment called Drone Watch And Rescue (DWR) [1][1] that has been used in previous works to build simple behavior models [9]. In this simulation environment the user must supervise the success of a surveillance

[1]Drone Watch And Rescue: http://savier.ii.uam.es:8888

mission performed by a group of UAVs, while avoiding the possible incidents that may occur during the course of the mission [12].

The rest of the paper is structured as follows: next section introduces the theoretical backgrounds on HMMs, with an emphasis on its extension to multichannel sequences. Then, in Section III a brief overview of DWR, the simulation environment used as a case study is given. Section IV describes the whole process for building this type of models based on data logs extracted from DWR, and in Section V an experiment is carried out by applying that process with inexperienced operators. Finally, Section VI summarizes the results obtained and proposes some future research lines.

II. Backgrounds on Hidden Markov Models (HMM)

HMMs are stochastic models mainly used for modeling and predicting sequences of symbols, and time series in general. They are characterized by a set of N discrete (*hidden*) states $S = \{S_1, \dots, S_N\}$, which can be interpreted as phases in a cognitive process that produce typical behaviors [13]. Although there are variants of these models in which time is considered continuous (the so-called Continuous-Time HMMs [14]), traditional HMMs are based on Discrete-Time Markov Chains (DTMCs), where the input time series are divided into equidistant *time steps*. The term *Markov* in a DTMC pertains to the time-dependence between the consecutive states S_t, which follows a Markov process. This means that the current state S_t only depends on the previous state S_{t-1} and not on earlier states, i.e:

$$P(s_{t+1} = S_j | s_t = S_i, \dots, s_1 = k) = P(s_{t+1} = S_j | s_t = S_i) \quad 1 \le i, k \le N$$

The transition probabilities between the states of this chain are denoted by a square matrix A (called *transition matrix*), with entries:

$$a_{ij}(t) := P(s_{t+1} = S_j | s_t = S_i), \quad 1 \le i, j \le N \tag{1}$$

This is a stochastic process, so we have that $\sum_{j=1}^{N} a_{ij} = 0$ for all $1 \le i \le N$. As in any Markov chain, we need to specify the set of initial state probabilities, Π, defined as:

$$\Pi_i := P(s_1 = S_i), \quad 1 \le i \le N \tag{2}$$

On the other hand, the term *hidden* in a HMM indicates that the underlying states S_t cannot be observed directly during the process, but what we see is the *emission* of that state. Although the observations that a HMM emit can be both continuous and discrete, this work has been focused on human interactions and simulation states, thus only discrete observations will be considered.

Let $O = \{O_1, \dots, O_V\}$ be the set of all the V possible *observation symbols* in our data domain (also called the *model dictionary*), the emission function, b, of a given state S_i is defined as a probability density function along the set O, i.e:

$$b_i(v) = P(o_t = O_v \mid s_t = S_i), \quad 1 \le v \le V, \quad 1 \le i \le N \tag{3}$$

In other words, $b_i(v)$ defines the probability of emitting O_v at any time step t in which the process is located in state S_i. Since the system can emit only one of the possible V observation symbols in each state at each time step, the function $b_i(v)$ is constrained to $\sum_{v=1}^{V} b_i(v) = 0$ for all $1 \le i \le N$. Gathering together the emission probabilities of each state into a $N \times V$ matrix we obtain the so-called *emission matrix* B.

Summarizing, any HMM λ can be defined as the tuple:

$$\lambda := \{S, V, A, B, \pi\} \tag{4}$$

Three main computational issues need to be addressed with HMMs:

- *Sequence Recognition*: It is related to how to compute the probability that a given T-length observation sequence $o = o_1 o_2 \dots o_T$ is produced by a model specified by a set of parameters λ (See Eq. 4). This allows us to decide whether some sequence belongs to some typical pattern or not. This probability, written as $P(o \mid \lambda)$ is called *sequence (log-)likelihood* and can be computed by the so-called forward-backward algorithm, which was introduced by Rabiner in [15].
- *Sequence Decoding*: It consists in determining, given a sequence of observation symbols o and a model λ, which corresponding sequence of hidden states $s = s_1 s_2 \dots s_L$ is most likely to produce it. This problem is addressed by the use of the popular *Viterbi* algorithm, also known as *decoding* [16].
- *Model Training*: In the majority of applications, the set of parameters λ of Eq. 4 cannot be inferred analytically but need to be estimated from recorded sample data. Although there are classical supervised learning techniques for HMMs, in many applications the hidden data is missing, so we no longer know which state to assign for each observation. The solution is to use *unsupervised learning* techniques, among which the *Baum-Welch* algorithm stands out for being the first to address the problem for classical HMMs [17]. In brief, it is a form of Expectation-Maximization (EM) which tries to maximize the likelihood of a set of observation sequences $o^1 \dots o^K$ to be produced by a model λ. Formally, this algorithm computes the optimal model $\hat{\lambda}$ as follows

$$\hat{\lambda} = argmax_{\lambda} \left(\sum_{K} \log P(o^K \mid \lambda) \right) \tag{5}$$

Convergence to a local optimum is proven in [15] with complexity $O(N^2 \cdot V)$ for a $N-$state HMM with a $V-$sized dictionary.

