Analyzing Temporal Performance Profiles of UAV Operators using Time Series Clustering

Víctor Rodríguez-Fernández[a,*], Héctor D. Menéndez[b], David Camacho[a]

[a] *Universidad Autónoma de Madrid (UAM) 28049, Madrid, Spain*
[b] *University College London (UCL), London, UK*

Abstract

The continuing growth in the use of Unmanned Aerial Vehicles (UAVs) is causing an important social step forward in the performance of many sensitive tasks, reducing both human and economical risks. The work of UAV operators is a key aspect to guarantee the success of this kind of tasks, and thus UAV operations are studied in many research fields, ranging from human factors to data analysis and machine learning. The present work aims to describe the behavior of operators over time using a profile-based model where the evolution of the operator performance during a mission is the main unit of measure. In order to compare how different operators act throughout a mission, we describe a methodology based of multivariate-time series clustering to define and analyze a set of representative temporal performance profiles. The proposed methodology is applied in a multi-UAV simulation environment with inexperienced operators, obtaining a fair description of the temporal behavioral patterns followed during the course of the simulation.

Keywords: UAVs, UAV Operators, Time Series Clustering, Performance measures, Simulation-Based Training

[*] Corresponding author.

Email addresses: `victor.rodriguez@inv.uam.es` (Víctor Rodríguez-Fernández), `h.menendez@ucl.ac.uk` (Héctor D. Menéndez), `david.camacho@uam.es` (David Camacho)

1. Introduction

Unmanned Aerial Vehicles (UAVs) have become a relevant area in the last decade. The main goal of this field is to replace human supervision in several sensitive tasks using UAVs in an accurate way. The automation of these tasks supposes an important step forward in several areas of our societies such as: agriculture, traffic, infrastructure inspection and forestry among others (Pereira et al., 2009).

In the current state of UAV research and development, there are some processes that can be almost totally automated with low risk, but others still require the role of the operator as a critical part of the entire system. A hard training of these operators is usually performed to guarantee that they have the appropriate attitudes to handle with this technology, specially in risky situations. The training process can also help to describe different features of the trainee, not only technical but also psychological aspects that might help to prevent dangerous circumstances.

This study focuses on UAV operators and takes information about how they evolve during a specific simulation, paying special attention to how their performance change during the process. With this information, we build a temporal performance profile of a simulation which will help to describe the decision abilities.

In previous works we were focused on describing a general profile of the operators, based on their behaviour during the whole simulation (Rodríguez-Fernández et al., 2015). Also, the temporal interaction patterns during a mission were modelled through the use of Hidden Markov Models in (Rodríguez-Fernández et al., 2015a). However, one of the most relevant aspects of the training process is the performance evolution during the simulation course. This work is focused on that attitude, creating temporal performance profiles for different simulations and then extracting and analyzing the most representative of all.

In order to achieve the purposes of this work, we combined clustering techniques with time series analysis (Liao, 2005), to define a set of representative simulation profiles, based on the evolution of a set performance measures that describe the attitude of the operator in specific moments of a simulation. To test the validity of the proposed methodology, an experiment with inexperienced operators is carried out, simulating a training mission in a lightweight multi-UAV simulation environment, developed as part of our previous work in the field (Rodriguez-Fernandez et al., 2015). Several experiments have been carried out to evaluate the quality of the results of the methodology and to compare those results against other clustering approaches. Furthermore, a qualitative analysis of the results have been made in the context of the experimental simulation environment.

In sum, this paper presents the following contributions:

- A new multi-variate time series clustering methodology is defined in the context of performance analysis for UAV operations. The proposed methodology is divided into two steps: the first focused on finding patterns in each dimension of the multivariate time series and the second focused on generating a multi-variate distance using the the patterns found in the previous step.

- The proposed methodology is scalable to the use of different time series dissimilarity metrics, different clustering methods and different number of clusters.

- A collective human judgement-based evaluation process is carried out to create ground truth information with which we are able to evaluate and compare the results of the proposed methodology.

- A quantitative and qualitative interpretation is given for the results obtained in a lightweight multi-UAV simulation environment.

The rest of the paper is structured as follows: next section presents the Related Work, after, Section 3 describes the proposed methodology, emphasizing on its division into two steps. Then, Section 4 provides a description of how to apply the proposed methodology to a specific simulation environment, detailing the environment itself, the defined performance measures comprising a simulation profile, and the evaluation criteria used to judge whether the results are right in an objective way. In Section 5 we carry out some experiments to evaluate and compare quantitatively the quality of the proposed methodology, and afterwards, Section 6 makes a qualitative analysis of the results obtained. Finally, Section 7 presents the conclusions and future work.

2. Related Work

This section aims to provide a general overview around the two main fields of this work: UAV's research and machine learning algorithms. We start by introducing the current problems that have been frequently studied in UAVs and after that, we describe clustering models that can be found in the literature.

2.1. UAVs Research

UAVs research aims to solve different problems related to this area in order to create a competitive field that can help in societies development, by automating complex human tasks. Several of the ideas are based on the design and development of these new vehicles, however, from this work perspective, we are more aware about the intelligence and the autonomy of these systems, specially for the new multi-UAVs systems.

Since the current state of the research do not allow fully independent and intelligent UAV operations, it is important to focus on the human factors associated to these technologies. Considering the importance of the operator work and, specially, the sensitiveness of their tasks and the costs of these technologies from both human and economical perspectives. It is critical to have appropriate means to measure and monitor the operator performance. For this reason, there are several works focused on analyzing behavioural features during UAV operations, specially in the fields of Human Supervisory Control (HSC) and Human-Robot Interaction (HRI) systems (McCarley & Wickens, 2004). These features are usually measured according to the performance standards on HRI systems, which focus on the operator workload and its *Situational Awareness* (Drury et al., 2003). In order to gather information related to direct measures of performance, as the ones used in this work, some ideas are taken from the video games field (Begis, 2000).

From a more general perspective, there are two main research lines in Unmanned Aircraft System (UAS) systems: those focused on the system design (Lemaire et al., 2004) and those developing efficient training processes for the operators (McCarley & Wickens, 2004). The former is relevant according to the number of operators needed to manage a single UAV (typically the model is many-to-one, where several operators manage a single UAV). The later, related to the former, is focused on how to prepare the new operators to deal with these complex tasks, ensuring that the trainee is highly qualified after this process. Due to these systems are currently evolving fast, the training systems need to be redesigned frequently, in order to meet the demands. Besides, in order to cope with the enormous future demand of UAVs operators, it is interesting to extend the availability of these technologies to new inexperienced but promising users, such as video game players (McKinley et al., 2011).

2.2. Machine Learning and Clustering Analysis

Machine Learning is the process of extracting knowledge-based models from data, identifying different patterns (Larose, 2005). Machine Learning techniques have been successfully applied to several different fields, such as *medicine* (Lavrač, 1999), *sports* (Menéndez et al., 2013), *security* (Portnoy et al., 2001) and *transport* (Liao et al., 2007), among others. There are several areas related to Machine Learning, however, in this work we focus on unsupervised learning, specifically clustering analysis (Larose, 2005).

Clustering is focused on discovering knowledge blindly with no labelled information (Larose, 2005). This process groups the data according to some criteria defined by the analyzer. The groups are named clusters and satisfies two main properties: the objects inside a cluster are related to each other, and objects of different clusters are different (Hruschka et al., 2009). These properties make the evaluation process a difficult task (Schaeffer, 2007), and it is still an open problem. However, there are some validation methods based on evaluation indexes (such as the Silhouette or the Dunn index) that provide an objective quality measure of the clustering discrimination process. There are lots of clustering algorithms, some of them based on different perspectives of the clustering problem and the information that can be extracted from the search space. Good and relevant examples are the centroid-based approaches (Macqueen, 1967), where the algorithm optimizes the position of a set of centroids in a known search space, and medoid-based approaches (Kaufman & Rousseeuw, 1987), where the features of the search space are unknown and only the distance between the data instances is known. Using this distance, the most relevant data instances (the so-called medoids) are chosen as the most representative elements of each cluster.

The most classical clustering algorithms are K-means (Macqueen, 1967), Expectation Maximization (Dempster et al., 1977) and Hierarchical Clustering. The first two algorithms are based on statistical iterations over the parameters of a specific estimator, while Hierarchical Clustering nests the clusters by hierarchical levels, describing degrees of similarity by level. Modern algorithms are based on other properties that can be extracted from data, such as continuity (von Luxburg, 2007) (i.e., the shape defined by the data in the space) or density (Navarro et al., 1997). These different ways of dividing the space increase the analyst choices when selecting the appropriate algorithm, and thus the validation process becomes a relevant step in order to determine which is the best solution for a given dataset with respect to the algorithm and metric. Furthermore, another important parameter that is commonly unknown during the clustering process is the number of clusters. Finding the optimum number of clusters is also an open problem, but nevertheless the validation process also provides a general idea about the quality of the cluster according to this parameter (Brock et al., 2008). In this work we are focused on developing a robust validation for the clustering results.

Clustering is also applied to time series. This area, also known as *time series clustering* (Liao, 2005) consists in finding similar time series, grouping them into clusters describing the general trends within the data, and predicting the evolution of a specific time series according to the group it belongs to. Authors working in these scenarios have been specially focused on solving missing values problems or large data volumes, as Iorio et al. who generate a simplified time series using P-splines, which are specially robust to missing values (Iorio et al., 2016). Some application domains of time series clustering are: anomalous event detection (Piciarelli et al., 2008), social media trends (Yang & Leskovec, 2011), and video game-user profiling (Menéndez et al., 2014).

The main goal of this work is to combine clustering algorithms and evaluation indexes to produce a robust process for clustering time series data. This algorithm will group UAV operators' profiles during their training process according to their evolution.

