Annual Reviews in Control 50 (2020) 13–28

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Vision Article

On the requirements of digital twin-driven autonomous maintenance

Samir Khan [a,*], Michael Farnsworth [b], Richard McWilliam [c], John Erkoyuncu [d]

[a] Department of Aeronautics and Astronautics, University of Tokyo, Japan
[b] Department of Automatic Control and Systems Engineering, University of Sheffield, United Kingdom of Great Britain and Northern Ireland
[c] School of Electronic and Electrical Engineering, Durham University, United Kingdom of Great Britain and Northern Ireland
[d] School of Aerospace, Transport and Manufacturing, Cranfield University, United Kingdom of Great Britain and Northern Ireland

ARTICLE INFO

Keywords:
Digital twin
Autonomous systems
Maintenance
Fault detection and isolation
Reinforcement learning

ABSTRACT

Autonomy has become a focal point for research and development in many industries. Whilst this was traditionally achieved by modelling self-engineering behaviours at the component-level, efforts are now being focused on the sub-system and system-level through advancements in artificial intelligence. Exploiting its benefits requires some innovative thinking to integrate overarching concepts from big data analysis, digitisation, sensing, optimisation, information technology, and systems engineering. With recent developments in Industry 4.0, machine learning and digital twin, there has been a growing interest in adapting these concepts to achieve autonomous maintenance; the automation of predictive maintenance scheduling directly from operational data and for in-built repair at the systems-level. However, there is still ambiguity whether state-of-the-art developments are truly autonomous or they simply automate a process.

In light of this, it is important to present the current perspectives about where the technology stands today and indicate possible routes for the future. As a result, this effort focuses on recent trends in autonomous maintenance before moving on to discuss digital twin as a vehicle for decision making from the viewpoint of requirements, whilst the role of AI in assisting with this process is also explored. A suggested framework for integrating digital twin strategies within maintenance models is also discussed. Finally, the article looks towards future directions on the likely evolution and implications for its development as a sustainable technology.

1. Introduction

From the work on massively redundant design (Von Neumann, 1956) to modern-day active fault management (McWilliam, Khan, Farnsworth, & Bell, 2018), the challenge of self-maintenance and repair by autonomous means is considerable and traverses multiple system design levels. Much of this activity has been centred on the microelectronics domain due to the compatible nature of configurable architectures. However, resilience is achievable at multiple design levels, whether through software or hardware techniques (Naghshbandi et al., 2020). A range of hardware-based techniques have recently emerged that must be better understood by hardware designers, made easier to evaluate against another and readily combined with other robust engineering techniques such as integrated health monitoring, autonomous systems and assisted fault diagnosis. Although the idea of systems resilience and autonomous maintenance has been an area of interest for many years, it has often been discussed in an abstract sense (Lee, Ghaffari, & Elmeligy, 2011). The manufacturing industry has highlighted the importance of having all-inclusive automated operational processes (Reynders, Houbrechts, & De Roeck, 2012), and similar trends have been seen in the automotive (Price, Snooke, & Lewis, 2006) and aerospace domains (Dale et al., 2007; MacDonnell & Clegg, 2007). This can be attributed to recent advances in technology that has enabled an array of innovations. However, much of these works often consider the term *autonomous* to be synonymous with *automation*; where an automated driving vehicle is labelled to be a feature of autonomous driving without much consideration of the requirements of autonomous systems. Instead, progress has focused on modernising components and streamlining processes by introducing novel sensors and computing technologies.

Achieving true autonomy certainly represents a quantum technological leap in contrast to automation, and even though there have been several instances of digitising the physical space (such as buying/selling stocks or auctions) as a process of automation, most of these applications do not warrant autonomy. Similarly, for system health management, a typical integrated design will have arisen from an automated design approach based on local system thresholds without knowledge of the dynamic state of the system. This often encompasses

* Corresponding author.
E-mail address: khan@ailab.t.u-tokyo.ac.jp (S. Khan).

https://doi.org/10.1016/j.arcontrol.2020.08.003
Received 10 June 2020; Received in revised form 24 August 2020; Accepted 24 August 2020
Available online 10 September 2020

static logic for detecting, isolating, and recovering from faults verified (and validated) through exhaustive testing prior to operation. Further, these processes remain anchored to the role of providing human-centric automation. The past few decades have shown a proliferation in this area, with research and solutions targeted to combat various faults occurring through the component level all the way up to the systems level (Huynh, Barros, & Bérenguer, 2014). However, even though these concepts have been studied extensively (Patton, Frank, & Clark, 2013; Zhang & Jiang, 2008), most methods often require triggering mechanisms that are intelligent enough to collect data about the failing component, the nature of the fault, and its severity on the overall system performance. Whilst these technologies are typically focused on fault detection and isolation within individual subsystems, the growing maintenance costs facing today's engineering industry have prompted further research into novel architectures that reduce maintenance, repair and overhaul of complex high value assets. As a consequence, future effort needs to look towards the integration of anomaly, diagnostic and prognostic technologies across autonomous systems and related platforms that bring about the capability to dynamically predict and isolate impending faults and failures. This will help to maintain system performance in a more cost-effective way whilst identifying in-operation issues. A knock-on effect of this will be increased importance of frameworks for data exchange within diagnostic technologies. By way of example, future generations of aerospace vehicles will depend upon reduced mass while being subjected to higher functional loads and more extreme service conditions over longer time periods than the present generation of vehicles (Glaessgen & Stargel, 2012). The associated requirements placed on thermal protection will range from propulsion and energy storage to avionics; which is greater than previously experienced while demands on long-term reliability will increase. Thus, the extensive legacy of historical flight information will likely be insufficient to either certify new vehicles or to guarantee mission success. To investigate this technological gap, the authors concentrate on self-learning technologies that are emerging within maintenance systems that exhibit a degree of autonomy. The transition to autonomous systems should bring with it the potential to approach zero-maintenance–a variety of strategies that ultimately provide the elimination of maintenance-centric costs–and its relation to through-life engineering services for a range of high-value products and assets (Farnsworth, McWilliam, Khan, Bell, & Tiwari, 2017; McWilliam et al., 2018). Within this paradigm, the overarching focus remains to carry out maintenance but limiting (or even removing) human intervention.

It should be noted that highly skilled knowledge is still valued across all industries because they can successfully anticipate risks and make critical optimisation decisions far sooner than any automated system. However, with ever increasing system complexities and process uncertainties, it has become critical to produce data-based solutions that map the overall evolution of all relevant systems and their processes. Also, problems associated with maintainability of equipment and the degrading overall system performance indicate that systems need some form of 'intelligence' by using reasoning testbeds, that should also have the capability to adapt in unknown situations. In this context, the concept of a digital twin has gained a lot of traction by leveraging a digital platform to quickly build high-fidelity simulations and models that can make informed decisions and create the ability to query/analyse results. The authors argue that if a digital twin is treated as a 'living' entity, it can offer the potential of monitoring and improving the functionality of interconnected complex engineering systems. This can be achieved by combining it with recent developments in machine learning to develop a data-driven digital twin that is not only capable of managing its health but also carry out maintenance and repair when required. This has a huge potential to disrupt traditional maintenance practices across all industries.

As a result, this article investigates the requirements of a digital-twin driven solution for autonomous maintenance, which warrants the development of a multi-modal digital/virtual model describing the system maintenance model at various levels of components, physics, process data, behaviours and rules. The digital twin enables detection of behavioural anomalies, which are potentially attributable to faults within the equipment.[1] Fundamentally, this can help to influence/determine rules defined by the system requirements. While many existing publications have focused on manufacturing applications, there is an increasing interest from the system health management domain. The authors have noted that most existing digital twin implementations are application- or equipment-specific, and as such, there seems to be no systematic way to select, design or implement them. As a result, the focused is placed on investigating basic questions such as: what is the relevant taxonomy in the field? what is its potential for autonomous maintenance applications? what are its enabling technologies? and what are the applications-specific open questions? This template provided an easier and succinct understanding of the concept. Whilst taking note of its strengths and limitations in the application domain, the authors aimed attention towards establishing:

- Recent trends on the application of autonomous maintenance;
- Requirements for autonomous maintenance using digital twin;
- Challenges with current technological capabilities;
- A suggested framework for integrating digital twin strategies within maintenance models.

1.1. Research methodology

One of the goals of this study is to understand the recent research progress in digital twin for autonomous maintenance. This is reflected through the following research questions:

- Can a digital twin introduce autonomy for in-situ self-maintenance and autonomous repair capability?
- What are the requirements for such a digital twin-driven framework?
- How can this concept be used for (and supported by) autonomous maintenance?

This is accomplished by investigating existing published material that yields insights into potential applications and academic interests that define the current major trends, significant works, and future directions. As research within this area is of practical importance, the scope of this investigation mostly covers publications over the past 10 years. To accomplish the study aims, this research is based on reviewing a variety of journal articles, all of which are directly related to the topic. Due to the scope and diversity of these methods, articles are found to be scattered across a range of sources, and thus a literature search was conducted using the electronic databases including: IEEE Xplorer and Scopus. The primary descriptor used is "digital twin", grouped with the following: "system health management", "fault diagnosis", "autonomous" and "maintenance". The authors have written this article in a way that allows the readers with a non-technical background to gain an understanding of the concept. In total, the authors shortlisted 30 published journals for this study as detailed in Table 1, which lists the journals that have been targeted for their publication. The authors also included an additional 14 articles on system health management relevant to the topic discussion.[2]

[1] These can also be used to identify any defects within the model, allowing it to be made more robust.

[2] These are Eti, Ogaji, and Probert (2006), Huynh et al. (2014), Khan, Liew, Yairi, and McWilliam (2019), Kobayashi and Simon (2005), Lee, Siu, Cruz, and Yetman (2016), Lv, Wen, Bao, and Liu (2016), McDuff, Simpson, and Gunning (1989), Nelles (2013), Patton et al. (2013), Russell and Benner (2010), Shin and Jun (2015), Sutharssan, Stoyanov, Bailey, and Yin (2015), Volponi, DePold, Ganguli, and Daguang (2003) and Zhang and Jiang (2008).

Table 1
Breakdown of the targeted journals in the past 10 years for publishing works related to digital twin and autonomous maintenance.

Journals	No of articles	References
IEEE Access	7	Cupek, Drewniak, Ziebinski, and Fojcik (2019), Khajavi, Motlagh, Jaribion, Werner, and Holmström (2019), Leonardi, Messina, and Santoro (2019), Liu, Chen, Zhang, Yang, and Chu (2019), Qi and Tao (2018), Xu, Sun, Liu, and Zheng (2019) and Zhao et al. (2019)
IEEE Transactions Journals	5	Dai, Xiang, Li, Xing, and Zhang (2011), Hegazy and Hefeeda (2014), Jain et al. (2019), Sujil, Kumar, and Bansal (2018) and Tao, Zhang, Liu, and Nee (2018b)
ACM Transactions Journals	3	Dutt, Jantsch, and Sarma (2016), Koch, Mancuso, West, and Bestavros (2019) and Shen, Tan, Lu, Wu, and Qiu (2013)
International Journal of Production Research	2	Ding, Chan, Zhang, Zhou, and Zhang (2019) and Wang, Ye, Gao, Li, and Zhang (2019)
CIRP Annals	2	Erkoyuncu, del Amo, Ariansyah, Bulka, Roy, et al. (2020) and Tao, Zhang, Liu, and Nee (2018a)
Robotics and Computer-Integrated Manufacturing	2	Leng et al. (2020) and Lu et al. (2020)
Microelectronics Reliability	1	McWilliam et al. (2018)
AIAA Journal	1	Li, Mahadevan, Ling, Choze, and Wang (2017)
Systems	1	Madni, Madni, and Lucero (2019)
IET Electric Power Applications	1	Venkatesan, Manickavasagam, Tengenkai, and Vijayalakshmi (2019)
IET Collaborative Intelligent Manufacturing	1	Lee, Azamfar, Singh, and Siahpour (2020)
IEEE Systems Journal	1	Lins, Givigi, Freitas, and Beaulieu (2016)
Systems Engineering	1	Araguz, Bou-Balust, and Alarcón (2018)
Mechanical Systems and Signal Processing	1	Booyse, Wilke, and Heyns (2020)
International Journal of Aerospace Engineering	1	Tuegel, Ingraffea, Eason, and Spottswood (2011)

1.2. Terminology

An early description of being *autonomous*, or having *autonomous properties*, was defined as a system's ability to self-control behaviour (Mele, 2001). Since then, this description has misguided many developments that classify systems as autonomous, but are rather automated — which is simply to use various predefined mechanisms to operate a system (or process). Another description of being ***autonomous*** is to have independence or self-governing capabilities (Calvo, Peters, Johnson, & Rogers, 2014). Absolute adherence to this interpretation indicates that *true autonomy* is something that (even today) does not exist and that any conventional developments have not gone beyond the automation concept.

