Certification of Safe and Trusted Robotic Inspection of Assets

Fateme Dinmohammadi
Smart Systems Group, School of Engineering and Physical Sciences, Heriot-Watt University
Edinburgh, EH14 4AS, United Kingdom
F.Dinmohammadi@hw.ac.uk

Michael Fisher
Department of Computer Science
University of Liverpool
Liverpool, L69 3BX, United Kingdom
Mfisher@liverpool.ac.uk

David Flynn
Smart Systems Group, School of Engineering and Physical Sciences, Heriot-Watt University
Edinburgh, EH14 4AS, United Kingdom
D.Flynn@hw.ac.uk

Michael Jump
Department of Mechanical, Materials and Aerospace Engineering, University of Liverpool
Liverpool, L69 3BX, United Kingdom
Mjump1@liverpool.ac.uk

Vincent Page
Department of Mechanical, Materials and Aerospace Engineering, University of Liverpool
Liverpool, L69 3BX, United Kingdom
eg0u7076@student.liverpool.ac.uk

Charles Patchett
Virtual Engineering Centre
Hartree Centre Sci-Tech Daresbury
Daresbury, WA4 4FS, United Kingdom
patchett@liverpool.ac.uk

Valentin Robu
Smart Systems Group, School of Engineering and Physical Sciences, Heriot-Watt University
Edinburgh, EH14 4AS, United Kingdom
V.Robu@hw.ac.uk

Wenshuo Tang
Smart Systems Group, School of Engineering and Physical Sciences, Heriot-Watt University
Edinburgh, EH14 4AS, United Kingdom
Wenshuo.Tang1@hw.ac.uk

Matt Webster
Department of Computer Science
University of Liverpool
Liverpool, L69 3BX, United Kingdom
matt@liverpool.ac.uk

***Abstract*—In future inspections of offshore assets utilizing robots, robots will not only be expected to collate new data from their payload of instruments, but they will also be expected to interact with the infrastructure being inspected, undertaking remedial tasks and engaging with embedded monitoring systems of the asset. This increasing level of interaction and deployment frequency of robot inspections requires an understanding of how we can embed safe and trusted operational architectures within robots. Currently, robots can undertake constrained semi-autonomous inspections, using predetermined tasks (missions) with minimum supervision. However, the challenge is that the state of the world changes with time as does the condition of the robot. Therefore, robots must be able to undertake adaptive measures to support optimal outcomes during autonomous missions. In this paper, we propose an initial architecture to the safe verification and validation of health condition and certification of robotic and autonomous inspection systems for offshore assets. Our first contribution relates to the verification and validation architecture, which takes into account risks associated with asset inspection, safety protocols, evolving ambient changes, as well as the inherent state of health of the robot. The second part of our paper looks to how prognostic analytics can be used to support robot resilience in terms of sensor drift and accurate state of health estimates of critical sub-systems. Initial results demonstrate that methods such as relevance vector machines and Bayesian networks can be used to accurately mitigate risks to autonomy.**

***Keywords*— Robotic inspection; verification and validation; asset certification; prognostic and health management (PHM).**

I. INTRODUCTION

Robotic inspection systems are becoming increasingly popular in a wide range of industry sectors including: offshore energy, nuclear, space exploration, etc., to improve productivity, quality and safety [1]. Adoption of such systems aims to support inspection and latterly maintenance services in remote and/or hazardous environments with a view to reducing operation and maintenance (O&M) costs, shortened diagnosis time, and improved health and safety for human inspectors.

The Robot Institute of America (RIA) defines inspection robots as a mechanical-electrical integration system, involving structure, control, communication, positioning system, sensors, power source, etc. which is used for the inspection of material, parts, and products to detect either inherent defects (e.g. fractures or cracks) or service induced defects (e.g. damage from use). These robots range from underwater vehicles such as autonomous underwater vehicles (AUVs) and remotely operated underwater vehicles (ROVs), to airborne vehicles such as unmanned aerial vehicles (UAVs) and drones [2, 3]. Figure 1 shows a typical ROV system and a drone that can be used for the inspection of subsea oil and gas assets and wind turbine nacelle components, respectively.

(a) (b)

Fig. 1. (a) Inspection ROV for subsea oil and gas assets (http://www.rov.org) (b) Inspection drones for wind turbine nacelle components (www.aerialmediascotland.co.uk).

Reliability, availability and maintainability (RAM), which is defined as the probability of a system operating satisfactorily in any time period and its capability of being repaired, is a key performance indicator for robotic inspection systems. Any unforeseen robot stoppage may cause an interruption in the inspection process, or even lead to a fatality, serious injury, or damage to property, equipment or environment. To minimize the probability of these incidents and improve asset availability and lower maintenance costs, it is critical to develop methodologies that are capable of verifying and validating robotic and autonomous inspection systems by means of computer vision, machine learning (ML) algorithms, as well as health monitoring, diagnostics, prognostics, and maintenance (collectively known as Prognostics and Health Management (PHM)) tools. PHM has gained considerable attention within the robot system domain as it can help inspection decision-makers increase the safety and reliability of robots while reducing their maintenance costs by providing accurate predictions concerning the remaining useful life (RUL) of critical components/systems.