A. Model Selection

One important aspect to consider when fitting a HMM to a given dataset is that the number of hidden states, N, must be known in advanced, which is often unrealistic. To choose an optimal number of states without prior knowledge about the model topology, several statistic metrics are used to compare

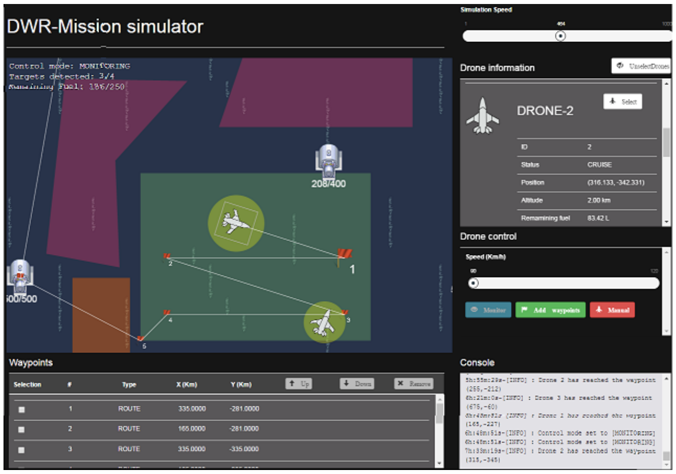

Fig. 1: Screenshot of a simulation in Drone Watch & Rescue.

and select models, of which Bayesian Information Criterion (BIC) is the best known [18]. This metric is defined as:

$$BIC(H) = -2(logLik(H)) + Plog(K),$$

where P is the number of parameters in the model, and K the number of observations used to train the model. The less the BIC scores, the better the model is considered. As it can be seen, BIC penalizes the likelihood of a model by a complexity factor proportional to number of parameters in the model and the number of training observations, so it gives advantages to simple and general models.

B. Multichannel Hidden Markov Models (MC-HMM)

A Multichannel Hidden Markov Model (MC-HMM), or Multivariate HMM, is a simple extension of a classical HMM, where the sequence data feeding the model is divided into C parallel sequences. The term "Multichannel" is adopted from the works of Helske et al. in [19], and makes reference to groups of categorical data sequences, rather than numerical or continuos time series, for which the term "Multivariate" is the most common [20]. Observations are now of the form $o_{tc}, t = 1, \ldots, T, c = 1, \ldots, C$, so that a complete observation sequence is $O = \{O^1, \ldots, O^C\}$. Unlike other complex HMM extensions as *Linked HMMs* or *Coupled HMMs* [21], where the observed states in different channels at a given time point t are interlaced with a special transition probability matrix, in the case of MC-HMMs the model has one transition matrix A, but several emission matrices $B_1, \ldots, B_C$, one for each channel. Sequence likelihood, decoding and model training issues are barely altered by this modification, as it can be read in [19].

III. DWR - A Lightweight Multi-UAV Simulation Environment

In this work we are interested in the utility of a simulator for research purposes, and its potential for low cost training, especially in terms of the data that can be retrieved during the training operations. For this reason, the simulation environment used as the basis for this work has been designed following the criteria of accessibility and usability. It has been named as Drone Watch And Rescue (DWR), and its complete description can be found in [12]. A screenshot of a simulation in DWR is shown in Figure 1.

DWR gamifies the concept of a multi-UAV mission, challenging the operator to capture all mission targets consuming the minimum amount of resources, while avoiding at the same time the possible incidents that may occur during a mission. To avoid these incidents, an operator in DWR can perform multiple interactions to alter both the UAVs in the mission and the waypoints comprising their mission plan.

Regarding the incidents that may occur during the execution of a simulation, three different types have been implemented in DWR:

- *Danger Area*: Due to a heavy storm or any other weather threat, a new danger area appears somewhere in the map. When a UAV overflies it, it will be automatically destroyed. To overcome this incident, the operator must change the flying path of the UAVs at risk of flying over these areas.
- *Payload Breakdown*: The sensors conforming the UAVs payload stop working. From this moment, the UAV is not able to detect any target. To overcome this incident, the operator must command the affected UAV to return to an airport, where it will be repaired.
- *Low Fuel*: When the fuel level of a UAV is lower than a predefined threshold, an alert will be displayed notifying about the incident. The operator must command the affected UAV to fly to the closest *refueling station* in the mission map.

