DOI: 10.1109/TITS.2018.2849266

Behaviour Monitoring Using Learning Techniques and Regular-Expressions Based Pattern Matching

Hyo-Sang Shin, Dario Turchi, Shaoming He and Antonios Tsourdos

***Abstract*—This paper addresses the problem of manoeuvre recognition and behaviour anomaly detection for generic targets by means of pattern matching techniques. The problem analysis is performed making specific reference to moving vehicles in a multi-lane road scenario, but the proposed technique can be easily extended to significantly different monitoring contexts. The potential extensions include, but are not limited to, public surveillance in train station or airport, road incidents and relative precursors detection and vehicle trajectories monitoring. The overall proposed solution consists of a trajectory analysis tool and a string-matching method. This allows integration of two different approaches, to detect both a-priori defined patterns of interest and generic manoeuvre/behaviour standing out from those regularly exhibited. This paper develops a new string matching method based on Regular Expressions. For generating reference patterns, a technique for the automatic definition of a dictionary of regular expressions matching the commonly observed target manoeuvres is presented. The advantages of the proposed approach are extensively analysed and tested by means of numerical simulations and experiments.**

I. INTRODUCTION

Airborne surveillance and monitoring systems have drawn increasing attention within the field of aerospace and robotics, thanks to dramatic improvements on unmanned aerial systems capabilities and associated sensing technology [1]. There have been numerous studies undertaken to develop such a system for many different application fields. Those include, but are not limited to, perimeters patrol [2], contaminant clouds extension estimation [3] and forest fires monitoring [4]. An application that has received an extremely wide consideration is airborne traffic monitoring [5], [6] using *Unmanned Aerial Vehicles (UAVs)*. Since UAVs can cover large/unfixed area and extend their monitoring functions to off-road zone, they could enable better and more flexible coverage, compared with traditional camera-based systems.

A specific application under this frame is the airborne monitoring of ground targets, which aims to detect and identify peculiar behaviours exhibiting hidden threats [7]. The current solution widely practised consists in two parts: UAVs implement tracking functionalities, and human operators assess and classify target behaviours. The issue with this current solution is that it requires highly skilled human operators and their workload could exponentially increase as the data collected from UAVs increases. Therefore, the current solution appears to be expensive and unsustainable.

It would be beneficial to reduce human operators' workload by providing them a filtered and summarising picture of the monitored scenario. This could be possible if UAVs signal human operators the level and nature of the associated target, and send the vehicle information when required. Hence, this paper aims to develop an efficient solution for automated airborne behaviour monitoring of ground targets. The main objective of the behaviour monitoring is to detect and identify 'possible threats' or, more generally, 'misbehaving targets', which will be indicated as *Targets Of Interest (TOIs)* in the paper.

The TOIs identification task can be framed within the more generic problem of '*anomaly detection*', which is a relevant research field extensively analysed over the last century. In its most general definition, an anomaly is a single data point or a points pattern that does not comply with a given definition of 'normality'; this is related to pieces of knowledge provided from experts in the field or inferred by a set of data constituting the available information.

There have been mainly two approaches that have been considered in literature for tackling the anomaly detection problem: detection of specific behaviour patterns of interest exhibited by the target; recognition of behaviours that do not comply with what is considered usual. The former approach is usually adopted in the cases where the monitoring action is focused on identifying a few well-known behaviours of interest or when a comprehensive description of all the unacceptable manoeuvres is provided from domain experts. With reference to this approach, techniques widely adopted in literature include probabilistic frameworks ([8], [9], [10]), fuzzy systems ([11]) and pattern matching ([7]). The issue with this approach is that it requires a prior knowledge of behaviours of interests or involvement of human experts in defining descriptions of unacceptable manoeuvres.

When no a priori specific knowledge about the monitored events is available, learning techniques are typically implemented to allow extraction of behaviour models from previous observations. Note that this approach is the main stream of current researches due to the large data-availability provided by modern computer systems. In this approach, any pattern that stands out with respect to other data-points is considered as anomaly, paying however the necessary attention in discriminating anomalies from novelties in the observed behaviours (referred as problem of '*novelty detection*', [12], [13], [14]). Various approaches involving different techniques, usually based on learning processes, have been used for discriminating anomalies from regular data points: classification of patterns by means of *neural networks* ([15], [16]), *Bayesian*

Hyo-Sang Shin, Dario Turchi, Shaoming He and Antonios Tsourdos are with Institute of Aerospace Sciences, STAM, Cranfield University, Cranfield, UK. e-mail: [h.shin, dario.turchi, Shaoming.He, a.tsourdos]@cranfield.ac.uk.

networks ([17], [18]), *SVM* ([19]) or *rule-based systems* ([20]), clustering of data for outliers identification ([21], [22]), distance or density analysis respect to *nearest neighbour* ([23]), statistical approaches leveraging parametric models (*Gaussian regression models*, [24], [25]) or *Kernel Functions* ([26]), information-theoretic techniques based on *entropy* ([27]) or *Kologomorov complexity* ([28]), spectral analysis performed, e.g., by means of *Principal Components Analysis* (PCA, [29]) or *wavelet transform* ([15]). There have been also interesting studies exploiting the recent developments in *Convolutional Neural Networks (CNNs)* and *Recurrent Neural Networks (RNNs)* for the anomaly detection, directly based on the images [30], [31], [32], [33]. A comprehensive survey on the anomaly detection problem is provided by Chandola et al. in [34], while Bolton et al. in [35] and Patcha et al. in [36] present surveys focused on fraud detection and intrusion detection, respectively.

As reference to traffic monitoring, *Knowledge-Based (KB)* pattern matching was proposed in [7], [37], [38]. This approach tries to identify suspicious targets by matching the driving behaviours with pre-defined suspicious patterns. Clearly, this approach requires domain experts to analyse and define prior knowledge of behaviours of interest, which is expensive and unsustainable under deluge of data and information. To reduce the workload of human operators, *Learning-Based (LB)* approaches, e.g., *Gaussian regression models* [24], [39], *unsupervised clustering* [22], *multi-feature clustering* [21], *nonparametric bayesian learning [18], were widely utilised in traffic monitoring in recent years.* The issue with the LB approaches is that it is hard to identify any behaviour of interest, which is not exhibited in the data. If some behaviours are not present in the data, it implies that those behaviours are most likely different from other data-points. In a logical sense, such a behaviour should be considered as anomaly. Moreover, although a certain type of behaviours is often presented from the data, it could be still considered as abnormal to human experts. However, such interpretation and identification of behaviours are not possible in the LB approaches.

To this end, this paper proposes a new TOIs identification approach that integrates the a prior knowledge based on learning based approaches to leverage the advantages of both approaches. The proposed algorithm relies on a novel pattern matching technique by means of regular expressions. As a preliminary step, we generate behaviour features from a manoeuvre detection using speed and curvature analysis of target trajectory using differential geometry techniques ([7], [40]). The behaviour features generated are stored as a sequence of discrete values. These results are then matched with reference patterns, representing behaviours of interest. The proposed integrated approach generates the reference patterns by specific behaviour patterns of interest or sets of patterns representing what is considered “normal” to various extents. This paper proposes to retrieve the latter patterns by applying automated learning. The learning objective is to infer, from observed sequences of target motion, a set of strings taking the form of regular expressions describing typical ‘legitimate’ behaviours; these then form a ‘dictionary’ of behaviour patterns, together with a measure of frequency of occurrence (‘regularity’) of target behaviours. The pattern matching algorithm then finds which pattern in the dictionary best matches to the exhibited behaviour and provides its regularity level. When a pattern observed during the monitoring process does not match any of the expressions in the dictionary, the relevant target is considered as TOIs, since it is exhibiting some unexpected behaviour which should reasonably raise the alert level. If domain experts consider a certain pattern, regularly exhibited by the target, abnormal, then they can simply place the pattern to an irregular position in the dictionary.

II. Overall Framework

The overall framework of the proposed approach is outlined in Fig. 1.

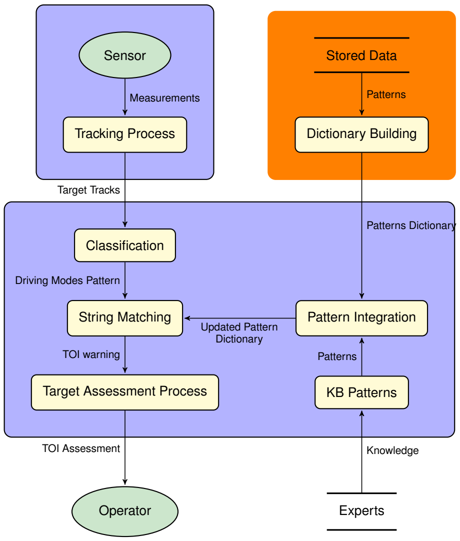


Fig. 1. Schematic representation of the proposed approach

Each enabling block and their inter-connection shown in Fig. 1 can be briefly described as follows. First, target tracks are processed based on measurements obtained from sensors such as vision camera, radar and LIDAR aboard monitoring platforms. Then, classification is performed using the tracks to obtain a string of target behaviour features, that is, *Driving Modes (DMs)*, in the form of *Regular Expression (regex)*. In this paper, track classification is done by trajectory analysis taking the monitoring scenario into account, but it could be done by other means depending on monitoring context. Next, the string of the behaviour features exhibited by the target is compared with reference strings by using pattern matching. This paper applies a string matching technique for the pattern matching. As discussed in Introduction, the reference patterns can be generated by two different, i.e. learning based and knowledge based, approaches. The learning based approach generates a set of reference patterns and ‘regularity’ by applying learning, typically performed off-line, on training data available. Here, regularity indicates occurrence frequency of reference patterns. The knowledge approach leverages the knowledge of domain experts for defining reference patterns that have to be spotted. The KB references can be not only generated off-line, but also updated and integrated to the LB references online during the online assessment if the experts consider necessary. When generating KB reference patterns, its regularity also needs to be produced. Note that depending on the availability and necessity, the reference patterns generated by the two approaches can be used individually or together after integration. The results obtained by pattern (string) matching are then forwarded to the next processing block indicated as 'Target Assessment Process': its role is to apply some kind of filtering action to the automatic *Target Of Interest Warnings (TOIWs)*, accordingly with the operator specification and requests. Details of each building blocks in the TOI assessment block will be given in the subsequent sections.