To address the challenges referenced above in asset and robot self-certification, the Offshore Robotics for Certification of Assets (ORCA) Hub (https://orcahub.org/) was established with the aim of bringing together internationally leading experts from five British universities (including: Heriot-Watt, Liverpool, Edinburgh, Oxford and Imperial College London) as well as more than thirty industry partners. This Engineering and Physical Science Research Council (EPSRC) funded project will provide remote solutions using robotics and artificial intelligence (AI) that can operate and interact safely in autonomous or semi-autonomous modes in complex and cluttered environments such as subsea, ground and aerial.

In this paper, an architecture is proposed for the safe verification and validation (V&V) of health condition and certification of robotic and autonomous inspection systems in the offshore energy sector. The proposed architecture takes into account risks associated with asset inspection, safety protocols, evolving ambient changes, as well as the inherent state of health of the robot. The second technical contribution relates to how prognostic and health management (PHM) methods can support assurance in the reliability of robot functionality and robot autonomy.

The organisation of this paper is as follows. In Section II, an overview of the V&V process for robotic and autonomous inspection systems is provided. In Section III, the requirements for the implementation of PHM for inspection robots in the offshore sector are described. In Section IV, case studies are presented to demonstrate the certification process of a robot and the role of prognostic analytics in supporting robot's reliability and safety. The use of Bayesian inference is also discussed to demonstrate how sensor drift, leading to a misrepresentation of the world to the robot, can be overcome. Finally, in Section V, we conclude this study with a brief discussion on topics for future research.

II. VERIFICATION AND VALIDATION PROCESS FOR ROBOTIC AND AUTONOMOUS INSPECTION SYSTEMS

A. Autonomy

By 'autonomy', we mean the ability of a system to make its own decisions and to act on its own, and to do both without direct human intervention. This general definition covers a range of variations from automatic, wherein decisions/activities are fixed in advance, through adaptive, wherein decisions/activities are optimised with respect to (and driven by) the environment, and on to autonomous, wherein decisions/activities can include internal motivations/beliefs allowing the system to not only be driven by its environment.

An additional dimension is the role that the human operator plays in making key decisions and taking key actions. Again this can range from the whole activity being undertaken by the operator, through remote control, to selection of options by the operator or to full autonomy where the operator sets general goals/guidance but the system actually decides how, when and where to act. There are a variety of such categorisations, an important one developed for aerospace being the Pilot Authority and Control of Tasks (PACT) categorisation [4]. As responsibility for the robot's decisions and certification lies with humans in less autonomous varieties, we are primarily concerned with how systems with high levels of autonomy can be handled.

B. Regulation and certification

In developing new systems for use in practical environments, each authority will have rules, policies and laws to ensure the safe design and operation of systems, i.e.

regulations. This then leads to certification, namely checking whether the systems in question comply with the relevant regulations. Note that this primarily is a legal, rather than scientific, assessment and usually involves external review, typically by some regulators. Note also that, once a system is certified, it does not guarantee it is safe – it just guarantees that, legally, it can be considered "safe enough" and that the risk is considered acceptable.

Certification processes and regulators which are documents (usually produced by a panel of experts) provide guidance on the proving of compliance. In the case of robotic systems there are many standards that might be deemed relevant by regulators. From general safety standards, such as ISO 61508, through domain specific standards such as ISO 10218 (industrial robots), ISO 15066 (collaborative robots), or RTCA DO-178B/C (aerospace), and even on to ethical aspects (BS8611).

While there are very many such approaches, regulators and standards almost uniformly ignore the issue of autonomy, particularly full autonomy wherein systems take crucial, safety critical, decisions on their own. Standards/regulations have little to say about intelligent software that make complex decisions about safety/ethics, yet this is a fundamental part of (semi) autonomous robotic systems. With these issues, and particularly where robotic systems need to carry out self-certification, it is clear that greater autonomy needs stronger V&V techniques.

C. *Verification and validation*

The aim of verification is to ensure that a system matches its requirements. These requirements may be informal, in which case it is hard to assess if, or how, our system does indeed correspond to them, or the requirements may be explicitly formal. The formal variety is often given in a clear, precise language with unambiguous semantics. Formal verification takes this further, not only having precise formal requirements in a mathematical form, but carrying out a comprehensive mathematical analysis of the system to 'prove' whether it corresponds to the formal specification of these requirements. Formal verification is particularly used for systems that are safety, business, or mission critical, and where errors can have severe consequences. Indeed, some of the aforementioned standards prescribe such formal methods, since informal verification techniques such as testing have little certainty in their results.

There are many varieties of formal verification, the most widespread being model checking [5], whereby the formal specification is checked (usually automatically) against all possible executions of the system. Verification, via model checking, is widely used especially for the analysis of critical systems. However, its use in autonomous software is relatively recent [6], while application to the verification of practical autonomous systems is still at a very early stage [7, 8]. Though these approaches are typically used before deployment, related techniques provide the basis for run-time monitoring and compliance testing. Such run-time verification [RosuH05] is important in assessing how complex systems evolve, and ensuring that unacceptable behaviours are detected and mitigated.

