Cumulative Damage Modeling with Recurrent Neural Networks

Renato Giorgiani Nascimento* and Felipe A. C. Viana†
University of Central Florida, Orlando, Florida 32816

https://doi.org/10.2514/1.J059250

Maintenance of engineering assets (for example, aircraft, jet engines, and wind turbines) is a profitable business. Unfortunately, building models that estimate remaining useful life for large fleets is daunting due to factors such as duty cycle variation, harsh environments, inadequate maintenance, and mass production problems that cause discrepancies between designed and observed lives. We model cumulative damage through recurrent neural networks. Besides architectures such as long short-term memory and gated recurrent unit, we introduced a novel physics-informed approach. Essentially, we merge physics-informed and data-driven layers. With that, engineers and scientists can use physics-informed layers to model well understood phenomena (for example, fatigue crack growth) and use data-driven layers to model poorly characterized parts (for example, internal loads). A numerical experiment is used to present the main features of the proposed physics-informed recurrent neural network. The problem consists of predicting fatigue crack length for a fleet of aircraft. The models are trained using full input observations (far-field loads) and very limited output observations (crack length data for only a portion of the fleet). The results demonstrate that our proposed physics-informed recurrent neural network can model fatigue crack growth even when the observed distribution of crack length does not match the fleet distribution.

Nomenclature

a	=	fatigue crack length
$\boldsymbol{a}$	=	state representing damage
C, m	=	Paris law coefficients
$\boldsymbol{h}$	=	states representing the sequence
t	=	time step
$\boldsymbol{x}$	=	input (observable) variables
$\Delta\boldsymbol{a}$	=	damage increment
ΔK	=	stress intensity range
ΔS	=	far-field stress

I. Introduction

PREDICTIVE models [1–4] are often used to model cumulative distress in critical components (diagnosis and prognosis) of engineering assets (e.g., aircraft, jet engines, and wind turbines). These models usually leverage data coming from design, manufacturing, configuration, online sensors, historical records, inspection, maintenance, location, and satellite data. In terms of modeling, we believe most practitioners would agree that 1) machine learning models offer flexibility but tend to require large amounts of data, and b) physics-based models are grounded on first principles and require good understanding of physics of failure and degradation mechanisms. In practice, the decision between machine learning and physics-based models depends on factors such as existing knowledge (maybe even legacy models), amount and nature of data, accuracy and computational requirements, timelines for implementation, etc. The interested reader can find a discussion on how these concepts apply to defense systems in Refs. [5,6] and examples of industrial and commercial applications.[‡,§,¶,**]

The literature on the use of traditional and modern machine learning methods for diagnosis and prognosis is rich. For example, Si et al. [7] reviewed statistical data-driven approaches for prognoses that rely only on available past observed data and statistical models (regression, Brownian motion with drift, gamma processes, Markovian-based models, stochastic filtering-based models, hazard models, and hidden Markov models). Tamilselvan and Wang [8] discussed a multisensor health diagnosis and prognosis method using deep belief networks. They demonstrate how deep belief networks can model the probabilistic transition between the health state and the damaged state in aircraft engines and power transformers. Interestingly, Son et al. [9] reported how they solved the same aircraft engine problem using the Wiener process combined with principal component analysis. Susto et al. [10] discussed how to approach the remaining useful life estimation using ensembles of classifiers. They based their work on traditional support vector machines and k-nearest neighbors, and they tested it on estimating the remaining useful life of tungsten filaments used in ion implantation (important in semiconductor fabrication). Khan and Yairi [11] reviewed the application of deep learning in structural health management (simple autoencoders; denoising autoencoder; variational autoencoders; deep belief networks; restricted and deep Boltzmann machines; convolutional neural networks; and purely data-driven versions of recurrent neural networks, including the long short-term memory and gated recurrent units). They found that most approaches are still application specific (unfortunately, they did not find a clear way to select, design, or implement a deep learning architecture for structural health management). They also advise that a tradeoff study

Received 19 November 2019; revision received 24 June 2020; accepted for publication 24 June 2020; published online 31 August 2020.