Note that on-line components are depicted in violet, while the off-line part is in orange in Fig. 1. Since the experts input can be given off-line and also online, the corresponding block, i.e. block Experts, is depicted neither in violet, nor in orange.

III. Trajectory Classification

In the proposed approach, the first step necessary to achieve behaviour recognition is the classification of behaviour features exhibited by the target. The purpose of this classification is to translate general behaviour features into a set of predefined motion categories; these are referred in the following as DMs. In behaviour monitoring of ground targets, one of the most intuitive features would be DMs. Therefore, this paper selected DMs as behaviour features. However, this doesn't constrain the selection of other types of behaviour features. Classification of DMs based on trajectory analysis is only a way to realise the approach proposed in this paper. Any features could be utilised if they present target behaviour features well. The identification of motion categories can be done by identifying and analysing a few, simple quantities (e.g., position, velocity, heading rate, etc.) capable of characterising the trajectories and using this information for selecting the most suitable DM.

Two main steps can thus be identified within the trajectory classification process:

- *Curvature Analysis* step allows to extract from the target track some quantities of interest: trajectory curvature, speed and acceleration;
- *Manoeuvre Classification* step produces a classification of the target trajectory within a predetermined set of DMs driving modes, on the basis of the quantities defined in the previous step.

A. Curvature Analysis

The proposed approach to curvature analysis is based on a moving-window-based trajectory approximation ([7]) and exploits a third-order polynomial function generating a trajectory with a virtually increased sampling frequency ([7], [24]). Given a window composed of N_T original samples taken at time instants[1]

$$[0, T_s, 2T_s, \ldots, (N_T - 1)T_s],$$

a new sample sequence is generated by means of a resampling process, taking place at instants

$$[0, T_n, 2T_n, \ldots, (N_T - 1)cT_n],$$

where the final instant of the two time sequences is the same, that is

$$(N_T - 1)cT_n = (N_T - 1)T_s. \tag{1}$$

In the previous description, T_s and T_n are the original[2] and "artificial" sampling times, and c is the ratio between the two times. In the following, the case is considered where $T_n < T_s$ ($c > 1$): this allows defining a sequence of data generated at 'virtually increased' sampling frequency. The resampling process has been performed with the purpose of applying the curvature analysis procedure to finer-grained, even if approximated, information, which leads to more effective identification of DMs. The polynomial function leveraged for the trajectory approximation has the following form:

$$p(x) = p_n x^n + p_{n-1}x^{n-1} + \ldots + p_1 x + p_0 \tag{2}$$

where n is the desired order, and the coefficients $p_0, \ldots, p_n$ are chosen so as to minimise the approximation error in a least squares sense. Such optimisation can be formulated as:

$$\tilde{P} = \underset{[p_0,\ldots,p_n]}{\operatorname{argmin}} \sum_{k=T-m+1}^{T} \left(p\left(k\right) - x(k)\right)^2 \tag{3}$$

where $\tilde{P} = [\tilde{p}_0, \ldots, \tilde{p}_n]$ is the vector of optimal coefficients for the polynomial, T is the current time step and $X = [x(1), \ldots, x(m)]$ is the history of the target position (single coordinate) within the considered time window of dimension m. Note that n is a design parameter that should be decided by considering the complexity of the pattern and computational efficiency. With this approach two polynomial functions are defined for approximating the target trajectory in the two coordinates of motion[3], referred as $p_x(t)$ and $p_y(t)$. The target position at discrete time instants within the time window $[T - N_T T_s, T]$ is then obtained by sampling from the two polynomial approximating functions. This process is performed at a "higher rate" with respect to the sensing frequency, that is with time step T_s instead of the original value T_n; in the following, indices k and i make reference to original and virtual samples, respectively, within the considered time window. After this, the velocity $(\dot{x}(i), \dot{y}(i))$ and acceleration $(\ddot{x}(i), \ddot{y}(i))$ profiles relative to the two coordinates of motion are derived by simple differentiation of the re-sampled position and velocity profiles. These are then used to calculate the forward acceleration $a_f(i)$, the orientation rate of change $\theta(i)$ and the minimum speed U of the vehicle for each i:

[1] Time instants are expressed as offset respect to the window initial time.

[2] It could be the time step of a tracking filter or simply the sampling time of a discrete measurement process

[3] Since our research is focused on ground vehicles, a bi-dimensional motion is assumed, but the approach proposed can be easily extended to the case of 3D trajectories.

$$U = \min v(i) = \min \sqrt{\dot{x}(i)^2 + \dot{y}(i)^2} \tag{4}$$

$$\theta(i) = v(i)\kappa(i)$$
$$= \sqrt{\dot{x}(i)^2 + \dot{y}(i)^2}\frac{\dot{x}(i)d\dot{y}(i) - \dot{y}(i)\ddot{y}(i)}{(\dot{x}(i)^2 + \dot{y}(i)^2)^{3/2}} \tag{5}$$

$$a_f(i) = \ddot{x}(i)\cos\psi(i) + \ddot{y}(i)\sin\psi(i) \tag{6}$$

where κ is the curvature, and $\psi = \tan^{-1}(\dot{x}/\dot{y})$ is the heading angle from North.

B. Manoeuvre Classification

Given the quantities defined in Section III-A, DMs are determined by checking how those quantities match with the conditions defined in each DM within a single time-window. This paper selected DMs after examination of the DMs from previous studies, especially [41], [7]:

- **Stopping(0)** : this state is detected when $U < U_{th}$, indicating that the target is stationary, stopping or moving;
- **Left Turn (1)**: recognised when $\max(\theta) > \theta_{\text{th},1} > 0$ and $\max(\theta)\min(\theta) > 0$.
- **Right Turn (8)**: $\min(\theta) < -\theta_{\text{th},1}$ and $\max(\theta)\min(\theta) > 0$;
- **Left Lane Change (2)**: this kind of manoeuvre is detected when

  $$\begin{cases} \max(\theta)\min(\theta) < 0 \\ \max|\theta| > \theta_{\text{th},1} \\ \theta(0) > 0 \end{cases}$$

  The difference with respect to a 'Left Turn' consists in the detection of a sign change for the orientation rate of change;
- **Right Lane Change (7)**: characterised by

  $$\begin{cases} \max(\theta)\min(\theta) < 0 \\ \max|\theta| > \theta_{\text{th},1} \\ \theta(0) < 0 \end{cases}$$

  The inspection of the sign change of θ is used to distinguish a lane change from a pure 'Right Turn' manoeuvre;
- **Straight (9)**: in this mode the monitored target is moving straight at constant speed, i.e. $\max(|\theta|) < \theta_{\text{th},1}$ and $\max(|a_f|) < a_{th}$;
- **Closing Gap (6)**: characterised by $a_f(0) > a_{th}$ and $\max(a_f)\min(a_f) < 0$. When the driver wants to close gap with the preceding vehicle, the monitored target will exhibit positive and negative accelerations at the beginning and in the end of the considered time-window respectively;
- **Widening Gap (3)**: $a_f(0) < -a_{th}$ and $\max(a_f)\min(a_f) < 0$. Considerations about this driving mode are similar to those for the 'Closing Gap' case, with the difference that the initial acceleration is negative over the whole time window;
- **Accelerating Ahead (5)**: this driving mode is recognised when $\max(a_f)\min(a_f) > 0$ and $a_f(0) > a_{th}$. The sign of acceleration is positive during the whole time-window;
- **Decelerating Ahead (4)**: in this case $\max(a_f)\min(a_f) > 0$ and $a_f(0) < -a_{th}$, i.e. the sign of acceleration stays negative;
- **U-Turning (A)**: detected when $\max|\theta| > \theta_{\text{th},2}$. A U-turn manoeuvre, where the vehicle inverts the direction of motion through a 180° rotation, can indeed be associated to large values for the orientation change.