Automation is as the ability by which a machine (continuous or digital) can reduce the work done by humans. This corresponds better to the original description and distinguishes itself from autonomy. The self-governing property of an autonomous system implies a degree of self-awareness to real-world consequences of any action taken, in a practical, data- or model-driven sense. The pursuit of *true autonomy* dictates that autonomous systems should be able to go beyond the analysis of historical data sets and use of decision-making models developed from, e.g., supervised rules, fuzzy logic, neural networks to include:

- Competitive survival strategies (e.g., adversarial networks) (Leng et al., 2019);
- Planning/scheduling for centralised optimisation, (e.g., using AI for data analytics, assuming that all the data is available and processed in a timely manner);
- Decentralised/self-organising (e.g., building a multi-agent system, assuming the data is distributed in different nodes) (Leng, Yan, Liu, & Xu et al., 2019).

Clear and consistent definitions for the degree of autonomy is an ongoing topic of debate within most technological areas.[3] However, the contemporary view of autonomy–and of greater relevance to this article–is that it is the capability that enables a system to function in the presence of uncertainty. This requires a degree of adaptation within a working environment, short and long term reasoning/planning in the presence of set goals, a generalisation of knowledge and skills, and importantly robustness to perturbations that arise from this uncertainty.

Maintenance is defined as a process of preserving a condition (or situation) and thus functional capability (McWilliam et al., 2018). It is the combination of all technical and administrative actions, including supervision actions, intended to retain an entity in, or restore it to, a state in which it can carry out a required function. A maintenance echelon, or maintenance line, is a physical location within an organisation where specified levels of maintenance are carried out, e.g. a repair shop. They are characterised by the skill of the personnel, the facilities available, the location, etc.

Maintenance, and its associated repair activities, are expected to achieve a high success rate in all modifications that take place during the system life cycle. This includes identification of a root cause, if there is one, or positive identification that there is no root cause. Only in this way can the correct and most appropriate maintenance activity be carried out, allowing the integrity of a removed unit to be established and hence for it to be safely returned to service. Therefore, its strategy looks to improve the overall process either by redesigning the asset itself (continuous design integration), or by paying attention to external factors that may contribute to the process in some way. These incorporate not only key characteristics that assist maintenance tasks, particularly with regard to automation, but also exhibit some independence and self-governing attributes.

Digital twin: There is still some ambiguity about the definition of a digital twin: is this just collected sensor data or does it encompass engineering data from, e.g., is it just CAD models that provide virtual sensor information from physics-based models? What about feedback to engineering, e.g., via CAD models and physics simulations, as part of the product life-cycle? Some authors simply put that a digital twin is essentially a model (or simulation) of the system, process or service, inside a computer that has the support for other technologies like cloud computing, IoT or machine learning (Wang et al., 2019). This view highlights the individual effort required to develop a comprehensive, integrated multi-physics model across technological disciplines. Provided the most accurate, physically realistic and robust models can be integrated, they will form a valid basis for certification of vehicles by simulation and real-time, continuous, health management. Ambiguity arising in this area is not only due to the wide breadth of application areas, but also because many studies have consisted of either highly conceptual/abstract aspects or extremely application-specific case studies (Cronrath, Aderiani, & Lennartson, 2019). Nonetheless, they have discussed the various challenges associated with understanding the underlying physics models whilst accounting for system state behaviour.

[3] A good example of this are standards in the automotive industry that defines automation levels from 0–5. According to the SAE classification, Level 5 defines a complete automation system. Also, self-driving is used synonymously with autonomous which typically leads to confusion (Inagaki & Sheridan, 2019; Lee, 2018).

Within the context of this article, the digital twin is regarded by the authors as a *simulation-of-simulations* that may be described by several representations, e.g., process graphs, space–time environments or statistical models. The more realistic the simulated environment, the better the expected gains from digital twin-based techniques. However, complexity will increase, often with diminishing returns from the output for each added dimension, indicating that there will still be model uncertainties associated with simulations, optimisation, control and hardware/software limitations.

2. Literature background

The need for autonomy in manually operated engineering applications can arise due to several reasons. E.g., the human involvement in monitoring and controlling an asset is either expensive or might not be possible due to communication delays (Hegazy & Hefeeda, 2014), or the maintenance personnel might not be able to operate directly within the external environments (Hancock, 2019), detecting and predicting system behaviour to maintain system health and management of distributed components at a system-level (Sridhar & Kopardekar, 2016). This section, therefore, underpins the constituent components and requirements for an autonomous maintenance system.

2.1. The role of autonomous maintenance

Autonomous maintenance is defined as a preventative maintenance strategy focusing on the system being ‘self-governing’, and effectively performing maintenance activities through the cooperation between maintenance personnel and operators to eliminate sources that affect system availability. The predominant aim is to reduce system breakdown and maintenance costs, which is achieved by fulfilling the following objectives:

- Understand the functions (and the components) of the system and detect the causes of abnormalities (Khan et al., 2019);
- Recognise possible quality issues and identify their root-causes (Eti et al., 2006);
- Timely detection of abnormalities to self-heal[4] (Frei et al., 2013).

This is not entirely a new concept: basic autonomy would enable a system to operate within a dynamic environment independent of external control. There is, however, a spectrum of autonomy that ranges from local autonomy within a subsystem, where actions may be executed in response to stimuli or local information, to a system-level autonomy, which manages actions and handles constraints across subsystems. A by-product of autonomous systems is improved performance with a reduced burden on operating personnel and achieving efficient control in dynamic environments. However, for maintenance purposes, autonomy should also aim to reduce both the mean time between human interventions and the number of functions performed per intervention. Several authors have advocated that this may be achieved by understanding co-dependencies of sub-systems, cross-domains and coupled properties[5] (Endsley, 2017), all of which constitute key capabilities required to enable support equipment to achieve cost-effective maintenance and proactive health management. Of course, the level of autonomy will depend on the number of humans needed to operate a system, but these capabilities help support system operational reliability, safety and maintainability. Despite this apparent complementary relationship between human and autonomous maintenance, consider the following practical limitations facing the successful realisation of such capability:

- Inherent uncertainty associated with the environment warrants added redundancy to address issues that might hinder certain actions. This should not prevent the system from achieving its main goal (McWilliam et al., 2018);
- Operating environment is an afterthought (Farnsworth et al., 2017). As a result, system capabilities (e.g., physical connections or data transfer) to interact with the environment can become limited;
- Existing health monitoring methods use a limited number of modelling parameters, only focus on the operational status information, have insufficient attention to production data and maintenance records. This results in low reliability in health assessment and prediction (Khan et al., 2019);
- Health supervision systems are designed for specific models, and the framework is not flexible enough to be put into operation. It is difficult to add new monitoring objects after installation (Cupek et al., 2019; Zhao et al., 2019);
- Physical implementation limitations, e.g., a lack of on-board computation and storage or scaling up the state of the technology to more complex scenarios (Koch et al., 2019);
- There is no particular level of target autonomy.[6] This can present engineering with options on how a maintenance system should be developed.
- Most existing mature health supervision technologies use broken-then-repair or planned maintenance strategies, which results in high cost, long cycle and poor reliability in equipment maintenance (Khan & Yairi, 2018);
- It is non-obvious how autonomous maintenance actions should liaise with human operators when this is expected or necessary.

This list is by no means complete, but it serves to highlight the key barriers that have hindered progress in the field. The level of autonomy depends on the complexity of decision-making in the chain and the operations required to bring back or maintain function.[7] Yet, autonomous maintenance can be a critical cross-cutting technology that will improve performance and reduce the risk factors faced by human-based exploration (crew vehicles, habitats), robotic (spacecraft, rovers, in-situ systems), and aeronautics (airspace, airport, and aircraft) applications. The obvious technological benefit comes from the resulting automated hardware and software systems able to identify off-nominal behaviours, analyse resulting data to identify probable causes and effects, take action to keep the system operating and alert ground control (diagnostics). The provision of on-board decision making capability enables greater access to real-time telemetry and system-state data.[8] As prognostic and diagnostic systems become an integral part of the system architecture, solutions will be sought based on model-based and data-centric techniques. These approaches amount to a paradigm shift when developing autonomous systems, which makes the barrier for their adoption seem greater.[9] However, the technology has broad applicability to most future robotic and crewed missions. For example, the complexity of operating a crewed interplanetary vehicle is perhaps

[4] Depending on the level of maintenance autonomy.

[5] This can be achieved through, e.g., AI, expert systems.

[6] Since autonomy is progressive, there can be five levels of autonomy as used by the automotive industry, i.e., basic, managed, predictive, adaptive, and autonomic (Lee et al., 2011). The higher the level, the more autonomous the system becomes.

[7] This is not a trivial task as it requires replacing the medium for decision making, that was previously performed by a human operator, by a machine.

[8] But with limited computation and storage. Conversely, ground software would have limited and non-real-time data due to communication delays and bandwidth limitations, but has excessive computation, given access to super-computing resources and data storage.

[9] This is because models need to ensure correctness, sufficient data should be available to identify nominal behaviour and given the probabilistic nature of these approaches, verification and validation of such capabilities can be notoriously challenging.

comparable to that of a nuclear submarine. The latter typically has over one hundred crew members. The former, the interplanetary vehicle, has to be managed by a crew of less than half a dozen, which would require a significant level of autonomy for system-health management given limited communication.

2.1.1. Challenges with current technology

Analysis models targeting heterogeneous data sets are limited by large uncertainties and conflicting information. This arises from the fact that, when only partial models are available to synthesise supporting analytics, human input is still needed to identify subtle correlations and final decisions. Meeting future performance expectations for quality of service, estimation of computation time for decision-making, and time to make a decision within on-board systems that involve large real-time sensory data streams requires next-generation on-board resources.[10] Therefore, the aim of autonomous systems is to offer intelligent platforms capable of: identify off-nominal behaviour, efficient analysis of data to identify probable causes/effect, initiation or recommendation of action(s) to maintain system operating envelope and bringing important metrics/actions to the attention of relevant personnel. These operational goals infer the following technological requirements:

- Record low false positives ($< 3\sigma$) and false negatives ($< 3\sigma$)[11];
- Respond in a timely fashion, which is context-specific but generally within the time window for corrective action (time to criticality);
- Adapt to novel situations (generalisation), such as failures or degradation in performance of subsystems;
- Learn from, and effectively integrate, past experience;
- Exhibit flexible and adaptable behaviour that is appropriate and scalable to mission complexity;
- Timeliness and accessibility to system data, with sufficient on-board computational resources to analyse and cache/archive current/historical data trends respectively;
- Anticipate future events and states, e.g., predictive modelling of impending faults and remaining useful life based on current/past data trends and inferred system health state. Prognostic data must be generated from available monitor/sensing capability.

These key attributes will dictate how well an autonomous maintenance system can be adapted to particular missions of increasing complexity. Many of them warrant timeliness and accessibility to system data, with sufficient on-board computational/storage resources for analysing and storing current and historical data trends. Having on-board components can help seamlessly access real-time telemetry data. In contrast, off-board components can be assumed to have limited access to data (due to communication delays and bandwidth limitations), but can still benefit from excessive computation capabilities. As a result, current technology should consider a fusion of model-based and data-centric techniques that ensure model correctness, whilst having sufficient data to identify/learn behaviours and perform validation. This amounts to a paradigm shift in developing such systems, making the barrier for their adoption greater.