3. Proposed Methodology for the Automatic Retrieval of Representative Simulation Profiles

This section is focused on describing the methodology proposed in this work to retrieve automatically the most representative simulation profiles from a simulation environment. A simulation profile comprises a set of time series (i.e, it is a multivariate time series) representing the evolution of a number of performance measures throughout the execution of a mission. Obtaining and analyzing the most representative simulation profiles is really useful for improving the quality of simulation-based training systems, since it can help not only to exploit general behavioral patterns among simulations, but also to detect off-nominal performances and to study whether the behavior of a specific operator changes when he/she is encountered in dangerous situations.

Given a log of simulations and a set of M performance measures, this process will blindly compute those measures for each simulation and extract the most representative profiles using a two-step clustering-based process. At the end of the process, several representative profiles will be generated, ready to be analyzed and described by a domain expert.

Below are detailed the two steps in which this methodology can be divided, namely the independent discrimination of each of the performance measures and the final extraction of the simulation profiles. In Figures 1 and 2, a graphical overview of this process is shown.

3.1. Step 1: Applying Time Series Clustering on every performance measure separately

Suppose we have a dataset composed of N simulations, each of them containing all the interactions and events happened during a simulation in a specific simulation environment. By using a set of M time-dependant performance measures, every simulation is processed and transformed into M time series, i.e, into a M-dimensional time series. Each dimension represents the evolution of a performance measure. This multivariate time series comprises the profile of that simulation, namely the *simulation profile.*

The first step in this methodology consists in extracting patterns among the M performance measures (i.e. among the M dimensions) **separately**. For this purpose, we will make use of *Time Series Clustering Techniques.* A graphical overview of this step of the methodology is shown in Figure 1.

In order to perform time series clustering, we need to fix three important parameters:

- *Time Series Dissimilarity Metric (μ)*: A crucial question in cluster analysis lies in establishing what we mean by "similar" data objects, i.e., determining a suitable similarity/dissimilarity metric between two objects. In the specific context of time series data, the concept of dissimilarity is particularly complex due to the dynamic character of the series. In this work, since the duration of two simulations usually differs, only those dissimilarity metrics which accept series of different length will be allowed to be part of the methodology. Once a dissimilarity metric is applied over a

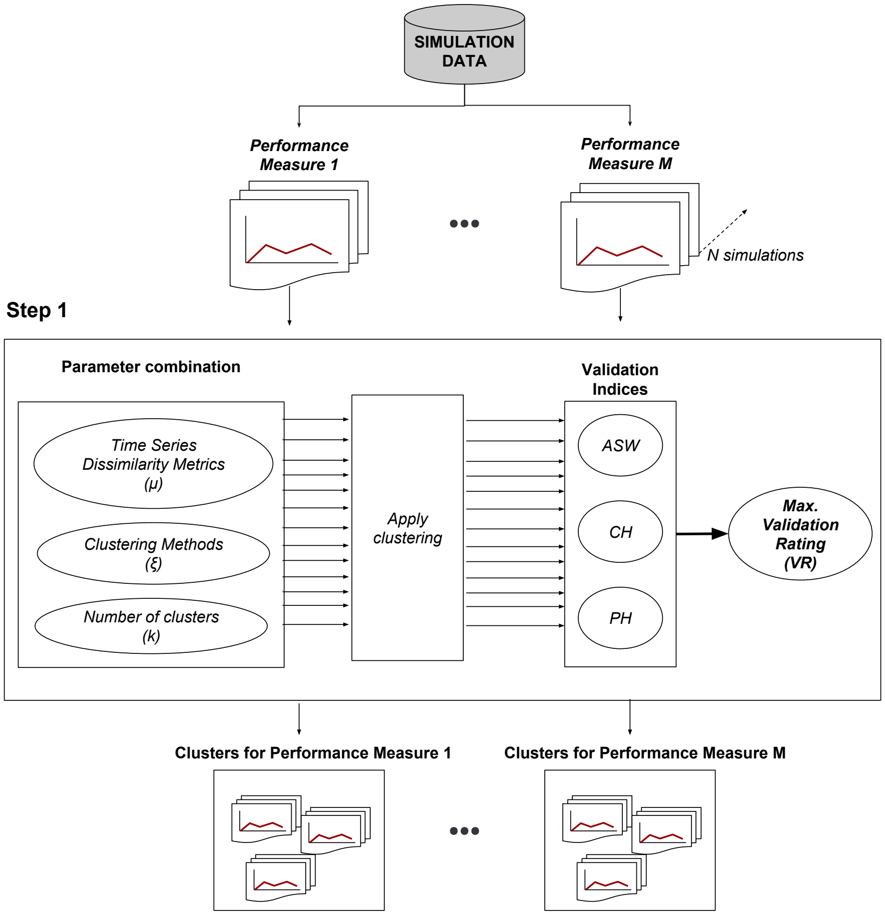


Figure 1: Step 1: Finding the best discrimination for each of the M performance measures separately.

set of time series, a pairwise dissimilarity matrix is obtained and taken as a starting point for a conventional clustering algorithm.

- *Clustering method (ξ)*: Choosing the best clustering method a priori for a given data is a difficult task. The only requirement imposed in this methodology over the algorithm to use is that it can be used with dissimilarities instead of raw data.

- *Number of clusters (k_1)*: Other critical point in many clustering-based systems is to establish the optimal number of clusters, namely k_1, given a dissimilarity matrix and a clustering method to use.

Since we have no prior information about the different groups in which each performance measure can be discriminated, we will compute different clustering solutions using different values of μ, ξ and k_1. Then, in order to automatically decide which is the best discrimination for each performance measure, the results of all those clusterizations will be assessed by three **internal validation indices** , based on the works of Hennig et al. (Hennig & Liao, 2013):

- *Average Silhouette Width (ASW)*: The silhouette of an observation in a specific clusterization measures the degree of confidence with which we can ensure that the observation really belongs to the cluster it is assigned (Rousseeuw, 1987). Given an observation i the silhouette for that observation, $s(i)$, is defined as:

  $$s(i) = \frac{b_i - a_i}{\max(b_i, a_i)},$$

  where a_i is the average intra-cluster distance for i, and b_i the average inter-cluster distance with respect to the nearest cluster to i, i.e:

  $$b_i = \min_{C_k \in C \setminus C(i)} \sum_{j \in C_k} \frac{dist(i,j)}{n(C_k)}, \tag{1}$$

  where $C(i)$ represents the cluster to which i is assigned, and $n(C_k)$ the number of observations contained in cluster C_k. The closer $s(i)$ gets to 1, the more confidence we have of i as well-assigned, and viceversa if $s(i)$ gets close to -1. Finally, to compute the silhouette width of a clusterization, we simply compute the average silhouette value for each observation:

  $$S(C) = \frac{\sum_{C_k \in C} \sum_{i \in C_k} s(i)}{|C|} \tag{2}$$

  The result lies in $[-1, 1]$, and should be maximized in order to achieve a good discrimination.

- *Calinski and Harabasz index (CH)*: Proposed in (Caliński & Harabasz, 1974) and popularized in (Milligan & Cooper, 1985), it establishes a ratio between the separation and cohesion of a partition, defined as:

$$\frac{B(k)(N-k)}{W(k)(k-1)},$$

where k denotes the number of clusters and $B(k)$ and $W(k)$ denote the between (separation) and within (cohesion) cluster sums of squares of the partition, respectively (see details in (Hennig & Liao, 2013)). An optimal clusterization maximizes this measure.

- *Pearson version of Huberts* Γ *(PH)*: This metric rates the Pearson correlation, $\rho(d, v)$ between the vector d of pairwise dissimilarities and the binary vector v that is 0 for every pair of observations in the same cluster and 1 for every pair of observations in different clusters. It was proposed by Hubert in (Baker & Hubert, 1975) and revised by Haldiki et al. in (Halkidi et al., 2001) to overcome some computational problems. Best discriminations are obtained when this value is maximized.

In order to automatically choose the best discrimination based on these validation indices, we define a final *Validation Rating* (VR), which balances the scores obtained for each of the indices defined above. Since all the indices defined denote better clusterizations when maximized, the Validation Rating (VR) is defined as:

$$VR(\mu, \xi, k) = \frac{ASW(\mu, \xi, k)}{\max_{\mu,\xi,k} ASW} + \frac{CH(\mu, \xi, k)}{\max_{\mu,\xi,k} CH} + \frac{PH(\mu, \xi, k)}{\max_{\mu,\xi,k} PH}, \tag{3}$$

where k_1 refers to a specific number of clusters tested in the validation process, μ refers to a time series metric and ξ to a clustering method. Using the criteria of Eq. 3 allows us to choose a discrimination that may not be the best in one of the validation indices, but guarantees reasonable values in all of them. The combination of parameters μ, ξ and k_1 whose clustering result maximizes the value of $VR(\mu, \xi, k)$ will be chosen and pass to the next step.

In this step, we have considered each performance measure as an independent value with respect to the rest (i.e., we only cluster time series using a specific performance measure in an univariate way). By applying the above validation

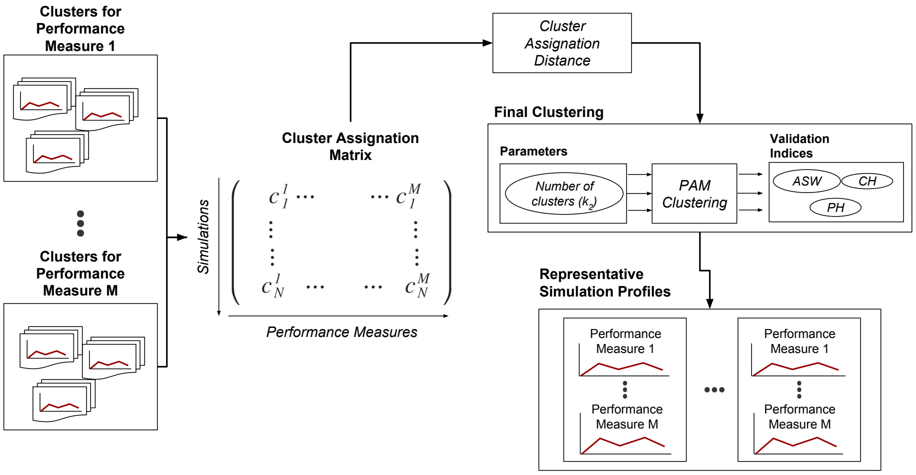


Figure 2: Step 2: Extracting the most representative simulation profiles based on the clustering results obtained by following the process described in Figure 1

process, we automatically obtain a set of M clusterizations containing the shared patterns found among each of the performance measures separately. Note that it may be the case that different performance measures are grouped with different values of (μ, ξ, k), depending on the nature of the data and the measure itself. In fact, the more values we try for each of these parameters, the more chances we have of finding a suitable discrimination of a given performance measure, which provides an easy and scalable framework for the use of this methodology in different simulation environments. In the next step, we will define a multivariate distance for a whole simulation profile using the results obtained in this step.