Validation is the process of confirming that the final system has the intended behaviour once it is active in its target environment, and is often concerned with satisfying external stakeholders. For example, does our system match safety standards or legal rules set by regulators? Does our system perform acceptably from a customer's point of view, and how well do users feel that it works? The are many approaches to carrying out validation, incorporating diverse aspects, but typically involving the assessment of accuracy, repeatability, usability, resilience, etc.

In our context, V&V necessitates a range of techniques, from formal safety verification, through testing, to in-situ evaluation and monitoring, and it is often difficult to delineate these phases. For example, since autonomous robotic systems typically interact with the "real world", we must ensure that verification is extended to this interaction. Yet it is impossible to accurately model the real-world, with its uncertain and continuous dynamics, in a finite way and so exploration of *all* possibilities via techniques such as model-checking is infeasible. This leads to several options. such as abstraction, testing, and monitoring. We can try to abstract from the complexity of the real world and provide a finite description of this abstraction that we can then use in formal verification; this abstraction is likely to be incorrect in some way and will need refinement. A practical alternative is to use sophisticated testing methods, appealing to Monte-Carlo techniques and dynamic test refinement in order to systematically 'cover' a wide range of practical situations. Such requirements-based testing is widely used in electronic systems design. A further option is to monitor the system as it runs, detecting if it ever performs unacceptable behaviour.

D. *Hybrid agent architecture and internal models*

In order to capture (and control) higher levels of autonomy, we utilise a modular architecture with a distinguished agent/executive taking the high-level decisions [9, 10]. Note that this agent not only takes critical decisions but, crucially, is able to explain and justify its decisions. In addition, the agent controls lower-level components (and their organisation). As well as being transparent in terms of high-level decision-making, the architecture exposes the behaviour of sub-modules within the system. This approach has been used to construct and verify a range of different autonomous systems, such as satellites [7], road convoys [9], and ethical decisions [11]. Within the robotic architecture we also have four models:

1. An interaction model, used to capture modes and preferences in user interaction;
2. A self-model, wherein the robot has a dynamic description of the (expected) behaviour of its own system components;
3. A task model, capturing the set of tasks/goals (including timing/resource constraints) that the robot must undertake (for example, inspection activities); and
4. A safety model [12], capturing the safety considerations identified in initial certification.

We will not consider (1) here but note that (2) and (4) (the self and safety models) are vital for self-certification. The former helps identify changes in robot capability, while the latter delimits issues that go beyond the bounds identified in the original certification process. Furthermore, (3) and (4) (the task and safety models) are central to carrying out safe and reliable inspection of external assets. The task model will capture the required inspections and actions that characterise

effective asset certification, while the safety model again ensures that this is carried out safely. All of these aspects can potentially be verified to ensure correctness and conformity to certification expectations.

III. PROGNOSTIC & HEALTH MANAGEMENT FOR ROBOTIC AND AUTONOMOUS INSPECTION SYSTEMS

Robots and autonomous systems are playing an increasingly important role in the inspection and maintenance operations of critical assets in remote locations and/or harsh environments. The growing demand for improving the reliability, survivability and overall success of robot supported missions, has led to an increased emphasis on robot safety and reliability. In this Section, the principles of Prognostics and Health Management (PHM) in robotic inspection are introduced, its specific requirements are described and the steps for implementation of the PHM are described in detail.

A. PHM principals in robotic inspection

PHM is an approach to the system life-cycle support that seeks to prevent unforeseen stoppages through accurate monitoring, incipient fault detection, and prediction of impending faults [13]. This approach has been an emerging discipline to scientifically manage the health condition of robotic systems and their critical components. The PHM aims at developing measurement tools to promote the design, test, V&V of inspection technology for robot systems.

An inspection robot is a complex system consisting of several interacting components including robot arms, sensors, control systems, end-effectors, process tooling, power supplies, and software. To successfully perform a task, the robot system needs to deliver the position and orientation accuracy of the tool centre position (TCP), the trajectory of the arm, the correct speed, force, and torque. The constituent sub-systems and components of an inspection robot degrade with use in harsh conditions [14]. The degradation of a robot system can lead to a decrease in inspection quality (e.g. in terms of probability of detection of a defect) and efficiency (e.g. in terms of duration and cost). One of the objectives of PHM systems in robotic inspection is to predict RUL of the robot system or its individual components as they degrade from an initial state to a failure state.

The RUL prediction is regarded as one of the key issues in PHM and it is generally defined as the time interval during which the asset's performance satisfies certain qualitative criteria. The RUL of an industrial robot can be predicted by extrapolating degradation patterns and by using three main approaches, namely model-based, data-driven and fusion-based (integration of model and data driven) approaches. The model-based approach is typically based on the utilization of physics of failure models of the degradation, while the data-driven approach is based on the transformation of the degradation data provided by the sensors into models that represent the pattern of the degradation. Figure 2 depicts an RUL prediction strategy for a robot system based on a pattern extracted from the observed degradation data.

