Drone Control to Major Tom: Anomaly Detection and Digital Twins

Eeshaan Verma
Thales UK
Crawley, UK
eeshaan.verma@uk.thalesgroup.com

ABSTRACT

As technology has improved, the use of drones has become commonplace, operating in a variety of domains from defence to environmental conservation. Among this are 'wingman' drones that are controlled by the pilot of an escorting vehicle. Their operational environment is often extreme and hence malfunctions can occur mid-flight which can vary from minor deviations in flight path to catastrophic failures. Dealing with these errors can lead to pilot overload and reduced situational awareness, especially within a complex domain such as congested or contested airspace. How does one predict the emergence of these faults and take action to mitigate them mid-flight autonomously?

Anomaly detection techniques allow for the identification of abnormal patterns within data and has been used for predictive maintenance. Generally, these techniques are trained on an ideal flight and sensor datasets which can be quite difficult to obtain. With the advent of the 4th industrial revolution, the boundaries between the physical and digital world have become blurred with technologies such as digital twins which can represent a physical system digitally.

This paper investigates using a drone digital twin and anomaly detection concurrently to predict and mitigate in-flight drone malfunctions. This will be demonstrated by the creation of a training dataset for anomaly detection techniques using a digital twin, the comparison of new anomalous data with past malfunction patterns and the use of the digital twin to inform mitigation strategies, both in training and operationally.

ABOUT THE AUTHOR

Eeshaan Verma is a part of the 2020 cohort of the engineering graduate scheme in Thales UK as a graduate engineer. He is currently working within Thales Training & Simulation looking into data processing and analytics platforms. Previously he was placed within Thales Research, Technology and Innovation exploring anomaly detection techniques and produced an internally published research note on the application of Growing Neural Gasses for anomaly detection.

Eeshaan has a dual-honours Bachelor's degree in Mathematics and Physics from the University of Warwick and is the Vice-Chair for Young Professionals in London branch of the Institute of Engineering and Technology (IET). He is currently pursuing a part-time MSc in Engineering from Cranfield University.

Drone Control to Major Tom: Anomaly Detection and Digital Twins

Eeshaan Verma
Thales UK
Crawley, UK
eeshaan.verma@uk.thalesgroup.com

INTRODUCTION

Unmanned Air Vehicles (UAV)s are increasingly being used in military applications especially where the mission has a long endurance, unfavourable conditions e.g. chemical, biological, nuclear and extreme weather or contested airspace (Ferramosca, 2021). UAV operation in such environments can lead to faults occurring during flight which may result in mission failure, serious injury or even death (ITT ENIDINE INC, 2019). Since 2001, the USA has had more than 400 large military UAV crashes around the world resulting in major accidents (Whitlock, 2014). To reduce this, it is necessary to plan system fault mitigation measures such as the installation of redundant hardware for mission critical parts in addition to predictive and corrective maintenance before a flight occurs. However, these measures often take up space and add weight to the overall system (Keipour, Mousaei, & Scherer, 2019). Additionally, a significant percentage of accidents are caused by human error e.g. Hunter (32%), Shadow (21%), Pioneer (28%) and Predator (67%) UAV platforms (Williams, 2004).

UAV operators receive sensor and system information that relays the health of the craft. If faults occur mid-operation human intervention is often required in an otherwise autonomous system. Operators that face many faults during a mission may be overloaded by the information relayed to them and unable to efficiently prioritise fault mitigation strategies. This may result in poor mission performance with (Porat, Oron-Gilad, Rottem-Hovev, & Silbiger, 2016) showing a threshold of 10 supervised or 3 controlled drones before performance degradation for a ground-based operator. A review by (Saini, Raju, & Chail, 2021) discussed that a minimum of 14% of ground-based drone operators were above the clinical cut-off for emotional exhaustion: a condition that could impair performance. This effect is exaggerated for UAVs in adverse environments and over long mission durations where a significant amount of concentration is needed for both. Furthermore, training operators to be able to identify faults and mitigate them is a lengthy process which will need to be repeated if the UAV platform changes significantly.

The Hebb-Yerkes-Dodson Law (Diamond, Campbell, Park, Halonen, & Zoladz, 2007) states that for complex tasks, peak cognitive performance occurs when a person is sufficiently mentally stimulated. Work by (Andrews, 2020) considers this and suggests that automation should affect both the system and an operator's cognitive workload. They suggest that there are 5 levels of autonomous abstraction (Table 1) and evaluate the simulated performance of a pilot manning an aircraft like an F-35 and controlling 3 UAVs concurrently. Furthermore, considering the results for cognitive workload and mission performance, an autonomous system at the level of Tactical Battle Manager showed the best results over 1000 trials. However, an overreliance on autonomous systems can lead to automation bias, a loss of situational awareness and shifting moral responsibility (Agrawal & Cleland-Huang, 2021).

Future generations of UAVs are likely to be more adaptive and intelligent to cope with changing safety and reliability requirements. Addressing increasingly complex missions will necessitate intelligent approaches to address the emergencies that will arise during operation. Not only the current increase in the use of UAVs and but also the projected surge of use in the future makes the real-time diagnosis of these systems a priority. Hence, based on the information above, this paper will focus on providing a framework for an autonomous system that is close to the level of a Tactical Battle Manager (Figure 1). This system should be able to predict and mitigate in-flight drone malfunctions by taking suitable actions autonomously. To achieve this, a digital twin of the system coupled with various anomaly detection schemes can be used.

