2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Detecting Anomalies in Unmanned Vehicles Using the Mahalanobis Distance*

Raz Lin, Eliyahu Khalastchi and Gal A. Kaminka
The MAVERICK Group, Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel 52900
{linraz,galk}@cs.biu.ac.il

Abstract— **The use of unmanned autonomous vehicles is becoming more and more significant in recent years. The fact that the vehicles are unmanned (whether autonomous or not), can lead to greater difficulties in identifying failure and anomalous states, since the operator cannot rely on its own body perceptions to identify failures. Moreover, as the autonomy of unmanned vehicles increases, it becomes more difficult for operators to monitor them closely, and this further exacerbates the difficulty of identifying anomalous states, in a timely manner. Model-based diagnosis and fault-detection systems have been proposed to recognize failures. However, these rely on the capabilities of the underlying model, which necessarily abstracts away from the physical reality of the robot. In this paper we propose a novel, model-free, approach for detecting anomalies in unmanned autonomous vehicles, based on their sensor readings (internal and external). Experiments conducted on Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) demonstrate the efficacy of the approach by detecting the vehicles deviations from nominal behavior.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) are finding increasing use in real-world applications. These include surveillance and patrolling ([1]), aerial search [8], and more. Increasing dependence on UAVs and UGVs (UVs, in general) for critical tasks makes it vital to remotely assure the nominal behavior of the vehicles, whether teleoperated or autonomous.

Yet, paradoxically, the prominent advantage of UVs being unmanned, raises the challenge of detecting deviations of the UV from its normal behavior. For example, when driving a car, the driver can use her own body perceptions to detect a failure (e.g., a flat tire can be detected when the wheel seems to "pull" to a particular direction, or the sound of the driving changes). However, in unmanned vehicles, a remote monitoring station no longer receives all the information available to a driver, and must instead rely on information gathered from (potentially faulty or inaccurate) sensors on the vehicle. Thus, it can be difficult for a remote operator to detect anomalies. The problem is further exacerbated with increasing autonomy of the unmanned vehicles, as this leads to reduced human monitoring, and therefore even further degradation in timely detection of failures.

A number of previous investigations have explored a variety of ways to improve monitoring. Model-based diagnosis methods (e.g., [10], [14]) rely on a detailed model of the vehicle's systems to note deviations between the vehicles actual behavior, and its nominal behavior (as generated by the model); these rely on the availability, resolution, and accuracy of a model. Fault models (e.g., rule-based) can be used to capture expert's experience in recognizing faults [6], but this approach is inherently tied to the scope of the expert's experience. Several approaches also attempt to identify outliers in the data based on the history of the vehicle's operation ([7], [9]).

In particular, a promising technique to determining anomalous values in data is based on the use of the Mahalanobis distance ([12], [9]), which measures the multi-dimensional distance between a sample point and a distribution, in units of standard deviation[1]. This allows, in principle, relatively straightforward discovery of outlier measurements. Unfortunately, the technique often fails in practice (due to both run-time and data availability issues) when the number of different variables (i.e., the number of dimensions) scales up [9]. Because of this, the Mahalanobis distance can only be used with a limited number of variables. Of course, a in monitoring UVs, we often have dozens, if not hundreds, of different variables whose values are reported to the monitoring station.

In this paper we propose a novel approach for detecting anomalies in the behavior of UVs, using the Mahalanobis distance. The approach consists of a pre-processing phase which finds dependencies between different internal sensors on the vehicles. This *dependency detection* (DD) phase uses an efficient search method—developed for data-mining applications and described in [13]—to identify sub-groups of variables that are statistically dependent (i.e., their values changes together in predictable ways). The results of this phase are therefore several distinct groups of variables—each of much smaller number of dimensions. The second phase, taking place during the execution of the mission, uses the Mahalanobis distance to identify anomalous values in each of the smaller-dimensional groups of variables.

We provide results of extensive experiments conducted using data from commercial UAVs, and in laboratory mobile

*We acknowledge partial support of IMOD. Thanks to K. Ushi.

[1]For one-dimensional data, the Mahalanobis distance is reduced to the standard z-score of a point.

ground robots. In the experiments, we investigate the efficacy of this approach in detecting anomalies in the UVs' behavior. We also demonstrate the critical role of the first phase.

II. Related Work

The problem of anomaly detection in the context of real-time data is not new to researches in robotics and autonomous systems. We cannot hope to cover all related work in this vast area, and so focus here on the most closely-related investigations.