From this, the expected gains to system maintenance and dependability are: **improved safety:** autonomous maintenance will not only reduce the burden on the maintenance staff but also improve safety during difficult environments (such as space) and poor visibility by preventing human errors. However, this also makes it prone to cyber-attacks and hacking; **increased reliability**: a byproduct of the technology is the built in reliability for maintenance operations using systematic technical problem-solving skills (centralised maintenance), rapid detection and warning, preventive maintenance and scheduling. This breaks away from human expert judgements which might still be important during mission critical situations; **lower infrastructure cost**: it is expected to bring down long-term operating costs due to the use of system-wide energy efficiency and optimised maintenance schedules, however, this also warrants continuous connectivity of data and the communication environment between systems components.

2.2. Requirements for system health management

Health management is the process of diagnosing and preventing system failures, whilst predicting the reliability and remaining useful lifetime (RUL) of its components (Shin & Jun, 2015). Its realisation requires three main constituents: fault detection, fault classification and fault prediction. In the past, process monitoring, equipment monitoring, and performance monitoring were performed independently, which would lead to inconsistent monitoring of equipment and the loss of process control performance due to a lack of effective maintenance practices. By contrast, today's technologies for system-health management benefit from a variety of tools that depend upon statistics-based (Khan et al., 2019), reliability-based (Zhao, Quan, & Cai, 2014) and model-based (Liao & Ḱottig, 2016) techniques, often including the use of data-centric and adaptive model-based approaches. Ignoring the possibility of coupled effects between different subsystems, state of the art systems can reconfigure themselves after diagnosing faults using reasoners that adapt through learning processes. Such systems have been developed and proven within testbeds at various centres, e.g., the Jet Propulsion Laboratory's (JPL) Beacon-based Exception Analysis for Multi-mission (BEAM) that uses an integrated, on-board or off-board data analysis for fault detection, anomaly detection, and prognostics (Mackey, James, Park, & Zak, 2001). Another JPL testbed is the Spacecraft Health Inference Engine (SHINE), a high-speed expert system (stateless rule-based system) and inference engine for the diagnosis of spacecraft health (James & Dubon, 2000). Model-based approaches are used for maintenance information while physics-based models are used to predict future failures. For example, the Ames Research Centre inductive monitoring system that uses data mining clustering techniques to isolate off-nominal interaction between parameters (Iverson et al., 2012). Another prominent example is the G2's real-time expert system, an artificial intelligence expert based system that has been demonstrated for on-board payload monitoring and is also in use within some commercial satellite facilities for control of formation systems (Moore, Rosenof, & Stanley, 1990). Such developments emphasise the importance of both physics-based and data-driven approaches for the development of robust health management platforms that enable informed decisions to be made and analysis of the outcome.

There is also a drive towards improving system performance by delivering more reliable assets, that possess a higher intrinsic availability.[12] In the midst of relentless operational pressures and reduced time available for diagnostic investigations, there is great value in arranging several data collection sources that may be used to provide context-rich information (e.g., operating variables, environmental conditions, etc.) upon the occurrence of a disruptive event (Russell & Benner, 2010). Despite these capabilities, these evolvable systems are still comparatively disparate and, with the ever-increasing size of operational data sets, remain coupled with the complexities of contextual components (Onori, Semere, & Lindberg, 2011). This creates barriers that were not anticipated during the design phase of the system life cycle can thus result in subsequent speculative replacements and higher overall levels

[10] Since even aircraft off-load such things to base monitoring, there is a concern for the computational/energy resource needed for this.

[11] Some might argue that $<3\ \sigma$ is a rather stringent criterion but it is close to optimal. Up to 99.9% of the samples in a Gaussian distribution will lie within $\pm 3\ \sigma$.

[12] Here we define availability as the percentage (typically measured in days per year) when the system is ready for operation. Time spent in the shop, either for scheduled maintenance or for an unscheduled (unexpected) event that requires special attention, detracts from the system's availability.

of uncertainty during the diagnosis process (Khan et al., 2019). The higher levels of interdependence between assets have made it difficult (if not impossible) to assess why certain failures appear and yet, the underlying engineering environment is expected to support the technological platforms as well as system availability requirements. Within this context, novel approaches are required that recognise and address anomalies during operation; as well as better decision-making mechanisms at the system-level. This requires new capabilities for monitoring in-service operation, recording and distributing expert knowledge, and securing the robust operation of critical on-board software.

In the past, researchers (McDuff et al., 1989) have used neural networks on F16 flight line data for diagnostics purposes to acknowledge the capability to carry out multiple fault diagnoses, prediction/reconfiguration, and the ability to work with inaccurate or incomplete rules. Adaptive resonant theory was then used to train the data due to its ability to learn faster than other methods. Since different fault scenarios can be used to verify the efficacy of a given approach, researchers found that integrating different diagnostic methods and developing a hybrid approach for this purpose to be effective. Volponi et al. (2003) used Kalman filters and neural networks methodologies to find the malfunction and deviations from the normal engine behaviour (Volponi et al., 2003). Another possible combination is the integration of neural networks (NN) with the genetic algorithm (GA) optimisation method. In Kobayashi and Simon (2005), a NN part of the scheme is applied to engine components fault diagnostics while the GA is applied to sensor bias detection and estimation. Integrating the two methods exploits their respective benefits; NN enable nonlinear estimation and GA methods bring increased robustness. The results indicated improved fault detection and reduced false alarms.[13] More recently, deep learning methods have been introduced to look at fault diagnosis and learn the deep architectures of fault data (Lv et al., 2016). The research makes use of stacked autoencoders to improve network learning capability with respect to classification accuracy and demonstrates the potential of deep learning. Importantly the authors noted that, unlike image data, fault characteristics can vary over time making them difficult to classify and hence they pursued deep learning architectures.

2.2.1. Neural networks as detectors vs. decision makers

The proliferation of NNs within AI applications has allowed practitioners to make system health management decisions beyond the classical "if–then and do" commands for complex actions. This brings major advantages for signal retrieval and analysis: the knowledge-base contains all possible architectures corresponding to the considered fault modes and can be used to account for various attributes required for learning. Hence the decision maker computes and stores tables or curves of diagnostic indexes for different faults whilst working across different operating conditions and modalities. This approach can also archive heuristic rules and expert knowledge gathered from in-field experience to help to overcome the common problem of incomplete system models. These techniques have been applied extensively for condition monitoring and fault diagnosis (Nelles, 2013). For health management activities, in particular, NNs are often employed as statistical modelling and prediction algorithms, which can be regarded as either density estimation and prediction or classification and regression (Sutharssan et al., 2015). Lee et al. investigated the use of Convolutional Neural Networks (CNN) for analysing acoustic signals in the midst of noise (Lee et al., 2016). The authors were motivated by the fact that most existing signal analysis methods are largely dependent on the physical behaviour/characteristics of the system being analysed, which warrants regular re-tuning of algorithms for new acoustic profiles. Although, training for a deep learning system can be slow; runtimes of these systems are usually quite fast, particularly when running on GPUs. By comparison, traditional methods are significantly slower than deep learning methods during test time. It should be noted that only recently organisations have been focusing on optimising neural based computations, and in the near future are expected to see silicon chips that are designed especially for these systems. Another strategy for reducing training time is to precondition the input data to extract (remove) features that are not considered to be important.

2.2.2. Challenges for autonomy

Regardless of the technique used, real-time requirements and theoretical formulations must be implemented by an efficient algorithm. There are predominately three issues in this respect and they occur in various forms: data sampling considerations, the size of data and the implementation architecture. In practice, these issues are not solely associated with AI implementation but for real-time systems in general. Nonetheless they are interdependent and therefore, it is important to first understand the nature of the problem. Firstly, many health management design engineers still lack the knowledge to develop machine learning architectures. A second challenge is the cost of design and implementation. The development of an autonomous health management system that is to be integrated into the entire system architecture is currently far more expensive than traditional distributed systems, especially since greater volumes of data is required train a network for varying fault characteristics. The use of synthetic fault data for training is not ideal and can result in incorrect performance attributes of real world applications.[14] However, this is a research field in its own right as it can be difficult to find explicit mathematical models due to system complexity and uncertainties. As a result, the accuracy of the results will decrease. Furthermore, it is important to ensure appropriate data quality and latency, i.e., the uncertain time stamps between when data is generated and when that data is available to use. This is a time-domain requirement for AI maintenance systems that is distinct from other domains.

2.3. Requirements for data processing

Assuming integration issues can be overcome, efforts should then focus on making delivering multi-source and multi-format data whilst preserving its relationship to multiple sources. This is crucial for AI detection and classification of subtle correlation patterns between data events that to an external observer exhibit no apparent obvious link. A traditional approach is to manually establish relationships through some logic. Unfortunately, this can only be achieved over a limited number of direct correlations and therefore does not adequately reflect complexities encountered in the real world. An alternative is to train machine learning models to automate this process. However, this typically addresses limited dimensions of the whole system complexity, i.e., correlating causal events in a predictive maintenance focus, and will not lead to sufficient control for the system to interpret its operation situation (independence), nor the ability to shape outcomes by adjusting control parameters (self-governance). From this, the following critical enablers for the development of autonomous maintenance are identified:

[13] This due to better direct recognition of events or rejection of noisy/non-correlated events.

[14] But is used in a lot of cases in the absence of a better alternative. It is worth making a distinction here between the more mainstream AI application areas such as image processing, where it is fairly common to synthesis data sets. This is important for high value data-driven application like medical AI since data acquisition is a really expensive step and is the reason why so many medical AI startups have multi-million investment value. In contrast, data is comparatively cheaper in maintenance applications but the problem lies in the cost of integrating adding acquisition hardware into the testbed, and/or end product, in a way that meets the requirements for AI training and on-line improvement.

- It should be assumed that required data will not (always) be available, not (always) accessible or not (always) of good quality. Furthermore, there will be incompatible data models that may not exist in digital formats. This can lead to a lengthy development phase whilst strategies will have to be developed that bring data into a useable and standardised format. Perhaps the required time to address this issue becomes prohibitive to consider developing an autonomous system. This becomes compounded by the preference for new vs. existing data;
- A requirement of maintenance support operations is the need for (almost) real-time response capability. This may be alleviated by relegating certain background tasks, e.g., relating to recent fault history, reorganising/maintaining redundant fabric to low priority for minimum energy draw. Optimal balancing of real-time operations may be accomplished using machine learning, but this depends heavily upon data associated with low-level redundant resources (i.e., more so than higher-level abstract data layers[15]);
- The need to improve data quality has been emphasised in many publications and is evidenced by the availability of several benchmarking datasets. However, the ability to train complex networks often depends on the availability of big data,[16] which has motivated the emergence of techniques capable of generating artificial data to reduce classifier variance and hence reduce the number of errors in the final result.[17]

2.3.1. Generative learning

Machine learning approaches learn from data. If data is limited, it has to be generated, e.g., for reinforcement learning.[18] Generative learning can be used to generate new (and often complex) data sets by using a generative model against an adversary. These models are particularly interesting in the context of autonomous maintenance since they examine which states of a physical system have been visited and trained upon and hence extending the model's ability to generalise in various conditions. Generative learning is being championed to improve AI reasoning capabilities by generating knowledge from a purely virtual or even semi-real environment. Applications in the field include driverless cars, autonomous factories, smart cities, gaming, robotics, natural language processing, finance, healthcare, intelligent transportation systems (Goodfellow et al., 2014). In these cases, reinforcement learning has been used to initiate actions that maximise the notion of cumulative reward within the given environment. Here, two components are often considered: attention and memory. Attention is the mechanism that focuses on the salient parts; whilst memory focuses on long-term data storage. This gives algorithms the capability to learn behaviours before an action is taken, which could be crucial to end performance. As an example, an autonomous helicopter model could learn fundamental mechanisms for a flight using generative data (in simulations) to achieve a high-level of attention using the memory required (Ng et al., 2006). Once the attention has reached a desired level, the model can be used in the physical world.[19]

A model-free algorithm can also be used to represent knowledge. This is achieved via generic value functions wherein several accurate predictions are made and learned in real-time For example, a single state representation can be used to accurately predict many different sensors (at many different time scales) using policies, termination functions, reward functions, and terminal reward functions, that maximise results though goal-oriented questions. Another notable development in the field is the Generative Adversarial Network (GAN). These estimate generative models via an adversarial process by training two models simultaneously (Han, Lu, Zhao, You, & Li, 2018); a generative model to capture the data distribution and a discriminative model to estimate the probability that a sample comes from the training data. This approach has been extended to train a generative model using virtual data (Salimans et al., 2016). A discriminative model is then used to estimate the probability function that determines whether or not a sample comes from the real world. This technique has helped address some issues associated with reinforcement learning within virtual environments and extension to the real world. By combining virtual data generative models and transferring the learning model to a discriminative model the system is able to accurately express what has been learned from the virtual learning environment to the real system.