[*] Graduate Research Assistant, Department of Mechanical and Aerospace Engineering; renato.gn@knights.ucf.edu.

[†] Assistant Professor, Department of Mechanical and Aerospace Engineering; viana@ucf.edu. Senior Member AIAA.

[‡] "TrueChoice™ Commercial Services," GE Aviation Collaborators, 2019, https://www.geaviation.com/commercial/truechoice-commercial-services [retrieved 14 February 2019].

[§] "Siemens Service Programs and Agreements," Siemens Collaborators, 2019, http://www.industry.usa.siemens.com/services/us/en/industry-services/services-glance/service-programs-agreements/pages/service-programs-agreements.aspx [retrieved 14 February 2019].

[¶] "Engine Services–Lufthansa Technik AG," Lufthansa Technik AG Collaborators, 2019, https://www.lufthansa-technik.com/engine [retrieved 14 February 2019].

[**] "Gemini Energy Services: Wind Turbine Services," Gemini Energy Services Collaborators, 2019, http://www.geminienergyservices.com [retrieved 14 February 2019].

should be performed when considering complexity and computational cost. Stadelmann et al. [12] brought an interesting discussion on deep learning applied to industry. Among the case studies, they discussed predictive maintenance with machine learning approaches such as support vector machines, Gaussian mixtures, principal component analysis. Authors also discussed how these approaches compare with deep learning methods such fully connected and variational as autoencoders.

There is also considerable research on the use of physics-based methods for diagnosis and prognosis. Although the literature tends to be very application specific, most authors use a physics-based model for damage accumulation and a statistical/machine learning technique for parameter estimation and uncertainty quantification. For example, Daigle and Goebel [13] formulated physics-based prognostics as a joint state-parameter estimation problem, in which the state of a system along with parameters describing the damage progression are estimated. This is followed by a prediction problem, in which the joint state-parameter estimate is propagated forward in time to predict the end of life and remaining useful life. They demonstrate their methodology in the estimation of the remaining useful life of a centrifugal pump used for liquid oxygen loading located at the NASA Kennedy Space Center. Li et al. [14] modeled fatigue crack growth on the leading edge of an aircraft wing. The model was updated through dynamic Bayesian networks with observed data (including both damage and loads). The updated model was used in diagnosis and prognosis of an aircraft digital twin (virtual representation of the physical aircraft). Ling et al. [15] also modeled fatigue crack growth on the leading edge of an aircraft wing. However, instead of only estimating the remaining useful life, they used the information gain theory to evaluate the usefulness of aircraft component inspection. This helps in deciding whether or not inspection is worthwhile (e.g., model improvement justifying the cost). A dynamic Bayesian network tracks and forecasts fatigue crack growth, and the detection of a crack is modeled through probability of detection. Information gain per cost of inspection is used to identify the optimal timing of the next inspection. Yucesan and Viana [16] modeled main bearing fatigue in onshore wind turbines, coupling damage models for bearing raceway and grease (lubricant). Their results demonstrated that, although bearing fatigue is a secondary life-limiting factor, it can still contribute significantly to bearing failures. They also showed how to use their physics-based cumulative damage model to promote component life extension. Berri et al. [17] proposed a framework for prognosis including signal acquisition, fault detection and isolation, and remaining useful life estimation. To keep computational cost manageable, they proposed using strategies for signal processing combined with physical models of different fidelity and machine learning techniques. They successfully tested their approach on an aircraft electromechanical actuator for secondary flight controls. Byington et al. [18] presented a study on the use of neural networks for prognosis of aircraft actuator components. The framework covered tasks such as feature extraction, data cleaning, classification, information fusion, and prognosis. They successfully demonstrated their approach on an F-18 stabilator electrohydraulic servo valves. Jacazio and Sorli [19] presented an enhanced particle filter framework for prognosis of electromechanical flight controls actuators. They achieved promising results and showed the benefits of their approach as compared to other published methods.