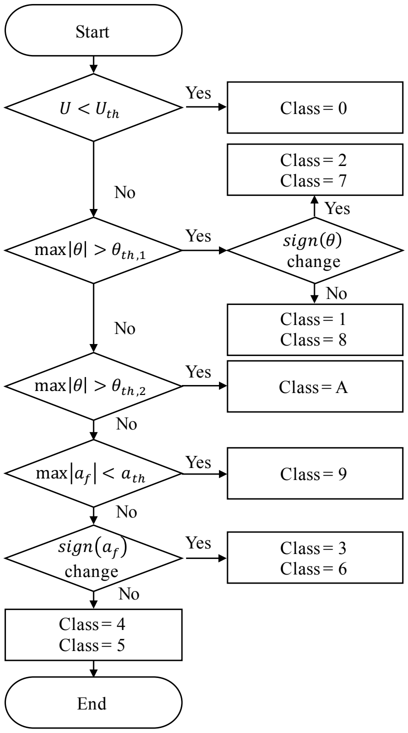


Fig. 2. Flow chart of the DM classification

In the previous classification, the quantities U_{th}, $\theta_{\text{th},1}$,$\theta_{\text{th},2}$ and a_{th} are threshold values that need to be tuned on the basis of the estimation accuracy and expected target dynamics.

Fig. 2 represents an overall flow chart of the classification of DMs

C. Test Patterns Definition

The classification introduced in Section III-B allows to define at each time step k a driving mode m_k. This is chosen among the set of pre-defined modes $M = [0, 1, \ldots, 9, A]$, which describes the behaviour of the monitored target during the considered time-window. These modes can then be leveraged for defining classes of complex behaviours that could draw the attention on the monitored target.

In the proposed approach, complex behaviours are defined as specific strings of DMs, and the behaviour detection relies on string matching techniques. More specifically, given the modes set M, a symbolic time-series of DMs $y_k = \{m_j \in M | j = k - N_w + 1, k - N_w + 2, \ldots, k\}$ is generated at each time-step k, where N_w is the adopted window dimension for behaviour detection. This last quantity depends on the expected duration of the events that have to be detected: assuming to be interested in events that last no longer than

t_w, N_w can be calculated as follows:

$$N_w = \left\lfloor \frac{t_w}{T_n} \right\rfloor - (N_T - 1) \tag{7}$$

IV. String Pattern Matching

The list of DMs extracted from the target trajectory is then compared with pre-defined strings associated to peculiar behaviours that need to be detected, referred in the following as *reference patterns*. Such comparison process is indicated as *pattern* (or *string*) *matching*, and it is implemented employing both basic strings of characters and regexes in the proposed approach. These two techniques are thoroughly described and analysed in Sections IV-A and IV-B, respectively.

A. Edit Distance Based Matching

The concept of ‘edit distance’ has been introduced as a generalised measure of the difference between the reference and an observed pattern: this allows defining a discrete level of similarity between patterns instead of a binary matching value. In [42] the edit distance between two strings S_r and S_t (reference and tested strings respectively) is defined as the minimum total number of changes C, insertions I and deletions R required to turn string S_t into S_r, that is:

$$D(S_r, S_t) = \min_j \left[C(j) + I(j) + R(j)\right] \tag{8}$$

where j runs over all possible combinations of symbol variations in order to obtain S_r from S_t. The main drawback of this simple approach is that the same relevance is given to missing, excess, wrong or misplaced terms within the analysed pattern.

The issue is that numerous patterns need to be recognised as a specific pattern of concern in practice. Setting a threshold on the edit distance values allows to recognise these slightly different patterns. However, at the same time, it does not consider properly the characterising peculiarities of each manoeuvre. Note that the concept of using string matching and edit distance with a threshold was already introduced in [7]. For more detail, the reader is referred to [7]. A possible solution to this problem could account for different coefficients for the components of the edit distance (C,R and I. However, it is important to note that weights-tuning is often a tricky process requiring deep insight into the considered problem. The previous considerations suggest that it is probably worthy to investigate for new, flexible approaches to reference patterns definition. To this end, this paper proposes to express reference patterns as regexes.

B. Regular-Expressions Based Matching

Expressing reference patterns as regexes allows defining a fixed binding structure for the behavioural patterns of interest, specifying mandatory and optional terms. With this approach, a reference pattern consists of any combination of the following components:

- **Specific Symbol, e.g. 1:** indicates a driving mode that must be present in the tested string;
- **Character Class, e.g. [65]:** gives multiple possibility, equally acceptable, for a term of the tested string (6 and 5 in the example);
- **Upper Unbounded Character Sequence, + :** one or more instances of the preceding element are allowed;
- **Unbounded Character Sequence, * :** zero or more of the preceding element are allowed;
- **Bounded Character Sequence, $\{n, m\}$:** a number between a minimum n and a maximum m of instances of the preceding element are matched.

In the cases where the reference string is expressed as regex, the matching result is a value representing an extension of the previously described edit distance:

- $\mathbf{D_r(S_r, S_t) = -1}$: the tested string does not match with the reference pattern
- $D_r(S_r, S_t) = \min_j I(j) \geq 0$: the edit distance indicates the number of insertion required for the test string to comply with the reference pattern;

With this approach, a possible reference pattern for an overtaking manoeuvre is:

$$S_{r,o} = \text{‘}\,[123456789] * 2 + [34569]\,\{4, 16\}7\text{’}$$

where the monitored vehicle moves to the overtaking (or *outer*) lane (‘2+’), maintains a straight direction for at least four time steps (regardless of the acceleration profile, ‘[34569] {4, 16}’) and then moves back to the right (or *inner*) lane (‘7’). The use of the unbounded sequence term (‘*’) allows decoupling the length of the reference and tested strings without affecting the edit distance. By adopting regexes, instead, the symbol ‘*’ can be used to reject the influence of driving modes detected before the manoeuvre of actual interest. The use of bounded and unbounded sequences appears to be particularly useful when the target dynamics is uncertain, and thus the length of the time interval needed by the target to exhibit the behavioural pattern of interest is not exactly known a priori. In the case of the overtaking manoeuvre, for example, the time interval between the two lane changes, and thus the number of characters between modes 7 and 2, depends on the relative velocity of the two vehicles involved. Consider the following three reference strings:

$$S_{r,o_1} = \text{‘255997’} \tag{9}$$

$$S_{r,o_2} = \text{‘}\,[13456789A] * 2 + [34569]\{4, 16\}7\text{’} \tag{10}$$

$$S_{r,o_3} = \text{‘}\,[13456789A] * 2 + [34569]\{4, \}7\text{’} \tag{11}$$

where S_{r,o_1} is expressed with the basic approach, while S_{r,o_2} and S_{r,o_3} are defined as regexes. S_{r,o_1} clearly represents a strict reference pattern: if it would be compared only with strings of the same length, the matching approach would not be robust with respect to little variations in the expected target dynamics since, for example, an overtaking that takes slightly longer, e.g. ‘25599997’, would not be recognised. This clearly shows how the proposed approach leads to greater flexibility in pattern recognition over the classic edit distance.

V. Approaches to Reference Patterns Definition

A. Knowledge-Based Reference Patterns

The simplest way to implement the pattern matching approaches is to define one or more patterns representing particular behaviours considered “concerning” or, more generally, “of interest”. In the traffic monitoring scenario, such behaviours could consist in any manoeuvre representing a danger for the regular traffic flow: e.g., to continuously switch from a lane to another. It is clear that this approach requires exploiting the knowledge of domain experts for defining the patterns of interest. On the one hand, a monitoring application designed for matching KB reference patterns is able to precisely identify particular behaviours of interest. On the other hand, this kind of approach cannot detect general unexpected manoeuvres that differ significantly from those regularly exhibited by the targets.

B. Learning-Based Reference Patterns

To overcome the limitations due to considering only KB reference patterns, this paper proposes to consider a new component: reference patterns defined on the basis of previously observed target behaviours.

1) Basic Concepts: The rationale behind the use of a LB component is that the detection of manoeuvres that are rarely observed, even if not ‘a-priori’ classified as “concerning”, should raise the attention level. This idea appears justified assuming that a sufficient large amount of target manoeuvres has been analysed, and thus that almost any legitimate manoeuvre has been observed a fair amount of time.

With reference to the string matching approach described in Section IV-B, the purpose of the learning component is to infer a set of regular expressions the observed sequences of driving modes. This set is referred as ‘*dictionary*’ in the following and is capable of summarising all the “frequent”, and thus assumed “regular”, target manoeuvres. When a pattern observed during the monitoring process does not match any of the expressions in the dictionary, the relative vehicle can be reasonably considered a TOI, since it has exhibited an unexpected behaviour that should reasonably raise the alert level.

2) Learning Technique: The objects of the proposed learning process are patterns of driving modes, that is sequences of alpha-numerical characters with no ordinal meaning. The proposed approach leverages learning approaches, especially in defining a dictionary, not competes with them. In this study, the learning accepts a sequence of driving modes of fixed length as input and produces a single value denoting to what extent the pattern can be assumed “regular”. Hence, any learning approach that can support this concept can be utilised. In this study, a *Neural Network (NN)* is applied for simplicity after considering various learning techniques. Note that there have also been some other interesting developments on reference driving pattern generation by Bayesian nonparametric approaches in recent years, for examples *Hierarchical Dirichlet Process-Hidden Markov Model (HDP-HMM)* [43], [44] and *Hierarchical Dirichlet Process-Hidden Semi Markov Model (HDP-HSMM)* [45]. With careful craft, those approaches can be also applied.

Implementing the learning process based on a NN enables defining groups of regular expressions constituting the regex dictionary. This NN is denoted as ‘*supporting neural network*’ in the following since its role is solely to support the creation of the dictionary.