The idea is, therefore, to begin with simple rules from which an AI can learn strategies using rewarding actions. This becomes a core characteristic to achieve intelligence whilst utilising prior knowledge. However, generalising from a purely training-based exercise to reach real-life scenarios is not a trivial task.[20] At first, models will be trained using rules or limited scenarios (depending on the data set) where it will learns to improve upon its competition (i.e., against other models). Interestingly, video games have emerged as one of the main sources of benchmarking for the training and testing of such problems, mostly due to their realistic, yet controlled environments, and the availability of large amounts of data.[21] The motivation to achieve better results has led to significant advances in NN architectures that are suitable for the reinforcement learning paradigm including Deep Q learning (Anschel, Baram, & Shimkin, 2017), Deep Successor Reinforcement learning (Kulkarni, Saeedi, Gautam, & Gershman, 2016) and Duelling networks (Wang et al., 2015). These areas of research are broadly known as Deep Reinforcement Learning. However, the subsequent problem of training these models to develop more complex policies inside realistic and highly specific environments remains an open challenge.

Most research effort seems to focus on specific attributes of the learning process together with observations on the way that they interact with their environment. To highlight this, the authors have examined factors that influence model exploration capability.[22] This

[15] Or perhaps some kind of data stack specific to fault management that provides high-level 'hooks' into the low-level resources and their health status.

[16] E.g., image processing and natural language processing fields have greatly benefited from deep learning methods (Khan & Yairi, 2018).

[17] For image processing, these include scaling of objects (Salvador & Perez-Pellitero, 2015), moving objects spatially (Vasudevan & Siegwart, 2008), adding noise (Li, Li, Liu, Li, & Jia, 2018), etc. For language processing, these include contortions in the temporal dimension, dynamic range compression, adding Gaussian noise, etc. (Bratko, Cormack, Filipič, Lynam, & Zupan, 2006; Nadkarni, Ohno-Machado, & Chapman, 2011).

[18] As advocated by many authors (Boone, 1997; Sallab, Abdou, Perot, & Yogamani, 2017; Sutton, Barto, et al., 1998), it requires a shift from learning to generalisation of spatial data. This can help react to continuous-time dynamical systems without *a priori* discretisation of time, state, and action.

[19] Simply put, reinforcement learning is made up of optimisation, exploration, generalisation and delayed consequences. Rather than if-then statements, there is a need for something more generalisable that learns from data directly with a high-level representation of the task. It involves an agent and an environment where the learning system perceives the state (of the environment) via a set of observations to take action. It then receives a new set of observations and a reward. Based on these, predictions for future rewards can be made, whilst also changing the learning system's policy (on how it selects actions). The key point here is that a single, scalar reward signal drives the learning process. Radical generality is achieved without any signal interpretation, no reference signals or labels, no human interpretation and no calibration.

[20] This often called the *reality gap* (Tremblay et al., 2018).

[21] For example, a model can be trained against its copy without any supervision. In such cases, a basic set of rules of the game are initialised for the model, which improves much faster using a vector of reward instead of the classical scalar quantity.

[22] I.e., factors that bring about independence and self-governing attributes.

has led to the conclusion that the definition of a realistic scenario is most important. This is because once a model has learnt to perform effectively within a specific environment it has also learnt the behaviour of the received experiences.[23] A universal problem occurs when a model learns from an overly limited set of scenarios, in which case it will not be prepared for adversarial situations when the environment changes. A general solution is to use a source of synthetic data generated from a diversity of sources to better simulate the properties of the real environment during training. In other words, such a model can learn independently but the environment should be controlled to adapt the model to operate within unexpected events. This process needs some generalisation[24] and emergent behaviour[25] but once a model interacts with its environment, it should be generalised to a set of given actions whilst behaviours will emerge independently (or due to the generalisation). This emergent behaviour will then interact with the environment to create a continuous exploration capability. Capturing this real-time machine learning capability can help remove some barriers to actual prediction whilst anticipating the technological evolution of autonomous maintenance systems. AI can be used to learn and improve through ongoing collaboration with users and the environment. Following this, the next challenge is to determine how autonomy can be realised by utilising rational decision-making and strategies to deal with the consequences of dynamic environmental conditions.

2.4. Current trends towards autonomous maintenance

For safety-critical applications there is a need to implement an effective health monitoring capability that: (i) collects relevant data from various sensory sources; (ii) carries out necessary signal processing–including the extraction of key features; (iii) performs fault diagnosis; and (iv) performs fault prediction. The drive towards Industry 4.0 concepts and large-scale information systems such as IoT and cloud computing have become instrumental technologies towards next-generation system up-time, performance optimisation and resilience (Lee et al., 2020). Yet, no matter how well a maintenance system is designed, there always exists the possibility of deficiencies in implementation (due to design decisions and trade-offs) and component/sub-system performance that leads to difficulties in ensuring quality of service and in-operation maintenance. The ultimate responsibility for recognising, interpreting, and compensating for deficiencies in diagnosis capability rests with human maintainers, who are likely to be operating independently and without sharing important information. Considering the size and complexity of assets operating in modern industrial domains–and the task of comprehending their physical behaviour–it seems unrealistic to believe that ubiquitous and integrated system-level decisions can be made entirely by humans. This is especially so when operating conditions, and even the maintenance environment, are subjected to unpredictable fluctuations that cause unforeseen consequences. A paramount property of the autonomous system is therefore that it should be capable of recommending (and taking) progressive actions according to its environment. This plays an important role in adding resilience to the system as a whole and for regulating its in-service availability. During the process of diagnosis, a number of recommended actions might be issued including fault alarms, alternatives to maintain availability, in-service feedback. Depending on the recommendation, the system may either choose to delay any action–if the failure can be tolerated until the next scheduled (human) maintenance–or to take immediate action in the case of recognised conditions that represent imminent failures or states that will compromise operating safety margins.

The autonomous maintenance system is therefore expected to collect/generate all health-related data that comprises the various sensors networks used to record system performance and health along with operating environmental information. A central repository (possibly on-board) is expected to store critical information and carry out necessary status processing before communication with other service platforms. Most importantly, the architecture must focus on maintaining seamless communication and collaboration services such that all human, environmental and other associated subsystems are linked and up-to-date. There are issues to consider here, such as anticipating bottlenecks and dynamic management of health management resources, e.g., to adapt to network congestion or sub-component failures. The increased data volume and quality of service requirements associated with autonomous systems management warrants dedicated next-generation network infrastructures.

A current trend in autonomous maintenance research is the relatively siloed nature of studies. Despite the existence of past findings relating to in self-healing and autonomy concepts progress to application demonstration has been hindered. We summarise hindering factors reflected by researchers and practitioners alike that have influenced today's disparity in an effort towards realising autonomous maintenance:

- A universal issue is that of only having access to publicly available data sets that are not directly suitable for the application in question. Deep learning is generally regarded as being dependent upon access to large data sets but its integration with a digital twin is even more uncertain since this requires even more specialised data sets or methods for artificial data synthesis;
- Engineering design and test resources are typically limited, especially for supporting the development of complex and large-scale systems. Development teams also require expertise in next-generation data networks, computational resources and their limitations;
- Machine learning techniques and frameworks depend upon many specialise tuning parameters and expertise is required to determine the most likely architectures that suit a given set of application requirements and constraints;
- Industry attitudes and perceptions towards trustworthiness are different. This can hinder research ideas from being tested at higher technology readiness levels;
- The problem of quality assurance is non trivial and is likely to involve extensive testing and continuous improvement;
- Even though autonomous maintenance has a broad apparent applicability future complex systems, a clear assessment of the technology readiness level for autonomous maintenance is difficult to assess. An important indicator in this context can be a system readiness level proposed for evaluating the complexity of integrating these techniques into existing applications (Sauser, Verma, Ramirez-Marquez, & Gove, 2006).

With this in mind, concepts such as on-board and real-time fault detection become appealing since they continuously monitor and detect faults/failures. Outside the vehicle, ground-based analysis can then process telemetry data (which is bandwidth limited and time-delayed) using ground-based computational resources to investigate faults, their root causes, and recommend actions for recovery. As hardware technology continues to evolve, it is likely that some of this ground-based analysis will converge towards on-board strategies. In aeronautics, the increasing use of autonomy is driven by requirements to improve the affordability, efficiency, reliability, and safety of civil airspace, airport, and aircraft (manned and unmanned) operations. Desired capabilities relating to civil aviation include: dynamic route planning in response to

[23] It should be mentioned that even though a model can only learn (or optimise) its parameters from a given data set, novel behaviours can emerge even without previous knowledge.

[24] E.g., using one-shot imitation learning (Duan et al., 2017).

[25] E.g., using multi-agent competition (Bansal, Pachocki, Sidor, Sutskever, & Mordatch, 2017).

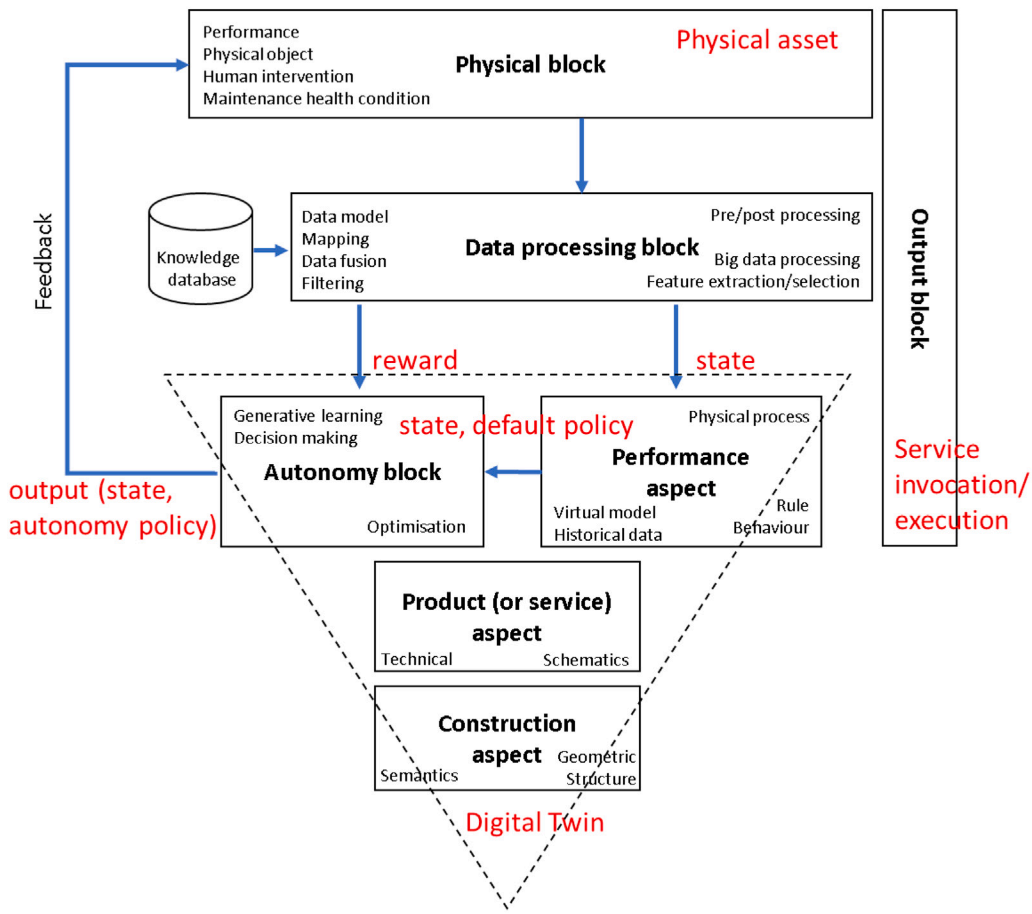

Fig. 1. Interrelations in the digital twin-driven autonomous maintenance framework. The digital twin comprises of three aspects: construction, product and performance. The autonomy block extends the application of the performance aspect.

traffic and weather, precision airport approach and departure management, payload directed flight, and in-flight compensation for degraded or failed aircraft systems.[26]

3. The digital twin-driven option

The concept of a digital twin has been applied for prognostics and health management by Tuegel et al. who used it to predict the structural life of an aircraft (Tuegel et al., 2011). Later, other researchers used it for damage detection, classification and isolation (Seshadri & Krishnamurthy, 2017). Gockel et al. used this to develop a twin model to isolate damage in real-time (Gockel, Tudor, Brandyberry, Penmetsa, & Tuegel, 2012). Effective damage detection is a key element here. However, despite developments in computation resources and mathematical optimisation, many actions are still undertaken by humans-in-the-loop. Autonomous maintenance systems require the ability to explore various training scenarios using their digital twin.