Digital Twins (DT) are a virtual representation of a physical entity that can mimic real world behaviour given certain data inputs with the advantages that it allows: real-time monitoring, fast simulation and troubleshooting and the ability to identify bottlenecks in a system (Wang, Cao, & Wang, 2021). Some uses include asset maintenance (Lu, Xie, Parlikad, & Schooling, 2020) and dataset creation (Ryad, et al., 2021). Anomaly detection refers to finding patterns in data that do not conform to expected behaviour. These non-conforming patterns are often referred to as anomalies. Anomaly detection is important since anomalies in data translate to significant actionable information: Refer to (Chandola, Banerjee, & Kumar, 2009) for a comprehensive overview on anomaly detection.

Table 1. Levels of autonomous abstraction, adapted from (Andrews, 2020)

Condition	Levels of Human Control Abstraction (LHCA)	Pilot's Role	UAV Role
Traditional Manned Wingman (Fully Manned)		Pilot performs all the planning and execution for the manned aircraft	No UAV involvement
Vector Steering	Parametric Control	Pilot flies manned aircraft; performs all planning and decision making for the UAVs	UAV follows specific pilot commands
Pilot Directed Engagement	Goal Oriented Control	Pilot flies manned aircraft; performs general planning and decision making for organisational movements	UAV autonomously decides how to execute general pilot commands
Tactical Battle Manager	Goal Oriented Control	Pilot flies manned aircraft; performs overarching planning and decision making with minimal interference	UAV decides and acts autonomously unless recommended action is vetoed by the pilot
No Manned Aircraft Engagement	Parametric and Goal Oriented Control	Pilot flies manned aircraft; offers no assistance in attacking the enemy targets	UAV executes pilot commands which are a combination of the above commands

OVERALL SYSTEM ARCHITECTURE

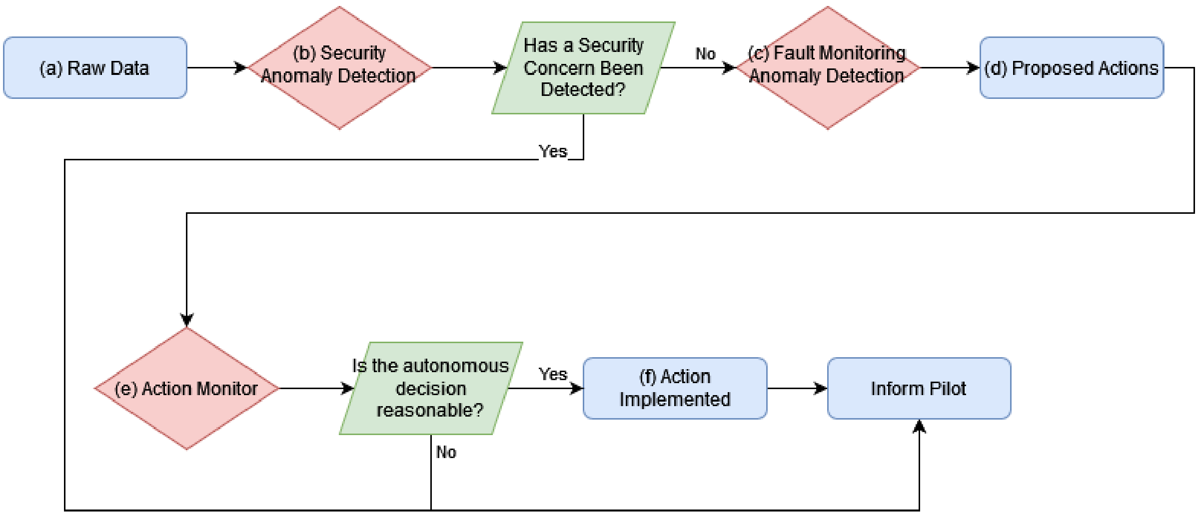

Figure 1. Proposed overall autonomous system architecture

Figure 1 shows a high-level architecture of an autonomous system that can detect faults within a UAV and then decide on a suitable action based on the fault it has found. The following subsections expand on each of the components of the architecture.

Figure 1a: Raw Data

All available data is collected from the system and sent for verification.

Figure 1b: Security Anomaly Detection Phase

Sending data directly for fault mitigation processing is unadvisable since discrepancies within the data may be due to adversarial attack. An example of this is the work by (Chen, Dong, & Duan, 2018) which injects false data that can result in improper geolocation, battery temperature etc. Directly ingesting this data would likely lead to an incorrect decision by the autonomous system resulting in the UAV operating outside of the acceptable mission parameters. Consequently, it is vital to examine incoming data to evaluate its integrity. If a security breach has been found, it should be directly reported to the pilot. If not, the raw data received by the security anomaly detection is sent for fault anomaly detection.

Figure 1c: Fault Anomaly Detection Phase

Here, raw data that has been approved for use is processed, cleaned and used to train the fault DT model. Additionally, anomaly detection that occurs here will inform the actions to be taken by the system. At this stage, it is important that the noise and discrepancies in the data is accounted for to prevent false positive results.

Figure 1d: Proposed Actions

Figure 2 shows a suggested framework for handling a detected anomaly. If there is no change to mission capability i.e. even with the anomaly the UAV is operating within mission parameters then the UAV should continue its current operation and store the pattern as a ‘precursor event’. In this instance, a precursor event is an anomalous pattern that does not affect mission performance itself but could be indicative of a fault that may occur in the future that will impact the mission.

If there is a change in mission capability but both the UAV and its flightpath report as being normal, pilot intervention will be required since the suggested fault mitigation strategy will not be able to provide a solution. An example of this would be a bug occurring within the detection software resulting in an anomalous reading. Alternatively, if there is a deviation in flightpath it may be possible to alleviate this by adjusting the UAV system.