In the context of unmanned vehicles, many investigations (e.g., [7], [15], [5]), propose the use of Kalman filters as a basic building block in detecting anomalies. Typically, one or more filters is use to make predictions of specific state variable values (e.g., sensors), and those predictions are compared against the observed values. Since often using a simple Kalman Filter usually yields a large number of false positives, additional computation is used to robustly decide on the failure state and its significance. For instance, Cork and Walker [5] present a non-linear model which, together with Kalman Filters, tries to compensate for malfunctioning sensors of UAVs. Goel *et al.* [7] use a classifying neural-network to determine when and which of several filter-based fault detectors to believe. The use of Kalman filters makes assumptions with regard to the behavioral nature of the data and noise (e.g., that the time series models are linear with additive Gaussian noise). The DD technique we introduce to detect subgroups of dependent variables can be useful to eliminate redundant variables from the Kalman filters, and thus simplify their design.

Oates *et al.* [13] and Lotze *et al.* [11] studied the problem of detecting anomalies in sequence of real-time data of patience and diseases. Our DD method is based on the work on multi-stream dependency-detection, described in [13]. However, we use only a subset of the results generated, and thus can potentially alleviate the computational load, compared to the original work. Machine learning methods are usually employed to model what constitutes a nominal behavior and deriving from the representation of the nominal behavior the abnormal behavior. For example, Ahmed *et al.* ([3], [2]) investigate the use of two distinct machine learning approaches, namely the block-based One-Class Neighbor Machine and the recursive Kernel-based Online Anomaly Detection algorithms, to detect network anomaly. Yet, as often happens in machine learning techniques, their models are constrained and cannot be easily adapted to other domains. Besides the fact that it is sensitive to different thresholds, to enable its use in different domains many parameters must be fine tuned. We, on the other hand, demonstrate that our proposed approach can be easily adapted to different domains, while preserving the high anomaly detection rates.

Recently, Daigle *et al.* [6] proposed an event based approach for diagnosis parametric faults in continuous systems. Their approach is based on a qualitative abstraction of deviations from the nominal behavior. Yet, in contrast to our proposed approach, their approach is aimed to diagnose an isolating single fault. Moreover, their technique is based on a finite automaton under the assumption that it is feasible to create a model that captures all relevant system behavior. Another approach for anomaly detection is based on model based reasoning (e.g., [10], [14]). Yet, this requires to have a model of the robot and its interactions with the environment. Such a model is expensive and complex to build.

Others have been using Mahalanobis distance for anomaly detection. Using the Mahalanobis distance allows testing whether a given sample point (say, a vector of all current values for all state variables) is "similar" to the nominal sample (defined by a distribution of such vectors), where the similarity accounts not only for the centers of each variable, but also for the variance of its values. For instance, Brotherton and Mackey [4] use the Mahalanobis distance as the key factor for determining whether signals measured from an aircraft are of nominal or abnormal behavior. However, they are limited in the number of dimensions (variables) across which they can use the distance, due to run-time issues. This is one challenge the DD approach addresses.

III. Problem Description

We define the problem of detecting anomalies in the behavior of unmanned autonomous vehicles. We deal with a multi-stream data which is measured by the UV and transmitted *online* to a remote monitoring computer. The data consists of various types of measurements collected by the UV and its sensors, internal and external, physical and virtual. For instance, the pose and altitude of the UV (e.g., location at the X, Y and Z axes, heading), odometry data, engine temperature, mass, and other telemetry. Let I denote the set of measurable inputs (called *attributes* below) the UV maintains, V_i the set of values (whether finite values or finite range of values) for each $i \in I$, and S a finite set of all joint values for all attributes ($V_1 \times V_2 \times \ldots \times V_{|I|}$). Therefore a *state* of the UV is denoted as a vector $\vec{s} \in S$. Let **Time** denote the set of time periods in which the UV is in motion, that is **Time** $= \{0, 1, ..., ep\}$, where *ep* denotes the end period, in which no more data is communicated to the ground station. A *stream* of data of the UV is then an ordered sequence of vectors, $\vec{s_0}, \ldots \vec{s_t}, \ldots s_{\vec{\textbf{Time}}}$, forming a matrix M of dimensions $\{|\textbf{Time}| \times |I|\}$ which consists of a state per each time unit. It is assumed that at each time unit (henceforth, *tick*) all attributes are transmitted at the rate of the highest frequency; that is, even if some attributes are measured in a lower frequency the state of the UV will always be sent in full.

We define a *nominal set* m of the UV as an uninterrupted sequence of vectors in M (i.e., an ordered set of consecutive vectors), in which the behavior of the UV is characterized as stable and following a predetermined logic. For example, we distinguish between the lift-off, the landing of the UAV and the constant speed flight. Each such stream can constitute a nominal behavior which characterizes the flight, and is captured by a different set m.