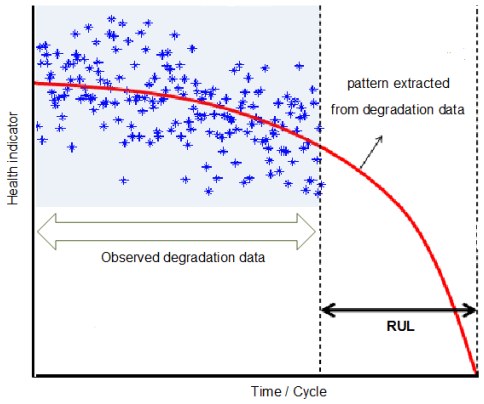


Fig. 2. RUL prediction model for a robot system based on the actual degradation data.

By prediction of the RUL, a plan for calibration and preventive maintenance of robot components and control equipment can be optimised. Furthermore, the insight provided by the RUL prediction model informs Design for Reliability (DFR) for next-generation robots.

B. PHM requirements for inspection robots

The application of PHM to inspection robots differs from that of traditional applications of PHM, such as simple rotating machinery. The use of PHM in robotic inspection systems brings new challenges to both uncertainty management and false alarm mitigation. There have been several recent advances for inspection robots to enhance their sensing, computing and fault simulation capabilities for hosting PHM functions. However, there are still several requirements needed to be fulfilled to increase the capability of PHM for inspection robots.

The PHM can be applied to inspection robots through the development of reference datasets, testing technologies, and analytics supporting tools (see Figure 3). Modern robotic and autonomous inspection systems are equipped with a variety of sensors, where each one monitors a part of the system's state. As a result, historical operational data from real-world systems are abundant. However, the majority of the data samples often belong to the healthy region of robots' operation, long before complete failure. Consequently, healthy and faulty data samples are not equally represented in the reference datasets, leading to a skewed dataset.

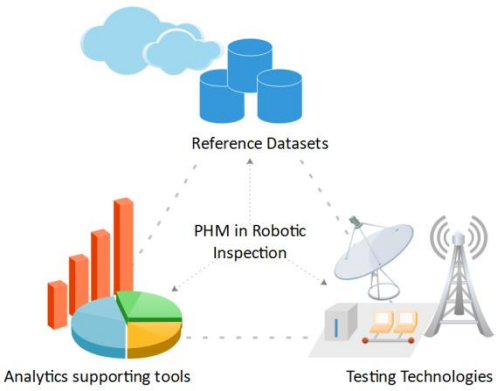


Fig. 3. The key requirements for the application of PHM to robotic inspection.

Over the past years, various analytical approaches have been proposed to support PHM of the robotic and autonomous systems. These tools deal effectively with high dimensional data, abstracting the important information, helping as such the user to extract the meaningful part of the data. Among these approaches, machine learning (ML) has been shown to be theoretically sound, generically applicable and demonstrating promising results especially on applications where online prognosis is required. Broadly, there are three types of ML algorithms [15]:

(*i*) Supervised learning algorithms in which a target variable (dependent variable) is predicted from a given set of predictors (independent variables). In other words, a function is generated to map inputs to desired outputs using a set of variables. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of such algorithms include regression, decision tree, random forest, k-nearest neighbours, logistic regression, etc.

(*ii*) Unsupervised learning algorithms in which no target or outcome variable is predicted but it is used for clustering the assets or failure modes in different groups for specific intervention. Examples of such algorithms are Apriori algorithm and K-means.

(*iii*) Reinforcement learning algorithms through which the robot is exposed to an environment where it trains itself continually using trial and error to capture the best possible knowledge to make accurate inspection decisions. An example of such an approach are Markov decision processes (MDPs).

Bayesian methods have been widely used for reinforcement learning, yielding principled methods for incorporating prior information into the robot's learning process [16]. A Bayesian Network (BN) is a graphical representation of conditional dependencies between a set of random variables. In other words, a BN is used to decompose the joint probability of a set of random variables (i.e., variables that are sampled from a probability distribution) to a product of simpler factors, exploiting the conditional dependences of the latter. A special case of BNs is that of Dynamic Bayesian Networks (DBNs). DBNs relate random variables at different instances in time. Assuming that X = $\{x_i\}$ and Y = $\{y_i\}$, i=1,…,k, the joint probability of all states and observations across time is written as follows:

$$P(X,Y) = \prod_{i=1}^{k} \overbrace{P(y_i|x_i)}^{update} \prod_{i=1}^{k} \underbrace{P(x_i|x_{i-1})}_{predict} P(x_0) \quad (1)$$

Given the joint probability distribution, any query combination can be answered by setting the variables that represent the observations and marginalising (summing over all possible events) the variables that are neither evidence nor query variables. Marginalisation is what hinders efficient computation of the queries, sometimes making them even computationally intractable. To this end, several algorithms have been developed both for answering the exact question or an approximate one, i.e., exact and approximate inference respectively [17, 18].

C. Implementation of PHM in robotic inspection

PHM can be applied to both the infrastructure being inspected and the robot that is used to support inspection services. PHM for infrastructure asset management has been extensively addressed in the past decade and several methodologies for implementation were presented, such as V-diagram, WEAR methodology, etc. A circular methodology to implement PHM in robotic inspection is illustrated in Figure 4.