When there are multiple anomalous results due to abnormal sensor data that change mission performance it is important to assess which has the greatest impact. One way is to use Failure Mode and Effect Analysis (FMEA) and assign each fault a Risk Priority Number (RPN).

$$RPN = Severity \times Occurence \times Detectibility \text{ (Franco \& Góes, 2007)}$$

Where the severity is the amount of harm or damage the failure mode may cause, the occurrence is the likelihood that the failure will occur, and detectability is the likelihood that the failure will not be detected (Franco & Góes, 2007). However, a disadvantage of this method is that it relies on having prior knowledge of the types and severity of faults that can occur, this makes it difficult to assign a numerical value to new and unexpected faults. One way to overcome this is by making the factors a function of the data itself. For example, the occurrence could be the number of times an anomalous pattern like the one being currently evaluated has occurred in past data. A similar pattern could be defined as having on average, less that one standard deviation between the error of each point in an anomalous pattern and patterns within historical data.

Once the RPN is determined, the fault with the highest value will be processed first. If a new fault has occurred, it needs to be determined what parameters must be adjusted to bring the UAV back to mission capability. Using the DT, it is possible to observe the effects any change will have on the system and potentially mitigate the fault. If a fault has been successfully mitigated it is stored along with the method of mitigation and is also used for further training the DT. If it is not possible to resolve a fault, this is relayed back to the pilot. Other more sophisticated methods for fault control can be found here: (Fourlas & Karras, 2021) (Zogopoulos-Papaliakos, Karras, & Kyrikopoulos, 2021).

Figure 1e: Action Monitor

An action monitor (Watkins, Hamilton, Kornegay, & Rubin, 2021) is a method of autonomous assurance to prevent a failure in mission capability due to errors within the autonomous code. It would evaluate each mitigation decision and check for statistical differences between decisions effecting the same fault and part. If there is a significant difference, this could be reported back to the pilot or take another appropriate action e.g. the UAV enters a degraded state whereby it returns to base.

Figure 1f: Action Implementation

Once approved by the action monitor the proposed mitigation strategy will occur. In order to prevent a pilot from receiving multiple messages about small corrections one could filter the messages so that only corrections on faults larger than a certain RPN value are displayed.

Figure 2. Flow diagram suggesting a method for handling a positive anomalous fault detection

DIGITAL TWIN-DRIVEN ANOMALY DETECTION ARCHITECTURE

Overview

This subsection looks further into the DT anomaly detection schemes of Figure 1b &1c with Figure 3 representing a general workflow for anomaly detection and DT updating. At this stage it is critical that noise and lapses in sensor measurement are corrected since they may result in a false positive detection or poorly train the DT model.

Flight data available from a fixed wing class ii UAV platform is used as a case study to discuss the section of Figure 3 highlighted in green with particular importance given to data pre-processing since poor data handling can cause issues further along the workflow. See (Ministry of Defence UK, 2017) for more on UAV classes. The aim is to process raw data in order to be close to ground truth data. Hence, one needs to understand, detect, and remove any redundant anomalies in the data, as well as transform the available data into the desired objective dataset.

The framework consists of five steps: Data Ingestion, Data Observation, Data Cleaning, Data Transformation and Data for Training.

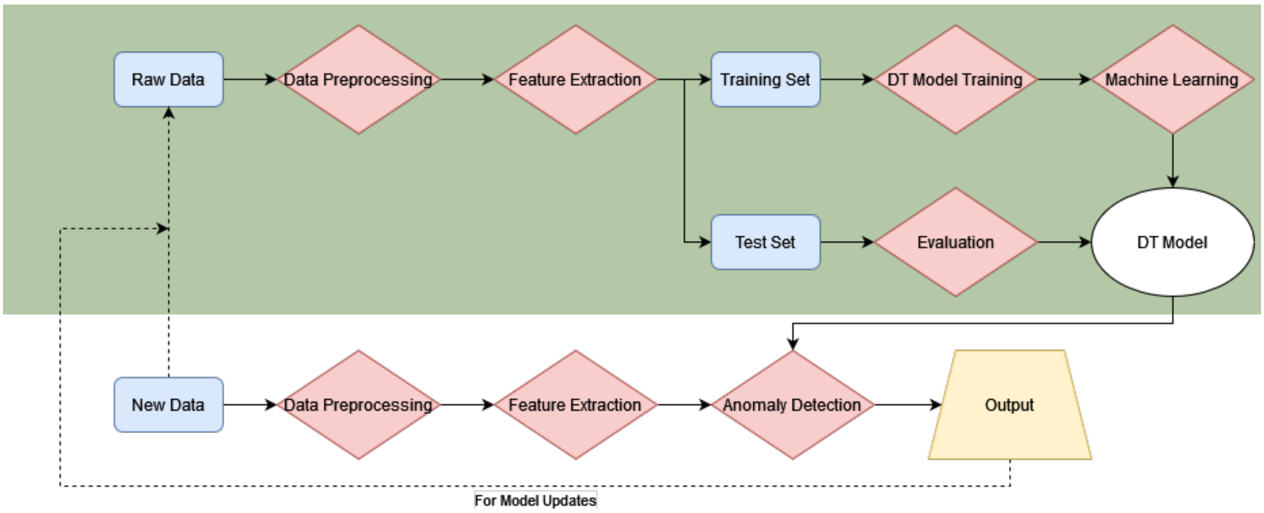

Figure 3. Data flow within anomaly detection schemes found in figure 1 (b & c), adapted from (Huang, Yang, Wang, Xu, & Lu, 2021)

Data Ingestion

The available UAV database contains 603 flights across several UAVs. Since it is likely that different tail numbers will have different platform configurations and performance profiles, data from the platform with the highest number of flights (60) was selected for analysis. The raw data was queried and gave over 230 hours of recorded data and from this the geolocation and Inertial Navigation System (INS) data was used.