3) Neural Network Implementation and Training: The structure of the neural network adopted is composed of three layers: the first one, the ‘input layer’, is made of n neurones, where n is the size of the input vectors. This layer has the sole purpose of ‘presenting’ the input data to the network and thus does not entail any calculation (passive nodes). The second layer, referred as ‘hidden layer’, is composed of n neurones as well, while the third, or ‘output’, layer has a single neuron connected to all those within the hidden layer.

In the proposed approach, the dataset used for training the neural network (*NN-training set*) entirely or partially consists of *synthetic data* and is composed as following:

- A set of “assumed regular” patterns, associated with output value 1. These can be obtained: i) by utilising a Markov Model, designed for producing sensible sequences of driving modes, ii) by logging patterns generated from a traffic simulator (described in Section VI-A2) or iii) by analysing real traffic data. Different sources of data have been considered for testing the effectiveness of the proposed methods under different conditions.
- A set of “irregular” sequences of driving modes, associated with the output value 0. These patterns are produced in an automated manner, by including within a random sequence of driving modes a sub-sequence representing a manoeuvre that is known to be concerning.

4) RegEx Dictionary Creation: Once the neural network has been trained to distinguish between “regular” and “suspicious” behaviours, it can be used to derive one or more sets of regular expressions that allow summarising and generalising the observed behavioural patterns. This can be done by providing a new set of patterns as input to the neural network. These patterns are denoted as *RD-training set* (where RD stands for *Regex Dictionary*) and are defined, following the same procedure used for the NN-training set. Considering that the network outputs lie in the $[0, 1]$ range, the RD-training patterns can be split among a pre-defined number of bins on the basis of the NN output associated with each of them:

$$\left\{ \begin{array}{l} \mathrm{nn}\left(p_i\right) > \gamma \\ \mathrm{nn}\left(p_i\right) \in \left[1-(j-1)\, s_b, \quad 1-j s_b\right) \end{array} \right. \Rightarrow p_i \in b_j \qquad (12)$$

where the ‘nn’ operator represents the neural network, p_i is the i-th pattern of the RD-training set, γ is the threshold on the NN outputs for the patterns to be considered within the regex creation process, $s_b = (1-\gamma)/N_b$ is the step between the values associated with two successive bins, N_b is the number of bins adopted and b_j denotes the j-th bins, with $j = 1, \ldots, N_b$.

The set of patterns associated with the j-th bin can thus be denoted as $p_{b_j} = \{p_i : p_i \in b_j\}$, and it can be used to define one or more regexes capable of expressing all the behaviours related to patterns p_{b_j}. This is done through a newly-developed algorithm that, given a set of patterns p_{b_j}, returns a set of regexes $r_{b_j} = \{r_{i,j}\}$, for $i = 1, .., n_{r_j}$ where n_{r_j} is the number or regexes generated for the j-th bin. Accordingly, a dictionary of regular expressions is built from the RD-training patterns and the output values provided by the supporting NN. Each of the N_b bins constituting the dictionary is associated with a different level of "regularity" assumed for the patterns matched by the regular expressions contained in the bin.

Given a regex dictionary d, the assessment process for a generic pattern p is simply to find the first bin from the top (index 1) containing a regex that matches the pattern. Assuming k as the result of the search, the assessment for the considered pattern can be formulated as follow:

$$\mathrm{l}(k) = \mathrm{l}(\mathrm{srch}(\mathrm{p}, \mathrm{d})) = 1 - (k-1)\, s_b \tag{13}$$

where the function $\mathrm{srch}()$ returns the index of the first bin containing a regex that matches the input pattern.

Novelty of the on-line monitoring approach developed consists in querying a dictionary of regexes as if it were the supporting NN[4]. The dictionary is created 'offline' on the basis of the NN outcomes and is characterised by a granularity that is a function of the number of bins N_b and the threshold γ adopted. This approach can be adopted under the assumption that

$$\mathrm{nn}(\mathrm{p}) \simeq \mathrm{l}(\mathrm{srch}(\mathrm{p}, \mathrm{d}))$$

This assumption implies that the dictionary-query technique represents a valid approximation of the neural network. Validity of the assumption is investigated through a comprehensive set of tests in Section VI-B.

5) Real-Time Dictionary Update: A relevant advantage of using a dictionary of regexes against a neural network is that the former can be easily updated in real-time. This means that the system can match patterns that are not accounted for during the system design phase, but appears to be of considerable importance during the operational phase. Therefore, the monitoring application can quickly adapt to different scenarios or temporary conditions influencing the targets behaviour.

For example, a prolonged slowdown on a motorway could be a priori considered as a suspicious manoeuvre, but if all the vehicles were doing that, e.g. because of a closed lane, the assessment for that pattern should be updated to "regular". Such result can be obtained by adding, during the monitoring process, new regular expressions to the dictionary or by moving an existing regex to a different bin in the dictionary hierarchy.

6) Assessment Framework structure: Given the two proposed approaches to TOI detection (KB and LB), an hybrid assessment framework for supporting operator decisions can be considered, whose structure is outlined in Fig. 1. It is possible to notice that the TOI warnings (TOIWs) produced by the pattern matching techniques are not directly forwarded to the operator but are provided to a further processing block indicated as 'Target Assessment Process': its role is to apply some kind of filtering action to the automatic TOIWs, accordingly with the operator specification and requests. From the practical point of view this component could:

- Implement the logic sum of the two input signal;
- Impose a threshold on the number of warnings necessary for classifying a vehicle as a TOI;
- Implement any arbitrarily complex policy.

The functionalities related to this component are not further analysed since they are strictly related to the operator choices and its personal interpretation of what represents an actual threat.

VI. Numeric Simulations

This section reports the results from an extensive set of simulations and tests performed for checking the effectiveness of the pattern matching approaches proposed in Section IV. KB and LB matching solutions have been tested separately, allowing to compare the proposed solutions with corresponding well-established techniques. The final target assessment step has not been taken into consideration in the following tests, since it entails a non-univocal interpretation of the problem and no labelled datasets or similar assessing techniques can be easily retrieved.

A. Knowledge-Based Matching

The two string-matching techniques described in the Section IV have been tested and compared for assessing whether the approach based on regexes performs better than the basic technique. Note that the basic technique is identical to the method proposed in [7]. For the simulation purpose, two different methods have been considered for generating the strings to be tested against the reference patterns. As first approach, a Markov chain has been leveraged, which produces sequences of DMs accordingly with some reasonable probabilities. Then, a dynamic system simulating a car moving on a two-lane highway has been implemented: in this case, the strings of driving modes have been produced through the analysis technique described in Section III.

1) Simulations Based on Markov Chain: The Markov chain described in Table II has been used for generating plausible sequences of driving modes: probability values are defined with a simple approach[5], taking into account some basic considerations:

- Manoeuvres time correlation suggests that the probability of remaining in the same state must be higher than switching to any other state;
- After the target has moved to the outer lane, it will not reduce its speed immediately;
- After the target has moved back to the inner lane, it will not reduce its speed immediately;

Lastly, notice that when in states $4, 5$ or 9 the probabilities of switching to modes 7 and 2 have been aggregated, since they depend on the lane occupied, i.e.:

$$\text{lane} = \text{left} \implies \begin{cases} \mathcal{P}(2) = 0 \\ \mathcal{P}(7) = P \end{cases}, \text{ otherwise} \begin{cases} \mathcal{P}(2) = P \\ \mathcal{P}(7) = 0 \end{cases}$$

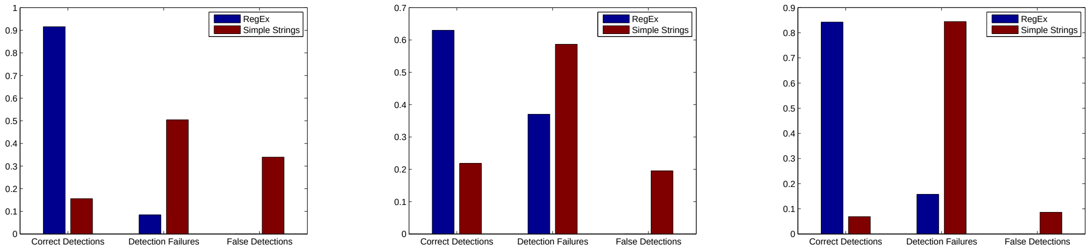
where P is the probability reported in the table. A sequence

[4] The NN used for the dictionary creation.

[5] Driving modes 'widening gap', 'closing gap' and 'U-turn', codified as '3','6' and 'A' respectively, are not considered in the following for simplicity.