An *abnormal reading* is a vector $\vec{m^*} \in S$ which is defined as such that the UV deviates from the nominal set m by at least some threshold. In other words, $\vec{m^*} \notin m$, and also

$Dist(\vec{m^*}, m) > h_0$, where $Dist$ is a distance function which measures the multi-dimensional distance between the vector $\vec{m^*}$ and the set m. It returns a number which is contrasted with the constant h_0. A value higher than h_0 indicates an abnormal reading.

IV. Detecting Anomalies in UVs' Behavior

Naively, to detect anomalies, we can use the matrix M from nominal run as the nominal state m, and the Mahalanobis distance [12] for the distance function $Dist$. We describe this method in Section IV-A. Unfortunately, for realistic UVs this fails due to the large number of attributes. We explain this—and describe a solution to this—in Section IV-B.

A. The Anomaly Detection Component

The anomaly detection component is the online mechanism which can be invoked at any given time on the streaming data. In this mechanism we also use the assumption that historical input is accessible, and thus it is used to compare the online data to it to allow finding which stream deviates from the nominal behavior of the UV.

To find a correlation between two sets of data a statistical model should be used. A simple distance metric, such as the Euclidean distance, for example, could be applicable if we are only comparing one vector to another. However, we need to compare a vector $\vec{m^*}$ to a set of vectors m.

Towards this end, we chose to use the Mahalanobis distance [12] as the distance metric that we invoke. Generally speaking, the Mahalanobis distance is the distance of the input stream from the centroid in the multidimensional space, where the centroid is built based on the distribution. Thus, it provides an indication of whether or not a given vector is an *outlier* with respect to the nominal set of vectors. Formally, we denote the known sample matrix as $M^{sample}_{t\times|S|}$, where $t \in T$ is the total time units in the sample matrix. The mean of all attributes in the sample matrix is a vector $\vec{\mu} = (\mu_1, \mu_2, ..., \mu_{|S|})$ and the covariance matrix is denoted as COV. Thus, the Mahalanobis distance of a new matrix X is denoted as:

$$D_{mahal}(X) = \sqrt{(X - \vec{\mu})^T COV^{-1} (X - \vec{\mu})} \qquad (1)$$

The output of the Mahalanobis distance is given in units of standard deviations. A large value is a stronger indication of the stream being an outlier than a smaller number. Note that this method has the benefit of not relying on domain knowledge with regard to the UV's behavior (e.g., motion or physical model of the flight/drive).

Unfortunately, the naive method described above does not work in realistic UVs (see Experiments section), for several reasons. The first obvious problem is that as the number of attributes is easily in the dozens, and often in the hundreds, the run-time increases to the point where processing cannot keep up with the incoming data.

However, a more fundamental problem exists with the use of the Mahalanobis distance with large-dimensional data. It relies on estimating a centroid of a distribution from the data available to it, and as a result, sparse data can complete throw off the estimation process. Thus a significant amount of data is needed to construct a good sample of the distribution, in order to allow the Mahalanobis distance to provide robust results. Now, as the number of dimensions increases, the amount of data required for such good sample grows combinatorially. For instance, having 50 different attributes means that the nominal distribution m is built from 50-dimensional vectors. To have each nominal vector appear once in m, we would require $|V|^{50}$ vectors (V is the set of possible values for each attribute) just as training data.

We believe this is the principal reason why methods based on Mahalanobis distance have only been used in UVs indirectly, in support of other methods. The Mahalanobis anomaly detector by itself simply does not scale with the number of attributes.

B. The Dependency Detection (DD) Component

We introduce a pre-processing mechanism that uses statistical dependency-detection methods to determine possible sub-groups of attributes which are statistically interdependent. These sub-groups are then used in the second (online) phase to form the basis for the Mahalanobis distance measurements. Thus, instead of using the Mahalanobis outlier detector on the entire input vector, we break the task into a set of outlier detectors, each focused on parts of the input vector, each using its own nominal distribution m, and each operating in a small-dimensional space (in our experiments, typically 2–3 attributes).

In this work we build on earlier work by Oates *et al.* [13], which have developed efficient data-mining algorithm called Multi-Stream Dependency Detection (MSDD). The algorithm finds statistically significant patterns of the type $A_x B_y \rightarrow C_z D_q$, which should be understood as follows: In an input vector $\vec{v}$, if the value A appears in attribute x, and the value B appears in attribute y, then the value C will likely appear in attribute z and the value D will likely appear in attribute q. In other words, MSDD is able to determine that attribute values are dependent on each other. The statistical strength of the patterns are measured by the G statistic [16][2]. MSDD uses an efficient heuristic search which guarantees finding the complete set of patterns, without examining the entire combinatorial search space.