Data Observation

Plotting the data, it is evident that discrepancies result in an abnormal pattern despite no such occurrence during the actual flight (Figure 4, Figure 5, and Figure 6). Another unexpected outlier is that the tested platform recorded an imbalanced flap profile, where the left flap is consistently several degrees below the right flap. This appears counter-intuitive, and it needs to be understood whether this is a biased measurement, or the platform is calibrated unexpectedly. Additionally, two flights have durations significantly higher than others. Observing the altitude over time reveals that the flight start and end times span multiple flights.

However, not all anomalies are a source of bad data. Inspecting the data in Figure 7, it appears that this data is true. The platform appears to be oscillating in roll around 15°-20° with a sporadic period while trying to maintain a loiter pattern. The anomalous data is not due to inaccurate measurements, but rather unfortunate control by the Flight Control Computer (FCS). Potentially due to the platform roll rate being different than expected as a result of the flaps misalignment found earlier.

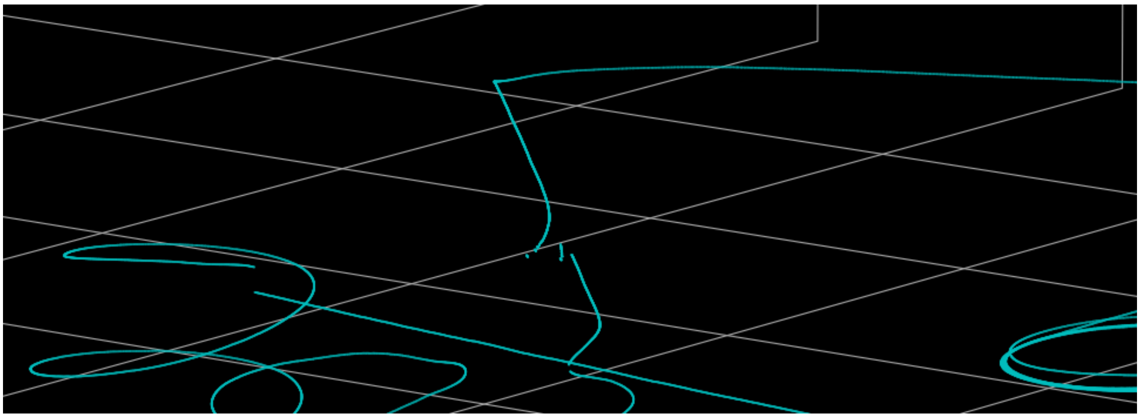

Figure 4. Discontinuity in flightpath observed in data

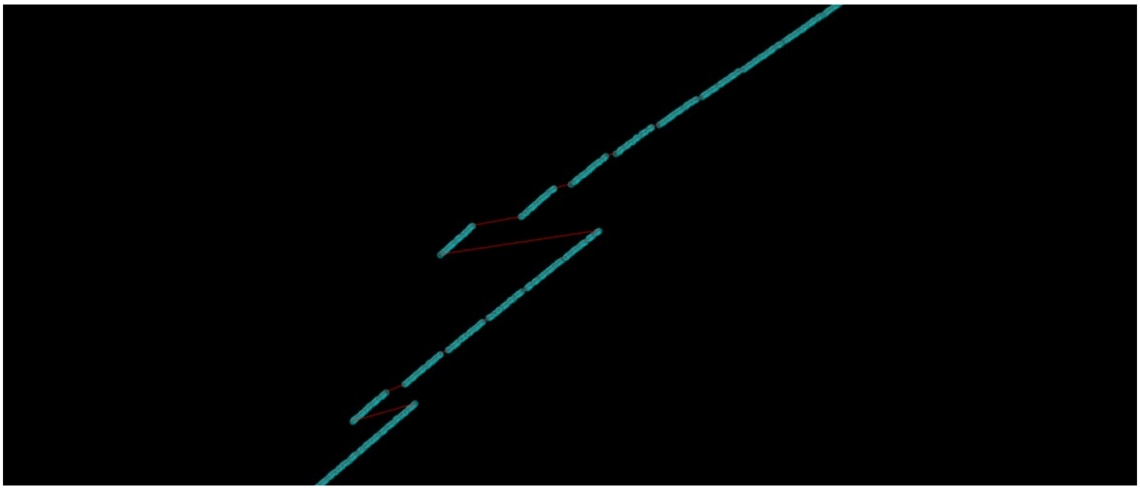

Figure 5. Discontinuity in flightpath observed in data (bird's eye view)

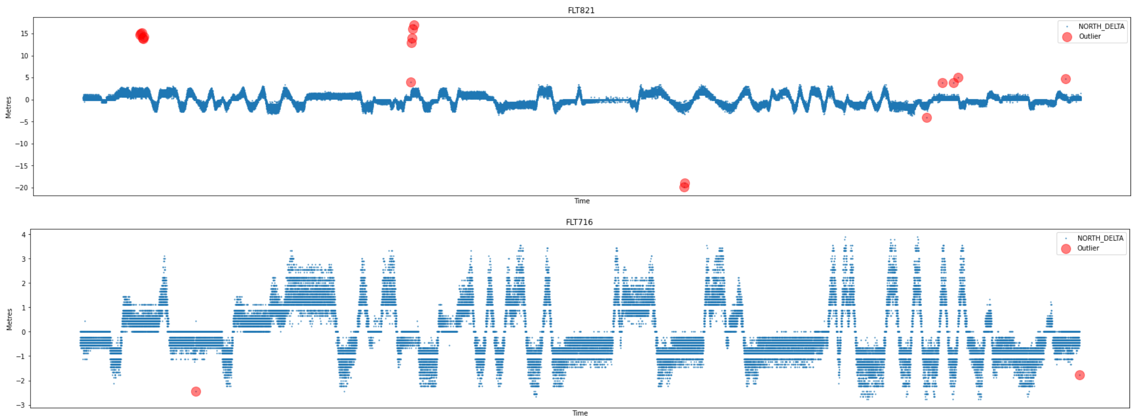

Figure 6. Distance deltas between consecutive positions in time along the North axis for two flights. As measured, sometimes the platform makes jumps in position that are out of context for the surrounding data (highlighted in red)