TABLE I
NUMERICAL RESULT FOR SIMULATIONS BASED ON MARKOV CHAIN AND DYNAMIC MODEL (%)

	Markov Chain				Dynamic Model	
	Consistent Knowledge		Inaccurate Knowledge		regex	Basic
	regex	Basic	regex	Basic		
Correct Detections	90.5	16	63.8	22	84.4	15.4
Missed Detections	9.5	49.5	36.2	58	15.6	84.6
False Detections	0	34.5	0	20	0	0



(a) (b) (c)

Fig. 3. Numeric results for regex and simple-strings based approaches: 3(a) "consistent" case simulation, 3(b) "inaccurate" case simulation, 3(c), dynamic system simulation

TABLE II
MARKOV CHAIN FOR PATTERN GENERATION, KB APPROACH

		Driving Modes				
		(2)	(7)	(5)	(4)	(9)
Driving Modes	Left Lane Change (2)	1/2	1/6	1/6	0	1/6
	Right Change (7)	1/6	1/2	1/6	0	1/6
	Acceleration Ahead (5)	1/6		1/2	1/6	1/6
	Deceleration Ahead (4)	1/6		0	2/6	2/6
	Constant Speed (9)	2/6		1/6	1/6	1/6

of one hundred thousand driving modes has been produced and analysed leveraging the two string-matching approaches, with the aim of identifying overtaking manoeuvres. The considered reference pattern for the simple-string approach [7] is $S_{r,o1}$ = '255997', assuming a threshold value of two for the edit distance. This means that only strings with edit distance smaller or equal to two have been considered matching the reference pattern. For the regex-based approach, the regular expression adopted for describing an overtaking manoeuvre is S_{r,o_2} = '$[13456789]*2+[345689]\{4,16\}7$', assuming a zero threshold on the edit distance. This is because the use of metacharacters and character classes already provides a sufficient degree of flexibility. The *Ground Truth (GT)* about the number of overtaking manoeuvres has been defined in two different cases:

Consistent Knowledge: just the sub-sequences with at least four "straight-direction" DMs between the left-lane-change and right-lane-change modes have been counted as actual overtaking manoeuvres. The term "consistent" is used since the assessment of the ground truth complies the definition of the reference patterns S_{r,o_1} and S_{r,o_2}. With this approach, the DMs sequence exhibits 3412 overtaking manoeuvres.

Inaccurate Knowledge: sub-sequences with two or more "straight-direction" DMs between the two lane changes have been considered as actual overtaking manoeuvres. In this case, a non-exact knowledge of target dynamics is simulated since overtaking manoeuvres are executed much faster than expected (the reference patterns account for four straight-direction driving modes). With this approach, 6253 overtaking manoeuvres have been counted.

The results for the two cases are shown in Fig. 3(a) and 3(b), and summarised in Table I: in the consistent case the correct detection rates are 90% and 16% for the regex-based and simple-string approaches, respectively. In the case of inaccurate knowledge the regex-based technique lowered its success ratio to 63.8% and the performance of the simple-string approach remains almost the same. False positive detections are not a problem concerning the regex-based technique (absent in both the cases), while in the simple-string case their fraction is not negligible: it is indeed of the same order of magnitude with respect to correct detections.

2) Simulations Based on Dynamic Model: A second test for the two string-matching techniques has been carried out using a dynamic system model representing a single car moving on a two-lane highway. An always-straight road is considered for simplicity reasons, but it can be readily extended to the more complex case of a generic road by making reference to differential geometry techniques ([40], [24]). The car trajectory has been simulated for fifty thousand seconds (corresponding to one hundred thousand driving modes, assuming a sample time $T_s = 0.5\,\text{s}$), on the basis of the following procedure:

I) The car moves along the inner lane for a randomly distributed time interval $t_{\text{in}} \sim \mathcal{N}(30, 16)$. The acceleration profile during t_{in} is defined dividing the interval into n_{in} randomly distributed[6] sub-intervals $t_{\text{in},j} \sim \mathcal{N}\left(2,\ 6.25 \times 10^{-2}\right)$ with $j = 1, 2, \ldots, n_{\text{in}}$. For each interval $t_{in,j}$ the vehicle can decide to accelerate, decelerate or maintain a constant velocity;

II) The car starts to move towards the outer lane with constant velocity;

III) The vehicle arrives on the outer lane and determines how long it will stay on this lane: $t_o \sim \mathcal{N}(5, 4)$. The acceleration profile during this period is defined using the same procedure as in step I;

IV) The car moves back to the lane on the right;

V) The procedure restarts from the first step.

Other simulation parameters are:

- Maximum forward speed: $30\,\mathrm{m/s}$;
- Minimum forward speed: $5\,\mathrm{m/s}$;
- Lateral speed: $\pm 1.75\,\mathrm{m/s}$;
- Acceleration: $\pm 2\,\mathrm{m/s}^2$

All the 'inner-to-outer lane' events followed by 'outer-to-inner lane' have been considered as overtaking manoeuvres; 1262 overtaking manoeuvres have been identified with this approach. The reference patterns for the two matching techniques are the same of the previous example (Section VI-A1), and the numeric results obtained from the simulation are reported in Table I and Fig. 3(c). Again, the regex-based approach undoubtedly outperforms the simple-string technique: 84% instead of the 15% of correct detections.

3) Computation Complexity Analysis: The time complexity for edit distance calculation is $\mathcal{O}(n \cdot m)$ ([46]) where n and m are the lengths of the compared strings. On the other hand, for classical regular expressions (the type here considered) the time complexity can be reduced to $\mathcal{O}(n)$, with n length of the tested string, by the use of a *Deterministic Finite Automata (DFA)*.

B. Learning-Based Matching

The learning approach has been tested under the same two cases considered in Section VI-A, and has been applied to real-world vehicle trajectories (made available by the NGSIM project). The objective of the simulations reported in the following is to establish whether or not the use of a RD represent a suitable solution for replacing a neural network, i.e., if the two methods have comparable outcomes. This analysis appears to be more meaningful than considering actual detection capabilities since: i) No GT is available for most[7] of the data ii) A neural network is assumed to be an effective tool for pattern recognition and thus its outputs are considered to be valid references.

1) Markov Model: Similarly to the KB case, a Markov model (Figure 4) has been defined for producing DM sequences associated with "regular" behaviours.

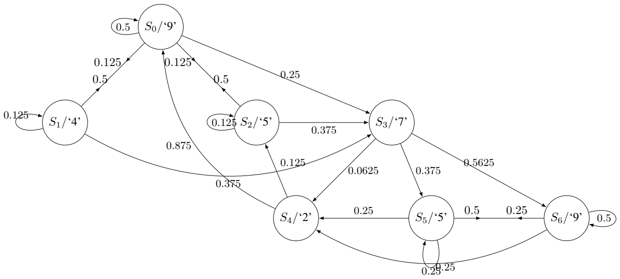


Fig. 4. Markov model for pattern generation, LB approach

However, unlike the KB case, "irregular" patterns are also required for properly training the neural network. Each irregular pattern is generated as a random sequence of DMs where a sub-sequence that represent a known irregular behaviour is added.

The following datasets have been produced on the basis of the criteria just described:

NN-training set: 20.000 patterns, used to train the neural network;

RD-training set: 20.000 patterns, used to query the trained network and populate the regex dictionary;

Test set: 2000 patterns, used to query both the neural network and the dictionary-based matching technique for validation and comparison.

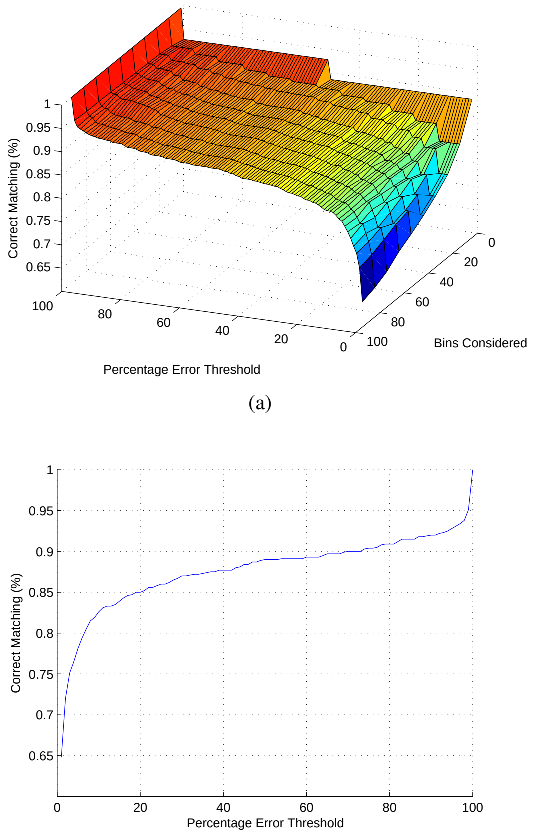
All the datasets are composed of regular patterns for the 75% and irregular patterns for the remaining part. This ratio for the two components of the dataset, clearly privileging the regular patterns, has been chosen considering that the simulation focus is in learning regular, while irregular patterns are provided to the network just for training purposes. Using these datasets, a test accounting for various numbers of bins for the dictionary-based technique have been performed, and the results are depicted in Fig. 5. X, Y, and Z axes in the figure represent the number of bins generated, error threshold, and correction matching ratio. The aim of the test consists in verifying for what percentage of the input patterns the regex and the NN approaches provide the "same" result. With "same", it is here meant that the difference in the outcomes of the two methods lies within a given error threshold. The higher the correction matching ratio is, the less the difference between the two approaches is. If the correct matching ratio is equal to 1, it implies that the outcome of the proposed approach well matches to the outcome of the NN approach. This interpretation is the same for Fig. 6.

The numeric results indicate that the proposed dictionary-query approach has been able to reproduce the NN outputs for a fair portion of the test patterns: approximately 85% of correct matching assuming an error threshold of 20%.