3.2. Step 2: Extracting the Most Representative Simulation Profiles based on the Clustering Results from Step 1.

Once Step 1 is finished, the M performance measures of all the simulations in the dataset have been clustered into groups of shared temporal behaviour , sharing some features such as the monotony or the minimum and maximum values reached. The next step consists in using those clusters to define the similitude between two simulation profiles. This part of the methodology is based on the work of Menéndez et al. (Menéndez et al., 2014).

Let $\{C_i^m\}_{i=1}^{k_m}$ be the clusters obtained after applying time series clustering on the mth performance measure. Note that the number of clusters, k_m, can vary depending on the measure referred. Each of the N simulations profiles will belong to one cluster per measure. Denoting by c_n^m, $1 \leq n \leq N$, $1 \leq m \leq M$ the assignation of the nth simulation profile to a cluster of the mth performance measure, with $c_n^m \in \left\{C_1^m, \ldots, C_{k_m}^m\right\}$, we can build a $N \times M$ matrix containing the cluster assignations for all the simulations in the dataset:

$$\begin{pmatrix} c_1^1 & \cdots & \cdots & c_1^M \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ c_N^1 & \cdots & \cdots & c_N^M \end{pmatrix} \tag{4}$$

Rows in Eq. 4 represent different simulation profiles and columns account for each of the M performance measures used. Given this matrix, we define a dissimilarity metric between two simulation profiles (rows) based on the number of cluster assignations shared among them. Formally, the *Cluster Assignation Distance* (CAS) between two simulation profiles s_i and s_j is defined as:

$$CAS\ (s_i, s_j) = 1 - \frac{\sum_{m=1}^{M} \delta_{i,j}^m}{M}, \tag{5}$$

where M is the number of performance measure considered, and δ is the *Dirichlet* delta (i.e., the coincidences) defined as:

$$\delta_{i,j}^m = \begin{cases} 1 & if \quad c_i^m = c_j^m \\ 0 & otherwise \end{cases}$$

Given the cluster assignation matrix from Eq. 4 and the dissimilarity metric from Eq. 5, the pairwise dissimilarity matrix among all the simulation profiles can be computed, and used as input for a conventional clustering algorithm. In this case, since we are interested in analyzing the most representative simulation profiles, we will perform a *medoid-based* clustering algorithm to gather the simulation profiles together based on the defined dissimilarity metric and extract the medoids of each of the resulting clusters.

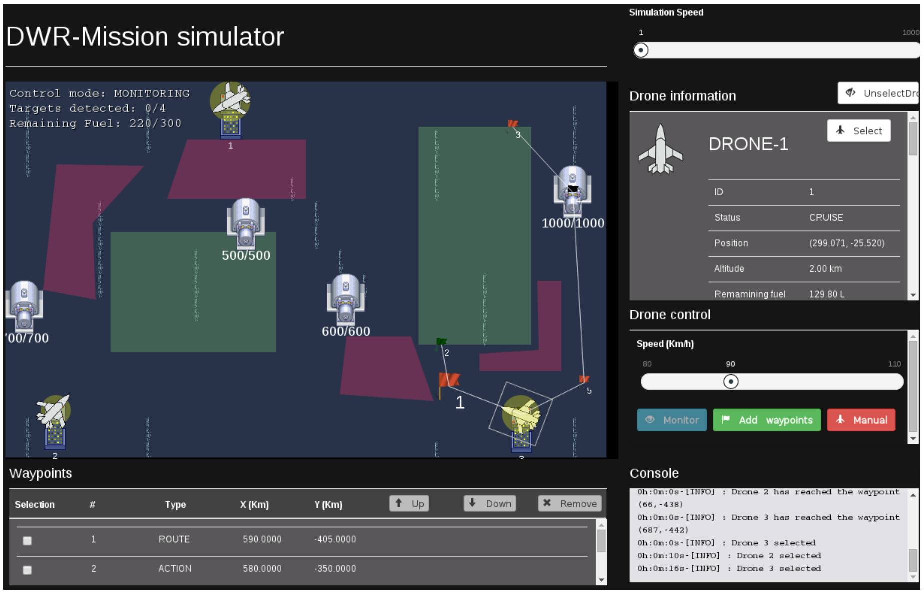
For this work, the *medoid-based* clustering algorithm used in this last step is fixed, and it is the classical Partition Around Medoids (PAM) method (Kaufman & Rousseeuw, 1987). However, as it happened in the first round of clustering of this methodology, an optimal number of clusters (or medoids in this case), namely k_2, needs to be established. The process to select this value will be the same as in the previous step, i.e, we will assess several possibilities via a set of validation indices and get the one maximizing a balance ratio among all of them (See Eq. 3). After that, the optimal medoids will be obtained and will conform the most representative simulation profiles in the dataset. The analysis of these medoids, carried out by a domain expert, will give helpful information about the behavioral patterns followed in the simulations and the causes that increase or decrease the performance of an operator over time.

4. Experimental Setup

In this section, the proposed methodology is tested using a lightweight multi-UAV simulation environment. Below are given all the necessary details to understand how the methodology has been applied, including a brief overview of the simulation environment used, a formal description of the 6 performance measures comprising a simulation profile in this environment, the process of creating ground truth information to evaluate the clustering results, the dataset used and the different parameters fixed for the whole process.

4.1. DWR - A Multi-UAV Simulation Environment

Retrieving data from the interactions of UAV operators during a multi-UAV simulation is a novel task, due to the premature state of the works in this field. This is causing an impediment to expand the analysis in this field towards an accessible place, where an inexpert user could be trained to become a potential expert in UAV operations (Cooke et al., 2006; McKinley et al., 2011).



```
Control mode: MONITORING
Targets detected: 0/4
Remaining Fuel: 220/300
```

ID	1
Status	CRUISE
Position	(299.071, -25.520)
Altitude	2.00 km
Remamining fuel	129.80 L

Selection	#	Type	X (Km)	Y (Km)
□	1	ROUTE	590.0000	-405.0000
□	2	ACTION	580.0000	-350.0000

```
0h:0m:0s-[INFO] : Drone 2 has reached the waypoint
(66,-438)
0h:0m:0s-[INFO] : Drone 3 has reached the waypoint
(687,-442)
0h:0m:0s-[INFO] : Drone 3 selected
0h:0m:10s-[INFO] : Drone 2 selected
0h:0m:16s-[INFO] : Drone 3 selected
```

Figure 3: Screenshot of Drone Watch And Rescue (DWR).

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 (Rodriguez-Fernandez et al., 2015). DWR gamifies the concept of a multi-UAV mission (see Figure 3), 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. One important aspect to remark about this type of simulations is that the level of user interaction is usually low. Operators are instructed to follow a restricted set of procedures in order to overcome incidents, but they are not supposed to interact with the system actively when the mission is going right (Boussemart & Cummings, 2011).

Below is listed all the possible interactions that an operator can perform in DWR:

- *Select UAV*: Allows the operator to focus, monitor and send commands to a specific UAV.

- *Set UAV Speed*: Change the speed of a selected UAV.

- *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. The minimum possible simulation speed is 1, which means that it is equal to real time. The maximum possible value is 1000, which means that it is 1000 times higher than real time.

- *Change UAV Path*: Add/edit/remove waypoints of any UAV. In the case of adding new waypoints, the behavior of the simulator varies depending on the active control mode (see control modes below).

- *Edit waypoint table*: Waypoints can be rearranged in a waypoints table, increasing or decreasing their order.

- *Set control mode*: Control modes in DWR manage how an operator can change the current path of a UAV. There are three control modes:

  1. *Monitor*: This is the default control mode and allows the operator to see and edit the position and order of the waypoints of the selected UAV, but not to add new waypoints.
  2. *Add waypoints*: This control mode allows the operator to view and edit the UAV waypoints, and also to add new waypoints at the beginning of the UAV path, maintaining the rest of the waypoints unchanged.
  3. *Manual*: This control mode allows the operator to define a new path, deleting the previous one.

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.

DWR saves data from a simulation whenever an event occurs during a simulation, 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, which is helpful for the analysis process.

4.2. Performance Measures on DWR

The simulation environment DWR, introduced in Section 4.1, retrieves information about the events triggered and the interactions performed by an operator throughout the execution of a mission. In this section, all the retrieved information will be used to define a set of performance measures which assess the performance of a user in a specific simulation. These measures form the basis for subsequent analysis.

In previous works (Rodríguez-Fernández et al., 2015), the performance measures were computed *globally*, hence every simulation was described as a numeric tuple $(m_1, m_2, \ldots, m_M)$ (assuming that a number of M metrics has been defined), where each metric m_i was represented by a value in the range $[0, 1]$, being 0 the worst performance for that metric, and 1 the best.

However, in this work every performance measure is defined as a time series, thus not only we are able to analyze the general performance of a simulation, but also to study the performance evolution and to detect the time intervals where the values of a specific measure tend to increase or decrease.

A total of six performance measures have been defined: Score(S), Agility(A), Attention (At), Cooperation (C), Aggressiveness (Ag) and Precision (P). All of them take values in the range $[0, 1]$, and are defined cumulatively over time. This means that, given an instant t in the simulation time, the value of a performance measure will depend on the information retrieved from time 0 (simulation start time) to time t. Following this, a simulation profile s is defined as a multivariate time series with the 6-tuple:

$$(S(s), A(s), At(s), C(s), Ag(s), P(s))$$

Below are described each of the performance measures developed for this work.