Besides, it is remarkable how DWR saves data from a simulation. Whenever an event occurs, DWR stores the simulation status in that moment, as a *Simulation Snapshot*. This snapshot contains information related to the current status of every element taking part in the simulation. Storing the data in this way allows to reproduce the entire simulation a posteriori, which is helpful for the analysis process.

IV. Building and Analyzing MultiChannel Hidden Markov Models in DWR

In this section, we will detail the process carried out in this work to build and analyze a MC-HMM exploiting the patterns found among UAV operators during a training session in the simulator DWR. A graphical overview of the whole process is shown in Figure 2. As it can be seen, not all the steps are automated, but there are some which require the human intervention of an expert in the domain, probably an instructor specialized in operations with UAVs. This is because here we are not interested in the predictive aspects of the HMM. Instead, we want to find a descriptive and interpretable model so that the training instructor can analyze it and extract conclusions, even if he/she knows nothing about the underlying modelling process.

After getting started in the DWR environment, a set of training operators (or trainees) is told to complete a specific training operation, designed by the instructor in the system. It

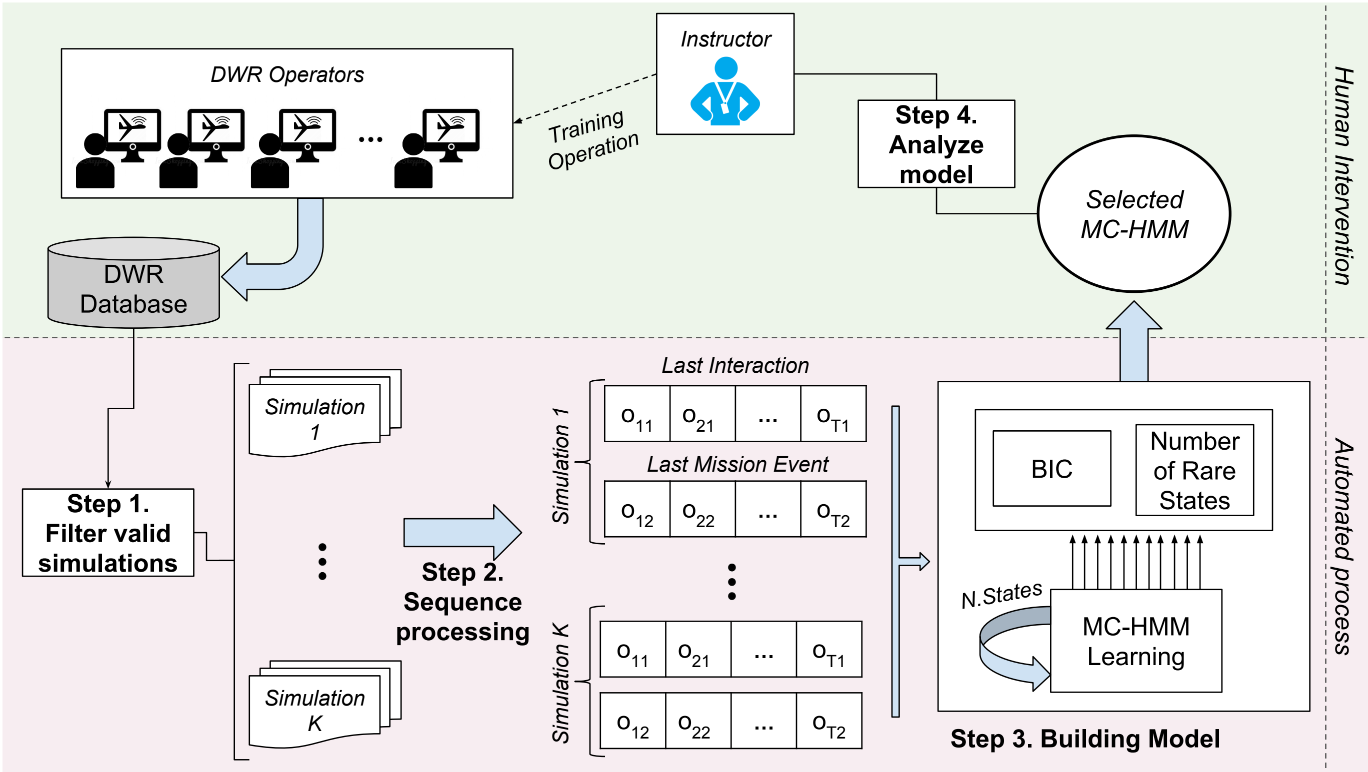

Fig. 2: General scheme of the analysis process developed in this work.

is important that all operators perform in the same operation environment, so that the instructor can better analyze the results. All the interactions and events happened during a simulation in DWR are stored into a database.