Using artificial neural networks for solving differential equations with applications in engineering is a relatively old concept [20–23]. Nevertheless, the unparalleled computational power available these days contributed to the popularization of machine learning in engineering applications [24–26]. The scientific community has been studying and proposing deep learning architectures that leverage mathematical models based in physics and engineering principles [27–31]. Differential equations are used to train multilayer perceptrons and recurrent neural networks. The idea is to use the physics laws (in the form of differential equations) to help in handling the reduced number of data points and to constrain the hyperparameter space. With incompressible fluids, for example, this is done by discarding nonrealistic solutions violating the conservation of mass principle. Raissi [32] approximated the unknown of the solution of partial differential equations by two deep neural networks. The first network acts as a prior on the unknown solution (enabling avoidance of ill-conditioned and unstable numerical differentiations). The second network works as a fine approximation to the spatiotemporal solution. The methodology was tested on a variety of equations used in fluid mechanics, nonlinear acoustics, gas dynamics, and other fields. Wu et al. [33] discussed, in depth, how to augment turbulence models with physics-informed machine learning. They demonstrated a procedure for computing mean flow features based on the integrity basis for mean flow tensors and proposed using machine learning to predict the linear and nonlinear parts of the Reynolds stress tensor separately. They used the flow in a square duct and the flow over periodic hills to evaluate the performance of the proposed method. Hesthaven and Ubbiali [34] proposed a nonintrusive reduced-basis method (using proper orthogonal decomposition and neural networks) for parametrized steady-state partial differential equations. The method extracts a reduced basis from a collection of snapshots through proper orthogonal decomposition and employs multilayer perceptrons to approximate the coefficients of the reduced model. They successfully tested the proposed method on the nonlinear Poisson equation in one and two spatial dimensions, and on two-dimensional cavity viscous flows, modeled through the steady incompressible Navier–Stokes equations. Swischuk et al. [35] demonstrated through case studies (predictions of the flow around an airfoil and structural response of a composite panel) that proper orthogonal decomposition is an effective way to parametrize a high-dimensional output quantity of interest in order to define a low-dimensional map suitable for data-driven learning. They tested a variety of machine learning methods such as artificial neural networks, multivariate polynomial regression, k-nearest neighbor, and decision trees. The interested reader can also find literature on Gaussian processes [30,36].

As we just discussed, there are several ways to build physics-informed machine learning models. In this work, we focus on neural network models suitable for solving ordinary differential equations (describing time-dependent quantities of interest). Chen et al. [29] demonstrated that deep neural networks can approximate dynamical systems for which the response comes out of integrating ordinary differential equations. In their approach, deep neural networks represent very fine discretization along the lines of a very fine Euler integration. This is particularly applicable to recurrent neural networks and residual networks. Interestingly, the loss function operates on top of the ordinary differential equation solver. Therefore, training data are generated through adjoint methods (automatic differentiation). Kani and Elsheikh [31,37] introduced the deep residual recurrent neural networks where a fixed number of layers are stacked together to minimize the residual (or reduced residual) of the physical model under consideration. To reduce the computational complexity associated with high-fidelity numerical simulations of physical systems (which generate the training data), they also used proper orthogonal decomposition. They demonstrated their approach with the simulation of a two-phase (oil and water) flow through porous media over a two-dimensional domain.

Our work is focused on building prognosis models that feed large amounts of data produced by fleets of engineering assets. The models proposed in this paper are hybrid models that implement physics-based kernels within deep neural networks and are designed specifically for prognosis. In this contribution, we propose a recurrent neural network cell derived from the concept of cumulative damage models [38,39]. These models, often used to describe the irreversible accumulation of damage throughout the useful life of components (or systems), are formulated through an initial value problem. We start by proposing a recurrent neural network architecture for cumulative damage models. Then, we break down the recurrent neural network model that defines the incremental damage over time into submodels and discuss how the approach is flexible and can mix physics-based and machine learning submodels. Finally, we demonstrate how to apply it to tracking fatigue crack growth at a fleet level. In principle, the proposed recurrent neural network cell could also be applied to model other failure mechanisms, such as corrosion, wear, and creep, among others.