4.2.1. Score

The *Score* (S) measure gives a global success/failure rate of a simulation. The main goal for an operator in DWR is to detect the maximum number of targets, while keeping safe all the UAVs in the mission. Based on this description, we define the score of a simulation s as:

$$S(s,t) = \frac{1}{2}\left[\frac{|tD(s,t)|}{|T(s)|} + \left(1 - \frac{|dUAVs(s,t)|}{|U(s)|}\right)\right] \tag{6}$$

where $tD(s,t)$ and $dUAVs(s,t)$ refer to the targets detected and the UAVs destroyed respectively up to time t, $T(s)$ is the set of all mission targets and $U(s)$ is the set of all UAVs participating in the mission.

4.2.2. Agility

Agility (A) measures how the speed of the operator interactions varies during a simulation. The speed of an interaction is given by the value of the *simulation speed* at the time when each interaction was performed. As it was mentioned in Section 4.1, the simulation speed in DWR can be set at any time to a value in the range $[1, 1000]$, which causes an acceleration or deceleration of the simulation dynamics. An operator is considered agile if he/she can interact when things are happening fast. Let $I(s,t)$ be the set of all interactions performed up to time t in a simulation s, the Agility at that time is computed as:

$$A(s,t) = \frac{1}{|I(s,t)|} \sum_{i \in I(s,t)} \frac{\mathit{simulationSpeed}(i)}{\mathit{MAX\_\ SPEED}} \tag{7}$$

where $\mathit{MAX\_SPEED} = 1000$ in this simulation environment and *simulationSpeed(i)* gives the speed in which the simulation was running at the moment when the interaction i was made. Note that computing Eq. 7 over time can be seen as calculating the average speed of the interactions cumulatively.

4.2.3. Attention

The *Attention* measure (At) is focused on assessing the progress of the operator intensity in terms of the number of interactions he/she performs throughout the simulation time. Let $I(s,t)$ be, as in the previous section, the set of interactions performed from the beginning of simulation s until time t, the Attention at that time is defined as:

$$At(s,t) = 1 - \frac{1}{1 + \sqrt{|I(s,t)|}}. \tag{8}$$

Note that the time series generated by computing Eq. 8 over time presents a monotonous increasing, since the number of interactions $|I(s,t)|$ always grows when t rises. A square root is introduced in the equation in order to avoid a fast convergence to 1.

4.2.4. Cooperation

Since the simulations carried out in DWR are focused on multi-UAV missions, it is important to measure how the operator has interacted with every available UAV. This concept is issued by the *Cooperation* measure, which is higher the more the interactions of a simulation are balanced among all UAVs. Assuming that a simulation s features a total of N UAVs (Ui), the set of interactions performed up to time t, $I(s,t)$, can be split into N subsets, $\{I_{Ui}(s,t)\}_{i=1}^{N}$, depending on which of the N UAVs was being monitored when the interaction was performed (Some interactions may belong to more than one subset). Let $I_U(s,t) = \{|I_{u1}(s,t)|, \dots |I_{uN}(s,t)|\}$ be the vector gathering the size of each of these subsets, i.e., the number of interactions per UAV, the *Cooperation* is defined as:

$$C(s,t) = \frac{1}{1 + \sqrt{Var(I_U(s,t))}},$$

where *Var*() describes the variance between the size of the different interaction sets. When the variance is low, the user is interacting in a similar way with all the UAV, therefore, the cooperation metric tends to 1.

4.2.5. Aggressiveness

The *Aggressiveness* measure analyzes how the operator changes the strength of his/her interactions during a simulation, in terms of what control mode it has been activated when some changes to the path of a UAV are made. Recall that the simulation environment used in this work features three control modes (*Monitor*, *Add waypoints* and *Manual*), and each of them allows the operator to change the waypoints of a UAV in a different way. In *Monitor* mode, the user is only allowed to move an existing waypoint, which is considered a "soft" interaction. Mode *Add waypoints* permits appending new waypoints to an existing path, while mode *Manual* allows the user to define a whole new path, which is an "aggressive" way of interacting with the simulation.

Since we will measure the Aggressiveness according to the waypoint modifications in the three different modes, we define the sets $W_{MO}(s,t)$, $W_A(s,t)$ and $W_{MA}(s,t)$ which represent the set of interactions with waypoints performed up to time t during the *Monitor*, *Add waypoints* and *Manual* mode, respectively. Using these variables, the measure at time t is defined as:

$$A(s,t) = \frac{\alpha|W_{MA}(s,t)| + \beta|W_A(s,t)| + \gamma|W_{MO}(s,t)|}{|W(s,t)|}, \quad \alpha, \beta, \gamma < 1, \quad \alpha > \beta > \gamma,$$

where $W(s,t) = W_{MO}(s,t) \cup W_A(s,t) \cup W_{MA}(s)$ represents the complete set of waypoint interactions until time t, used to normalize the metric in the range

$[0, 1]$. Parameters α, β, γ are weight coefficients used for balancing the aggressive factor of each type of interaction. Obtaining values of this metric close to 1 indicates that the user is performing mostly aggressive interactions at that time, i.e, he/she is probably defining new paths. On the contrary, values close to 0 designate moments of quick and soft waypoint handling.

4.2.6. Precision

The *Precision* (P) measure studies the replanning skills of a operator on a simulation, rating how he has reacted to the mission incidents. The design of this measure is based on the following assumption: a precise operator should only alter the path of a UAV when an incident occurs. Therefore, the waypoints added when no incident is happening should penalize the precision rate. Based on this, we can divide the computation of this measure into two parts: the precision in times of incidents (*Incident Precision*, P_I) and the precision when nothing is altering the normal execution of the simulation, i.e, when the operator must only monitor the simulation status (*Monitoring Precision*, P_M).

The *Incident Precision* (P_I) supposes that every waypoint added/edited/removed during a specific interval time from the beginning of an incident (10 seconds for this work) is placed in order to avoid that incident, so it is considered as a precise interaction. Let $In(s, t)$ be the set of incidents triggered up to time t on simulation s, we can compute $P_I(s, t)$ as follows:

$$P_I(s,t) = \frac{\sum_{i \in In(s,t)} p_I(i,s,t)}{|In(s,t)|} \qquad p_I(i,s,t) = 1 - \frac{1}{1 \quad + \quad |W_i(s,t)|},$$

where $p_I(i, s, t)$ gives the precision for an specific incident i. In this last equation, $W_i(s, t)$ is the set of all *waypoint interactions* (add/edit/remove) performed since the incident i started until a maximum of 10 seconds after, i.e, interactions within the interval $[startTime(i), min(startTime(i)+t, startTime(i)+10)]$. The more waypoints changed during this interval, the more the precision increases for that incident.

The *Monitoring Precision* (P_M) is conceptually contrary to P_I, in the sense that it penalizes the waypoint interactions performed out of the scope of incidents, i.e, during *monitoring time*. Here, the less interactions the more precision obtained. It is computed as

$$P_M(s,t) = \frac{1}{1 + |W_M(s,t)|}, \qquad W_M(s,t) = \overline{\bigcup_{i \in In(s,t)} W_i(s,t)},$$

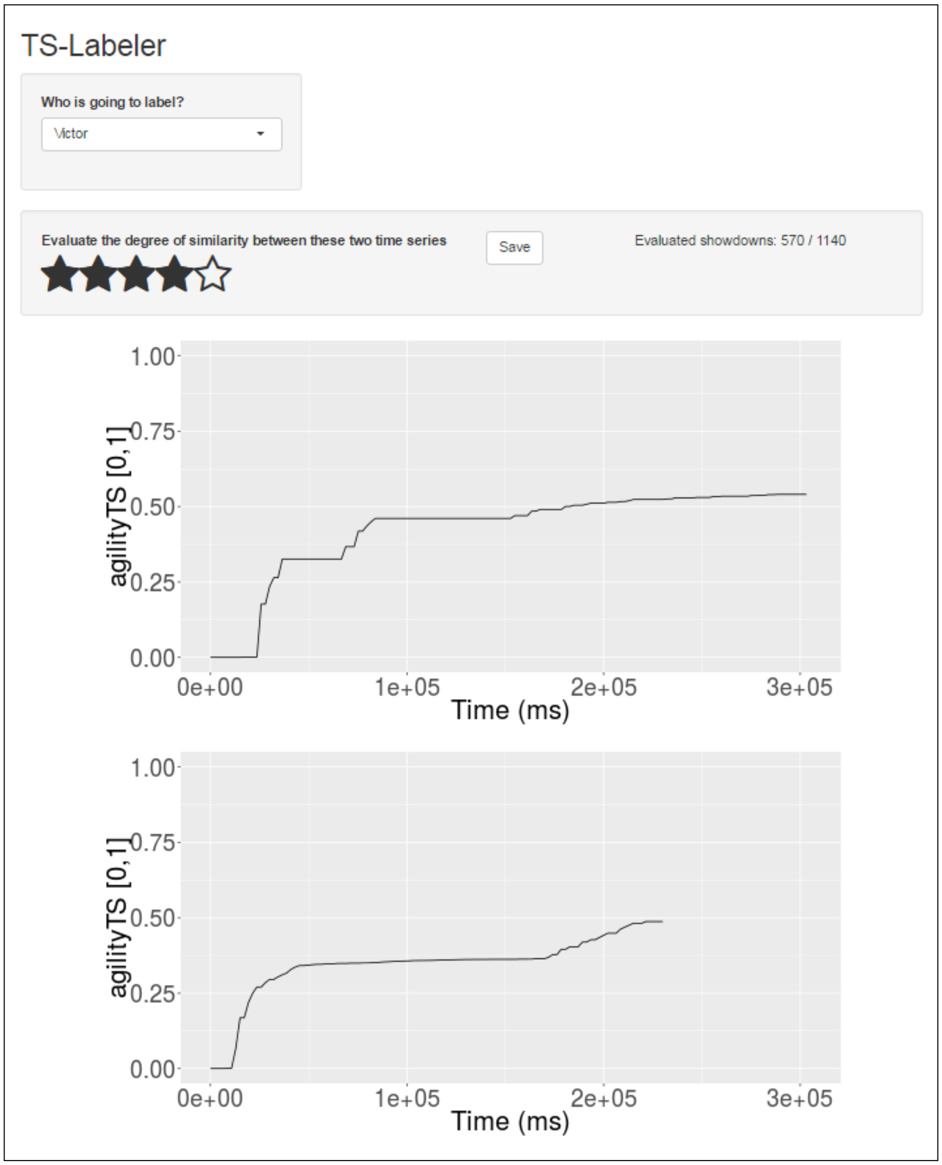
where $W_M(s,t)$ is the set of all waypoint interactions performed during monitoring time up to time t, i.e, the complementary of the union of all waypoint interactions made to avoid any of the incidents triggered until that moment. Averaging the values of the *Incident Precision* and the *Monitoring Precision*, we finally get the expression for the Precision measure:

$$P(s,t) = \frac{P_I(s,t) + P_M(s,t)}{2} \tag{9}$$

4.3. Evaluation Criteria

In order to perform an external evaluation of the clustering results obtained in this work, and to compare them objectively against other clustering approaches, we have created a ground truth dataset based on collective human judgement, inspired by the work of Afnan et al. in (Al-Subaihin et al., 2016), where the similarity of a set of mobile apps is rated manually by several users. Human judgement as a way to create ground truth data is also typical from the field of sentiment analysis. Here, a group of expert users categorize the opinion expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic is positive, negative or neutral (Liu, 2012).

In this work, the ground truth is created by asking users to rate the similarity of pairs of time series, corresponding to the evolution of a specific performance measure between two randomly selected simulations executed in DWR. Ratings are given on a 5-star rating system (Pang & Lee, 2005), where 1 star indicates the lowest possible similarity and 5 the highest. Note that, although in this work the unit of analysis is a *simulation profile*, which is a multivariate time series, the item to rate by humans is a pair of 1-dimensional time series. This is because comparing a pair of multivariate time series is much more difficult, and thus, the resulting ground truth would be less reliable.



TS-Labeler

Who is going to label?
Victor

Evaluate the degree of similarity between these two time series
★★★★☆

Evaluated showdowns: 570 / 1140

Figure 4: Screenshot of the app developed to create a ground truth dataset by labeling the similarity between pairs of time series. In the screenshot, the user is asked to rate the similarity between two time series representing the evolution of the agility performance measure in two randomly selected simulations executed in DWR.

To measure the degree of consistency among the evaluations from multiple raters, many statistical measures have been studied, depending on the number of participants and the type of scale used. Some examples are the Cohen's Kappa and Weighted Kappa, when there are two raters (Cohen, 1968), the Fleiss Kappa (Fleiss, 1971) when multiple raters use a nominal or categorical scale, or the Interclass Correlation Coefficient (ICC) (Bartko, 1966) for semantic-differential scales. Since we use a ordinal scale with multiple raters, we select the Kendall's Coefficient of Concordance (W) (Kendall & Smith, 1939). Kendall's W assigns a value of consistency among the raters that ranges between 0 and 1. Low values indicate high variations of the scores given to each item by the raters, and high values indicate more consensus.

The process of rating has been automated by the use of a web-app, whose graphical user interface can be seen in Figure 4. This app simply takes two random simulations from the simulations dataset, and choose a random performance measure to show (as a time series). Once the user has rated their similarity, the app stores the corresponding information and show a new pair of time series. For every submitted similarity rating, we store the following information:

- Rater's name
- Identifier of the two simulations that have been faced in the rating process.
- Name of the performance measure that has been rated (Score, Agility...).
- Value of the similarity rating assigned (one of $\{1, 2, 3, 4, 5\}$)

Given this data, the Average Similarity Rating (ASR) between two simulations s_i and s_j is computed by averaging the ratings values for all the stored data between this pair of simulations. The more evaluations we have between every possible pair of simulations, the more reliable will be the ground truth dataset. Formally, let $S^p = \{(s_i, s_j)\}_{i,j=1}^{N}$ be the set of all possible pairs (ignoring order) of simulations in our simulation dataset S, the ground truth of this work can be defined as a function ASR: $S^p \to [1, 5]$.

Algorithm 1 Pairwise Accuracy (P-Acc)

Input: $C = (C_{s_1}, \ldots, C_{s_N})$ is a the clustering solution to evaluate. $ASR\colon S^p \to [1,5]$ is the ground truth function. θ^A is the match acceptance threshold. θ^R is the match rejection threshold

Output: Value between 0 and 100 indicating the pairwise accuracy of the clustering solution

```
function P-Acc($C, ASR, \theta^A, \theta^R$)
    $P\text{-}Acc \leftarrow 0$
    $decisivePairs \leftarrow 0$
    for $(s_i, s_j) \in S^p \ (i \neq j)$ do
        if $ASR(s_i, s_j) \geq \theta^R$or$ASR(s_i, s_j) \leq \theta^A$ then    ▷ $ASR(s_i, s_j) \in [1,5]$
            $decisivePairs \leftarrow decisivePairs + 1$
            if $ASR(s_i, s_j) \geq \theta^A \ \& \ C(s_i) = C(s_j)$ then
                $P\text{-}Acc \leftarrow P\text{-}Acc + 1$
            if $ASR(s_i, s_j) \leq \theta^R \ \& \ C(s_i)! = C(s_j)$ then
                $P\text{-}Acc \leftarrow P\text{-}Acc + 1$
    return $(P\text{-}Acc/decisivePairs) * 100$
```

To measure the accuracy of a clustering result against this type of ground truth, we check whether the pairs of simulations rated with high similarity ratings are assigned to the same cluster or not. If the ASR between a pair of simulations in the ground truth is greater than a given *match acceptance threshold* (θ^A) (a value between 1 and 5), then the ground truth is saying that they are very similar, so a given clustering solution should locate them at the same cluster. On the contrary, if the ASR between the two simulations is lower than a given *match rejection threshold* (θ^R), the clustering solution is expected to place the simulations into different clusters. If the ASR falls between the two thresholds, then we consider that the human judgment is not decisive, and that pair is not taken into account.

Parameter	**Value**
Map extension	800 x 500
UAVs	3
Surveillance Areas	2
Number of Targets	4
Preplanned Incidents	4 (2 Danger Area and 2 PayloadBreakdown)
No Flight Zones	2
Refueling Stations	4

Table 1: Summary of the main features of the test mission used to carry out the simulations conforming the dataset of this work.

With this, let $C = (C_{s_1}, \ldots, C_{s_N})$ be a given clustering solution for those simulations, we define the *Pairwise-Accuracy (P-Acc)* as the percentage of concordance between the clustering solution and the ground truth over every pair of simulations in S^p. Algorithm 1 shows in detail the process to calculate this value. Basically, it consists in looping over every pair of simulations checking, firstly, whether the ASR value for that pair is decisive or not, and finally, whether the clustering solution classify in the same cluster or not both elements in the pair.

4.4. Dataset

In this work, 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. All users completed a brief tutorial before using the simulator, explaining the mission objectives and the basic controls. After that, they were told to execute a test mission prepared for this experiment. That mission (see Figure 3) features 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. Although all the incidents were planned to be triggered at the same simulation intervals, a user could receive an incident sooner or later depending on the speed with which he/she was running the simulation. A summary of the contents featured in this mission is given in Table 1. For more information about the mission elements involved in the simulation see (Rodriguez-Fernandez et al., 2015).

The dataset resulted from this experiment comprises 87 distinct simulations, executed by a total of 40 users. To achieve a robust analysis of the data extracted, we must clean the dataset by removing those simulations which can be considered as useless. Taking into account the difficulty level of the test mission, we have considered as useless those simulations aborted before 20 seconds of duration or those which presented less than 10 interactions. From the 87 simulations composing our initial simulations dataset, only 55 of them are considered as useful for this experiment, hence $N = 55$.

Regarding the ground truth dataset, a total of 3 raters, namely the authors of this paper, has contributed to the rating of time series pairwise similarities. Due to the abundance of possible combinations of the tuple (simulation 1, simulation 2, performance measure) to rate, we draw a random sample of 20 simulations, $\widetilde{S}$, from the original dataset S. Thus, the set of possible simulation pairs $\widetilde{S}^p$ to be rated contains 190 unique elements $(20 * 19/2)$. Since there are a total of 6 performance measures for each simulation, the number of possible cases to rate amounts to 1140 $(190 * 6)$.

After several days of using the web-app for creating the ground truth (See Figure 4), a total of 1742 evaluations were gathered, covering 936 out of the 1140 possible rating cases. That means that for many pair of simulations there are some performance measures that have not been rated. Depending on the *minimum number of rated performance measures* that we establish as necessary to achieve an accurate analysis, and on the threshold values of θ^A and θ^R needed to decide whether the human judgement for a given pair of simulations is decisive or not, the number of useful ground truth data will vary.

With regard to the Kendall's coefficient of concordance (W), there are a total of 45 rating cases that have been evaluated by all the raters. Using this common cases, the achieved coefficient is 0.58 (*p-value* $= 0.0018$), and thus, according to the common criteria to judge this value (Remøy, 2010), there is a **moderate agreement** among raters.

4.5. Parameter Tuning

In this section, all the free parameters seen in the proposed methodology will be assigned to a value or a set of values in the context of this experiment. A summary of this parameter tuning is shown in Table 2.

Once the dataset has been created, the simulation profile for every simulation in the dataset has to be processed. To do this, we will use the set of performance measures defined in Section 4.2, so we have a total of $M = 6$ time series comprising each simulation profile.