The first step (See Figure 2, step 1) consists in filtering the stored data, removing those simulations considered as "useless" for modelling. Several filters are applied to select only the most relevant simulations, including a *duration filter*, which removes the shortest simulations, an *interaction filter*, which takes only those simulations where the user has been active, and an *incident filter*, which considers only simulations in which the operator was threatened with several mission incidents. After this step, only a number of K simulation logs are taken as input for the rest of the process.

The next step (See Figure 2, step 2) is to process the filtered simulation logs, transforming them into multi-channel sequences of categorical data, suitable for MC-HMM modelling. A graphical overview of the process is detailed in Figure 3. As it was mentioned in Section III, DWR stores a simulation as a list of asynchronous and timestamped snapshots. Despite HMMs model time information, they do not consider time as a continuous variable, but a discrete one. For that reason, every log must be discretized into *equidistant time steps*, where each one contains, at the most, one log entry. Then, the discretized log is divided into two aligned categorical data sequences:

- *Last Interaction*: It contains, for every time step, a symbol identifying the last command performed by the operator.
- *Last Mission Event*: It contains, for every time step, the last event happened in the mission course.

By having this multi-channel sequence representation, not only we are able to analyze the operator response, but also we can relate the concepts of "What happened" to "What was done". The different interactions and events taking part of each

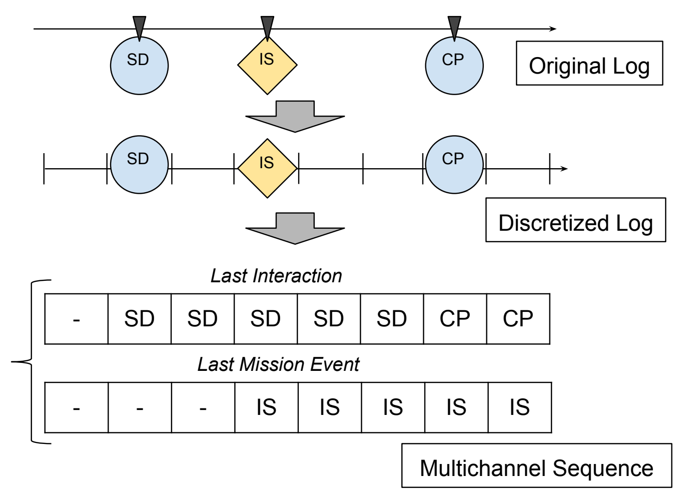

Fig. 3: Details of the sequence processing step shown in Figure 2. The original log extracted from DWR is discretized and divided into two aligned sequences, one containing the last operator interaction (circles) and the other indicating the last mission event occured (diamonds).

	Symbol	Name	Description
Interactions	SD	Select Drone	Allows the operator to focus, monitor and send commands to a specific UAV
	ST	Set Simulation Speed	Increase or decrease the simulation speed. Usually, UAV missions last many hours, thus sometimes it is desirable to accelerate the process to allow a fast simulation-based training
	CP	Change UAV Path	Add/edit/remove waypoints of any UAV.
	CM	Change Control Mode	Control modes in DWR manage how a user can change a UAV path (*Monitor mode, Add waypoints, Manual mode*).
	IS	Incident Started	A new incident starts
Mission Events	IE	Incident Ended	An incident ends (either because the operator opposed it or because it had an scheduled end time).
	TD	Target Detected	A UAV detects a target.
	DD	Drone Destroyed	A UAV is destroyed
	AS	Action Started	A UAV starts an action (Refueling, Landing...) or a mission task (Surveillance)

TABLE I: Summary of the interactions and events from the simulator DWR, used for the analysis carried out in this work.

channel are detailed in Table I. It is remarkable that, although the simulator DWR extracts more events and interactions , we have selected those considered as most "relevant" for the course of a mission.

The next and most important step is the model creation (See Figure 2, step 3). In some applications, the number of states of the model and other type of initial parameters, neccesary for executing the model training algorithm, are known in advanced [22]. However, in this work we do not rely on any prior-knowledge to build the MC-HMMs, thus we must train models along the entire search space of the unknown parameters and select which of them fits best to the data.