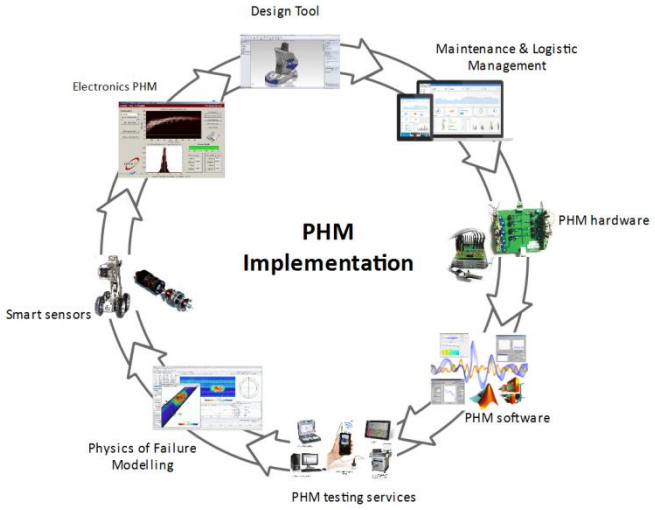


Figure 4. Steps for implementation of PHM in asset management.

PHM for robotic inspection systems can be implemented in two levels: the robot system level and robot component level. Robot level PHM assesses the health of the overall system by taking into account its architecture, function, and field-related parameters (such as depth of water in which the ROV operates). Component level PHM is typically focused on monitoring the health of individual components (e.g., arms, sensors and electronic devices) to determine if the health of the monitored component is degraded.

The key building blocks of the PHM for an inspection smart pig are shown in Figure 5.

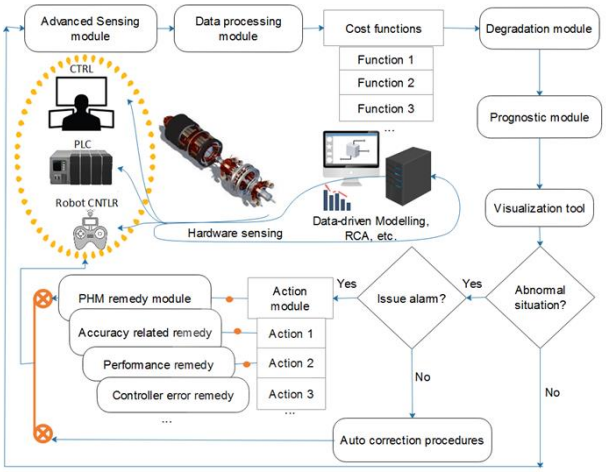


Figure 5. Key elements of the PHM for an inspection smart pig.

As can be seen, the first key building block is the advanced sensing module for PHM (shown in the upper left of Figure 5). Advanced sensing technologies are developed to measure and monitor the robot's health status, with three sensing layers: a system layer, a component layer, and an add-

on layer [13]. The system layer aims to support the overall system's health assessment, including repeatability, accuracy, velocity, force, and torque; the component layer extracts data from the robots' controllers and/or embedded sensors to perform the on-line monitoring; the add-on layer promotes the inclusion of additional sensors to provide information that the component and system layers may be neglecting. The second key building block is the data processing module. This module will focus on the development of reference algorithms to fuse data captured from multiple sensors employed in the advanced sensing module. The third key building block is the development of algorithms for robot system health assessment and PHM V&V methods.

The PHM model accuracy depends on the robust prediction of the system's state and its RUL. Moreover, the system's model needs to be adaptive, to account for variance in dynamic parameters, as well as for diverse operating conditions. To accomplish that, locally weighted projection regression (LWPR) can be used, given part of the state history and the current input.

$$y_n = f(y_{n-k:n-1}, u_{n-m:n-1}) \tag{2}$$

Equation (2) describes the general case of an Auto-Regressive model with Exogenous inputs (ARX). In typical ARX models $f(.,.)$ is a polynomial of the inputs and the previous states of the system.

In the following Section, case studies are used to demonstrate the role of prognostic analytics in supporting robot reliability and safety.

IV. CASE STUDIES

A. A robot's lithium-ion battery

The majority of autonomous robots utilise a battery power supply, typically a Lithium-ion battery. Although these batteries are rechargeable, cell-aging caused by irreversible chemical reactions during usage leads to reduction in capacity overtime. Therefore, how the robot is used will affect the depth of discharge which has a distinct influence on battery RUL. As a critical sub-system within the robot, the state of health of the lithium-ion battery needs to be monitored during operation in order to help assure safe robot operation, functionality and reliability. Andoni *et al.* [19] applied a Relevance Vector Machine (RVM) technique which was a Bayesian treatment of support vector machine for battery's RUL prediction. The Bayesian treatment led to probabilistic predictions and allowed arbitrary kernel functions to be utilised. In the study, an iterative expectation–maximization (EM) algorithm for RVM training was implemented. The EM algorithm can directly avoid the step of optimizing hyperparameters.

The battery data used to conduct this experiment are obtained from the open-source, life cycle test data repository of the National Aeronautics and Space Administration (NASA) Ames Prognostics Centre of Excellence (PCoE). To measure the RUL prediction error of the algorithm, we define error AE and relative error RE as:

$$AE = \left\|R - \check{R}\right\| \; and \; RE = \frac{\left\|R - \check{R}\right\|}{R} \tag{3}$$

where R and $\check{R}$ represent the actual and the predicted RUL values, respectively. These values are shown in Figure 6.