The remainder of the paper is organized as follows. Section II reviews the basic concepts behind cumulative damage models and fatigue crack growth. Section III gives an overview on physics-informed neural networks (focusing on recurrent neural networks) and presents the proposed recurrent neural networks cell, illustrating how to apply it to fatigue crack growth modeling. Section IV presents and describes the numerical experiments and presents the results along with discussion. Finally, Sec. V closes the paper, recapitulating salient points and presenting final conclusions.

II. Cumulative Damage Models and Fatigue Crack Propagation

Cumulative damage models [38,39] are very useful for tracking and forecasting damage through discrete time series. These models are often used to describe the irreversible accumulation of damage throughout the useful life of components (or systems). A cumulative damage model represents damage at time t as

$$\boldsymbol{a}_t = \boldsymbol{a}_{t-1} + \Delta \boldsymbol{a}_t \tag{1}$$

where $\boldsymbol{a}_{t-1}$ is the damage level at time $t-1$, and $\Delta \boldsymbol{a}_t$ is the damage increment, which is often a function of both $\boldsymbol{a}_{t-1}$ and inputs $\boldsymbol{x}_t$ at time t. The characterization of the damage $\boldsymbol{a}_t$ and the inputs $\boldsymbol{x}_t$ is highly specific to the problem. The damage $\boldsymbol{a}_t$ is usually associated with a failure mechanism and $\boldsymbol{a}_t$ is ideally an observable quantity. Although this is not a requirement, it significantly facilitates the modeling task [39–41]. For example, if fatigue is the failure mechanism, fatigue crack length is the observable quantity. The inputs $\boldsymbol{x}_t$ usually express time-dependent loading and boundary conditions (e.g., pressures, temperatures, torques, mechanical and thermal stresses, etc.) or even operating points (e.g., altitude, thrust, angle of attack, etc.).

In this paper, we use fatigue crack growth as an example of a cumulative damage mechanism. From a physics of failure standpoint, and quoting Dowling (Ref. [42] p. 399) for a definition, the "process of damage and failure due to cyclic loading is called *fatigue*." Fatigue crack propagation is usually modeled through Paris law [43]. Mathematically, the crack length a is modeled through the following ordinary differential equation:

$$\frac{da}{dt} = C \Delta K^m \tag{2}$$

where C and m are material properties, and ΔK is the stress intensity range (which depends on factors such as localized geometry, current crack length, and far-field cyclic stress). The Paris law coefficients, C and m, can be obtained through coupon data; and many engineering materials have constants documented in handbooks such as Ref. [44]. In its discrete form, the Paris law can be written as a cumulative damage model:

$$a_t = a_{t-1} + C \Delta K_t^m \tag{3}$$

where the subscripts t and $t-1$ define the current and previous time stamps, respectively.

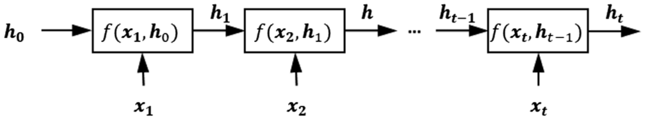


Fig. 1 Recurrent neural network. The function $f(h_{t-1}, x_t)$ (also known as the RNN cell), implements the transition from step to step throughout the time series.