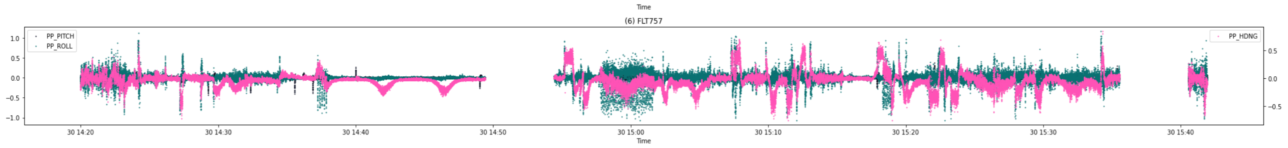

Figure 7. Contextual anomaly in roll measurements at around the 15:00 mark (roll measurements appear to fluctuate sporadically)

Data Cleaning

Since there is a large volume of data, the simplest approach is to drop flights which have fluctuations in Weight on Wheels (WOW) while airborne as well as filtering out pre-ascent and post-descent stages of a mission i.e. high WOW values since this is indicative of being on the ground. Flights which have anomalous time gaps between recordings i.e. multiple flights captured as a single flight are also dropped. Plotting the distribution of flight durations after this looks more typical (Figure 8).

Anomaly detection can also be used for data cleaning. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm which can be applied to time-series anomaly detection (Ester, Kriegel, Sander, & Xu, 1996).

Sections of track that are deemed anomalous (e.g. Figures 4, 5 & 6) are isolated and filtered out within a given time window around the anomalies to avoid any surrounding data corruption. This is done in three ways:

1. Running DBSCAN over the measured values for anomaly detection
2. Apply it to parts of track with time deltas greater than the expected 0.051s
3. Generally, all of the INS measurements update at the same frequency and interval. However, there are occasional sections of data where some measurements have become out of sync. As there is a lack of time and data to understand this fully, these instances are detected, marked as anomalies and filtered out.

After applying the above processing, the flight profiles are plot in Figure 9, with filtered sections highlighted in red. Most of the filtered sections are caused by recordings > 0.051s, filtering out entire flights in some cases.

Interestingly, the majority of other anomalies (e.g. position jumps) occur around points where there is a longer than expected time delta. Seemingly as the delay is affecting the real-time computations on board the platform, and computations and/or recordings miss intervals. Partly due to the accumulation of error over missed intervals, but also in the surrounding data (i.e. not just the next immediate time step).

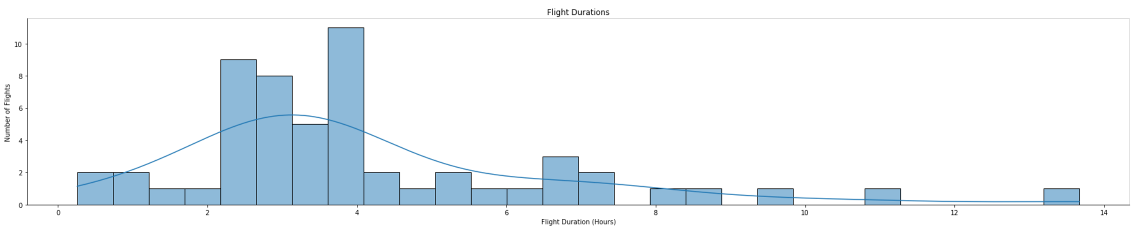

Figure 8. Histogram of flight durations

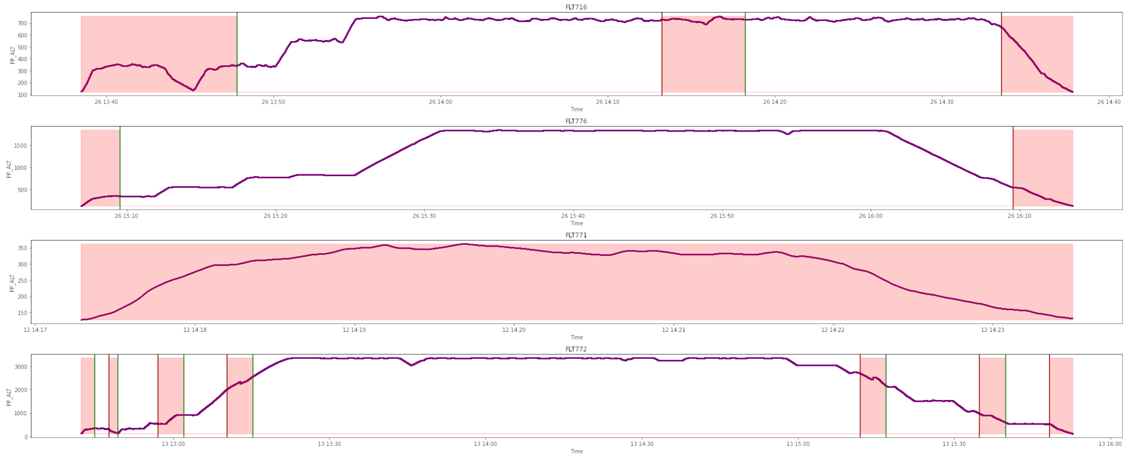

Figure 9. Altitude plots for 4 flights, with filtered sections highlighted in red

Data Manipulation

Using data available from the INS (Figure 10a) as a source of approximation and by filtering out the redundant recordings, Figure 10b is obtained. However, if it is assumed that the up-to-date measurements were sampled from a ground truth function, f(x), leads to potentially a very poor approximation with large errors (Figure 11a). Attempting to learn the response of this function would have far too much variance to be useful.