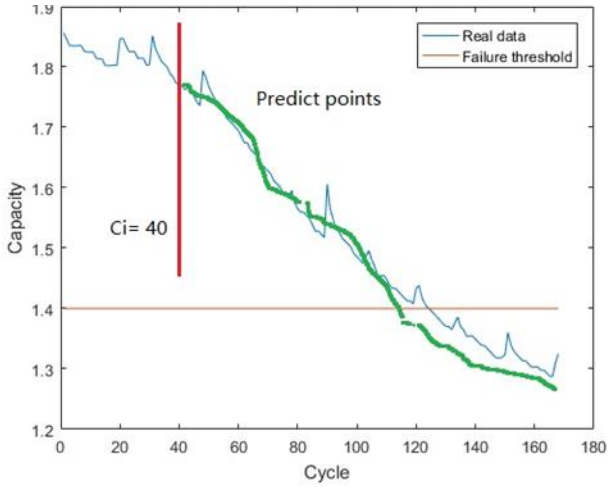


Figure 6. Real data (blue) and predicted (green) remaining useful life estimates based on 40 learning cycles.

In terms of sensing, all autonomous agents are equipped with various sensors that measure physical quantities of interest. The information from the sensors is used to yield an estimation of the state of the world; specifically, the part of the world that is relevant to a prescribed task (mission). Based on this estimation, the autonomous agent deliberates how to proceed towards mission completion. Next, the result of this deliberative process, namely the "plan", is executed.

In most applications the environment is dynamic; i.e., the operational conditions are subject to change without prior notice. In this case, the agent needs to adapt the original plan to account for the new condition. However, the agent may receive erroneous data that leads to a misrepresentation of the world around it due to sensor drift or failure. Long-term autonomy entails sophisticated inference mechanisms to cope with the dynamic traits of a real-world environment. It is of profound importance to ensure that inference, as described above, is based on accurate information. To accommodate sensor drift and failure, Fagogenis [16] implemented an incremental kernel regression for dynamic modelling. Algorithms of this sort are easy to train and are highly adaptive. Adaptivity allows for model adjustments, whenever the environment of operation changes. Bayesian reasoning provides a rigorous framework for addressing uncertainty. Moreover, using Bayesian Networks, complex inference regarding hardware degradation can be answered. Specifically, adaptive modelling is combined with Bayesian reasoning to yield recursive estimation algorithms that are robust to sensor failures. In this work, two solutions are presented by extending the existing recursive estimation algorithms from robotics literature. The algorithms are deployed on an underwater vehicle and the performance is assessed in real-world experiments. A comparison against standard filters is also provided. Next, the previous algorithms are extended to consider sensor and actuator failures jointly. An algorithm that can detect thruster failures in an AUV has been developed. Moreover, the algorithm adapts the dynamic model online to compensate for the detected fault. The performance of this algorithm was also tested in a real-world application. One step further than hardware fault detection, prognostics predict how much longer can a particular hardware component operate normally. Ubiquitous sensors in modern

systems render data-driven prognostics a viable solution. However, training is based on skewed datasets; datasets where the samples from the faulty region of operation are much fewer than the ones from the healthy region of operation. A prognostic algorithm that tackles the problem of imbalanced (skewed) datasets is developed through the implementation of RAS boosting and linear regression analysis.

B. Certification of UAVs

One of the key components of any autonomous robotic system is the decision-making system, which makes decisions based upon what is being sensed around and within the robot. If the robot is part of, or is, a safety critical system, demonstrating that the robot is 'safe enough' through physical testing is likely to be impractical. This issue was addressed at for an unmanned aerial system [20]. Here, the pilot was replaced by a rational agent with the agent being programmed to follow a small number of key Rules of the Air [21]. A simulation environment was created that linked high-fidelity multi-body flight dynamics models with rational agents performing pilotage and air traffic control functions as well as a route planner and a visualisation engine [22]. The subsequent mission simulations were then able to be run in real-time or, using co-located high performance computing facilities, faster than real time.

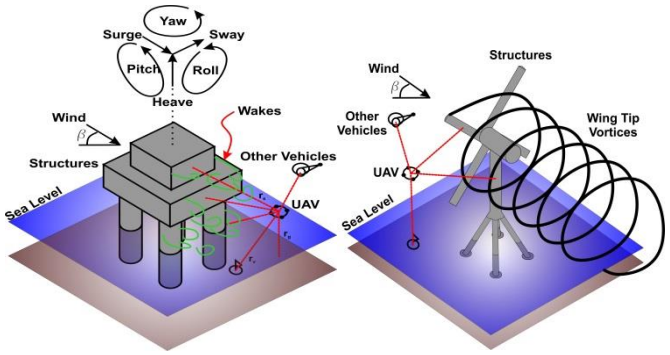
The advantage of using a rational agent is that it can be programmed in a way that allows the logic within it to be formally verified using a formal method called model checking [20]. The Rules of the Air have been developed over at least 100 years of manned flight (and are based upon the Rules of the Sea, developed over an even longer timescale) and so can be inconsistent. Two specific rules were included in the agent that are, by inspection, in conflict with one another. The model checking process was able to detect this conflict and the decision-making logic was modified accordingly.