MSDD is both too crude and too good for our needs. On one hand, MSDD has the capability for finding such patterns even when they are spread over time (i.e., to find patterns of the form "if attribute x has value A, then in two ticks, we expect attribute y to have value B"), and can thus produce finer-grained information than what is needed for the Mahalanobis distance. Indeed, using this finer grain may be interesting for anomaly detection by itself, but we leave this for future work. On the other hand, rather than simply outputting a single pattern for each set of dependent attributes, MSDD very often detects redundant dependencies,

[2]Similar in principle to the χ^2 statistic.

by finding different variations on the same basic dependence:

$$\begin{aligned} A_x B_y &\to C_z D_q \\ C_z A_x B_y &\to D_q \\ C_z D_q &\to A_x B_y \\ &\dots \end{aligned}$$

We therefore used a modified version of MSDD—called simply Dependency Detection (DD)—in which redundant patterns of the type above are merged together to form groups of dependent attributes (in this case, the set would be x, y, z, q). This modified version, in effect, tells us what attributes are correlated, and this, in turn, allows to run the Mahalanobis distance only on the dependent attributes, thus significantly reducing the dimensionality of the space. Note that often more than one group of dependent attributes would be identified, in which case multiple Mahalanobis outlier detectors should be used, one for each group.

V. Experiments

To evaluate the efficacy of our approach for detecting anomalies in UVs we conducted several sets of experiments. The different experiments demonstrate the strength of the combination between the pre-processing dependency detection mechanism and the online anomaly detection mechanism, as well as the generality of the approach. We begin by describing the experiment setup, and then continue to describe the different experiments and results.

A. Experiment Setup

We chose two different unmanned vehicles to demonstrate the generality of our approach. The first set of data came from actual commercial unmanned aerial vehicles. The UAV is equipped with several sensors and actuators, as well as a communication system. The communication system transmits the information, along with monitoring information, to the ground station.

The information which is measured by the UAV sensors and is relevant for the anomaly detection process includes more than 50 attributes. The different attributes can be categorized to different families: *air data* (includes telemetry data that the UAV measures), *inertial data* (includes information about the inertial navigation system (INS)), *engine data* (includes information about the engine's air and water temperature), *servo* information, and *other* information, including the UAV mass, the air temperature and other information. The data is measured by the sensors either in a 1Hz or 10Hz frequencies, yet the whole data is downloaded from the UAV at a frequency of 10Hz.

The second set of experiments was conducted on a commercial vacuum-cleaning mobile robot (the Friendly Robotics RV-400) fitted with our own control software.

The RV-400 robot is equipped with many less sensors and actuators than the UAV. It has 22 attributes measured by ten sonar sensors which measure ranges, four bumper sensors, and various other measurements including the target velocity and the actual velocity (based on wheel motor encoder data), etc. The data itself is recorded in a 10Hz frequency.

In the course of evaluations, we utilized data from several nominal runs of the robots, as well as from failure runs. We refer to these runs in the discussion of the experiments below.

For the UAV we the following errors were recorded:

- *Descend*: In this error, one of the sensors is malfunctioning and thus causes the sensor's input to decrease rapidly from a valid input to a constant value of zero.
- *Constant*: In this error, one of the sensors is malfunctioning and reports a constant value for a period.

For the UGV the following errors were recorded:

- *Weight Drag Halt*: In this error, the robot was attached to a cart via a fishing string which was loose. Then, the robot started its movement away from the cart, causing the string to stretch, until it was completely stretched. This caused the robot to completely halt.
- *Direction Deviation*: In this error, a coin was attached to one of the robot's wheels. This caused the robot to divert from nominal behavior every time the coin touched the floor (which was about every 5 seconds). It also changed its heading, etc.

For each UV we had a nominal behavior file which was used for two purposes. First, it was used in the pre-processing phase to obtain the strongest dependent attributes for the domain of the UV. Then, it was used in the online anomaly detection process for finding deviations in the behavior of the vehicle from the nominal behavior recorded in that file (i.e., as the basis for the nominal set m). The experiment data sets are summarize in Table I.

B. Successfully Detecting Anomalies

As we mentioned, the anomaly detection process requires that we first find the attributes that are considered correlated. To this end, we ran the pre-processing mechanism described in Section IV-B on a nominal data file, Nominal UAV A (Nominal UGV A for the UGV domain). Out of the 56 (22 for the UGV) different attributes that are measured several attributes were found to be significantly strongly correlated (based on the G statistic). We chose to use 2 of the correlated attributes in our anomaly detection process (both in the UAV and the UGV domains). It is notable to mention that in the case of the UGV domain, the MSDD pre-processing mechanism returned somewhat unsurprising correlation between the odometry sensors, yet it also returned a surprising correlation between two sonars on-board the robot. Later we found this dependency highly useful for detecting the anomalies in the UGV experiments.