In order to perform the model selection, a pool of MC-HMM candidates, with different number of states are trained using the popular *Baum-Welch* algorithm. To avoid falling into a local optimum, we execute the algorithm multiple times with different initialization parameters (randomly chosen), and select the result with the highest likelihood to the data. Then, those candidates are scored on two different measures:

1) *Bayesian Information Criterion (BIC)*: It tries to give a balance between the likelihood of a model and its complexity. Generally, the less the BIC scores, the better the model performs (See section II).
2) *Number of Rare States* (NRS): In addition to BIC, we add a second rating that counts the number of "rare" hidden states in a HMM, to ensure the simplicity and interpretability of the model. Given an input T-length data sequence o and a N-state HMM (or MC-HMM) λ, we can compute, via the *Viterbi algorithm*, the most probable hidden state path $s^o = s_1 s_2 \dots s_T$ for that sequence, where $s_t \in \{S_1, S_2, \dots S_N\}$ is the value of the state at time t. Then, we compute the *State Visit Frequency* (SVF) for every state in sequence o as follows:

$$\mathit{SVF}(S_j, o) = \frac{|\{t \in T \mid s_t^o = S_j\}|}{T}, \quad j \in 1 \dots N$$

By averaging the values of $\mathit{SVF}(S_j, o)$ for every sequence in the whole dataset o^K, we obtain a measure, namely $\overline{\mathit{SVF}}(S_j, o^K)$, that indicates the prevalence or rareness of a hidden state in the overall dataset. Finally, given a *State Visit Frequency Threshold* (SVFT), the Number of Rare States for a N-state HMM λ is computed as a simple filter for low-visited states:

$$\mathit{NRS}(\lambda) = |\left\{j \in 1 \dots N \mid \overline{\mathit{SVF}}(S_j, o^K) < \mathit{SVFT}\right\}|$$

The candidate model which minimizes both selection measures (BIC and NRS) will be selected as the best model to fit the dataset. Finally, the selected MC-HMM will be presented to the training instructor (See Figure 2, step 4), who is responsible for analyzing it, giving sense to each of the hidden states and interpreting the most representative patterns found in the underlying Markov chain.

V. Experimentation

In this section, the modelling process detailed above is applied to the data obtained after training a set of inexperienced operators in a simple mission in DWR. Below is a description of the retrieved dataset and the proper experimental setup, along with the results for both the selection and analysis of the resulting MC-HMM.

A. Dataset

In this experiment, the simulation environment (DWR) was tested by Computer Engineering students of the Autonomous University of Madrid (AUM), all of them inexperienced in this type of systems. Every user completed a brief tutorial before using the simulator, explaining the mission objectives and the basic controls. Recall that the main goal of a surveillance mission in DWR is to detect all the mission targets and make the available UAVs return to the base airports safely, while avoiding the possible incidents that may arise during the mission course.

The mission prepared for this experiment featured a total of 3 UAVs performing 4 Surveillance tasks in 2 different areas, in order to detect 4 mobile targets. The map also presented 4 No-Flight-Zones and 4 Refueling Stations. During the simulation, 4 scheduled incidents were triggered, affecting both the UAVs and the environment. Every UAV started the mission with a pre-loaded mission plan (route), so, a priori, the operator is only expected to supervise that route and possibly perform minor changes in it. For more information about the mission elements involved in the simulation see [12].

The dataset resulted from this experiment comprises 108 distinct simulations, executed by a total of 38 users.

B. Experimental Setup

The choice of the parameters involved in the process described in the previous section is very important for the success of the analysis. Table II gathers the chosen values for all the necessary parameters for the whole experiment. Below are some remarks about this parameter tuning:

- The Time Step Resolution (TSR) is set to a $1000\ ms$, in a way that none of the events/interactions in a simulation log overlap with each other.
- Since we want the MC-HMM to be interpretable, the maximum number of possible states is set to 10.
- A hidden state is considered "rare" when it is visited less than 5% in average, i.e, we set the State Visit Frequency Threshold to 0.05.

The whole experimentation has been implemented in the R Statistical Environment [2], making use of the package *seqHMM* [19], designed to fit HMMs to sequences of categorical data.