However, for both the INS measurements of orientations and velocities, the first order INS measurements of body rates and accelerations are available. As the body rates and accelerations are in body axis, transforming the coordinate system into world coordinates to match the orientations and velocities, gradient information for the orientation and velocity measurements can be obtained. In other words, information on orientation and velocity values, as well as the rates at which they were changing at those points in time.

Using this to fit a Cubic Hermite spline to the data, with respect to both the data points, and the gradient at those points' results in figure 11b. This gives a better approximation of the ground truth data. However, the INS data doesn't have sufficient information available to accurately reconstruct either a high amplitude or high frequency function. Therefore, in training a model, it should be expected that any model produced will lose some degree of high amplitude and/or high frequency behaviour. Applying this process to pitch data results in the approximation found in Figure 12.

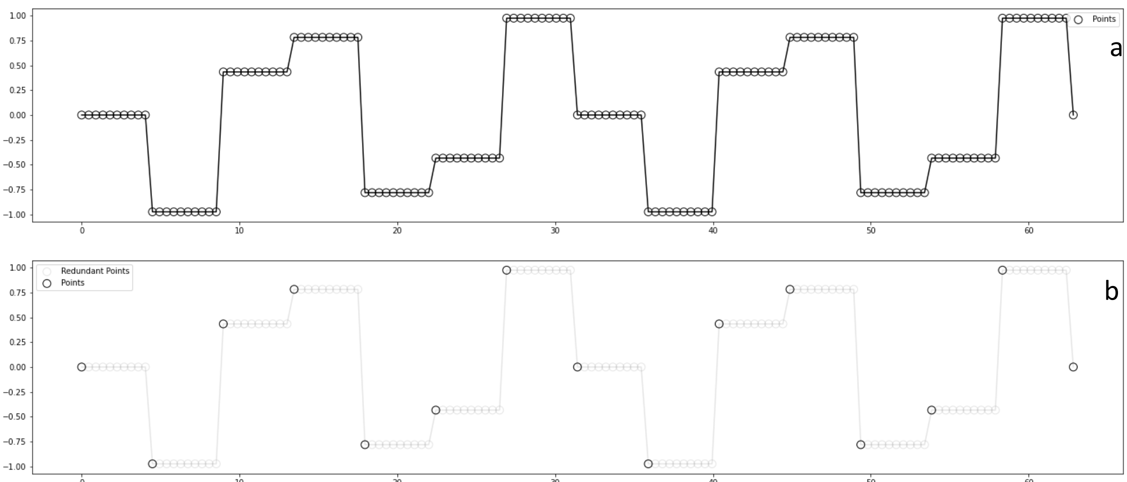

Figure 10. INS data x - axis is the average deviation from normal, y - axis is the time in seconds. a: INS Data, b: INS passed through a filter to remove redundant data

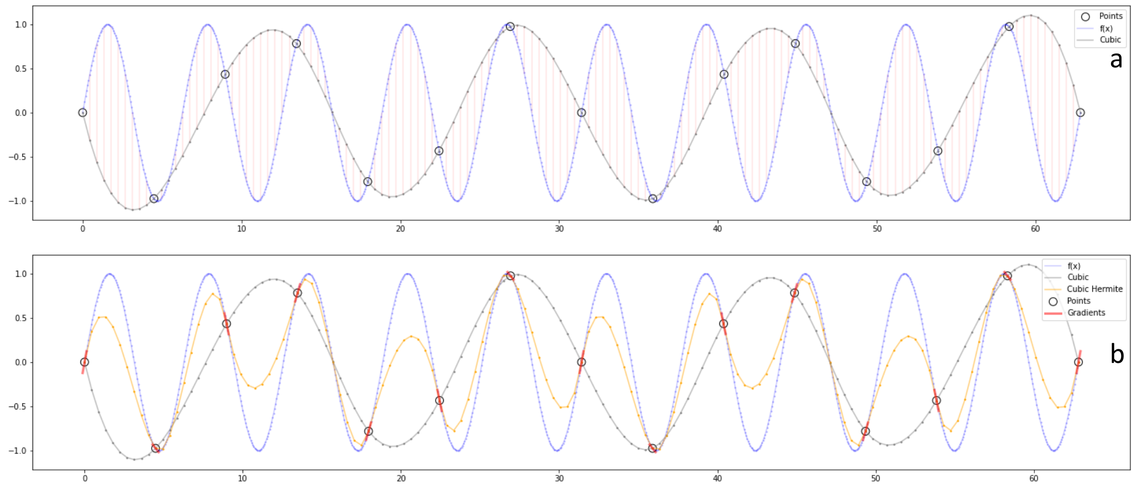

Figure 11. INS data x - axis is the average deviation from normal, y - axis is the time in seconds a: Ground truth function f(x) (blue), up-to-date measurements of the ground truth (black circles), Cubic spline fitted to the up-to-date measurements (grey), and error bars between the ground truth and the Cubic spline (red), b: Ground truth data (blue), measured values (black circles), gradients at measured values (red lines), Cubic spline (grey), Cubic Hermite spline (orange)