In the process of detecting the anomalies we need to determine the threshold above which an anomaly is flagged. To this end we first run the Mahalanobis distance algorithm on the nominal file and create a histogram of the standard deviations that are the output of the algorithm. The threshold is then determined in such that at least 93% of the measurements are below it. For the UAV and UGV domains, this generates a threshold of 15 and 0.081 standard deviation

Data Type	Description
Nominal UAV A	Contains nominal flight behavior. This file was also used for the pre-process phase and the comparison of other UAVs' behavior.
Nominal UAV B	Contains an additional nominal flight behavior.
Descend UAV C	Contains an error in a sensor, which rapidly decreases its value until a constant zero. The error is between time units 15,990 and 16,054.
Constant UAV D	Contains an error in a sensor, which value is stuck constant. The error is between time units 8,105 and 8205.
Nominal UGV A	Contains nominal driving behavior. This file was also used for the pre-process phase and the comparison of other UGVs' behavior.
Weight Drag Halt UGV	Contains an error in the nominal driving behavior: The UGV attempts to drag a heavy load, which causes to comes to a complete halt at time unit 100.
Direction Deviation UGV	Contains an error in the nominal driving behavior: The UGV has an object stuck in one of its wheels, causing it to bounce every 5 seconds.

TABLE I
DESCRIPTION OF EXPERIMENT DATA.

units, respectively. We begin by describing the results on the UAV domain and finish with the description of the UGV experiments.

1) Detecting Anomalies in UAVs: Figure 1 shows the results of the Mahalanobis distance algorithm when applied on the Descend UAV C data. Disregarding the end of the flight, in which the behavior of the UAV changes, we can see from the figure that in the exact times of the error, the output of the Mahalanobis distance is significantly higher than the threshold. Out of the 64 time units of the error, a total of 59 (92%) were above the 15 threshold.

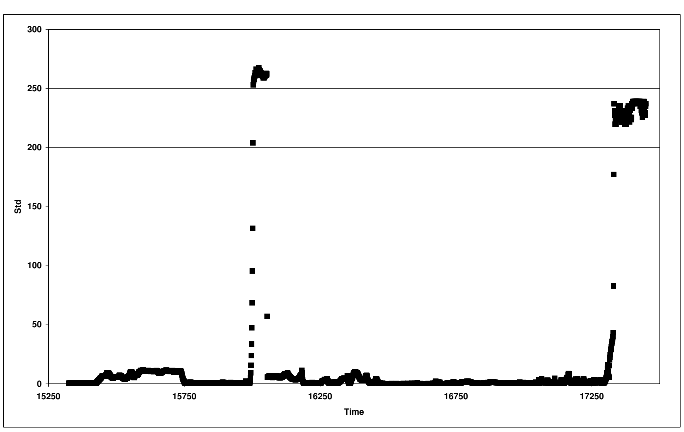

Fig. 1. Descend UAV C: Mahalanobis distance (in std units) as a function of flight time.

Figure 2 shows the results of the Mahalanobis distance algorithm when applied on the Constant UAV D data. Again, we disregard the start and end of the flight time periods (we discuss them later). Unfortunately, though, the algorithm found no evidence of deviations from the nominal behavior in this case. The explanation for this is the fact that "freezing" a sensor on a constant value does not cause deviations from nominal behavior, since the value is legit, and thus the Mahalanobis distance cannot detect these kinds of errors.

Trying to overcome this issue we ran an additional experiment. In this experiment we took the differential of the data per each attribute, and now ran the anomaly detection mechanism to find whether there are deviations from the nominal behavior of the UAV based on the differential of the data. Figure 3 now shows the results of this experiment (note that for display purposes we omitted the start and end periods of the flight from the figure's scale). Now we can see that the algorithm was indeed able to find a deviation from the nominal behavior at the end of the error period, just before the sensor re-started reporting normal behavior.

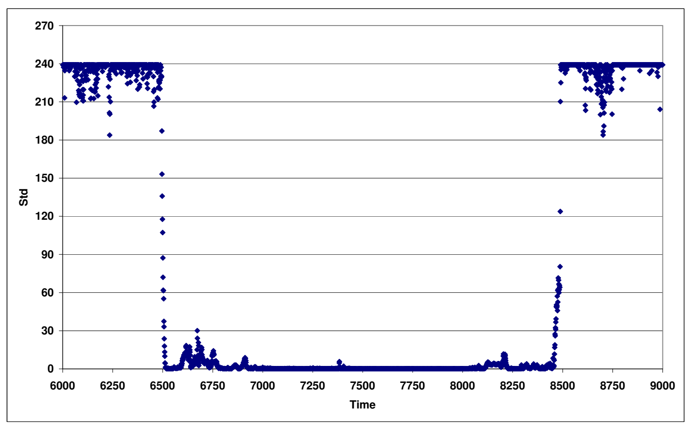

Fig. 2. Constant UAV D: Mahalanobis distance (in std units) as a function of flight time.