C. Experimental Results

After applying the simulation filters to clean the useless simulations (See Section II), only 55 of them were considered as useful for this experiment, hence $K = 55$. All those simulations are introduced into the sequence processing step, as detailed in the previous section, resulting in 55 multichannel sequences describing the simulation interactions and events. The duration of these sequences goes from 57 time steps (minimum) to 1000 (maximum), achieving the average in 261.8 time steps. Note that in this experiment, 1 time step is equal

Context	**Parameter**	**Value**
General	Time Step Resolution	1000 ms
Simulation Filters	Min. number of mission incidents	2
	Min. number of interactions	10
	Min. mission duration	20 s
HMM	Possible number of states	2..10
	Number of restarts of the Baum-Welch algorithm	50
Baum-Welch algorithm	Max. iterations	1000
	Relative tolerance for convergence	$1e{-10}$
Model Selection	State Visit Frequency Threshold	0.05

TABLE II: Parameter tuning for all the variables involved in the experimentation carried out in this work.

Nstates	*LogLik*	*BIC*	*Number of Rare States (NRS)*
2	-26326.87	52814.3	0
3	-23063.94	46401.76	1
4	-19846.67	40099.45	1
5	-17545.55	35648.31	2
6	**-15621.96**	**31971.14**	**0**
7	-15207.23	31330.55	4
8	-13679.73	28483.32	6
9	-13235.95	27822.44	6
10	-12160.1	25916.29	7

TABLE III: Results for the model selection. The bolded row indicates that the 6-state model is chosen, since it obtains great values for both the BIC and NRS measures.

to 1 second, so it is clear that the duration of the test mission is short in comparison to a real mission involving UAVs. This is something characteristic from gamified environments as DWR.

1) Model Selection: The results for the model selection process are shown in Table III. It is important to keep in mind that for every possible value of N (the number of hidden states), multiple models are trained, and from them, we choose as candidate the one maximizing the (log)-likelihood to the training data. It can be seen that, from state 2 to state 10, the BIC measure is always decreasing. However, the rate of decline is lower from models with 6 states, which can be seen as an "elbow" in the BIC decreasing. Furthermore, the model with 6 states also minimizes the Number of Rare States (NRS), which is a sign that, by having 6 states, we achieve a simple and fair description of the patterns hidden throughout the sequences. For these reasons, the 6-state MC-HMM is the one selected and it will be analyzed below.

2) Model Analysis: The last step of the experiment consists in analyzing the resulting model and describing the hidden patterns it contains, in the context of the simulation environment DWR. As it was shown in Figure 2, the main responsibility of this step lies in the instructor of the experiment, which must be an expert in the system in question. Since the development of DWR is part of our previous work [12] and the training session carried out for this experimentation was also our responsibility, here we are in the position to perform the model analysis.

A graphical presentation of the 6-state MC-HMM selected as the best one for this experiment is shown in Figure 4. To allow a better model analysis, the state emission probabilities are combined across channels, and drawn as a pie chart within each of the states (nodes). The first part of the analysis consists in examining those emissions, using them to describe and label the behavior (or *latent class*) hidden in each state:

- *State 1: Monitoring*. In this state, the prevalent interaction is "Select Drone" (SD), performed during the events "Action Started" (AS) and "Incident Ended" (IE). This behavior is characteristic of those parts of the mission course where the operator does not need to alter the UAV