In order to compute a simulation profile, the measures will be computed for different *time steps* throughout the whole simulation duration. These time steps are obtained from sampling the whole simulation time into equidistant time slots, fixing a time slot resolution. That sampling resolution, in this experiment, is computed automatically as the average distance between subsequent interactions in all the simulation dataset. For the dataset used in this work, this results in a sampling resolution of 2000 ms. Thus, every 2000 ms we will compute the values of each of the performance measures defined in Section 4.2 and create the performance time series for each simulation.

In order to perform time series clustering for each of the performance measures (Step 1 of the proposed methodology), different dissimilarity metrics (values of μ), different clustering methods (values of ξ) and different number of clusters (values of k_1) are tested during the validation process. Recall that the time series dissimilarity metrics to use must accept series of different length, and the clustering methods must work with a pairwise dissimilarity matrix as input.

Regarding time series metrics, two examples are tested and compared. Both

Table 2: Parameter tuning for all the variables involved in the experimental setup of this work.

Context	**Parameter**	**Value**
DWR	Number of performance measures (M)	6
	Sampling Resolution	2000 ms
Proposed Methodology	Time Series Metrics	Frechet DTWarp
	Clustering methods	AGNES DIANA PAM
	Possible number of clusters for step 1 (K_1)	$2 \dots 8$
	Possible number of clusters for step 2 (K_2)	$3 \dots 8$
Ground truth and Clustering Evaluation	Minimum number of performance measures rated for each pair of simulations	4
	Match Acceptance Threshold (θ^A)	3.5
	Match Acceptance Threshold (θ^R)	2.5

of them allow to recognize similar shapes among time series, even in the presences of signal transformations such as shifting or scaling:

1. *Fréchet Distance*: This distance has been extensively used in the time series framework for both continous and discrete cases (Eiter & Mannila, 1994). It does not just treat the series as two point sets, but it has into account the ordering of the observations and can be computed on series of different length. Denote by X and Y two discrete time series and by P the set of all possible sequences of p pairs preserving the data order in the form

   $$\left((X_{a_1}, Y_{b_1}), \ldots, (X_{a_p}, Y_{b_p})\right),$$

   then the Fréchet distance is computed as:

   $$\mathit{Frechet}(X,Y) = \min_{P}\left(\max_{i=1,\ldots,p} |X_{a_i} - Y_{b_i}|\right)$$

2. *Dynamic Time Warping Distance (DTWarp)*: This distance, very popular in the field of time series pattern recognition is aimed to minimize the sum of distances between the sequence of pairs as defined for the Fréchet distance. The definition of the $DTWarp$ distance is given by:

   $$DTWarp(X,Y) = \min_{P}\left(\sum_{i=1,\ldots,m} |X_{a_i} - Y_{b_i}|\right)$$

Regarding the clustering methods to test, three classical algorithms will be applied, both of them allowing dissimilarity matrices as input data:

1. *Agglomerative Nesting (AGNES)*: This is one of the most frequently used clustering algorithms (Kaufman & Rousseeuw, 2009). It is a bottom-up, non-parametric hierarchical algorithm. Each observation is initially placed in its own cluster, and the clusters are iteratively joined together according to their closeness. This closeness of any two clusters is measured by a dissimilarity matrix between sets of observations, usually achieved by use of an appropriate metric (Euclidean distance in this case). The results

of this algorithm (and all hierarchical methods) are usually presented in a *dendrogram*. This dendrogram can be cut at a chosen height to produce the desired number of clusters.

2. *DIANA*: DIvisive ANAlysis Clustering (DIANA) (Kaufman & Rousseeuw, 2009) is a divisive hierarchical algorithm that constructs the hierarchy in the inverse order (*top-down*). It initially starts with all observations in a single cluster, and successively divides the clusters until each cluster contains a single observation. Although it is usually less efficient than the agglomerative nesting, DIANA stands out as a competitive clustering algorithm for many fields (Datta & Datta, 2003).

3. *Partition Around Medoids (PAM)*: Proposed by Kaufman et al. in (Kaufman & Rousseeuw, 1987), this algorithm is similar to the popular K-means. In contrast to the k-means algorithm, PAM chooses data points as centers (called *medoids*) instead of centroids.

In order to choose the optimal number of clusters in the first step of the methodology, we will test different values of K_1, from 2 to 8. For the second step, we search among values of K_2 from 3 to 8.

Regarding the creation of the ground truth dataset, we set the minimum number of rated performance measures for every pair of simulations in 4, over a total of 6. This way, the amount of useful rated simulation pairs in our dataset is reduced from 190 to 132. With regard to the values of θ^A and θ^R, we establish that any pair of simulations with an average rating above $\theta^A = 3.5$ or below $\theta^R = 2.5$ will be decisive for the clustering evaluation process. On this basis, only 128 simulation pairs will conform our decisive ground truth information.

Every process of clustering, validation and evaluation described in this work has been implemented in the *R* Statistical Environment, using the packages *TSclust* for the computation of time series dissimilarities (Montero & Vilar, 2014), and *fpc* for the computation of the validation indices (Hennig, 2010).

The final code is available on Github. [1]

5. Experimentation

In this section, we will deepen into the results obtained after applying the proposed methodology to extract the most representative simulation profiles in DWR. First of all, we will detail the intermediate and final validation results of our two-step methodology, and check the evaluation results against the ground truth data. Finally, a comparative study will be carried out in order to compare the results of the proposed methodology against other clustering approaches.

5.1. Results from applying the proposed methodology in DWR

Due to the large number of parameter combinations tested to find a good cluster discrimination for each performance measure in the first step of the methodology, only the best results are summarized in Table 3 for legibility purposes. As it can be seen, all dissimilarity metrics and clustering methods tested are selected as "best" at least once. The *Validation Rating* introduced in this work allows an easy comparison among clusterizations and avoids the differences in the range of each of the validation index. The optimal number of clusters chosen is, excluding the *Attention* and *Cooperation* measures, always the minimum or maximum value of k tested. This gives us general information about the variance in the temporal evolution of each of the metrics and must be taken into account when analyzing the simulation profiles: those time series grouped into 8 different clusters will define a richer set of behaviors and must be given more importance than those with only 2 different patterns detected (best k is 2).

Based on the best clusterizations given by the results of Table 3, a $N \times M$ cluster assignation matrix is built following the structure described in Eq. 4. After applying Eq. 5 over this matrix and cluster the resulted dissimilarity matrix using a PAM algorithm with values of k_2 from 3 to 8, we select $k_2 = 7$ as

[1] The code will be published once this work is accepted due to copyright issues

	PERFORMANCE MEASURES					
	S	**A**	**At**	**C**	**Ag**	**P**
Dissimilarity Metric	*Frechet*	*Frechet*	$DTWarp$	*Frechet*	*Frechet*	*Frechet*
Clustering Method	PAM	PAM	AGNES	AGNES	DIANA	PAM
Number of Clusters (k_1)	8	8	3	7	8	8
ASW	0.586	0.581	0.708	0.606	0.580	0.587
CH	608.725	596.207	1354.357	529.529	510.083	611.322
PH	0.389	0.397	0.423	0.436	0.443	0.398
Validation Rating (VR)	2.332	2.335	2.243	2.376	2.390	2.344

Table 3: Summary of the best validation results for the time series clustering of each of the performance measures used, corresponding to Step 1 of the methodology proposed in this work.

optimal number of clusters to separate the simulation profiles. Table 4 shows the validation results for each of the values of k_2 tested in this last clustering process. As it can be seen, the selected k_2 not only get the best general rating (represented by the Validation Rating), but also maximizes each of the validation indices independently.

The 7 medoids of this clusterization represent the most representative simulation profiles for this dataset. Section 6 will focus on analyzing those profiles (medoids) and give some ideas about the typical behaviours followed by the users of this experiment.

With regard to the external evaluation, we calculate the Pairwise Accuracy (P-Acc) of the clustering results as detailed in Algorithm 1. The result marks 84.09%, which is quite a good result taking into account the accuracy values

Table 4: Validation results for the final clustering process of the proposed methodology, corresponding to Step 2. Bolded cells represent the best results obtained.

	ASW	**CH**	**PH**	**VR**
$K_2 = 3$	0.451	22.919	0.592	1.417
$K_2 = 4$	0.628	44.298	0.828	2.086
$K_2 = 5$	0.701	53.13	0.843	2.273
$K_2 = 6$	0.73	67.724	0.897	2.498
$K_2 = 7$	**0.788**	**112.873**	**0.924**	**3.000**
$K_2 = 8$	0.782	103.035	0.807	2.779

usually obtained when using human judgement-based ground truth data. As an example, in the world of sentiment analysis, values of accuracy above 70% are considered more than acceptable (Pak & Paroubek, 2010).

5.2. Comparative study between the proposed methodology and other multivariate time series clustering approaches.

In this section, we are interested in finding out if the proposed methodology performs better than other clustering approaches. Due to the unit of analysis that we want to cluster is a simulation profile, which is a multivariate time series composed of the evolution of several performance measures, we will compare our approach against a PAM clustering applied over different multivariate time series distances from the literature. As a requisite, we need that the distance accepts time series of different length, so that we can compare simulations with different durations. Below is detailed the list of multivariate time series metrics used for this comparison:

- *Mean Frechet*: This metric computes the Frechet distance for each component of the multivariate time series and averages the results.
- *Mean DTW*: The same than *Mean Frechet* but using DTW as metric.
- *Penrose Distance*: Proposed by Penrose in (Penrose, 1952), it computes a distance based on means, variance and covariances for each sample based

Table 5: Comparative results, in terms of Pairwise Accuracy (P-Acc), between the proposed methodology against direct clustering approaches based on multivariate time series distances. The results are compared for different number of clusters (K_2). While the bolded cell indicates the result obtained for the proposed methodology, cells in italics show the results that surpasses our best value.