Encouraged by these results, we moved to apply this technique to the UGV experiment data. The results of the approach on the UGV domain is given below.

2) Detecting Anomalies in UGVs: Figure 4 shows the results of the Mahalanobis distance algorithm when used with the UGV Weight Drag data, causing the UGV to halt . In Figure 4 we can see that the approach accurately detected the stop movement of the UGV around time unit 100.

Finally, Figure 5 depicts the results of the Mahalanobis distance algorithm when applied on the Direction Deviation UGV anomaly. We can see that approximately every 5 seconds the standard deviation units leap to a value larger than 0.08. An operator at the control station watching this data online (or rather, being notified as the measures pass the threshold) would have been able to detect that there is some malfunction with the robot, which is taking place every 5 seconds.

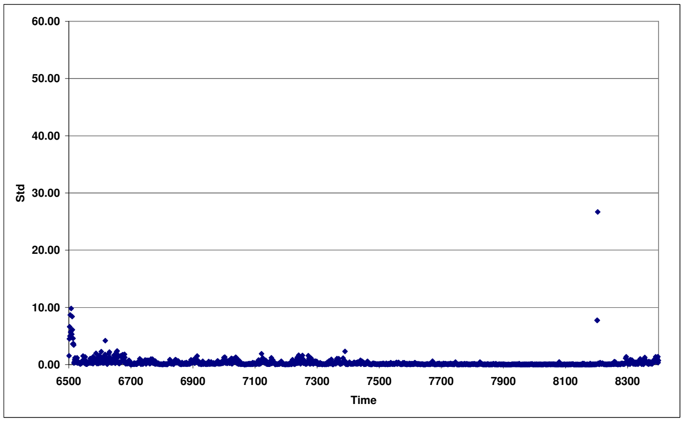

Fig. 3. Constant UAV D, analysing differential data: Mahalanobis distance (in std units) as a function of flight time.

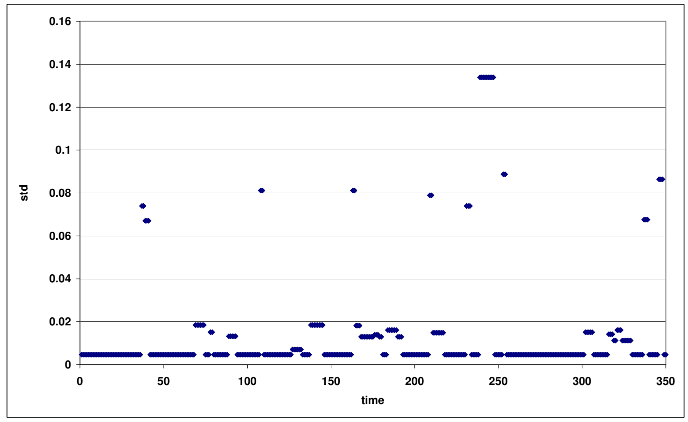

Fig. 5. Direction Deviation UGV: Mahalanobis distance (in std units) as a function of movement time.

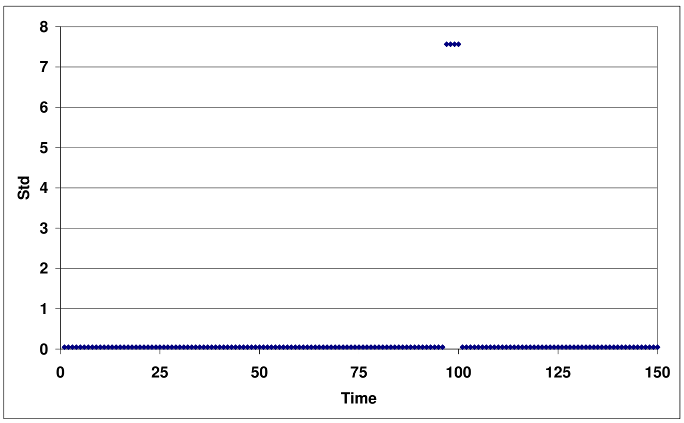

Fig. 4. Drag Weight Halt UGV: Mahalanobis distance (in std units) as a function of movement time.