In real time, the efficacy of the rational agent acting as pilot was able to be assessed through a simulated flight from Aberporth to Sumburgh in the UK. A hazardous situation was introduced whereby a second aircraft was introduced into the environment on a flightpath that would, if no action were taken, result in a head-on collision. It could then be demonstrated that the agent correctly turned the aircraft to the right to avoid the oncoming intruder aircraft.

The detection of the intruder aircraft, of course, relied upon the sensor models within the simulation. An added benefit of using the simulation environment is that the sensor model can be up- or down-graded at will. Using high-performance computing facilities, thousands of simulations can be run with different effective sensor ranges (for example). In this way, it was possible to determine the sensor range and flight conditions for which the system would no longer be able to maintain a pre-defined safe separation distance from the intruder aircraft.

The case study briefly described above showed that the coupling of formal verification methods and traditional engineering modelling, simulation and testing can provide a powerful means either to generate evidence that the operation of a robotic system (in this case, an unmanned aircraft) can be achieved as required, or, to show under what conditions it would not. We combine two strands of existing work. The first is the simulation environment described above for UAV certification evidence; the second is work carried out on the simulation of ship air wakes, with a particular interest in manned helicopter operations [23]. Ship air wakes are generated using computational fluid dynamics techniques and introduced into the multi-block flight dynamics models as air flow perturbations. The aim of this work is to be able to perform initial helicopter-ship trials in simulation such that the overall cost of such trials can be reduced.

In the context of the offshore case, to achieve a particular inspection mission, the UAV rational agent pilot must plan a safe route through a potentially turbulent environment. Unwanted collisions with the offshore platform and any other vehicles must be avoided, whilst ensuring that the UAV only enters areas of disturbed air flow for which it has sufficient control power and disturbance rejection capability. Figure 7 illustrates this for two offshore asset cases. In the lee of the oil rig, there will be a wake, which is dependent on the geometry, wind speed, wind direction, sea state, and (if the oil rig is a floating one) the motion of the platform. For the wind turbine case, the blade tip vortices create a rotating flow structure of vortices, which forms in a helix in the lee of the turbine. This wake is dependent on the geometry of the turbine, wind speed, wind direction, and turbine speed. For both cases, the wake dissipates as it travels downstream of the structure, which means that the hazard presented by these flow structures reduces with the distance from the structure.



(a) (b)

Figure 7. (a) Factors when operating around an oil rig (b) Factors when operating around a wind turbine (only one vortex core shown for clarity)

Pilots currently avoid this problem by following simple rules, such as separation distances and times. However, these are unlikely to be applicable or helpful for this mode of operations. The challenge when operating in these environments will be to provide the UAV with the internal risk model that allows it to know where the dangerous areas are and integrating that information into its path planning and monitoring functions. Such models form part of both the system's safety model and its task model, as described above.

If the systematic testing and verification of the autonomous system and the simulation of the interaction between wakes and aircraft can be combined, then this will allow a simulation system to be built that will allow both the decision-making elements and the engineering components of the system to be tested to generate evidence to regulators to allow them to be confident that a given operation/mission is 'safe enough'.

Due to the high computational requirements the above risk model cannot feasibly be recomputed on-board the inspection

vehicle. Instead, the model will contain broad risk maps, pre-computed for a range of different wind speed/directions and structures. These are then used as part of the system's safety model and its task model; the former to ensure that the vehicle remains safe in its environment, the latter to ensure that risk of damage is taken into account when planning inspection routes.

This then brings the two threads together: a 'safe' autonomous vehicle, based on a broad risk model; with prognostics and asset inspections that it must carry out, taking into account safe operations. This is clearly a cyclic process – the result of inspection and prognostic procedures will input to the rational agent's decisions about how important it is that various structures be inspected, and also how much `risk' is acceptable. The potential for verifiability of all these components is crucial in ensuring that these important systems remain within acceptable bounds.

V. CONCLUSION AND FUTURE RESEARCH

Global investment in robots to support inspection services in remote locations and/or hazardous environments is driven by factors such as the reduction in operation and maintenance (O&M) costs, minimizing breakdown time, and eliminating risks to humans and hazards to assets. The adoption of robotics into many spheres of life, especially when interacting with critical assets, will be dependent on the safe and trusted inspection. In this paper, we present an architecture for the safe verification and validation (V&V) of health condition and certification of robotic and autonomous inspection systems in the offshore energy sector. A number of PHM analytics tools including Machine Learning (ML) techniques such as kernel regression for dynamic modelling when the environment of operation changes, or the Bayesian reasoning to provide a rigorous framework for addressing uncertainty were briefly introduced. In addition, Bayesian Networks (BNs) were explained to provide complex inference regarding hardware degradation. Finally, it was discussed how Relevance Vector Machines were used to provide accurate predictions of the remaining useful lifetime (RUL) of robotic assets. Our future research will explore the role of online probabilistic modelling and the role of self-learning systems in autonomous operations.

ACKNOWLEDGMENT

The authors acknowledge the funding of the EPSRC Offshore Robotics for Certification of Assets hub (https://orcahub.org) (EP/R026173/1).