[2]The R project for Statistical Computing: https://www.r-project.org/

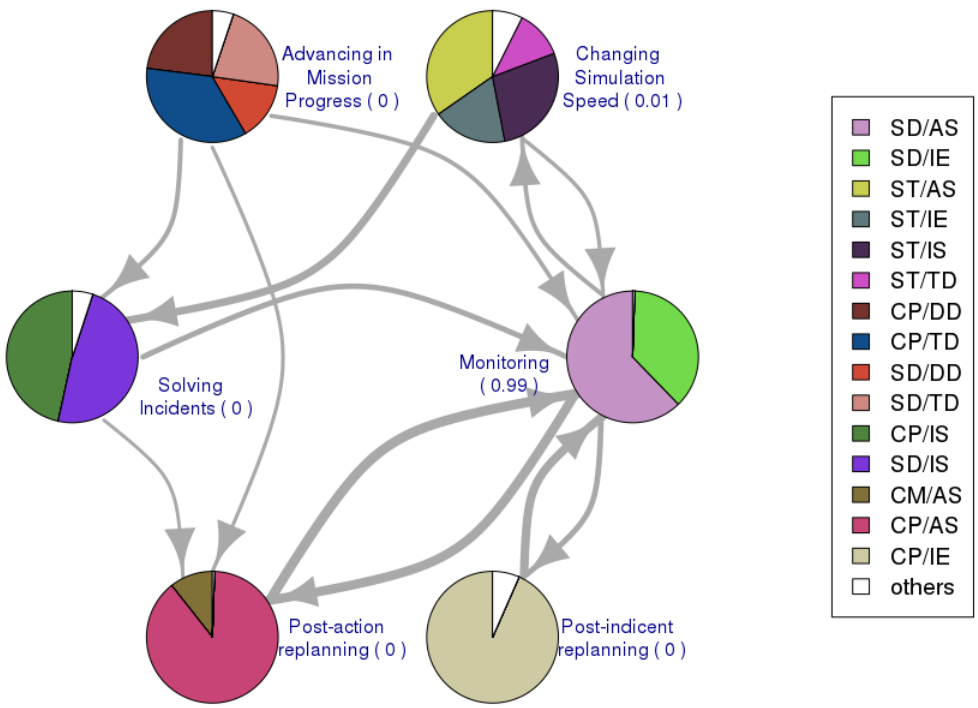

Fig. 4: The 6-state MC-HMM has been selected as the best one for the experiment carried out in this work. Low-probability transitions and self-transitions are omitted for legibility purposes. Slices in the pie charts represent the emission probabilities for a combination of the data channels (*Interaction/Event*). The edge width is proportional to the transition probability. State labels comprise the state name (assigned after the model analysis) along with the initial state probability (in brackets). Legend symbols make reference to the interactions and actions described in Table I.

path and simply selects and monitors the status and the trajectory of the different UAVs.

- *State 2: Changing Simulation Speed*. In this state, the common factor in all the probability slices is the prevalence of the interaction "Set Simulation Speed" (ST), regardless of the mission event. Operator is changing the simulation speed, whether to make it faster or slower.
- *State 3: Advancing in Mission Progress*. In this state, the main mission events are: "Target Detected" (DD) and "Drone Destroyed" (DD), both concerning to key points in the mission progress and the likelihood of missions success. The combination of "Select Drone" and "Change UAV Path" regarding to the operator interactions indicate that, after these events, operators usually adjust the plan of the UAVs to adapt to the new mission status.
- *State 4: Solving Incidents*: This state clearly represents the operator responses in the moments when an incident arises (prevalence of the event "Incident Started" (IS)). As it can be seen, the response consists, again, in a combination of selecting UAVs and changing paths, which is a typical pattern of path adjustment with multiple UAVs.
- *State 5: Post-action Replanning*: This state indicates that operators usually redefine the plan of a specific UAV (Interactions "Change Control Mode" (CM) and "Change UAV Path" (CP)) when it starts a new action (Event "Action Started" (AS)), probably a surveillance task throughout some area. This is logical, considering that in DWR the detailed path that a UAV follows during a surveillance task is not computed until the UAV starts the task, so that may lead operators to change it just after the task beginning.
- *State 6: Post-incident Replanning*: This state is likely to be visited during the latter parts of the mission. When the mission is coming to an end, and the last remaining incidents begin to disappear (Event "Incident Ended" (IE)), operators tend to hasten the pre-scheduled mission course and redefine the plans of the remaining UAVs (Prevalence of interaction "Change UAV Path (CP)"), making the return to the base airports.

Apart from the hidden states, the MC-HMM holds interesting patterns and issues within the underlying Markov chain:

- The initial probabilities are approximately equal to zero in all states except *Monitoring*, which indicates that, as it can be expected, operators always start the mission by exploring the status of every UAV in the mission, and checking their initial plans.
- The probability of going to the state *Advancing in Mission Progress* is very low (even not drawn in the plot), due to the number of times that the events of that state, TD and DD, happen, is low with respect to the rest of event that can happen throughout the mission.

- It is remarkable that the probability of going from state *Changing Simulation Speed* to state *Solving Incidents* is high, which indicates that these operators tend to increase the simulation speed until an incident appears, which is a clear sign of a restlessness behavior, characteristic of novice operators.

VI. Conclusions and Future Work

This work has presented a new way to find and model behavioral patterns among UAV operators in a simple multi-UAV simulation environment. It is based on Multichannel (or Multivariate) Hidden Markov Models, which allow to gather in the same model multiple data sequences, such as the combination of operator interactions and mission events. The different steps for preprocessing data, creating, selecting and analyzing the model have been detailed, and an experiment has been carried out using data from a set of inexperienced operators.

The resulting model for this experiment turns out to be fairly descriptive, and reveals several behavioral patterns, some of them representative of the inexperience of the operators tested, such as the way they control the simulation speed, or the general tendency they have to hasten and change the prescheduled mission plan, specially at the latter parts of a mission. In sum, by adding extra information in the model apart from the operator interactions, we achieve more robust and informative models than those from previous works in the field.