The authors propose the three aspects for data-driven digital twin:

- The construction aspect: this includes all the information of the structure of the asset. Not just the outer information, but the semantic information of how everything fits and works together. This can be used for planning, for system engineering, for simulation, for commissioning;
- The product (or service) aspect: this is delivered by manufacturers, which is a representation of components. Unfortunately, this does not usually come with semantic information such as controller firmware details, how many inputs/outputs, compatibility requirements, etc. These can be used (by the product manufactures) for simulations of behaviours.
- The performance aspect: This is the object that gathers all the real-time data that is generated by the system. They collate sensor data, process it through simulations to model all interactions (and reactions) of the real-world system. It offers an insight into a systems current environmental conditions, its service life and internal/external loads. Past data is stored and used to optimise future designs, predict various aspects of operations, increase availability by reducing downtime/costs and improve throughput. Machine learning algorithms are used to make better predictions which are then sent to an expert system to control the system operation.

These three aspects highlight why technology data is required during all phases of the system life cycle. It was mentioned earlier that the ability to generate and store information (from physical assets) is an important requirement for autonomous maintenance. This is represented by extending the performance aspect in Fig. 1 using the autonomy block (more discussion to follow). All this knowledge has to be stored in a standard format and must be fed back to other relevant interfaces. The digital twin will integrate sensor data from the vehicle's on-board health management system, maintenance history and all available historical data obtained using data/text mining. It continuously forecasts the health of a system, its remaining-useful-life or the probability of mission success. Such a platform will mitigate

[26] Although this article primarily focuses on large assets, there are benefits towards consumer products and general socioeconomic factors.

damage (or degradation) by recommending changes during the mission. But this goal can only be achieved by handling nonlinear system behaviour. Reinforcement learning can be used for this, as well as other deep learning methods (Khan & Yairi, 2018).

The aerospace domain has started to investigate the application of digital twin to accomplish the goal of reducing maintenance downtime for engines and other systems to receive an advance warning and generate a plan of actions based on simulated scenarios whilst accounting for weather conditions, the performance of the asset, and several other variables (Glaessgen & Stargel, 2012). These developments would enable operators to keep aircraft in-service for a longer duration, increase platform operational availability and efficiency, extend its useful life cycle, reduce its life cycle cost, mitigating damage/degradation and recommend changes in the mission profile.

The potential for digital twin-driven applications has been discussed by some publications for:

- Mirroring the actual flight of its flying twin (Glaessgen & Stargel, 2012). Once the vehicle is in flight, the continuous updates of actual load, temperature and other environmental factors will be input to the model enabling continuous predictions for the flying twin. Additionally, updates of the flying twin's health parameters, such as the presence and extent of damage or the temperature of the engine, can be incorporated to reflect flight conditions. Since the algorithms comprising the digital twin are modular, the best-physics models of individual systems or subsystems can be upgraded throughout the life of the vehicle;
- Performing in-situ forensics in the event of a potentially catastrophic fault or damage (Glaessgen & Stargel, 2012). Because the digital twin closely mirrors the state of health of the flying twin, it is well suited to analysing potentially catastrophic events. Once the sensor suite on-board the flying twin has communicated the degraded state of health to the digital twin, the digital twin can begin to diagnose the causes of the anomaly;
- Serve as a platform for learning (Xu et al., 2019). If, for example, mission control wants to determine how the health management system will deal with novel fault modes in a failed actuator and the best mitigation, the digital twin can be used to learn and determine new load distributions throughout the structure, the fatigue life of the structure under the new loads and the corresponding remaining life.
- Flying the actual vehicle's future mission(s) before its launch (Boschert & Rosen, 2016). Even without the benefit of continuous sensor updates, the digital twin will enable the effects of various mission parameters to be studied; effect of various anomalies to be determined; and fault, degradation and damage mitigation strategies to be validated. Additionally, parametric studies can be conducted to determine the flight plan and mission parameters that yield the greatest probability of mission success. This application becomes the foundation for certification of the flying twin;

Again, this list demonstrates a big reliance upon data and model quality. The question to ask here is: even if we had great AI-based monitoring/decision making capability, is the multi-physics model side of digital twin good enough? If the digital twin can predict when one of its systems is about to fail and be replaced, this could significantly reduce downtime. Maintenance is carried out only when required. At the centre of all this is collected sensor data that is used for training/testing an algorithm for fault detection, isolation and prediction. However, it is not always possible to acquire data from physical systems in the field under all fault conditions,[27] whilst generating faults, for example, from a probability distribution, may not provide the adequate test regime. A possible solution is for the digital twin is to create the various fault conditions through simulation(s). This would help test all facets (and severity) of a problem during the predictive maintenance workflow. Tao et al. (2018) have published a tentative framework to achieve this for classifying gradual faults (due to component degradation) and abrupt faults (due to disturbances) (Tao et al., 2018a). It divided the problem into three stages which included modelling the digital twin[28] and its interactions for detection, isolating and prediction of faults and, finally, deciding the maintenance action. The process needs both the physical and digital counterparts to measure any deviations from the expected states/behaviour to indicate potential fault occurrence. However, the authors did not discuss any limitations of this setup for system optimisation and predictive maintenance.

If a digital twin is considered as the "world/environment" within the generative learning environment, it would be able to describe how different components are coupled together. From this, it can be hypothesised: "if the simulation and physical environments are shaped in the same way (i.e., as a digital twin), can this enable a self-learning algorithm to steer itself without manual intervention?". This can allow leveraging in-house knowledge from experts without having to rely on new roles (e.g., data scientists) and also fulfils some of the requirements discussed earlier. It is a promising solution as no prior knowledge of the environment physics will be required to tune the system parameters. This is something that can even be done without having any technical background, as the algorithm will be able to find out what action to take, whilst accounting for increasing complexities; by adding their state information from previously unknown instances.

3.1. Towards a "reward engineering" environment

The authors propose a generic architecture in Fig. 1, with seven essential blocks. Five are in real-time: the physical, data communication, performance aspect, autonomy, output, and two offline blocks: the construction and product (or services) aspects:

- The physical asset: To map the physical representation, several considerations are needed. Starting with operational processes, sensors serve as the input.[29] It is represented as the physical block and also accounts for human intervention and feedback;
- The data communication block is the means of communication between the physical and virtual layers. It comprises of various elements and models such as data, fusion, acquisition, feature extraction/selection, pre/post processing, filtering or big data processing;
- The performance aspect of the digital twin is separated from the physical object; perhaps it is virtually located (e.g., in a cloud-computing environment). Using such infrastructures, the block accounts for various models such as geometric, physical/behaviour, rules/fuzzy, processes or technical CADs. This block also processes the information to diagnose/prevent system failures and predicting the reliability of the components in the physical block. Its functions include fault warning, querying, diagnosis, equipment health information, life prediction, flight condition monitoring and recovery status monitoring;
- The autonomy block is an extension of the performance aspect and is used to learn system uncertainty and allow to take action/feedback in real-time. This includes the management of unknown instances, provisioning of resources and optimisation of the various elements in the other layers;

[27] Some faults can even lead to catastrophic failures and result in massive business losses.

[28] If there were any variations in the model, then parameters are tuned using least squares method.

[29] Of course, not all necessary data can be retrieved from already installed sensors, e.g., to record environmental conditions, additional sensors and data loggers are needed.

- The output block serves as an interface to the framework. It can either be humanly readable as a web-based interface or an Application Programming Interface (API). One general purpose of this block is the generic access to the digital twin data. Hereby, an integrated interface to any possible service application is allowed;
- The remaining two offline aspects of the digital twin, the product (or service) and construction, form the rest of the architecture.

This framework illustrates the benefits of combining the concepts of digital twins and reinforcement learning to solve core issues in the ecosystem. The key requirement satisfied here is the ability to compensate for model uncertainties in the digital twin (during control and optimisation). In addition to providing health information, the digital twin triangle can observe a physical system state/behaviour/fault, and make a decision for relevant control action. This action will be based on a default policy as per the health management system requirement. The autonomy block observes both the system state/behaviour/fault and the default policy. It then makes a decision if the feedback should be applied as is (or be modified) to the physical asset. This setup would generate feedback (or reward) for the autonomy block, as the next states will be observed by the digital twin. Therefore, the reward is simply used to improve the autonomy block's policy. A similar idea was discussed by Cronrath et al. (2019) who used reinforcement learning as a means to adapt the control policy (of the twin) to impose a constraint on the learner performance. However, the question remains that in the use of these black-box techniques, how to account for uncertainty and guarantee safety requirements, and the implications of its realisation.

Of course, the problem of complexity associated with real-time implementation is still an issue. There is a need to place some limits on it and perhaps aim to achieve a reasonable system performance. Here, the long-term goal should be to appreciate these limitations that a digital representation would bring and then attempt to simplify the problem. Understanding these trade-offs is an ongoing motivation for many authors in the field (Leng et al., 2020; Leng, Yan, Liu, & Zhang et al., 2019). Complexity requirements are therefore categorised in terms of limitations in the structure of the computation, selecting a criterion for acceptable solutions, the cost of reasoning and task predictability:

- Structure of the computation: To limit complexity, a limit needs to be imposed on how many subsystem interactions are required for the architecture. This will dictate the number of computations involved;
- Selecting criterion for acceptable solutions: The assumption here is to satisfy a requirement rather than to find an optimal solution. These 'satisfying' solutions often take the form of using heuristic problem solving technique and are only useful if non-optimal solutions exist for a problem;
- The cost of reasoning: A good way to describe this to consider the cost of control against the cost of safety. Within safety-critical applications, this assumption will be difficult to justify unless the cost of control is predictable, e.g., does having a digital twin in the loop result in feedback delays which can affect system stability?
- Task predictability: The digital twin architectures need to make accurate system health management predictions to help avoid excessive runtime costs.

In addition, it is important to establish synchronisation between the physical asset and the virtual model of a mechatronic component. This can be done by identifying the various component model states as well as relations/inter-dependencies between the specific domains. This is because, over time, the states of the simulation and the physical asset will drift apart, perhaps due to degradation. It can be achieved using an online-optimisation (Leng et al., 2019) or an anchor point method (Talkhestani et al., 2019) to compare measurable states of the physical system and interdisciplinary models (and their relations in the digital twin). However, this needs to be extrapolated across the entire system life-cycle. This warrants a standardised semantic description of these models (or data and services) to establish a uniform understanding across the architecture. To enable this data exchange, standards for industrial communication for machine-to-machine or PC-to-machine communication can be used.[30] Ontologies have been widely used in context modelling, as they are independent of programming languages and enable context reasoning. Even though digital twins are used in different contexts, limited research efforts have evaluated how a digital twin and its architecture can accommodate changes occurred to the asset during its life-cycle. Lack of adaptive approaches is one of the main reasons for preventing industrial adoption. Generally, software integration in a digital twin should be achieved using some standard format for data exchange. However, this might not be feasible for data-driven digital twins, which require real-time updates due to uncertainties across its life-cycle. Erkoyuncu et al. had proposed an ontology-based approach for designing a data architecture for a digital twin, that would semantically link data and models to represent the asset (Erkoyuncu et al., 2020). Compared to other shared languages (e.g., structured query language - SQL), this offered advantages as the ontologies:

- Organise data semantically according to knowledge domains, helping to share information with the same meaning for both software systems and users;
- Are based on the 'open-world' assumption. Since data not declared is not implied to not exist, data changes can spread easier as interfaces are prepared to receive new data schemas;
- Can provide inferencing capabilities to the data stored, offering additional capabilities to reason over existing data. Therefore, the architecture would enable to generate or update ontologies so that diverse sources of information can communicate with each other using a common language.