2) Dynamic Model: The test described in Section VI-B1 has also been performed using data from the dynamic simulator described in Section VI-A2 instead of patterns from the Markov model. The test results for this different case are depicted in Fig. 6. From a qualitative point of view, the results are not substantially different from those obtained for the Markov model data. The resulting curve (Figure 6(b)) exhibits a trend similar to that of Fig. 5(b), but it results in a considerably less pronounced slope. This could be due to the fact that the dynamic-model simulator has probably produced

[6] The length of the last sub-interval $t_{\text{in},n_{\text{in}}}$ is constrained by the length of interval t_{in}.

[7] Only patterns produced by means of the Markov model are associated with an actual GT.



(a)

(b)

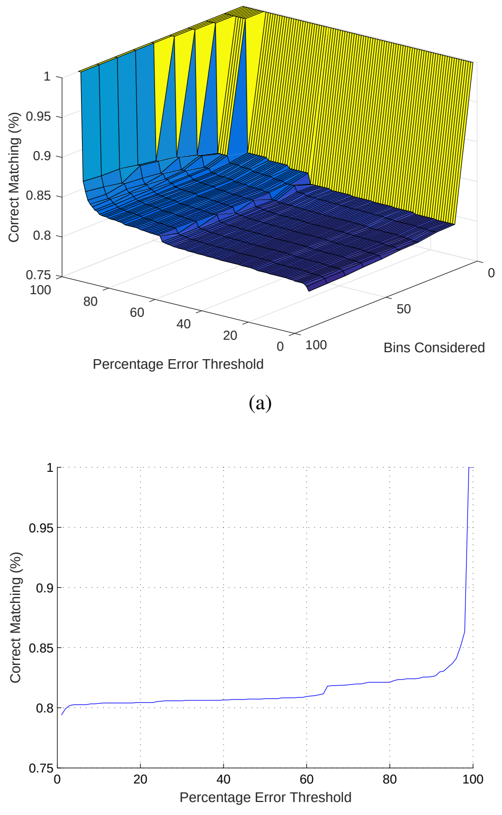
Fig. 5. Results for RD against neural network approaches confrontation, Markov-model data: 5(a) multiple bins test, 5(b) detail of the results for the highest number of bins considered (100)

a set of patterns with lower ‘entropy’, that is a smaller variety of DM sequences in training.

3) Real Data Analysis: For the purpose of testing the proposed approach in real-world conditions, the data collected for the *Federal Highway Administration (FHWA) Next Generation SIMulation (NGSIM)* project ([47], [48]) has been considered. This is available to researchers in the field of transportation and traffic flow theory and incorporates a collection of single-vehicle related data, including processed trajectories in global and road-local reference frames.

For the simulations, the original NGSIM data have been corrected by applying the technique described by Montanino et al. in [49] to allowing removal of outliers and infeasible acceleration profiles caused by measurement errors.

We have considered 5200 trajectories from the NGSIM US101 dataset, within the period of time between 7.50am-8.35am. On the basis of these target trajectories and by employing the analysis technique described in Section III, 3563 driving patterns have been identified. These have been then leveraged for the definition of a dictionary accounting for 30 regular expressions distributed among 100 bins. For this example test, the dictionary threshold γ has been set to 0, thus allowing to have a granularity of $1/100$ in the representation of the NN outputs by means of the RD. The test results are depicted in Fig. 7, where the ratio of correct matching between NN and RD outcomes are plotted against the error threshold (X-axis) and the training set portion[8]. Note that percentage values in the legend of Fig. 7 represent the percentage of data dedicated to the training process. The numeric results show that the proposed RD based approach provides the results almost 90% identical to the NN results when only 15% of the data is used in training. The correct matching ratio between the



(a)

(b)

Fig. 6. Results for RD against neural network approaches confrontation, dynamic model: 6(a) multiple bins test, 6(b) detail of the results for the highest number of bins considered (100)

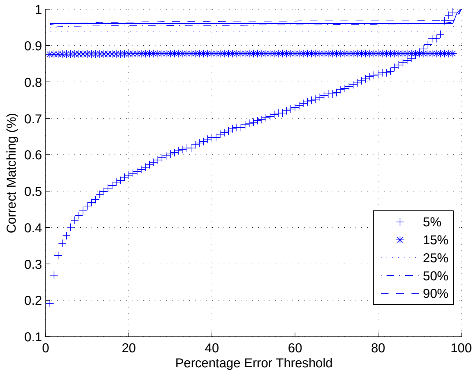


Fig. 7. Results for RegEx dictionary against neural network approaches confrontation

[8] Notice that, for the test here reported, the training set portion is equally distributed between the NN-training and RD-training sets

proposed approach and NN gets higher as more data is utilised in training. This clearly demonstrate even with a relatively small amount of data dedicated to the training process, the proposed RD based approach is capable of replicating the outcomes of the neural network. Furthermore, it is shown that in the case of extremely small training sets, the matching performance is not sensibly affected by the error threshold. This implies that for most of the patterns in the test set, the dictionary-query technique either provides a perfect match or does not recognise the pattern at all.

4) Computation Time Analysis: Let us now compare the computational efficiency for the dictionary-query and neural network techniques.

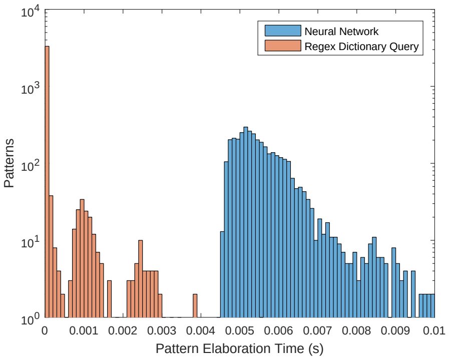


Fig. 8. Distribution of the computation times for the 3563 patterns obtained from NGSIM data

The test has considered the real-word data case (Section VI-B3), and the neural network and the regex dictionary have been defined on the basis of a training set portion equal to 0.5. Then, each pattern in the dataset has been assessed through both the considered techniques and the actual computation times[9] have been recorded. The neural network outputs have been calculated using the Matlab built-in *net()* function, while the dictionary-query approach relies on a newly-implemented function that browses the dictionary and checks for the first regex matching the pattern under assessment.

The measured time intervals only for pattern evaluation between the two approaches are reported in the histogram in Fig. 8. This figure clearly shows that the RD based technique is always faster than the neural network. Furthermore, it is shown that the use of a neural network leads to quite evenly distributed time values, while the dictionary-query has taken the minimum observed elaboration time for most of the patterns. This aspect can be attributed to the fact that the regular patterns are usually matched almost immediately by a regex located in the first few bins as they are most frequent. On the other hand, irregular patterns take longer since they require most or all of the dictionary to be browsed. Regarding the overall performance, however, this does not seem to be a problem, since irregular patterns are expected to be observed far less frequently than regular ones. The measured mean value and standard deviation for the elaboration times in the two cases are $\bar{t}_{NN} = 5.9\,\text{ms}$, $\sigma_{NN} = 1.5\,\text{ms}$ and $\bar{t}_{RD} = 0.12\,\text{ms}$, $\sigma_{RD} = 0.49\,\text{ms}$ for the neural network and RD approaches, respectively.

It is clear that the performance of the proposed approach depends on the number of bins defined. Moreover, the computational time clearly relies on the number of bins. The proposed approach can trade off computational complexity against performance by adjusting the design parameter, that is, the number of bins. On the other hand, it is hard, if not impossible, to achieve the trade-off in NN approaches. When the data becomes much bigger and more complicated, the pattern elaboration time could significantly increase. In this case, the advantage of having the trade-off capability could become more significant.

VII. Experiments

To validate the proposed approach, especially in the integrated form, this paper considers a military scenario and performs experiments. This section presents test results under the military scenario, with comparison to KB pattern matching and LB approaches. Note that the KB approach chosen for comparison is identical to the method in [37] and NN is selected for the LB approach as it is one of the most well-known learning techniques.

A. Indoor Test

We evaluate the proposed behaviour monitoring algorithm based on experiments in an indoor flight arena. The flight arena is equipped with a VICON motion capture system, which can be utilised to provide real-time six-degree-of-freedom pose parameters. Figure 9 shows the layout of the indoor flight arena.

Fig. 9. Indoor flight arena

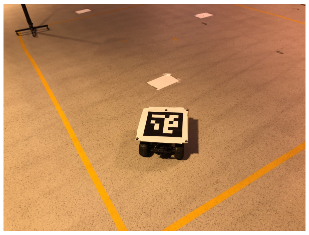


Fig. 10. Ground moving vehicle with marker

In the indoor experiment, one ground moving vehicle with a QR code marker attached on the top for target detection (shown in Fig. 10) is leveraged as the target to be tracked by a stationary GoPro camera. The ground moving vehicle is connected to the *Robot Operating System (ROS)* through WIFI mode. Waypoint following guidance commands are sent to the vehicle by ROS such that they can move in some specified trajectories for the purpose of behaviour monitoring test. A GoPro 5 camera is mounted on a stationary platform to provide a certain height for target tracking. The GoPro camera is set as video mode with a fixed frequency of 30Hz and is accessible by a Linux-based Nvidia TX2 single board computer, which is also connected with the ROS system.

[9] The reported time values make reference to a Matlab 2015 environment on an Intel i7-3770 16 GB RAM machine, with no parallel computation enabled.