In engineering applications (for example, prognosis and health management of industrial assets such as aircraft, jet engines, wind turbines, etc.), the cyclic loads are either measured or estimated. Then, engineering models map the cyclic loads and current crack length into a stress intensity range. For example, assuming that fatigue damage accumulates under a mode I loading condition (perpendicular to the crack plane), the stress intensity range ΔK_t can be expressed as

$$\Delta K_t = F \Delta S_t \sqrt{\pi a_{t-1}} \tag{4}$$

where ΔS_t is the far-field cyclic stress time history; and F is a dimensionless function of geometry and, to some extent, the crack length relative to its width. For example, for crack in infinite plate $F = 1$ and for a single-edge crack in finite-width sheet $F = 1.12$ (as long as the sheet width is much larger than the crack length). Discussion on ΔK_t and F for different conditions can be found in Ref. [42] (p. 399), Ref. [45] (p. 550), and Refs. [46,47].

Building accurate estimates of ΔS_t might be just as challenging as modeling damage accumulation itself. Most of the time, ΔS_t is not measured directly but, instead, it is obtained with the help of engineering models (e.g., through finite element modeling). Even if the instantaneous far-field stresses are available, converting far-field stress time histories into far-field cyclic stresses is usually done through cycle counting approaches such as the rainflow method [48]. Not surprisingly, the cycle counting approaches are application/industry dependent. Even though this is an interesting topic, we consider the discussion on how to obtain ΔS_t outside the scope of this paper.

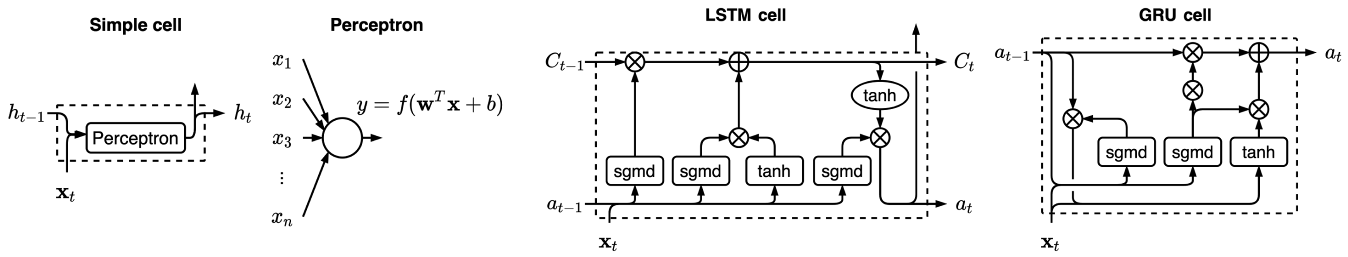
III. Recurrent Neural Networks and Physics-Informed Machine Learning

This work is focused on artificial neural networks that represent the solutions of ordinary differential equations. As we will demonstrate, recurrent neural networks (RNNs) are ideal for this task.

A. Brief Overview of Recurrent Neural Networks

Recurrent neural networks [49] have been successfully used to model time-series data [50–52], speech recognition [53], text sequence [54], and many other applications. As illustrated in Fig. 1, in every time step t, recurrent neural networks apply a transformation to a state $\boldsymbol{h}$ in the following fashion:

$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) \tag{5}$$



a) Simple cell and perceptron

b) LSTM and GRU cells

Fig. 2 Detailed recurrent neural networks cells. The arrow pointing upward indicates the state can be observed at time step t. In LSTM and GRU cells, squares are perceptrons with predefined activation functions, and the oval shape is just tanh activation.

where $t \in [0, \ldots, T]$ represents the time discretization, $\boldsymbol{h} \in \mathbb{R}^{n_h}$ are the states representing the quantities of interest, $\boldsymbol{x} \in \mathbb{R}^{n_x}$ are input variables, and $f(.)$ is the transformation to the state. Depending on the application, $\boldsymbol{h}$ can be available (i.e., actually observed) in every time step t or only at specific observation times.