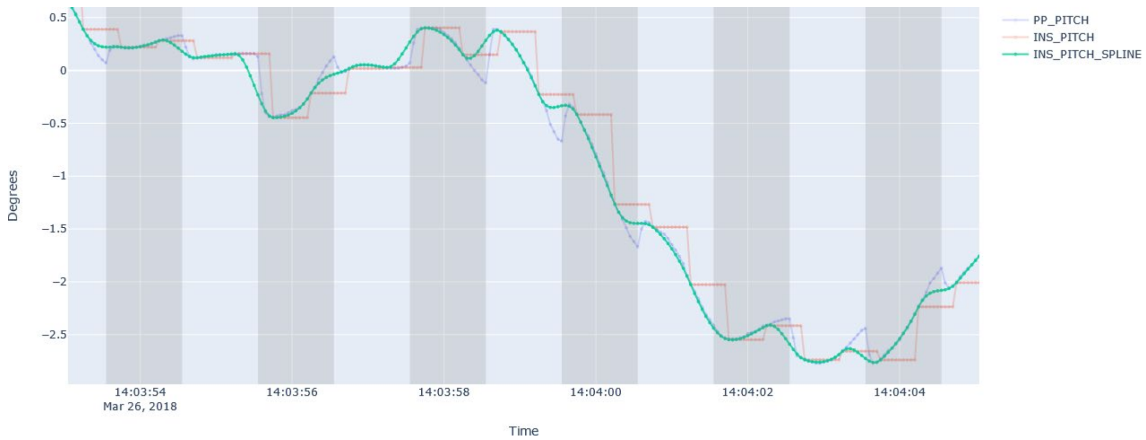

Figure 12. Real-time calculation of pitch (PP_PITCH) (blue), INS measured pitch (combined) (INS_PITCH) (red), and Cubic Hermite spline fitted with respect to pitch measurements and transformed body rate measurements (INS_PITCH_SPLINE) (green)

Data for Training

Training a neural network to approximate the function f(s, a) = s' using the processed observed data, allowed the neural network to be used as a flight dynamics model for the platform. A section of the processed recorded flight data is set aside that does not train the model with the test data. Instead, the model is queried using this unseen test data during training to see how well it can predict this data. This indicates deployment performance and shows how well the trained model has learned the underlying flight dynamics.

The model makes sensible predictions on the response to the test data. One would expect some small fluctuations in the test data around the model predictions due to observation stochasticity, as well as the low frequency of measurements causing approximation over high frequency and amplitude. In general, the model appears to be working well from the test plots (Figure 13).

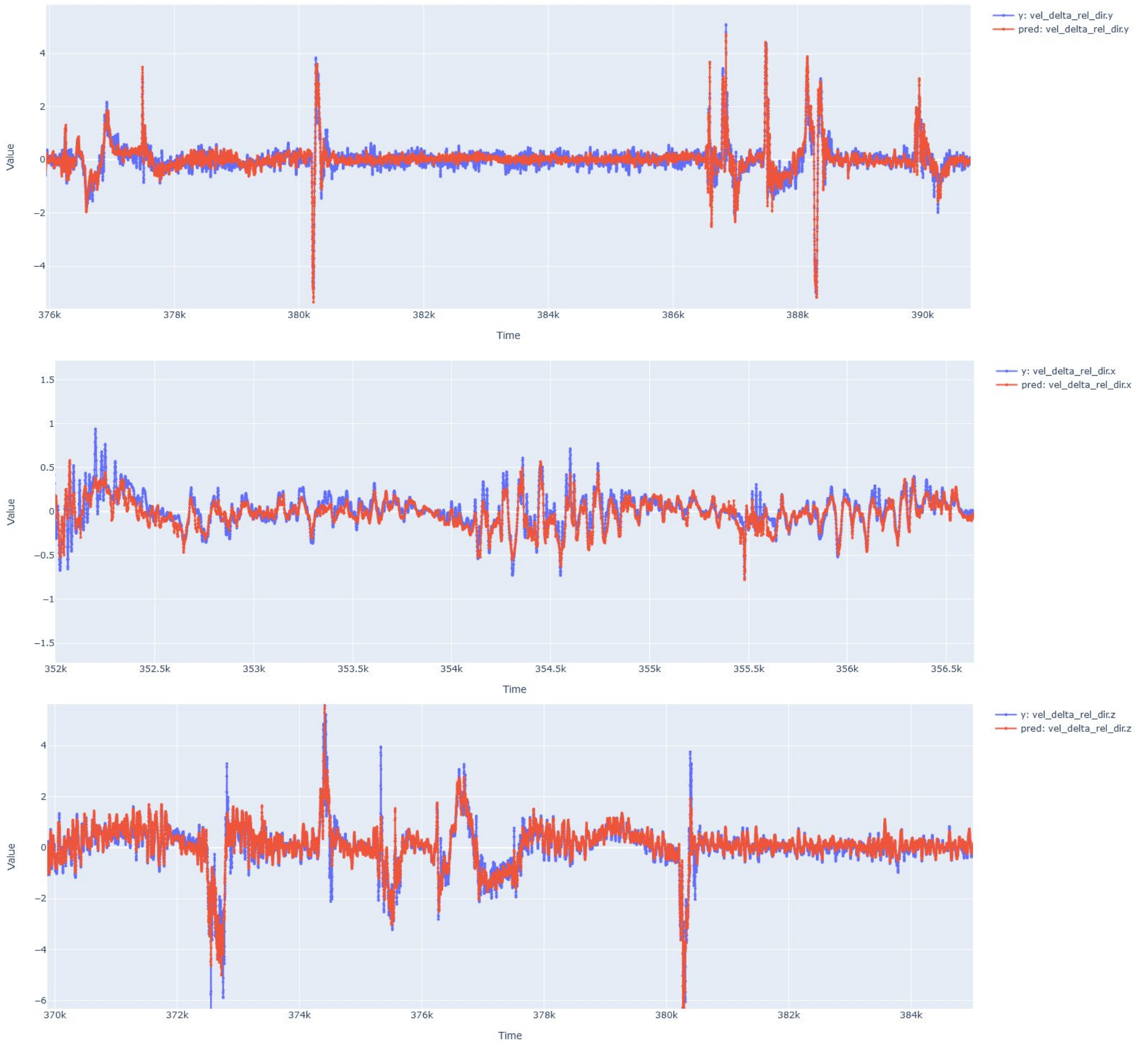

Figure 13. Longitudinal (Top), vertical (Middle) and lateral (Bottom) velocity direction delta response over time in seconds. Test data (blue), model predictions (red)

CONCLUSION

This paper has provided a high-level framework on the overall system architecture of an autonomous fault mitigation scheme. From this, the paper takes a deeper look into the architecture that would be required for both security and fault anomaly detection units. Using the UAV platform as a case study the paper then investigates the work required in order to pre-process data for it to be used for the detection units and training a DT model. Overall this paper concludes that such a system for handling faults autonomously is likely to reduce the chance of pilot overload preventing a degradation in mission performance.