In the scenario, a *Unmanned Ground Vehicle (UGV)* is assumed to move in a circular trajectory to monitor a military base which is suspicious. In the experiment scenario, the UGV also performs a deceleration and stop/slowly moving manoeuvre at around $7s$ for about $15s$ and stops at around $33s$ for about $35s$ and these behaviours are considered as abnormal. Therefore, the abnormal behaviour pattern can be defined as $S_r =' [123456789] * 4\{2, 10\}0\{2, 10\}'$. To implement KB pattern matching [37], the reference string is defined as $'999444440000'$. Note that it is difficult for the human expert or operator to provide an exact form of reference string under the given mission scenario. Therefore, we can expect that the KB approach chosen for comparison might not be able to identify suspicious target behaviours.

The specific trajectory exhibiting such a behaviour is generated by using waypoint-following guidance. The ground truth provided by VICON and estimated UGV trajectories for the considered scenario are shown in Fig. 11.

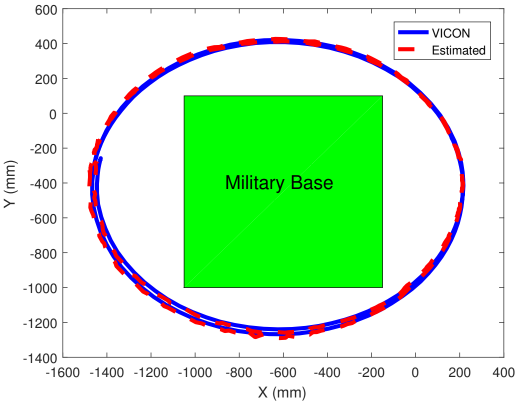


Fig. 11. Real and estimated UGV trajectories for the considered scenario.

To demonstrate the advantage of the integrated approach developed, it is assumed that the abnormal behaviour pattern is not exhibited by the target before and hence data containing this pattern is not available for NN training. This means that this specific string pattern cannot be generated in the dictionary. However, human operators considered such a behaviour is suspicious and thus inserted the pattern to a lower bin with a score close to 0 using the proposed approach. Fig. 12 shows the assessment results of pattern matching, NN and the proposed approach. As expected, the results show that KB pattern matching cannot identify the suspicious behaviour of the vehicle. This is because no string exhibited by the target matches to with the specific reference string, albeit the target behaviour can be considered as suspicious. The results also clearly demonstrate that the regularity level of the normal vehicle in the proposed monitoring approach is close to zero at around $7s$ and $33s$, but it is very close to 1 in the NN method. This confirms that the proposed monitoring approach successfully identifies anomaly exhibited by the target at around at around $7s$ and $33s$, whereas the NN approach cannot. To solve this problem in the NN approach, the NN has to be retrained for different scenarios and thus could become a case-by-case solution. Note that NN training is usually time consuming. Moreover, the running time of the proposed RD based approach is approximately 50% of that of the NN method.

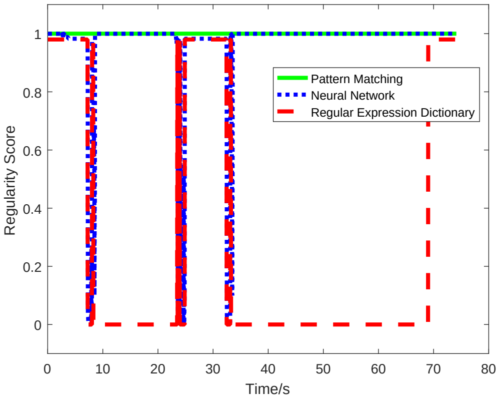


Fig. 12. UGV regularity level assessment (1 for normal and 0 for abnormal).

B. *Outdoor Test*

To further validate the proposed behaviour monitoring algorithm, we also performed outdoor experiments at the airport of Cranfield University.

In the outdoor experiment, one ground moving vehicle with remote control is leveraged as the target for test. The ground moving vehicle is connected to the ground ROS through WIFI mode such that we can get the real-time position data. The vehicle is manually controlled such that it can move in some specified trajectories for the purpose of behaviour monitoring test. The targets are monitored by two UAVs and tracked by GoPro cameras aboard the UAVs.

Like in the indoor experiment, a UGV moves in a circular trajectory, monitoring a military base. The UGV performs a 'deceleration and stop/slowly moving' manoeuvre at around 7s until 15s . We use the same neural network (without retraining) and regular expression dictionary, generated for the indoor test, for the regularity level assessment. The query results are depicted in Fig. 13. From the figure, it is clear that the proposed RD based approach can successfully identify the suspicious behaviour exhibited by UGV. On the other hand, the results show that pattern matching and neural network approaches cannot identify abnormal behaviour. The chattering at around 14s is due to the estimation accuracy of the target tracks. The results clearly prove that the proposed algorithm could be an effective way to assess the regularity level of ground moving vehicles.

VIII. Conclusion

This paper proposes a new behaviour monitoring approach that integrates two main anomaly detection trends, a priori KB approach and LB approach, to exploit advantages of both the approaches. In general sense, the proposed method consists of two main steps: classification of target behaviours in sequences of behaviour features and comparison of the behaviour feature sequences with reference patterns. As the

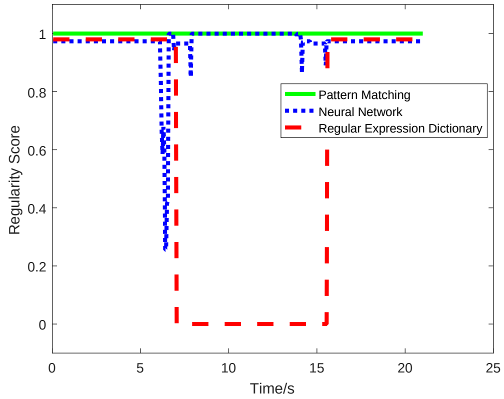


Fig. 13. UGV regularity level assessment (1 for normal and 0 for abnormal).

main monitoring context considered in this study is monitoring of ground vehicles, DMs are selected as behaviour features and trajectory analysis is performed for the classification. Integrating the two types of approaches, the reference patterns are either defined a priori on the basis of specific knowledge or automatically learnt by means of a supporting NN. The proposed method with each way of defining reference patterns was tested through numerical simulations and compared with well-established techniques in the field, leading to promising matching results. To demonstrate its potential, the proposed approach with the two types of reference patterns combined together was also tested through experiments. The experiment results clearly show that the proposed approach is an effective way of assessing the regularity level of target behaviours and could potentially overcome the limitations of a priori knowledge based on learning based approaches.

The main contributions of the work presented in this paper are: i) The application of a regex-based matching technique to the field of automated monitoring, specifically to the problem of behaviour classification, ii) Development of a procedure for the automatic derivation of a dictionary of regular expressions representing the behaviours commonly observed in monitoring context. The RD enables assessing to what extent an observed pattern can be considered regular and thus to detect uncommon and unexpected manoeuvres that should be considered as concerning or even possible threats.

It is worthy to remark that the flexibility and generality of the proposed approach: the proposed method can thus be easily applied to various monitoring scenarios by defining appropriate reference patterns and driving modes. Note that in a general context, the term ‘driving’ should be replaced by ‘moving’, thus fitting to other monitoring problems such as boat or person monitoring. Defining reference patterns and moving modes will depend on the specific application field (e.g., maritime, highways, urban roads) and on the behaviours of interest (e.g., fickle speed, suspicious stops, periodic behaviours).

Acknowledgment

This work has received funding from the European Union under the Pilot Project on Defence Research managed by the European Defence Agency under grant agreement No PP-15-INR-01 05 EuroSWARM (’Unmanned Heterogeneous Swarm of Sensor Platforms ’).

This work reflects only the authors’ views. The European Commission and European Defence Agency are not responsible for any use that may be made of the information it contains (Art. II.9 of the Model Grant Agreement)

References

[1] L. C. Trost, *Unmanned air vehicles (UAVs) for cooperative monitoring*. Sandia National Laboratories, 2000.

[2] A. Girard, A. Howell, and J. Hedrick, “Border patrol and surveillance missions using multiple unmanned air vehicles,” in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1, pp. 620–625 Vol.1, Dec 2004.

[3] B. A. White, A. Tsourdos, I. Ashokaraj, S. Subchan, and R. Zbikowski, “Contaminant cloud boundary monitoring using network of uav sensors,” *Sensors Journal, IEEE*, vol. 8, no. 10, pp. 1681–1692, 2008.

[4] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, “Cooperative forest fire surveillance using a team of small unmanned air vehicles,” *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.

[5] H. Oh, S. Kim, A. Tsourdos, and B. White, “Road-map assisted standoff tracking of moving ground vehicle using nonlinear model predictive control,” in *American Control Conference (ACC), 2012*, IEEE, 2012.

[6] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, “Survey of unmanned aerial vehicles (uavs) for traffic monitoring,” in *Handbook of unmanned aerial vehicles*, pp. 2643–2666, Springer, 2015.

[7] H. Oh, S. Kim, H.-S. Shin, A. Tsourdos, and B. A. White, “Behaviour recognition of ground vehicle using airborne monitoring of unmanned aerial vehicles,” *International Journal of Systems Science*, vol. 45, no. 12, 2014.

[8] P. Angkititrakul, R. Terashima, and T. Wakita, “On the use of stochastic driver behavior model in lane departure warning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 174–183, March 2011.