REFERENCES

[1] V. Robu, D. Flynn, and D. Lane, "Train robots to self-certify their safe operation", *Nature*, 2018, 553(7688), pp. 281.

[2] G.P.S. Carvalho, M.F.S. Xaud, I. Marcovistz, A.F. Neves, R.R. Costa, "The DORIS offshore robot: recent developments and real-world demonstration results", *IFAC-PapersOnLine*, 2017, 50(1), pp. 11215-11220.

[3] L. Tang, E. Hettler, B. Zhang, and J. DeCastro, "A testbed for real-time autonomous vehicle PHM and contingency management applications", In: *Annual Conference of the Prognostics and Health Management Society*, 2011, pp. 1-11.

[4] M. C. Bonner, R. M. Taylor, and C. A. Miller, "Tasking interface manager: affording pilot control of adaptive automation and aiding". In: *Contemporary Ergonomics*, pp 72-76, CRC Press, Taylor & Francis, 2000.

[5] E. M. Clarke, O. Grumberg, and D. Peled, "Model Checking", MIT Press, 1999.

[6] L. A. Dennis, M. Fisher, M. Webster, and R. H. Bordini, "Model checking agent programming languages", *Automated Software Engineering*, 19(1):5-63, 2012.

[7] M. Fisher, L. A. Dennis, and M. Webster, "Verifying autonomous systems", *ACM Communications*, 56(9):84-93, 2013

[8] L. A. Dennis, M. Fisher, N. K. Lincoln, A. Lisitsa, and S. M. Veres, "Practical verification of decision-making in agent-based autonomous systems", *Automated Software Engineering*, 23(3):305-359, 2016.

[9] L. A. Dennis, M. Fisher, N. Lincoln, S. M. Veres, , and A. Lisitsa, "An agent based framework for adaptive control and decision making of autonomous vehicles", In: *IFAC Proceedings Volumes*, 43(10): 310-317, 2010.

[10] R. Woodman, A. Winfield, C. Harper, and M. Fraser, "Building safer robots: safety driven control", International Journal of Robotics Research 31(13):1603-1626, 2012.

[11] M. Kamali, L. A. Dennis, O. McAree, M. Fisher, and S. M. Veres. Formal Verification of Autonomous Vehicle Platooning. Science of Computer Programming, 148:88-106.

[12] L. A. Dennis, M Fisher, M. Slavkovik, and M. Webster. Formal Verification of Ethical Choices in Autonomous Systems. Robotics and Autonomous Systems, 77, 1-14, 2016.

[13] G. Qiao, C. Schlenoff, and B.A. Weiss, "Quick positional health assessment for industrial robot prognostics and health management (PHM)", In: IEEE International Conference on Robotics and Automation, 29 May-3 June 2017, Singapore.

[14] G. Qiao, and B.A. Weiss, "Accuracy degradation analysis for industrial robot systems", In: Proceedings of ASME International Manufacturing Science and Engineering Conference, June 4-8, 2017, pp. 1-9, Southern California, USA.

[15] A. Dey, "Machine learning algorithms: a review", International Journal of Computer Science and Information Technologies, 2016, 7(3), pp. 1174-1179.

[16] G. Fagogenis, "Increasing the robustness of autonomous systems to hardware degradation using machine learning", PhD thesis, 2016, Institute of Sensors, Signals and Systems, School of Engineering and Physical Sciences, Heriot-Watt University.

[17] G. Fagogenis, V.D. Carolis, and D.M. Lane, "Online fault detection and model adaptation for Underwater Vehicles in the case of thruster failures", In: IEEE International Conference on Robotics and Automation, 2016, pp. 2625–2630.

[18] G. Fagogenis, D. Flynn, and D.M. Lane, "Improving Underwater Vehicle navigation state estimation using Locally Weighted Projection Regression", In: IEEE International Conference on Robotics and Automation, 2014, pp. 6549–6554.

[19] M. Andoni, W. Tang, V. Robu, and D. Flynn, "Data analysis of battery storage systems", In: International Conference on Electricity Distribution, 2017, Glasgow, United Kingdom.

[20] M. Webster, M., Cameron, N., Fisher, M., and Jump, M. (2014). Generating certification evidence for autonomous unmanned aircraft using model checking and simulation. *Journal of Aerospace Information Systems*, 11(5), 258-278.

[21] Civil Aviation Authority, "The Air Navigation Order 2016 and Regulations (CAP 393)," 2016. Available at: https://publicapps.caa.co.uk/docs/33/CAP393_E5A3_MAR2018(p).pdf. Last accessed 23/4/18.

[22] N. Cameron, M. Webster, M. Jump, and M. Fisher, "Certification of a civil UAS: a Virtual Engineering approach. In: *AIAA Modelling SImulation and Technologies Conference and Exhibition*, 8-11 August 2011, pp. 1-15, Portland, Oregon.

[23] C.H. Kaaria, Y. Wang, M.D. White, and I. Owen, "An experimental technique for evaluating the aerodynamic impact of ship superstructures on helicopter operations," *Ocean Engineering*, Vol. 61, 2013, pp. 97-108.