Finally, readily mapping sensor data to model parameters can become an issue. This is due to the large number of sensors that are used to map real-time information; it is necessary to properly interpret and reuse sensor data from different domains, which places an emphasis on the construction of semantic maps (that illustrate the relations between heterogeneous domain ontologies), which becomes important for knowledge reuse. A possible solution is to establish a connection between sensor data and domain ontologies by classifying sensor data using a Semantic Sensor Network (SSN) ontology, and then mapping the corresponding instances to the concepts in the domain ontology (Liu, Li, Tian, Sangaiah, & Wang, 2019). A machine learning model can also be used to reduce the workload of manually labelling the enormous heterogeneous sensor data.

4. Trends for the future

The previous sections have delineated the main requirements and given an interpretation of where autonomy for maintenance is today. But what about the future? How are these requirements likely to evolve? What are the implications for the domain? What sort of processing will be available? This final section is an attempt to answer these questions, at least in part.

It is important to identify technology/capability milestones when looking into the future, and Fig. 2 provides a roadmap for what is needed if autonomous maintenance (according to the vision set out in this article) is to be realised. Predicting exact dates is of course difficult. Instead, the authors have opted to classify future developments as near, medium and long term goals:

[30] This includes technologies such as the Open Platform Communications United Architecture (OPC UA) or OWL ontologies.

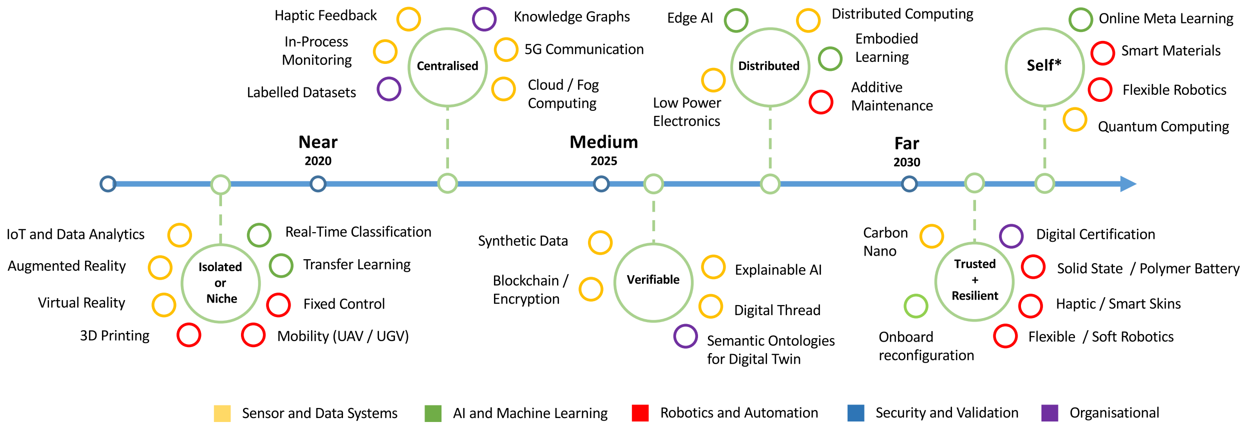

Fig. 2. Important technology milestones for enabling digital twin-driven autonomous maintenance.

- 'Isolated or niche' technologies focus on the development of strategic technology *niches* which satisfy certain commercialisation challenges. Various tools and processes are being developed to address current technological gaps, whilst aiming at elevating the core competencies of these solutions up to a competitive level;
- 'Centralised' technology is built around a single system that handles all the major processing and addresses client requests. A key benefit of their application includes consistency, efficiency, and affordability. However, with the rise of technologies like Blockchain and the cloud, a decentralised network will eventually become the norm;
- 'Verifiable' highlights the overwhelming progress of the large-scale impact from AI systems. It recognises that existing regulations and industry standards are insufficient to ensure responsible AI development. To engineer trust for their users, clients, society, governments, and other stakeholders, the technology needs to make verifiable claims to which it can be held accountable. This places an emphasis to provide evidence about the safety, security, fairness, and privacy protection of these solutions;
- 'Distributed' technology enables distributed workloads to offer high performance whilst making scalability a lot simpler;
- 'Trusted and resilient' milestone examines the impending challenges in situations of disrupted technology infrastructures and yet continuing to provide our society numerous benefits, including high-quality and energy efficient solutions;
- 'Self-' is the final milestone which focuses on achieving some form of singularity by demonstrating intelligent characteristics like self-resilience, self-adaptation, self-organising and self-healing. It will help to solve physics problems where quantum mechanics and the interrelation of materials or properties are important.

With these technological milestones in mind, it is clear that there is a need for significant investment and technology breakthroughs. Despite many large technology companies currently heralding many of these developments as the next breakthrough in engineering applications, their current investments suggest there is some disagreement as to how they will be achieved, especially when the amount of investment (public and private) keeps shifting (Kim & Min, 2020). To successfully realise an autonomous solution, various developments from these milestones are required. Each effort is going to be unique, even if it (seamlessly) reuses the same data. For a digital twin-driven solution, most of the challenges will be attributed to addressing the increasing complexity because:

- There is a need to break the system requirements down and scrutinise their existence. This will help analyse (and describe) the complexities of the physical process in terms of its constituents at a more fundamental level, whilst paying attention to the integration of these parts at system-level (Glaessgen & Stargel, 2012);
- Using the traditional approach to integrate components would reduce the agility of the twin (Uhlemann, Lehmann, & Steinhilper, 2017). This is because it rather focuses on loosely coupling components together in a way (generally) that lets them share or reuse data. This often leads to centralised architectures and deployments. There are no flexible ways for services to communicate with each other in ways that enable scalability, nor the ability to add (or remove) any service functionality;
- Dealing with data heterogeneity, redundancy, interoperability and datasets evaluation, along with an ontology that supports this (Lu et al., 2020);
- Programming all interrelationships between many data sets (Tao et al., 2018a).

Other challenges will be based on the assumptions that (Amaeshi & Crane, 2006; Kiritsis, 2013; Ueda, Takenaka, Váncza, & Monostori, 2009):

- Data is not owned but shared across all stakeholders;
- There is an active shift from transactional systems (and organisations) to more cooperative processes;
- The incremental development of intelligence is directly proportional to value creation;
- Asset life-cycle management is not only process-centric but also data-centric;

For the case of autonomous capabilities, any delay in retrieving and treating some data can lead to delays in action, affecting the expected benefits of using the digital twin in the first place. This encourages efforts to focus solutions on:

- Seamless connectivity: It is important to understand data origins to create a source of truth for a digital twin and defining the outlets for collecting knowledge which is missing. This includes techniques used to analyse data from IoT devices and simulating real-world scenarios in a virtual environment;
- Data intelligence: Analytics are important to transform data into meaningful (and manageable) form;
- Multi-model simulation: The platform should enable the simulation of large-scale business scenarios so that it can be debugged without affecting the physical device;
- Human machine interfacing: Visualisation and interactions with the processed information with increased efficiency;
- Knowledge graphs: For dynamic and semantic integration of data from complex systems.

While these opportunities are emerging, the key to creating system autonomy (based on digital twins) will be through the integration of technology infrastructure and focusing on the resilient milestone of Fig. 2. This is a challenging task and its implementation will require the right mix of technologies, domain expertise, and partnership ecosystems. Investing efforts here will help overcome the various complexities during system interactions, humans, the lack of agility, organisational cultures, comprehensive data sets and the relationship between data. Most technical challenges are related to data availability and access, while others are related to ensuring that it is an adequate representation of reality. In fact, extended reality[31] has already demonstrated its utility for training and equipment maintenance (Fast-Berglund, Gong, & Li, 2018), its role in autonomous maintenance will ensure that processes can be better analysed, serviced and updated in real-time by humans. Here, the digital twin visualisation will be a combination of virtual, augmented and mixed reality to simulate the real process (or service, machine, etc.) using real sensor data and models that enable autonomy and interaction in a virtual manner.

Another major technology area will focus on networking capability to manage the vast number of unattended sensor sources. All mobile systems, both manned and unmanned, will be networked while on the move. Most of these sensors will be in small devices that are dispersed throughout the asset (or fleet, business, etc.). This raises issues such as size, weight, power consumption, orientation when making decisions during operation, signal propagation in various environments, and security for these unattended devices as they pass information into a classified network. Such technology needs to permit communication on the move and need to account for a temporary loss in network connections. It should be able to recover the link and resume the communications stream immediately. Therefore, requirements such as bandwidth restrictions, reporting times and overhead will have to be satisfied. Moreover, the technology that will be at the core of all these developments will maintain all network operations. This involves combining network management, security management and information dissemination management into a single function. It would allow a system to help filter out and profile information to a specific commander based on parameters arising from missions/needs. The emergence of blockchain technology is one possible future trend which could also find itself integrated into the digital twin pipeline where security and trust are an issue. The ability to maintain an immutable data ledger, which could house sensor data, decisions and actions undertaken by the autonomous system could provide many benefits, though the computational cost to maintain such a ledger could be a challenge to overcome for a real-time system such as this.

None-the-less, autonomous maintenance will require an industry to test it during its early stages, i.e., the operator is willing to take the risks and accept the costs. This cultural impact can be minimised by introducing standards and bringing together various organisations, early in its development with a clear understanding of its operational benefits and use of unified terminology. The development of a specific ontology built around the core idea of autonomy and maintenance could also aid in this endeavour, but once again requires cross-organisation support.

Finally, the capability to test various simulations in each digital twin can become overwhelmingly large for traditional computing. Quantum computing is being championed to satisfy these requirements that make real-time machine learning (nearly) unlimited in capacity. In this case, a *quantum digital twin* would be the complex and accurate simulation of the real world. Perhaps it can utilise data from other digital twins models and postulate thousands (and millions) of variables interacting with each other. In this way, future digital twins platforms can use the processing power of quantum computers to simulate various scenarios in the least time and guide the autonomous decisions on the most optimal strategy.

[31] It is an umbrella term for all real-and-virtual combined environments and human–machine interactions. It encompasses virtual, augmented, and mixed reality.

5. Conclusions

Autonomous maintenance is still at its nascent stage and requires a synergy of algorithmic and computing techniques from intelligent systems and machine learning to establish its decision making capability. It is a promising solution that aims to minimise maintenance effort and reduce cost. At present some systems already feature basic autonomy. However, achieving full autonomy is still far from being accomplished. Most publications have approached the problem from a conceptual perspective and limited efforts have been made to create a universal view of the key components involving full autonomy. This article acknowledges an avid curiosity to understand the development process of autonomous maintenance systems. The route toward autonomy is complex as it requires the realisation of ambitious goals that must consider dynamic environments. This indicates that true autonomy might ultimately be an unattainable goal. However, this article has expressed this pursuit in the context of operating envelopes (such as performance margin, redundant resource allocation achieved for some defined period), that incur additional investment cost at the outset.

In particular, this article investigated autonomous maintenance using recent developments in machine learning and the concept of the digital twin. The resulting knowledge highlighted key requirements and how they can be met. A potentially important enabling concept is generative learning, which is gaining traction in engineering applications. Practitioners have also made use of several (digital) models to monitor and troubleshoot system-level problems. However, the quality of data used in this process impacts accuracy (and the meaning) of models and it seems that many potential adopters do not know how to find or exploit the right data. This highlights the two types of data, one which is visible that comes directly from maintenance problems, or is based on experiences.[32] As for the data that is not immediately visible, system maintainers often rely on trial-and-error approaches to diagnose a problem. This approach does work in some cases, but it cannot necessarily be reproduced by others, indicating the need for a more systematic approach to address data quality issues in autonomous maintenance applications.