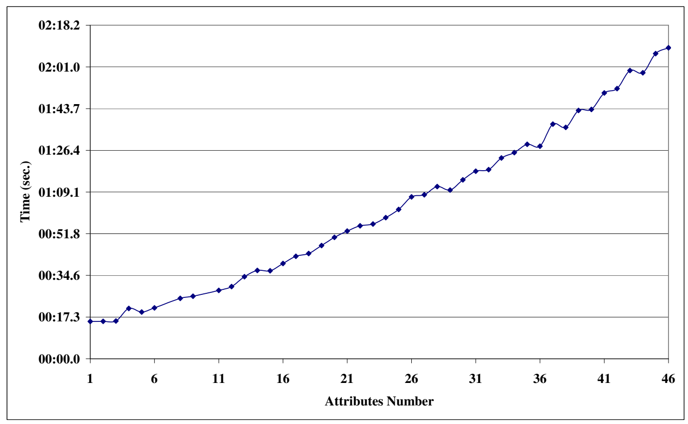

Fig. 6. Mahalanobis distance's run-time (in minutes) as a function of attributes number.

C. The Importance of DD Pre-Processing

The Mahalanobis distance cannot stand on its own to detect anomalies. Its success in detecting anomalies above lies in the fact that the dependency detection pre-processing mechanism was invoked prior to the anomaly detection algorithm, and chose specific attributes on which to focus the Mahalanobis distance measure. Here, we demonstrate the importance of invoking this mechanism.

First, let us examine the run-time of using the Mahalanobis distance as the number of attributes increases. Figure 6 demonstrates the importance of narrowing down the input to the Mahalanobis distance algorithm. The figure shows the run-time (in minutes) of the algorithm as a function of the number of attributes in each stream of data it uses to detect the deviation from the nominal behavior. The results demonstrate that the algorithm's run-time increases quickly as a function of the number of attributes. This is a part of the motivation for allowing the DD process to select a smaller set of attributes.

However, it is not simply a case of reducing the number of attributes. To demonstrate the importance of running the Mahalanobis distance on dependent attributes we ran the following experiment.

Here, we built on a predefined knowledge of the UAV domain and chose several attributes which are independent of each other (we verified also that they do not appear in the results of the DD process). We then applied the Mahalanobis outlier detector based on these attributes, to see if we could detect the failures using these attributes instead of those selected by the DD process. We hypothesized that both on the nominal files and the simulated error files the results would generate high rates of false alarms (detecting anomalies even though there is none), making the algorithm useless.

We started by running the algorithm on two different data files which describe a nominal behavior (Nominal UAV A and Nominal UAV B). Then, we applied the same mechanism on a data file which simulated errors in predefined times (Descend UAV C). Figure 7 show the percentage of false alarms detected when running the Mahalanobis distance on uncorrelated attributes as compared to running it on correlated attributes.

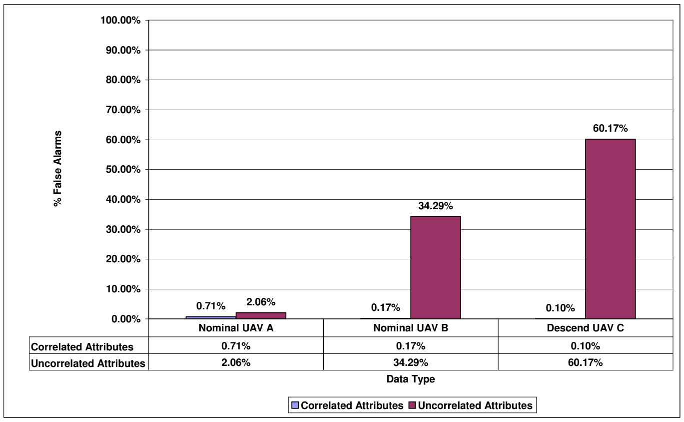

Fig. 7. False alarm rates when applying the Mahalanobis distance on correlated and uncorrelated attributes.

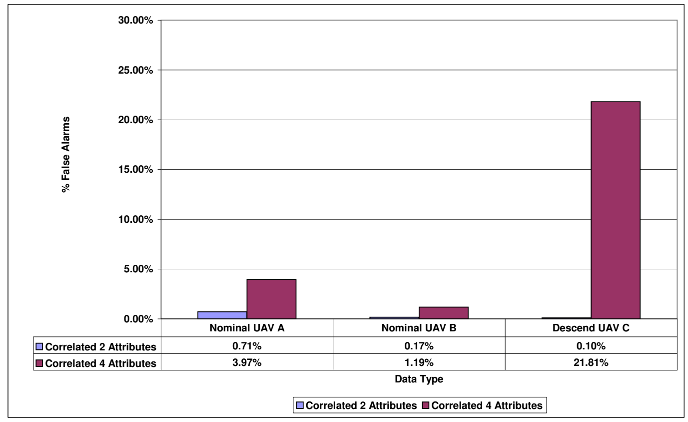

Fig. 8. False alarm rates when applying the Mahalanobis distance on two and four correlated attributes.