	Penrose Distance	**Mahalanobis Distance**	**Mean DTWARP**	**Mean Frechet**	**Proposed Methodology**
$K_2 = 3$	60.61	46.97	62.12	68.94	-
$K_2 = 4$	66.67	73.48	73.48	71.97	-
$K_2 = 5$	73.48	75.76	76.52	81.82	-
$K_2 = 6$	75.00	78.03	77.27	84.09	-
$K_2 = 7$	83.33	*84.85*	78.79	*87.88*	**84.09**
$K_2 = 8$	*87.12*	*88.64*	*86.36*	*89.39*	-

on p variables. It takes into account within population variation by weighting each variable by the inverse of its variance, but does not account for correlations among variables.

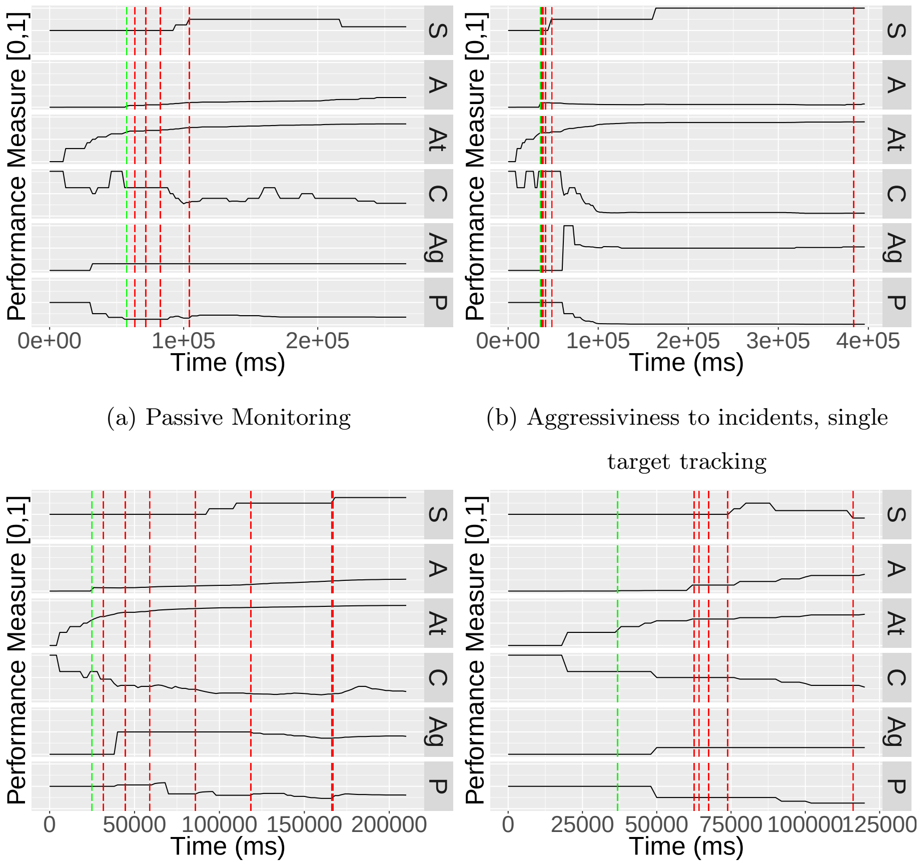
- *Mahalanobis Distance*: Described in (De Maesschalck et al., 2000), this distance is very similar to the Penrose distance, except for the fact that in this case, the contribution of each pair of variables is “weighted” by the inverse of their covariance.

The results of this comparison are shown in Table 5. The value used to compare the clustering results is the Pairwise Accuracy (P-Acc) against our ground truth dataset (See Algorithm 1). Results for the other clustering approaches are given for different number of clusters, ranging from 3 to 8, exactly the same range of values used in the last step of the proposed methodology. Note that the results of the proposed methodology for values of K_2 different than $K_2 = 7$ have a low interest, since these values were not chosen as best in the internal validation process.

From the results we can appreciate that, on a total of six occasions, some other clustering result has surpassed the P-Acc value obtained with respect to the proposed methodology. Also, it can be noted that in general, the *Mean Frechet* distance is the most suited for this experiment, probably because of the nature of the performance measures defined for this specific experimental setup. However, these results have to be analyzed with the sights set on a bigger picture, due to the proposed methodology is clearly intended for being applied in any simulation environment. Thus, achieving a $P - Acc$ value of 84.09% is clearly above the mean accuracy for the rest of the methods, and this is achieved without the need of selecting any parameter a priori. In fact, since the proposed methodology is scalable, it may be the case that adding more clustering methods or more dissimilarity metrics to the first step of the methodology would lead to an increase of the accuracy. In conclusion, summarizing the pros and cons of applying this the proposed methodology, we conclude that this methodology is quite accurate and interesting for open and new environments where the nature of the time series is unknown, and though, one does not know a priori which clustering configuration is optimal for the problem.

6. Discussion - Analysis of the Most Representative Simulation Profiles

Generally, the knowledge of a domain expert is required when developing a cluster analysis, specially when the clusters represent time series data. In our case, due to our experience with the simulation environment DWR gained from previous works (Rodriguez-Fernandez et al., 2015), we are able to carry out an analysis of the most Representative Simulation Profiles (RSPs) obtained by applying the methodology proposed in this work. In other works, when static profiles were used, this analysis was automated by using a set of Fuzzy Control Systems (Rodríguez-Fernández et al., 2015b), but due to the complexity introduced in this paper by the temporal nature of the performance measures, the analysis is carried out by observing in detail each of the profiles.



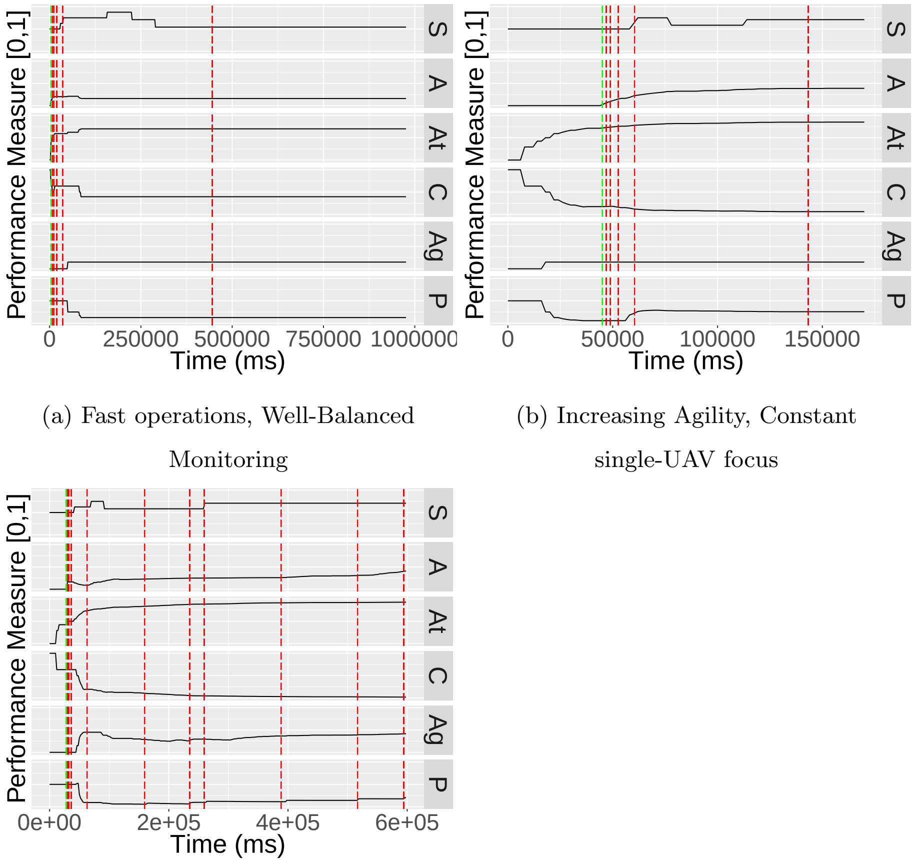
(a) Passive Monitoring

(b) Aggressiviness to incidents, single target tracking

(c) Well-balanced behavior, cautious and relaxed after incidents

(d) No waypoint interactions, Aborted Mission

Figure 5: Plots of the most representative simulation profiles (I). Red lines mark times when an incident was triggered, and the green line indicates the moment when the mission preparation phase finishes and the execution phase starts. Each subplot contains the evolution of the six performance measures (S,A,At,C,Ag,P) for comprising a simulation profile.



(a) Fast operations, Well-Balanced Monitoring

(b) Increasing Agility, Constant single-UAV focus

(c) Cautious, passive before incidents, single UAV-focus

Figure 6: Plots of the most representative simulation profiles (II). Red lines mark times when an incident was triggered, and the green line indicates the moment when the mission preparation phase finishes and the execution phase starts. Each subplot contains the evolution of the six performance measures (S,A,At,C,Ag,P) for comprising a simulation profile.

Figures 5 and 6 show a grid with the evolution of the performance measures defined in work for each RSP found in this experiment. In order to facilitate the analysis, red lines mark the instants when some incidents are triggered, and a green line marks the moment when the operator started to accelerate the simulation speed for the first time, denoting the end of the *mission preparation phase* and the beginning of the *mission execution phase*. During the mission preparation phase, the simulation is paused, and the operator can spend some time overviewing the scenario and making changes in the initial mission plan.

Based on the plots of Figures 5 and 6, the RSPs are described as follows:

1. *Passive Monitoring* (Figure 5a): This simulation profile features a constant level of attention once the mission preparation phase has ended. This means that there have been scarcerly a few interactions during the mission execution phase. Thus, all the performance measures which depend directly on the interactions will remain constant. Incident times are close to each other, which means that the simulation speed set to start the mission execution phase is high. Despite all this, the *Score* does not decrease until the end of the mission, which suggests that operators within this simulation profile trust in the pre-loaded mission plan in order to detect all targets.

2. *Aggressiveness to overcome incidents* (Figure 5b): This simulation profile features a type of aggressive operation. After a soft mission preparation, only dedicated to overview the map (no paths are changed because aggressiveness marks 0), the simulation begins with high speed, and to overcome the incidents, the paths of the UAVs are completely redesigned (maximum aggressiveness). The rest of the simulation maintains the path of one single UAV, ensuring that it detects all targets. The mission finishes with maximum score, which indicates that all targets have been detected and none of the UAVs were destroyed.