Autonomous maintenance has the potential to boost automation to a completely new level and have numerous applications across all industries. However, many current attempts to create autonomy do not systematically address the two main challenges of complexity and uncertainty. The authors would agree that, while machine learning can (and will) play an important role, service robots executing maintenance tasks will not master the real-world complexity without significant prior knowledge for structuring their (almost) infinitely complex work and action spaces. Solutions will have to be specific, indicating that no single architecture will dominate, but rather achieving autonomy in unknown environments will require the safe, robust and verifiable composition of machine learning, perception, mapping, planning and control with training feedback. From this perspective, it is useful to draw out some key findings:

- Autonomy is the ability of a system to achieve predefined goals while operating independently of external control. The underlying spectrum ranges from local autonomy within subsystems where actions may be executed in response to stimuli (or local information), to system-level autonomy, which manages actions and handles constraints across subsystems.
- A fully autonomous maintenance system has the following aims:
  - Be able to undertake maintenance tasks independently and intelligently in dynamic and uncertain environments;

[32] General questions for the visible data are: First, how to find the useful data? Second, how to evaluate which data is useable? Third, which data is most critical?

– Seek to improve performance with a reduced burden on maintenance personnel;
– Achieve safe and efficient control of the system;
– Enabling decisions in complex and dynamic environments;

These aims can be realised by the creation of the three aspects of a data-driven digital twin which satisfying the following requirements:

• Determining in-situ implementation complexity requirements in terms of limitations of the structure of the computation, selecting criterion for acceptable solutions and the cost of reasoning and task predictability;
• Data processing requires continuous improvement of data quality and the ability to deal with big data. This can help reduce classification errors;
• To generalise a maintenance action (with continuous exploration capability) that requires more synthetic data to generalise the problem. This can assist in applying deep reinforcement learning algorithms to compensate for model uncertainties in the digital twin (during control and optimisation);
• Determining the required level of autonomy from a system will depend on:
  – The number of humans needed to operate a system;
  – The mean time between human interventions and the number of functions performed per intervention;
• Addressing primary technological barriers:
  – Lack of on-board computation and storage;
  – Network performance issues;
  – Knowledge representation;
  – Addressing the challenges associated with scaling up the state of technology to more complex scenarios where they can handle unanticipated anomalies and learn from past experiences.
• Appropriate interfacing methods.

Addressing these requirements will influence the development of the next-generation system health management capabilities. From this perspective, the data-driven digital twin should be considered to enable robust autonomy for maintenance services. This will also provide specific run-time optimisation for processor intensive requirements demanded by AI-based solutions. However, considering the high initial costs associated with the technology, it will only be successful when these solutions can be spread over several systems and implementations. This indicates that early adoption is going to be based on risk[33] and building trust. Yet, autonomous maintenance offers great potential for the maintenance industry and practices and the associated research communities. These concepts require joined efforts between manufacturers, maintenance organisations and regulators in order to mutually design, develop and control effective and trustworthy systems. Clarity in taxonomy will emerge at the same time. In conclusion, this article has discussed an important emerging topic in systems design and maintenance that requires extensive research and industrial collaboration to fully realise systems capable of autonomous maintenance. From a practical viewpoint, the most notable requirement is effective laboratory demonstrations that demonstrate the viability of high-fidelity models that capture enough information to support autonomous maintenance. As a result, research effort is needed in the areas of model development and computing architectures. Overall the authors advocate an incremental adoption of proven autonomous maintenance technologies that stimulate widespread industry adoption, for which the digital twin-driven approach is a key enabler.

[33] Perhaps introducing it as a decision aid to systems where decisions are made based on limited information seems like a natural starting point.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Amaeshi, Kenneth M., & Crane, Andrew (2006). Stakeholder engagement: A mechanism for sustainable aviation. *Corporate Social Responsibility and Environmental Management*, *13*(5), 245–260.
Anschel, Oron, Baram, Nir, & Shimkin, Nahum (2017). Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 176–185). JMLR.
Araguz, Carles, Bou-Balust, Elisenda, & Alarcón, Eduard (2018). Applying autonomy to distributed satellite systems: Trends, challenges, and future prospects. *Systems Engineering*, *21*(5), 401–416.
Bansal, Trapit, Pachocki, Jakub, Sidor, Szymon, Sutskever, Ilya, & Mordatch, Igor (2017). Emergent complexity via multi-agent competition. arXiv preprint arXiv:1710.03748.
Boone, Gary (1997). Efficient reinforcement learning: Model-based acrobot control. In *Proceedings of international conference on robotics and automation. vol. 1* (pp. 229–234). IEEE.
Booyse, Wihan, Wilke, Daniel N., & Heyns, Stephan (2020). Deep digital twins for detection, diagnostics and prognostics. *Mechanical Systems and Signal Processing*, *140*, Article 106612.
Boschert, Stefan, & Rosen, Roland (2016). Digital twin—the simulation aspect. In *Mechatronic futures* (pp. 59–74). Springer.
Bratko, Andrej, Cormack, Gordon V., Filipič, Bogdan, Lynam, Thomas R, & Zupan, Blaž (2006). Spam filtering using statistical data compression models. *Journal of Machine Learning Research (JMLR)*, *7*(Dec), 2673–2698.
Calvo, Rafael A, Peters, Dorian, Johnson, Daniel, & Rogers, Yvonne (2014). Autonomy in technology design. In *CHI'14 extended abstracts on human factors in computing systems* (pp. 37–40).
Cronrath, Constantin, Aderiani, Abolfazl R., & Lennartson, Bengt (2019). Enhancing digital twins through reinforcement learning. In *2019 IEEE 15th international conference on automation science and engineering (CASE)* (pp. 293–298). IEEE.
Cupek, Rafał, Drewniak, Marek, Ziebinski, Adam, & Fojcik, Marcin (2019). "Digital twins" for highly customized electronic devices–case study on a rework operation. *IEEE Access*, *7*, 164127–164143.
Dai, Yuanshun, Xiang, Yanping, Li, Yanfu, Xing, Liudong, & Zhang, Gewei (2011). Consequence oriented self-healing and autonomous diagnosis for highly reliable systems and software. *IEEE Transactions on Reliability*, *60*(2), 369–380.
Dale, Daniel R., et al. (2007). *Automated ground maintenance and health management for autonomous unmanned aerial vehicles* (Ph.D. thesis), Massachusetts Institute of Technology.
Ding, Kai, Chan, Felix TS, Zhang, Xudong, Zhou, Guanghui, & Zhang, Fuqiang (2019). Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors. *International Journal of Productions Research*, *57*(20), 6315–6334.
Duan, Yan, Andrychowicz, Marcin, Stadie, Bradly, Ho, OpenAI Jonathan, Schneider, Jonas, Sutskever, Ilya, et al. (2017). One-shot imitation learning. In *Advances in neural information processing systems* (pp. 1087–1098).
Dutt, Nikil, Jantsch, Axel, & Sarma, Santanu (2016). Toward smart embedded systems: A self-aware system-on-chip (soc) perspective. *ACM Transactions on Embedded Computing Systems*, *15*(2), 1–27.
Endsley, Mica R. (2017). From here to autonomy: Lessons learned from human–automation research. *Human Factors*, *59*(1), 5–27.
Erkoyuncu, John Ahmet, del Amo, Iñigo Fernández, Ariansyah, Dedy, Bulka, Dominik, Roy, Rajkumar, et al. (2020). A design framework for adaptive digital twins. *CIRP Annals*.
Eti, Mark C., Ogaji, S. O. T., & Probert, S. D. (2006). Reducing the cost of preventive maintenance (PM) through adopting a proactive reliability-focused culture. *Applied Energy*, *83*(11), 1235–1248.
Farnsworth, M., McWilliam, R., Khan, S., Bell, C., & Tiwari, A. (2017). Design for zero-maintenance. In *Advances in through-life engineering services* (pp. 349–365). Springer.
Fast-Berglund, Åsa, Gong, Liang, & Li, Dan (2018). Testing and validating extended reality (xR) technologies in manufacturing. *Procedia Manufacturing*, *25*, 31–38.
Frei, Regina, McWilliam, Richard, Derrick, Benjamin, Purvis, Alan, Tiwari, Asutosh, & Serugendo, Giovanna Di Marzo (2013). Self-healing and self-repairing technologies. *International Journal of Advanced Manufacturing Technology*, *69*(5–8), 1033–1061.
Glaessgen, Edward, & Stargel, David (2012). The digital twin paradigm for future NASA and US Air Force vehicles. In 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA (p. 1818).

Gockel, Brian, Tudor, Andrew, Brandyberry, Mark, Penmetsa, Ravi, & Tuegel, Eric (2012). Challenges with structural life forecasting using realistic mission profiles. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA (p. 1813).

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Han, Xiaofeng, Lu, Jianfeng, Zhao, Chunxia, You, Shaodi, & Li, Hongdong (2018). Semisupervised and weakly supervised road detection based on generative adversarial networks. *IEEE Signal Processing Letters*, *25*(4), 551–555.

Hancock, P. A. (2019). Some pitfalls in the promises of automated and autonomous vehicles. *Ergonomics*, *62*(4), 479–495.

Hegazy, Tamir, & Hefeeda, Mohamed (2014). Industrial automation as a cloud service. *IEEE Transactions on Parallel and Distributed Systems*, *26*(10), 2750–2763.

Huynh, Khac Tuan, Barros, Anne, & Bérenguer, Christophe (2014). Multi-level decision-making for the predictive maintenance of k-out-of-n: F deteriorating systems. *IEEE Transactions on Reliability*, *64*(1), 94–117.

Inagaki, Toshiyuki, & Sheridan, Thomas B. (2019). A critique of the SAE conditional driving automation definition, and analyses of options for improvement. *Cognition, Technology & Work*, *21*(4), 569–578.

Iverson, David L, Martin, Rodney, Schwabacher, Mark, Spirkovska, Lilly, Taylor, William, Mackey, Ryan, et al. (2012). General purpose data-driven monitoring for space operations. *Journal of Aerospace Computing, Information, and Communication*, *9*(2), 26–44.

Jain, Palak, Poon, Jason, Singh, Jai Prakash, Spanos, Costas, Sanders, Seth R, & Panda, Sanjib Kumar (2019). A digital twin approach for fault diagnosis in distributed photovoltaic systems. *IEEE Transactions on Power Electronics*, *35*(1), 940–956.

James, Mark L, & Dubon, Lydia P (2000). An autonomous diagnostic and prognostic monitoring system for nasa's deep space network. In *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484) 2*, 403–414.

Khajavi, Siavash H, Motlagh, Naser Hossein, Jaribion, Alireza, Werner, Liss C, & Holmstŕom, Jan (2019). Digital twin: Vision, benefits, boundaries, and creation for buildings. *IEEE Access*, *7*, 147406–147419.

Khan, Samir, Liew, Chun Fui, Yairi, Takehisa, & McWilliam, Richard (2019). Unsupervised anomaly detection in unmanned aerial vehicles. *Applied Soft Computing*, *83*, Article 105650.

Khan, Samir, & Yairi, Takehisa (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, *107*, 241–265.

Kim, Wonjoon, & Min, Sungjin (2020). The effects of funding policy change on the scientific performance of government research institutes. *Asian Journal of Technology Innovation*, 1–12.

Kiritsis, Dimitris (2013). Semantic technologies for engineering asset life cycle management. *International Journal of Productions Research*, *51*(23–24), 7345–7371.

Kobayashi, Takahisa, & Simon, Donald L. (2005). Hybrid neural-network genetic-algorithm technique for aircraft engine performance diagnostics. *Journal of Propulsion and Power*, *21*(4), 751–758.

Koch, William, Mancuso, Renato, West, Richard, & Bestavros, Azer (2019). Reinforcement learning for UAV attitude control. *ACM Transactions on Cyber-Physical Systems*, *3*(2), 1–21.

Kulkarni, Tejas D, Saeedi, Ardavan, Gautam, Simanta, & Gershman, Samuel J (2016). Deep successor reinforcement learning. arXiv preprint arXiv:1606.02396.

Lee, John D. (2018). Perspectives on automotive automation and autonomy. *Journal of Cognitive Engineering and Decision Making*, *12*(1), 53–57.

Lee, Jay, Azamfar, Moslem, Singh, Jaskaran, & Siahpour, Shahin (2020). Integration of digital twin and deep learning in cyber-physical systems: Towards smart manufacturing. *IET Collaborative Intelligent Manufacturing*, *2*(1), 34–36.

Lee, Jay, Ghaffari, M., & Elmeligy, S. (2011). Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems. *Annual Reviews in Control*, *35*(1), 111–122.