As we hypothesized, we can see that the approach does not scale well if the input is not fine tuned. While the rates of false alarms when applying the algorithm on dependent attributes is relatively low (0.71%, 0.17% and 0.10% for Nominal UAV A, Nominal UAV B and Descend UAV C, respectively), the rates increase significantly when applied on uncorrelated attributes (2.06%, 34.29% and 60.17% for Nominal UAV A, Nominal UAV B and Descend UAV C, respectively). That is, the algorithm "found" that the nominal flights actually deviated from the nominal behavior, which, of course, was not the case.

As we argued in Section IV-B, a small number of attributes is also important because it facilitates increased accuracy. Figure 8 demonstrates that the number of false alarms dramatically increases (compared to the DD-based runs) if too many attributes are used (3.97%, 1.19% and 21.81% for Nominal UAV A, Nominal UAV B and Descend UAV C, respectively, when four attributes are used—compare to 0.71%, 0.17% and 0.10% when using the two strongly-dependent attributes). From the figure we can see the difference in the false alarm ratio when only two of the strongest correlated attributes are used as compared to using four strongest attributes. Thus, using the DD algorithm to find the K strongest correlated attributes can also allow minimizing false alarms in the anomaly detection process.

VI. Conclusions and Future Work

In this work we presented a novel approach for detecting anomalies in unmanned autonomous vehicles. Experimenting with two different domains we showed that applying our approach allows detecting anomalies successfully in the different domains, encouraging us with regard to the efficacy and adaptability of our approach.

As discussed, future work also warrants careful investigation due to different behavioral characteristics and dynamics of the motion of the autonomous robots, to allow the differentiation between the anomaly detection when a change is occurring in the behavioral settings of the robot. Future work in this field will also focus on an efficient automated way for determining the threshold for the Mahalanobis distance algorithm, above which anomaly should be detected. While negative examples might be a scarce resource, another research direction would be to understand how to utilize positive examples, which might be abundant, and reflect from them on nominal behavior and the deviation from it.

References

[1] N. Agmon, S. Kraus, and G. A. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *ICRA*, pages 2339–2345, 2008.

[2] T. Ahmed, M. Coates, and A. Lakhina. Multivariate online anomaly detection using kernel recursive least squares. In *IEEE INFOCOM*, pages 625–633, 2007.

[3] T. Ahmed, B. Oreshkin, and M. Coates. Machine learning approaches to network anomaly detection. In *SysML*, 2007.

[4] T. Brotherton and R. Mackey. Anomaly detector fusion processing for advanced military aircraft. In *IEEE Proceedings on Aerospace Conference*, pages 3125–3137, 2001.

[5] L. Cork and R. Walker. Sensor fault detection for UAVs using a nonlinear dynamic model and the IMM-UKF algorithm. *IDC*, pages 230–235, 2007.

[6] M. Daigle, X. Koutsoukos, and G. Biswas. A qualitative event-based approach to continuous systems diagnosis. *IEEE Transactions on Control Systems Technology*, 17(4):780–793, 2009.

[7] P. Goel, G. Dedeoglu, S. I. Roumeliotis, and G. S. Sukhatme. Fault-detection and identification in a mobile robot using multiple model estimateion and neural network. In *ICRA*, 2000.

[8] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey. Supporting wilderness search and rescue using a camera-equipped mini UAV. *Journal of Field Robotics*, pages 89–110, 2008.

[9] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[10] M. Hofbaur, J. Köb, G. Steinbauer, and F. Wotawa. Improving robustness of mobile robots using model-based reasoning. *Journal of Intelligent and Robotic Systems*, 48(1):37–54, 2007.

[11] T. Lotze, G. Shmueli, S. Murphy, and H. Burkom. A wavelet-based anomaly detector for early detection of disease outbreaks. In *ICML*, 2006.

[12] P. C. Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Institute of Science*, pages 49–55, 1936.

[13] T. Oates, M. D. Schmill, D. E. Gregory, and P. R. Cohen. *Learning from Data: Artificial Intelligence and Statistics*, chapter Detecting Complex Dependencies in Categorical Data, pages 185–195. Springer Verlag, 1995.

[14] G. Steinbauer, M. Mörth, and F. Wotawa. Real-time diagnosis and repair of faults of robot control software. In *RoboCup 2005: Robot Soccer World Cup IX*, pages 13–23. 2006.

[15] P. Sundvall and P. Jensfelt. Fault detection for mobile robots using redundant positioning systems. In *ICRA*, 2006.

[16] T. D. Wickens. *Multiway Contingency Tables Analysis for the Social Sciences*. Lawrence Erlbaum Associates, 1989.