Further Work

1. While individual components described by the high-level system architecture do exist, further work is required to create an end-to-end anomaly detection platform following this architecture
2. A flexible method for categorising and ordering faults in terms of their impact needs to be investigated as well as a deeper investigation into other fault tolerant control schemes
3. A fault prediction scheme should be elaborated upon to ensure faults do not occur in the future
4. A framework for the action monitor needs to be established
5. Further analysis, processing, and inclusion of available data is likely to give performance improvements. For example, in this work perceived lower impact features such as fuel and oil levels, state of electrical systems, current positions of control surfaces etc. Including further relevant features are likely to give an additional benefit

6. It needs to be tested whether the proposed high-level architecture would significantly reduce operator overload

ACKNOWLEDGEMENTS

A big thanks to Ian Henderson (Thales UK) for his support and excellent work on the UAS the UAV platform's data analysis and modelling.

REFERENCES

Ministry of Defence UK. (2017). *Joint Doctrine Publication 0-30.2 - Unmanned Aircraft Systems.* MoD.

Agrawal, A., & Cleland-Huang, J. (2021). Explaining Autonomous Decisions in Swarms of Human-on-the-loop Small Unmanned Aerial Systems. *Association for the Advancement of Artificial Intelligence*.

Andrews, J. M. (2020). *Human Performance Modeling: Analysis of the Effects of Manned-Unmanned Teaming on Pilot Workload and Mission Performance.* Ohio: Air Force Institute of Technology (USAF).

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection : A Survey. *CM Computing Surveys*.

Chen, W., Dong, Y., & Duan, Z. (2018). Manipulating Drone Dynamic State Estimation to Compromise Navigation. *IEEE*.

Diamond, D. M., Campbell, A. M., Park, C. R., Halonen, J., & Zoladz, P. R. (2007). The Temporal Dynamics Model of Emotional Memory Processing: A Synthesis on the Neurobiological Basis of Stress-Induced Amnesia, Flashbulb and Traumatic Memories, and the Yerkes-Dodson Law. *Neural Plast*.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *AAAI*.

Ferramosca, M. L. (2021). *Safety Assessment of UAV Systems: Field Data Analysis.* Politecnico di Torino.

Fourlas, G. K., & Karras, G. C. (2021). A Survey on Fault Diagnosis and Fault-Tolerant Control Methods for Unmanned Aerial Vehicles. *Machines*.

Franco, B. J., & Góes, L. C. (2007). Failure Analysis Methods in Unmanned Aerial Vehicle Applications. *COBEM.*

Huang, H., Yang, L., Wang, Y., Xu, X., & Lu, Y. (2021). Digital Twin-driven online anomaly detection for an automation system based on edge intelligence. *Manufacturing Systems*.

ITT ENIDINE INC. (2019, 2 18). *FLIGHT / PRIMARY / SAFETY CRITICAL PARTS PROGRAM MANUAL.* Retrieved from https://www.enidine.com/CorporateSite/media/itt/Resources/Distributors/EndUserDocuments/Suppliers_Documents/QAM03_Rev_E.pdf

Keipour, A., Mousaei, M., & Scherer, S. (2019). Automatic Real-time Anomaly Detection for Autonomous Aerial Vehicles. *International Conference on Robotics and Automation.*

Lu, Q., Xie, X., Parlikad, A. K., & Schooling, J. M. (2020). Digital Twin-enabled anomaly detection for built asset monitoring in ooperation and maintenance. *Automation in Construction*.

Porat, T., Oron-Gilad, T., Rottem-Hovev, M., & Silbiger, J. (2016). Supervising and Controlling Unmanned Systems: A Multi-Phase Study with Subject Matter Experts. *Frontiers in Psychology*.

Ryad, I., Zidan, M., Rashad, N., Bakr, D., Yehia, N., Ismail, Y., . . . Salem, A. (2021). Using Path Planning Algorithms and Digital Twin Simulators to Collect Synthetic Training Dataset for Drone Autonomous Navigation. *IEEE*.

Saini, R. K., Raju, M., & Chail, A. (2021). Cry in the sky: Psychological impact on drone operators. *Industrial Psychiatry Journal*.

Wang, Y., Cao, Y., & Wang, F.-Y. (2021). Anomaly Detection in Digital Twin Model. *IEEE*.

Watkins, L., Hamilton, D., Kornegay, K., & Rubin, A. (2021). Triaging Autonomous Drone Faults By Simultaneously Assuring Autonomy and Security. *CISS.*

Whitlock, C. (2014, June 20). *When drones fall from the sky*. (The Washington Post) Retrieved 06 13, 2022, from https://www.washingtonpost.com/sf/investigative/2014/06/20/when-drones-fall-from-the-sky/

Williams, K. W. (2004). *A Summary of Unmanned Aircraft Accident/Incident Data: Human Factors Implications.* Federal Aviation Administration.

Zogopoulos-Papaliakos, G., Karras, G. C., & Kyrikopoulos, K. J. (2021). A Fault-Tolerant Control Scheme for Fixed-Wing UAVs with Flight Envelope Awareness. *Intelligent and Robotic Systems*.