3. *Well-balanced behavior, cautious and relaxed after incidents* (Figure 5c): Unlike the previous simulation profile, this one features a more relaxed

behavior in terms of the way the operator acts to solve the incidents. The simulation speed is set lower, due to the wide time intervals between subsequent incidents, and whenever something alters the simulation, only a few and soft interactions, possibly waypoint editions, are performed, maintaining the cooperation among all the UAVs in the mission. The result of this behavior in terms of score is also achieving the maximum score possible, but this time the process is made without taking drastic decisions and having into account all the available UAVs and resources.

4. *No waypoint interactions* (Figure 5d): This simulation profile is notable for having no interactions with the paths of the UAVs. Operators within this profile just monitor the mission execution during short periods of time, and abort the mission when some of the UAVs are lost. This is suggested by the decrease of the Score metric just before the end of the simulation time.

5. *Fast operations* (Figure 6a): This simulation profile represents fast operations where the mission preparation phase is practically nonexistent. At the beginning of the simulation execution, when all incidents are triggered, the operator tries to manage them by altering as little as possible the path of each of the UAVs.

6. *Increasing Agility, Constant single-UAV focus* (Figure 6b): In this simulation profile, we see how the agility metric constantly increases over time, which indicates that the operator is gradually taking control of the mission and that allows him/her to speed up the simulation speed. Also, it can be seen that from the very beginning of the mission preparation phase that the cooperation metric goes down drastically, suggesting that the focus of the control is always located on one single UAV.

7. *Cautious, passive before incidents, single UAV-focus*: (Figure 6c): This profile is very similar to the one from Figure 5c, except for the fact that in this case, the precision measure maintains low values during all the

simulation, which is a sign of passivity before alerts.

7. Conclusions and Future Work

This work presents an analytical methodology based on time series clustering to extract representative simulation profiles from UAVs operators during their training processes. Assuming that we have defined the profile of a simulation as a multivariate time series composed of the evolution of several performance measures, the proposed methodology begins by grouping the data for each measure separately, validating different clustering configurations. The clustering results for each measure are used to define the similarity between two simulation profiles, which is used in a last medoid-based clustering process to extract the most representative profiles.

This methodology has been applied in a lightweight multi-UAV environment where a total of 6 performance measures comprises the profile of a simulation. To evaluate the results, a human judgement-based ground truth dataset has been created by asking users to rate the similarity between pairs of time series. The results obtained from the experimentation show that the proposed methodology gets good accuracy scores, specially from a general perspective, due to the scalability offered to the use of different time series metrics and clustering methods. Furthermore, the different representative profiles obtained in the experimentation have been qualitatively analyzed, according to the decisions that operators take during a training session. This shows how this methodology can be applied to describe real cases, where the performance needs to be evaluated with a high granularity level.

The future work will be focused on: 1. Applying the proposed methodology in different simulation environments using different performance measures, and verify objectively the scalability of the solution. 2. Trying to use this information to predict significant reductions in the performance of an operator. 3. Developing different performance measures, ensuring that all of them offer valuable information. 4. Assessing the evolution of an operator not only during a single simulation, but during a whole training process. 5. Extending this methodology to large data using robust methods for missing values problems such as P-splines (Iorio et al., 2016).

Acknowledgements

This work has been supported by the next research projects: EphemeCH (TIN2014-56494-C4-4-P) Spanish Ministry of Economy and Competitivity, CIBER-DINE S2013/ICE-3095, both under the European Regional Development Fund FEDER, SeMaMatch EP/K032623/1 and Airbus Defence & Space (FUAM-076914 and FUAM-076915). 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

Al-Subaihin, A., Sarro, F., Black, S., Capra, L., Harman, M., Jia, Y., & Zhang, Y. (2016). Clustering mobile apps based on mined textual features. In *2016 International Symposium on Empirical Software Engineering and Measurement* (p. In press). IEEE.

Baker, F. B., & Hubert, L. J. (1975). Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association*, *70*, 31–38.

Bartko, J. J. (1966). The intraclass correlation coefficient as a measure of reliability. *Psychological reports*, *19*, 3–11.

Begis, G. (2000). Adaptive gaming behavior based on player profiling. US Patent 6,106,395.

Boussemart, Y., & Cummings, M. L. (2011). Predictive models of human supervisory control behavioral patterns using hidden semi-markov models. *Engineering Applications of Artificial Intelligence*, *24*, 1252–1262.

Brock, G., Pihur, V., Datta, S., & Datta, S. (2008). clvalid: An r package for cluster validation. *Journal of Statistical Software*, *25*, 1–22.

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, *3*, 1–27.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, *70*, 213.

Cooke, N. J., Pedersen, H. K., Connor, O., Gorman, J. C., & Andrews, D. (2006). 20. acquiring team-level command and control skill for uav operation. *Human factors of remotely operated vehicles*, *7*, 285–297.

Datta, S., & Datta, S. (2003). Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, *19*, 459–466.

De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, *50*, 1–18.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*, 1–38. doi:10.2307/2984875.

Drury, J. L., Scholtz, J., & Yanco, H. A. (2003). Awareness in human-robot interactions. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on* (pp. 912–918). IEEE volume 1.

Eiter, T., & Mannila, H. (1994). *Computing discrete Fréchet distance*. Technical Report Citeseer.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, *76*, 378.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, *17*, 107–145.

Hennig, C. (2010). fpc: Flexible procedures for clustering. *R package version*, *2*, 0–3.

Hennig, C., & Liao, T. F. (2013). How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *62*, 309–369.

Hruschka, E., Campello, R., Freitas, A., & de Carvalho, A. (2009). A survey of evolutionary algorithms for clustering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *39*, 133 –155. doi:10.1109/TSMCC.2008.2007252.

Iorio, C., Frasso, G., DAmbrosio, A., & Siciliano, R. (2016). Parsimonious time series clustering using p-splines. *Expert Systems with Applications*, *52*, 26–38.

Kaufman, L., & Rousseeuw, P. (1987). *Clustering by means of medoids*.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis* volume 344. John Wiley & Sons.

Kendall, M. G., & Smith, B. B. (1939). The problem of m rankings. *The annals of mathematical statistics*, *10*, 275–287.

Larose, D. T. (2005). *Discovering Knowledge in Data*. John Wiley & Sons.

Lavrač, N. (1999). Selected techniques for data mining in medicine. *Artificial intelligence in medicine*, *16*, 3–23.

Lemaire, T., Alami, R., & Lacroix, S. (2004). A distributed tasks allocation scheme in multi-uav context. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on* (pp. 3622–3627). IEEE volume 4.

Liao, L., Patterson, D. J., Fox, D., & Kautz, H. (2007). Learning and inferring transportation routines. *Artificial Intelligence*, *171*, 311–331.

Liao, T. W. (2005). Clustering of time series dataa survey. *Pattern recognition*, *38*, 1857–1874.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, *5*, 1–167.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*, 395–416. doi:10.1007/s11222-007-9033-z.

Macqueen, J. B. (1967). Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–297).

McCarley, J. S., & Wickens, C. D. (2004). Human factors concerns in uav flight. *Journal of Aviation Human Factors*, .

McKinley, R. A., McIntire, L. K., & Funke, M. A. (2011). Operator selection for unmanned aerial systems: comparing video game players and pilots. *Aviation, space, and environmental medicine*, *82*, 635–642.

Menéndez, H., Bello-Orgaz, G., & Camacho, D. (2013). Extracting behavioural models from 2010 fifa world cup. *Journal of Systems Science and Complexity*, *26*, 43–61.

Menéndez, H. D., Vindel, R., & Camacho, D. (2014). Combining time series and clustering to extract gamer profile evolution. In *Computational Collective Intelligence. Technologies and Applications* (pp. 262–271). Springer.

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*, 159–179.

Montero, P., & Vilar, J. A. (2014). Tsclust: An r package for time series clustering. *Journal of*, .

Navarro, J. F., Frenk, C. S., & White, S. D. (1997). A universal density profile from hierarchical clustering. *The Astrophysical Journal*, *490*, 493.

Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc* (pp. 1320–1326). volume 10.

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (pp. 115–124). Association for Computational Linguistics.

Penrose, L. S. (1952). Distance, size and shape. *Annals of Eugenics*, *17*, 337–343.

Pereira, E., Bencatel, R., Correia, J., Félix, L., Gonçalves, G., Morgado, J., & Sousa, J. (2009). Unmanned air vehicles for coastal and environmental research. *Journal of Coastal Research*, (pp. 1557–1561).

Piciarelli, C., Micheloni, C., & Foresti, G. L. (2008). Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for video Technology*, *18*, 1544–1554.

Portnoy, L., Eskin, E., & Stolfo, S. (2001). Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001*. Citeseer.

Remøy, H. T. (2010). *Out of office: a study on the cause of office vacancy and transformation as a means to cope and prevent*. IOS Press.

Rodríguez-Fernández, V., Gonzalez-Pardo, A., & Camacho, D. (2015a). Modeling the behavior of unskilled users in a multi-uav simulation environment. In *Intelligent Data Engineering and Automated Learning–IDEAL 2015* (pp. 441–448). Springer.

Rodríguez-Fernández, V., Menéndez, H. D., & Camacho, D. (2015b). Automatic profile generation for uav operators using a simulation-based training environment. *Progress in Artificial Intelligence*, *(In press)*.

Rodriguez-Fernandez, V., Menendez, H. D., & Camacho, D. (2015). Design and development of a lightweight multi-uav simulator. In *Cybernetics (CYBCONF), 2015 IEEE 2nd International Conference on* (pp. 255–260). IEEE.

Rodríguez-Fernández, V., Menéndez, H. D., & Camacho, D. (2015). User profile analysis for uav operators in a simulation environment. In *Computational Collective Intelligence* (pp. 338–347). Springer.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, *1*, 27–64.

Yang, J., & Leskovec, J. (2011). Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining* (pp. 177–186). ACM.