As future work, several issues will be extended and improved, including: 1) An extension of the alphabets feeding the MC-HMMs in order to add more precise information about interactions and events. 2) The use of covariates in the model creation to compare behavioral patterns with respect to specific operator features, such as the age or the previous experience with UAVs. 3) A formal comparison among single channel HMMs, multichannel HMMs, and other HMM extensions in terms of the quality of the behavioral patterns found. 4) The use of the resulting model as an online predictive tool to detect abnormal behaviors during a mission.

Acknowledgments

This work has been supported by the next research projects: Airbus Defence & Space (FUAM-076914 and FUAM-076915), EphemeCH (TIN2014-56494-C4-4-P) Spanish Ministry of Economy and Competitivity, CIBERDINE S2013/ICE-3095, both under the European Regional Development Fund FEDER, and RiskTrack (JUST-2015-JCOO-AG-723180). The authors would like to acknowledge the support obtained from Airbus Defence & Space, specially from Savier Open Innovation project members: José Insenser, Gemma Blasco, Juan Antonio Henríquez and César Castro.

References

[1] M. DeGarmo and G. M. Nelson, "Prospective unmanned aerial vehicle operations in the future national airspace system," in *AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum*, 2004, pp. 20–23.

[2] V. V. Klemas, "Coastal and environmental remote sensing from unmanned aerial vehicles: An overview," *Journal of Coastal Research*, vol. 31, no. 5, pp. 1260–1267, 2015.

[3] J.-C. Pomerol and S. Barba-Romero, *Multicriterion decision in management: principles and practice*. Springer Science & Business Media, 2012, vol. 25.

[4] W. D. Gray, B. E. John, and M. E. Atwood, "The precis of project ernestine or an overview of a validation of goms," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1992, pp. 307–312.

[5] Ş. Gündüz and M. T. Özsu, "A web page prediction model based on click-stream tree representation of user behavior," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 535–540.

[6] L. Hawarah, S. Ploix, and M. Jacomino, "User behavior prediction in energy consumption in housing using bayesian networks," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2010, pp. 372–379.

[7] V. Rodríguez-Fernández, H. D. Menéndez, and D. Camacho, "Automatic profile generation for uav operators using a simulation-based training environment," *Progress in Artificial Intelligence*, vol. 5, no. 1, pp. 37–46, 2016.

[8] M. A. Awad and I. Khalil, "Prediction of user's web-browsing behavior: Application of markov model," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1131–1142, 2012.

[9] V. Rodríguez-Fernández, A. Gonzalez-Pardo, and D. Camacho, "Modeling the behavior of unskilled users in a multi-uav simulation environment," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2015, pp. 441–448.

[10] ——, "A method for building predictive hsmms in interactive environments," in *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, Accepted.

[11] Y. Boussemart and M. L. Cummings, "Predictive models of human supervisory control behavioral patterns using hidden semi-markov models," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 7, pp. 1252–1262, 2011.

[12] V. Rodríguez-Fernández, H. D. Menéndez, and D. Camacho, "Design and development of a lightweight multi-uav simulator," in *Cybernetics (CYBCONF), 2015 IEEE 2nd International Conference on*. IEEE, 2015, pp. 255–260.

[13] I. Visser, "Seven things to remember about hidden markov models: A tutorial on markovian models for time series," *Journal of Mathematical Psychology*, vol. 55, no. 6, pp. 403–415, 2011.

[14] W. Wei, B. Wang, and D. Towsley, "Continuous-time hidden markov models for network performance evaluation," *Performance Evaluation*, vol. 49, no. 1, pp. 129–146, 2002.

[15] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[16] G. D. Forney Jr, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[17] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state markov chains," *The annals of mathematical statistics*, pp. 1554–1563, 1966.

[18] K. P. Burnham and D. R. Anderson, "Multimodel inference understanding aic and bic in model selection," *Sociological methods & research*, vol. 33, no. 2, pp. 261–304, 2004.

[19] S. Helske and J. Helske, "Mixture hidden markov models for sequence data: the seqhmm package in r," 2016.

[20] J. Chiang, Z. J. Wang, and M. J. McKeown, "A hidden markov, multivariate autoregressive (hmm-mar) network framework for analysis of surface emg (semg) data," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 4069–4081, 2008.

[21] V. Raghavan, G. Ver Steeg, A. Galstyan, and A. G. Tartakovsky, "Coupled hidden markov models for user activity in social networks," in *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[22] J. O'Connell, F. A. Tøgersen, N. C. Friggens, P. Løvendahl, and S. Højsgaard, "Combining cattle activity and progesterone measurements using hidden semi-markov models," *Journal of Agricultural, Biological, and Environmental Statistics*, vol. 16, no. 1, pp. 1–16, 2011.