Lee, Dean, Siu, Vincent, Cruz, Rick, & Yetman, Charles (2016). Convolutional neural net and bearing fault analysis. In *Proceedings of the international conference on data mining (DMIN)* (p. 194). The Steering Committee of The World Congress in Computer Science, Computer ...,.

Leng, Jiewu, Liu, Qiang, Ye, Shide, Jing, Jianbo, Wang, Yan, Zhang, Chaoyang, et al. (2020). Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model. *Robotics and Computer-Integrated Manufacturing*, *63*, Article 101895.

Leng, Jiewu, Yan, Douxi, Liu, Qiang, Xu, Kailin, Zhao, J Leon, Shi, Rui, et al. (2019). Manuchain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing. *IEEE Transactions on Systems, Man & Cybernetics, A (Systems & Humans)*, *50*(1), 182–192.

Leng, Jiewu, Yan, Douxi, Liu, Qiang, Zhang, Hao, Zhao, Gege, Wei, Lijun, et al. (2019). Digital twin-driven joint optimisation of packing and storage assignment in large-scale automated high-rise warehouse product-service system. *International Journal of Computer Integrated Manufacturing*, 1–18.

Leng, Jiewu, Zhang, Hao, Yan, Douxi, Liu, Qiang, Chen, Xin, & Zhang, Ding (2019). Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop. *Journal of ambient intelligence and humanized computing*, *10*(3), 1155–1166.

Leonardi, Fabio, Messina, Fabrizio, & Santoro, Corrado (2019). A risk-based approach to automate preventive maintenance tasks generation by exploiting autonomous robot inspections in wind farms. *IEEE Access*, *7*, 49568–49579.

Li, Tong, Li, Jin, Liu, Zheli, Li, Ping, & Jia, Chunfu (2018). Differentially private naive Bayes learning over multiple data sources. *Information Sciences*, *444*, 89–104.

Li, Chenzhao, Mahadevan, Sankaran, Ling, You, Choze, Sergio, & Wang, Liping (2017). Dynamic Bayesian network for aircraft wing health monitoring digital twin. *Aiaa Journal*, *55*(3), 930–941.

Liao, Linxia, & Ḱottig, Felix (2016). A hybrid framework combining data-driven and model-based methods for system remaining useful life prediction. *Applied Soft Computing*, *44*, 191–199.

Lins, Romulo Gonçalves, Givigi, Sidney N, Freitas, Arthur DM, & Beaulieu, Alain (2016). Autonomous robot system for inspection of defects in civil infrastructures. *IEEE Systems Journal*, *12*(2), 1414–1422.

Liu, Zhifeng, Chen, Wei, Zhang, Caixia, Yang, Congbin, & Chu, Hongyan (2019). Data super-network fault prediction model and maintenance strategy for mechanical product based on digital twin. *IEEE Access*, *7*, 177284–177296.

Liu, Jin, Li, Yunhui, Tian, Xiaohu, Sangaiah, Arun Kumar, & Wang, Jin (2019). Towards semantic sensor data: An ontology approach. *Sensors*, *19*(5), 1193.

Lu, Yuqian, Liu, Chao, Kevin, I, Wang, Kai, Huang, Huiyue, & Xu, Xun (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, *61*, Article 101837.

Lv, Feiya, Wen, Chenglin, Bao, Zejing, & Liu, Meiqin (2016). Fault diagnosis based on deep learning. In *2016 American control conference (ACC)* (pp. 6851–6856). IEEE.

MacDonnell, Michael, & Clegg, Ben (2007). Designing a support system for aerospace maintenance supply chains. *Journal of Manufacturing Technology Management*.

Mackey, Ryan, James, Mark, Park, Han, & Zak, Michail (2001). BEAM: Technology for autonomous self-analysis. In *2001 IEEE aerospace conference proceedings (Cat. No. 01TH8542). Vol. 6* (pp. 2989–3001). IEEE.

Madni, Azad M., Madni, Carla C., & Lucero, Scott D. (2019). Leveraging digital twin technology in model-based systems engineering. *Systems*, *7*(1), 7.

McDuff, Richard J., Simpson, Patrick K., & Gunning, David (1989). An investigation of neural networks for F-16 fault diagnosis. I. system description. In *IEEE automatic testing conference. the systems readiness technology conference. Automatic testing in the next decade and the 21st century. conference record* (pp. 351–357). IEEE.

McWilliam, Richard, Khan, Samir, Farnsworth, Michael, & Bell, Colin (2018). Zero-maintenance of electronic systems: Perspectives, challenges, and opportunities. *Microelectronics Reliability*, *85*, 122–139.

Mele, Alfred R. (2001). *Autonomous agents: From self-control to autonomy*. Oxford University Press on Demand.

Moore, R. L., Rosenof, Howard, & Stanley, Gregory (1990). Process control using a real time expert system. *IFAC Proceedings Volumes*, *23*(8), 241–246.

Nadkarni, Prakash M., Ohno-Machado, Lucila, & Chapman, Wendy W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544–551.

Naghshbandi, S Neda, Varga, Liz, Purvis, Alan, Mcwilliam, Richard, Minisci, Edmondo, Vasile, Massimiliano, et al. (2020). A review of methods to study resilience of complex engineering and engineered systems. *IEEE Access*.

Nelles, Oliver (2013). *Nonlinear system identification: From classical approaches to neural networks and fuzzy models*. Springer Science & Business Media.

Ng, Andrew Y, Coates, Adam, Diel, Mark, Ganapathi, Varun, Schulte, Jamie, Tse, Ben, et al. (2006). Autonomous inverted helicopter flight via reinforcement learning. In *Experimental robotics IX* (pp. 363–372). Springer.

Onori, Mauro, Semere, Daniel, & Lindberg, Bengt (2011). Evolvable systems: An approach to self-X production. *International Journal of Computer Integrated Manufacturing*, *24*(5), 506–516.

Patton, Ron J., Frank, Paul M., & Clark, Robert N. (2013). *Issues of fault diagnosis for dynamic systems*. Springer Science & Business Media.

Price, Chris J., Snooke, N. A., & Lewis, S. D. (2006). A layered approach to automated electrical safety analysis in automotive environments. *Computers in Industry*, *57*(5), 451–461.

Qi, Qinglin, & Tao, Fei (2018). Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *IEEE Access*, *6*, 3585–3593.

Reynders, Edwin, Houbrechts, Jeroen, & De Roeck, Guido (2012). Fully automated (operational) modal analysis. *Mechanical Systems and Signal Processing*, *29*, 228–250.

Russell, B. Don, & Benner, Carl L. (2010). Intelligent systems for improved reliability and failure diagnosis in distribution systems. *IEEE Transactions on Smart Grid*, *1*(1), 48–56.

Salimans, Tim, Goodfellow, Ian, Zaremba, Wojciech, Cheung, Vicki, Radford, Alec, & Chen, Xi (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (pp. 2234–2242).

Sallab, Ahmad EL, Abdou, Mohammed, Perot, Etienne, & Yogamani, Senthil (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, *2017*(19), 70–76.

Salvador, Jordi, & Perez-Pellitero, Eduardo (2015). Naive bayes super-resolution forest. In Proceedings of the IEEE international conference on computer vision (pp. 325–333).

Sauser, Brian, Verma, Dinesh, Ramirez-Marquez, Jose, & Gove, Ryan (2006). From TRL to SRL: The concept of systems readiness levels. In Conference on systems engineering research, Los Angeles, CA (pp. 1–10).

Seshadri, Banavara R., & Krishnamurthy, Thiagarajan (2017). Structural health management of damaged aircraft structures using digital twin concept. In 25th AIAA/AHS adaptive structures conference. (p. 1675).

Shen, Hao, Tan, Ying, Lu, Jun, Wu, Qing, & Qiu, Qinru (2013). Achieving autonomous power management using reinforcement learning. *ACM Transactions on Design Automation of Electronic Systems*, *18*(2), 1–32.

Shin, Jong-Ho, & Jun, Hong-Bae (2015). On condition based maintenance policy. *Journal of Computational Design and Engineering*, *2*(2), 119–127.

Sridhar, Banavar, & Kopardekar, Parimal (2016). Towards autonomous aviation operations: What can we learn from other areas of automation? In 16th AIAA aviation technology, integration, and operations conference (p. 3148).

Sujil, A., Kumar, Rajesh, & Bansal, Ramesh C. (2018). Multiagent-based autonomous energy management system with self-healing capabilities for a microgrid. *IEEE Transactions on Industrial Informatics*, *15*(12), 6280–6290.

Sutharssan, Thamo, Stoyanov, Stoyan, Bailey, Chris, & Yin, Chunyan (2015). Prognostic and health management for engineering systems: A review of the data-driven approach and algorithms. *Journal of Engineering*, *2015*(7), 215–222.

Sutton, Richard S., Barto, Andrew G., et al. (1998). *Introduction to reinforcement learning. vol. 135*. MIT press Cambridge.

Talkhestani, Behrang Ashtari, Jung, Tobias, Lindemann, Benjamin, Sahlab, Nada, Jazdi, Nasser, Schloegl, Wolfgang, et al. (2019). An architecture of an intelligent digital twin in a cyber-physical production system. *at-Automatisierungstechnik*, *67*(9), 762–782.

Tao, Fei, Zhang, Meng, Liu, Yushan, & Nee, A. Y. C. (2018a). Digital twin driven prognostics and health management for complex equipment. *CIRP Annals*, *67*(1), 169–172.

Tao, Fei, Zhang, He, Liu, Ang, & Nee, Andrew Y. C. (2018b). Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, *15*(4), 2405–2415.

Tremblay, Jonathan, Prakash, Aayush, Acuna, David, Brophy, Mark, Jampani, Varun, Anil, Cem, et al. (2018). Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 969–977).

Tuegel, Eric J, Ingraffea, Anthony R, Eason, Thomas G, & Spottswood, S Michael (2011). Reengineering aircraft structural life prediction using a digital twin. *International Journal of Aerospace Engineering*, *2011*.

Ueda, Kanji, Takenaka, Takeshi, Váncza, Jozsef, & Monostori, László (2009). Value creation and decision-making in sustainable society. *CIRP Annals*, *58*(2), 681–700.

Uhlemann, Thomas H.-J., Lehmann, Christian, & Steinhilper, Rolf (2017). The digital twin: Realizing the cyber-physical production system for industry 4.0. *Procedia Cirp*, *61*, 335–340.

Vasudevan, Shrihari, & Siegwart, Roland (2008). Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, *56*(6), 522–537.

Venkatesan, Suchitra, Manickavasagam, Krishnan, Tengenkai, Nikita, & Vijayalakshmi, Nagendran (2019). Health monitoring and prognosis of electric vehicle motor using intelligent-digital twin. *IET Electric Power Applications*, *13*(9), 1328–1335.

Volponi, Allan J, DePold, Hans, Ganguli, Ranjan, & Daguang, Chen (2003). The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: A comparative study. *Journal of Engineering for Gas Turbines Power*, *125*(4), 917–924.

Von Neumann, John (1956). Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata Studies*, *34*, 43–98.

Wang, Ziyu, Schaul, Tom, Hessel, Matteo, Van Hasselt, Hado, Lanctot, Marc, & De Freitas, Nando (2015). Dueling network architectures for deep reinforcement learning. arXiv preprint arXiv:1511.06581.

Wang, Jinjiang, Ye, Lunkuan, Gao, Robert X, Li, Chen, & Zhang, Laibin (2019). Digital twin for rotating machinery fault diagnosis in smart manufacturing. *International Journal of Productions Research*, *57*(12), 3920–3934.

Xu, Yan, Sun, Yanming, Liu, Xiaolong, & Zheng, Yonghua (2019). A digital-twin-assisted fault diagnosis using deep transfer learning. *IEEE Access*, *7*, 19990–19999.

Zhang, Youmin, & Jiang, Jin (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, *32*(2), 229–252.

Zhao, Zhiyao, Quan, Quan, & Cai, Kai-Yuan (2014). A profust reliability based approach to prognostics and health management. *IEEE Transactions on Reliability*, *63*(1), 26–41.

Zhao, Rongli, Yan, Douxi, Liu, Qiang, Leng, Jiewu, Wan, Jiafu, Chen, Xin, et al. (2019). Digital twin-driven cyber-physical system for autonomously controlling of micro punching system. *IEEE Access*, *7*, 9459–9469.