[9] A. Gerdes, “Automatic maneuver recognition in the automobile: the fusion of uncertain sensor values using bayesian models,” in *Proceedings of the 3rd International Workshop on Intelligent Transportation (WIT 2006)*, pp. 129–133, 2006.

[10] Y. Xuan and B. Coifman, “Lane change maneuver detection from probe vehicle dgps data,” in *2006 IEEE Intelligent Transportation Systems Conference*, pp. 624–629, Sept 2006.

[11] T. Hülnhagen, I. Dengler, A. Tamke, T. Dang, and G. Breuel, “Maneuver recognition using probabilistic finite-state machines and fuzzy logic,” in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 65–70, June 2010.

[12] M. Markou and S. Singh, “Novelty detection: a review—part 1: statistical approaches,” *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[13] M. Markou and S. Singh, “Novelty detection: a review—part 2:: neural network based approaches,” *Signal processing*, vol. 83, no. 12, pp. 2499–2521, 2003.

[14] C. M. Bishop, “Novelty detection and neural network validation,” in *Vision, Image and Signal Processing, IEE Proceedings-*, vol. 141, pp. 217–222, IET, 1994.

[15] S. Thajchayapong, E. S. Garcia-Trevino, and J. A. Barria, “Distributed classification of traffic anomalies using microscopic traffic variables,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 1, pp. 448–458, 2013.

[16] G. Lokman and G. Yilmaz, “A new method for anomaly detection and target recognition,” in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pp. 577–583, IEEE, 2014.

[17] W.-K. Wong, A. Moore, G. Cooper, and M. Wagner, “Bayesian network anomaly pattern detection for disease outbreaks,” in *ICML*, pp. 808–815, 2003.

[18] V. Bastani, L. Marcenaro, and C. S. Regazzoni, “Online nonparametric bayesian activity mining and analysis from surveillance video,” *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2089–2102, 2016.

[19] S.-J. Horng, M.-Y. Su, Y.-H. Chen, T.-W. Kao, R.-J. Chen, J.-L. Lai, and C. D. Perkasa, “A novel intrusion detection system based on hierarchical clustering and support vector machines,” *Expert systems with Applications*, vol. 38, no. 1, pp. 306–313, 2011.

[20] T. Fawcett and F. Provost, “Adaptive fraud detection,” *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 291–316, 1997.

[21] N. Anjum and A. Cavallaro, “Multifeature object trajectory clustering for video analysis,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1555–1564, 2008.

[22] H. Xu, Y. Zhou, W. Lin, and H. Zha, “Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4328–4336, 2015.

[23] R. Laxhammar and G. Falkman, “Online learning and sequential anomaly detection in trajectories,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1158–1173, 2014.

[24] H. Oh, H.-S. Shin, S. Kim, A. Tsourdos, and B. A. White, “Airborne behaviour monitoring using gaussian processes with map information,” *IET Radar, Sonar & Navigation*, vol. 7, no. 4, pp. 393–400, 2013.

[25] H. Li, A. Achim, and D. Bull, “Unsupervised video anomaly detection using feature clustering,” *IET signal processing*, vol. 6, no. 5, pp. 521–533, 2012.

[26] C. Oh, J.-S. Oh, and S. G. Ritchie, “Real-time hazardous traffic condition warning system: framework and evaluation,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 3, pp. 265–272, 2005.

[27] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, “An empirical evaluation of entropy-based traffic anomaly detection,” in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pp. 151–156, ACM, 2008.

[28] A. Wagner and B. Plattner, “Entropy based worm and anomaly detection in fast ip networks,” in *Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on*, pp. 172–177, IEEE, 2005.

[29] Q. Du and J. E. Fowler, “Hyperspectral image compression using jpeg2000 and principal component analysis,” *Geoscience and Remote Sensing Letters, IEEE*, vol. 4, no. 2, pp. 201–205, 2007.

[30] P. Christiansen, L. N. Nielsen, K. A. Steen, R. N. Jørgensen, and H. Karstoft, “Deepanomaly: combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field,” *Sensors*, vol. 16, no. 11, p. 1904, 2016.

[31] O. Gorokhov, M. Petrovskiy, and I. Mashechkin, “Convolutional neural networks for unsupervised anomaly detection in text data,” in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 500–507, Springer, 2017.

[32] W. Luo, W. Liu, and S. Gao, “A revisit of sparse coding based anomaly detection in stacked rnn framework,” *ICCV, Oct*, 2017.

[33] B. Meng, L. Zhang, F. Jin, L. Yang, H. Cheng, and Q. Wang, “Abnormal events detection using deep networks for video surveillance,” in *International Conference on Cognitive Systems and Signal Processing*, pp. 197–204, Springer, 2016.

[34] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[35] R. J. Bolton and D. J. Hand, “Statistical fraud detection: A review,” *Statistical science*, pp. 235–249, 2002.

[36] A. Patcha and J.-M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” *Computer networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[37] S. Kim, R. W. Żbikowski, A. Tsourdos, and B. A. White, “Airborne monitoring of ground traffic behaviour for hidden threat assessment,” in *Information Fusion (FUSION), 2010 13th Conference on*, pp. 1–8, IEEE, 2010.

[38] S. Kim, R. Żbikowski, A. Tsourdos, and B. A. White, “Behaviour recognition of ground vehicle for airborne monitoring by uav swarm,” *IFAC Proceedings Volumes*, vol. 43, no. 16, pp. 455–460, 2010.

[39] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, “Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5288–5301, 2015.

[40] B. O’neill, *Elementary differential geometry*. Academic press, 2006.

[41] C.-F. Lin and A. Ulsoy, “Calculation of the time to lane crossing and analysis of its frequency distribution,” in *American Control Conference, Proceedings of the 1995*, vol. 5, pp. 3571–3575, IEEE, 1995.

[42] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Elsevier Science, 2008.

[43] T. Taniguchi, S. Nagasaka, K. Hitomi, N. P. Chandrasiri, T. Bando, and K. Takenaka, “Sequence prediction of driving behavior using double articulation analyzer,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1300–1313, 2016.

[44] W. Wang and D. Zhao, “Extracting traffic primitives directly from naturalistically logged data for self-driving applications,” *IEEE Robotics and Automation Letters*, 2018.

[45] W. Wang, J. Xi, and D. Zhao, “Driving style analysis using primitive driving patterns with bayesian nonparametric approaches,” *arXiv preprint arXiv:1708.08986*, 2017.

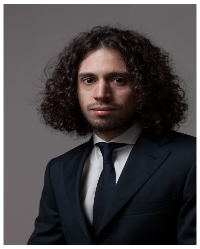
[46] R. A. Wagner and M. J. Fischer, “The string-to-string correction problem,” *J. ACM*, vol. 21, no. 1, pp. 168–173, 1974.

[47] U.S. Department of Trasportation, FHWA, “Ngsim-next generation simulation.” http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm, 1999.

[48] V. Alexiadis, J. Colyar, J. Halkias, R. Hranac, and G. McHale, “The next generation simulation program,” *Institute of Transportation Engineers. ITE Journal*, vol. 74, no. 8, p. 22, 2004.

[49] M. Montanino and V. Punzo, “Making ngsim data usable for studies on traffic flow theory: Multistep method for vehicle trajectory reconstruction,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2390, pp. 99–111, 2013.

Hyo-Sang Shin received his BSc from Pusan National University in 2004 and gained an MSc on flight dynamics, guidance and control in Aerospace Engineering from KAIST and a PhD on cooperative missile guidance from Cranfield University in 2006 and 2011, respectively. He is currently Reader on Guidance, Control and Navigation Systems in Autonomous and Intelligent Systems Group at Cranfield University. His current research interests include multiple target tracking, probabilistic target detection and distributed control of multiple agent systems.



Dario Turchi received Laurea (BSc.) degree in computer engineering and Laurea Specialistica (MSc.) degree (cum laude) in control engineering from University of Pisa, Pisa, Italy, in 2007 and 2011, respectively. Since 2013 he is a PhD student at the Division of Engineering Sciences, School of Engineering, Cranfield University, UK. His research interests cover information consensus, sensor and data fusion, cooperative control and path planning of autonomous systems, guidance, and navigation systems.

Shaoming He was born in 1991. He received his MSC degree in Aerospace Engineering from Beijing Institute of Technology in 2016. He is currently a PhD student in the School of Aerospace, Transport and Manufacturing, Cranfield University. His research interests include multi-target tracking, guidance and control.

Antonios Tsourdos obtained a MEng in electronic, control and systems engineering from the University of Sheffield (1995), an MSc in systems engineering from Cardiff University (1996), and a PhD in nonlinear robust missile autopilot design and analysis from Cranfield University (1999). He is a Professor of Control Engineering with Cranfield University, and was appointed Head of the Centre for Autonomous and Cyber-Physical Systems in 2013. He was a member of the Team Stellar, the winning team for the UK MoD Grand Challenge (2008) and the IET Innovation Award (Category Team, 2009).

2018-08-02

Behavior monitoring using learning techniques and regular-expressions-based pattern matching

Shin, Hyo-Sang

IEEE

Hyo-sang Shin, Dario Turchi, Shaoming He and Antonios Tsourdos. Behavior monitoring using learning techniques and regular-expressions-based pattern matching. IEEE Transactions on Intelligent Transportation Systems, Volume 20, Issue 4, 2019

https://doi.org/10.1109/TITS.2018.2849266