The repeating cells for a recurrent neural networks implement the function $f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$ in Eq. (5), which defines the transformation applied to the states and inputs in every time step. Cells such as the ones illustrated in Fig. 2 are commonly found in data-driven applications. Figure 2a shows the simplest recurrent neural network cell, where a fully connected dense layer (e.g., the perceptron) with a sigmoid activation function maps the inputs at time t and states at time $t-1$ into the states at time t. Figure 2b show two other popular architectures: the long short-term memory (LSTM) [55] and the gated recurrent unit (GRU) [56]. These architectures have extra elements (gates) to control the flow and update of the states through time, and they aim at improving the recurrent neural network generalization capability and training by mitigating the vanishing/exploding gradient problem [49].

Recurrent neural networks are trained in a very similar way to traditional neural networks. The inputs are fed forward for every time step through the cell. Then, the loss value, calculated with the cell output and ground truth values, and its gradient are used to adjust the network weights through the process called backpropagation through time [49]. Here, we used the mean square error (MSE) as the loss function:

$$\mathrm{MSE} = \frac{1}{n_{\mathrm{OBS}}} \sum_{i=1}^{n_{\mathrm{OBS}}} \left(h_i^{\mathrm{RNN}} - h_i^{\mathrm{OBS}} \right)^2 \tag{6}$$

where n_{OBS} is the number of observations, h_i^{RNN} is the ith RNN prediction, and h_i^{OBS} is the ith observation.

B. Proposed Cumulative Damage Cell for Recurrent Neural Networks

In this paper, we propose the cell illustrated in Fig. 3a to be used for modeling cumulative damage through recurrent neural networks. In other words, Fig. 3a represents the recurrent neural network implementation of Eq. (1), and the block denoted by "MODEL" implements the damage increment $\Delta \boldsymbol{a}_t$ as a function of $\boldsymbol{a}_{t-1}$ and $\boldsymbol{x}_t$ at time t. As illustrated by Fig. 2, the design of the recurrent neural network cell is not limited to a simple perceptron and can use multiple tensor operations and gates. Instead, the design is limited only by the computational cost of the forward pass (prediction) and the availability of gradients with respect to trainable hyperparameters (backward propagation). Therefore, the main idea behind the design illustrated in Fig. 3a is the numerical integration of the differential equation that describes the physics of failure in case. As a matter of fact, the design shown in Fig. 3a is equivalent to implementing first-order Euler integration using recurrent neural networks.

From an implementation perspective, there is nothing preventing Model from assuming the form of any artificial neural network structure, such as the traditional multilayer perceptron. Nevertheless, we believe an interesting possibility for building the MODEL block is using a hybrid approach, where some parts of MODEL are physics informed while others are data driven. To clarify the nomenclature, here, we use "physics informed" to refer to functional forms, transformations, equations, or laws based on physics. In such cases, computational efficiency is an important point to consider and strategies using reduced-order modeling might become attractive. Although the discussion is outside of the scope of this paper, the interested reader is referred to Refs. [57,58]. We believe the decision between using a physics-informed versus a "data-driven" layer depends on the application, and we will use the fatigue crack growth example to illustrate it.

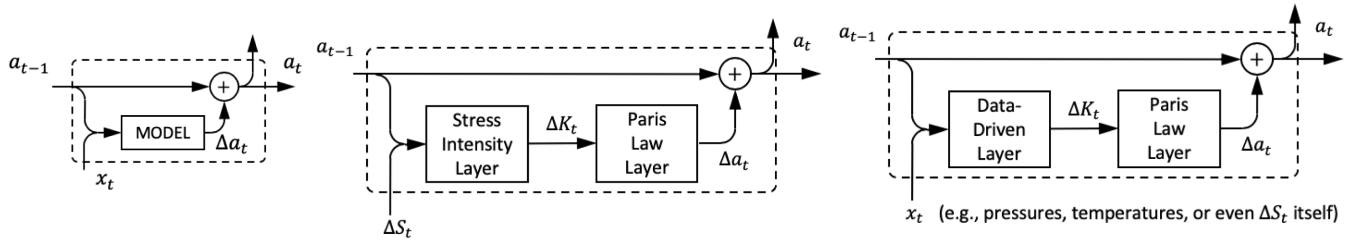
Figure 3b illustrates the fully physics-informed implementation of the cumulative damage cell for fatigue crack growth. In other words, Fig. 3b illustrates how to implement the model described by Eq. (3) through a recursive neural network. The stress intensity layer implements $\Delta K_t = F \Delta S_t \sqrt{\pi a_{t-1}}$, where ΔS_t is the input, a_{t-1} is the state, and F is the geometry-dependent factor, which can be obtained through engineering analysis (or, alternatively, it can be implemented as trainable parameter estimated during training of the recurrent neural network). The Paris law layer implements $\Delta a_t = C \Delta K_t^m$, where ΔK_t comes from a previous layer, and C and m are the Paris law coefficients, which should be readily available for many engineering materials (or, alternatively, they can be implemented as trainable parameters estimated in the recurrent neural network training).

Figure 3c illustrates the hybrid physics-informed neural network model implementation of the cumulative damage cell for fatigue crack growth. In this case, an artificial neural network layer, such as a multilayer perceptron, substitutes the stress intensity range layer shown in Fig. 3b. This is very powerful in certain real life applications, for example, when ΔK_t from physics is misled by poorly estimated ΔS_t, or when it is plainly difficult to model ΔK_t as a function of observed inputs $\boldsymbol{x}_t$. Therefore, the training of the recurrent neural network is such that the artificial neural network layer will learn how to map $\boldsymbol{x}_t$ and $\boldsymbol{a}_{t-1}$ into $\Delta \boldsymbol{K}_t$. In this case, the artificial neural network layer is compensating for the lack of knowledge about the stress intensity range.

As we discussed, Fig. 3 has the sole purpose of implementing the numerical integration of the ordinary differential equation that describes the target physics of failure [Eq. (2)]. As we will further discuss in Sec. IV, input data are available throughout time and initial damage is known. Therefore, this is an initial value problem and it makes sense using first order Euler integration, which leads to Fig. 3a. The blocks used in Fig. 3 come out of the discretization of Eq. (2). We believe the decision of which block will be physics informed and which one will be data driven depends on the application. However, in this particular paper, we argue that $\Delta a_t = C \Delta K_t^m$ is relatively well characterized by coupon data, whereas $\Delta K_t = F \Delta S_t \sqrt{\pi a_{t-1}}$ carries the uncertainty due to engineering models used for ΔS_t estimation. Therefore, for the sake of our study, we suggest implementing ΔK_t as a data-driven layer.

IV. Numerical Experiments and Discussion

First, we discuss the specifics of fatigue crack growth at a specific control point and the extension to the fleet of aircraft. Then, we present two scenarios for fatigue crack growth estimation of aircraft fuselage panels and prediction at a fleet of aircraft. In the first scenario, we consider that the control point in the fuselage panel is instrumented and continuously monitored through dedicated structural health



a) Cumulative damage RNN cell **b) Fully physics-informed cell** **c) Hybrid physics-informed neural network cell**

Fig. 3 Cumulative damage and examples of RNN cells for crack growth. Stress intensity range layer implements $\Delta K_t = F \Delta S_t \sqrt{\pi a_{t-1}}$, and Paris law layer implements $\Delta a_t = C \Delta K_t^m$.

monitoring sensors (e.g., comparative vacuum monitoring [59], fiber Bragg grating sensors [60], etc.). In the second scenario, we consider the control point in the fuselage panel is inspected in regular intervals through nondestructive evaluation approaches (e.g., eddy current [61], ultrasound [62], dye penetrant inspection [63], etc.).

These applications impose two major challenges for recurrent neural networks:

1) The sequences are very large (thousands of steps).

2) Output values are known at the beginning of the sequence but are observed only at few time stamps throughout the sequence.

The large sequences can lead to a significant increase in the norm of the gradients during training, which can harm the learning process. Moreover, by having only a few observations throughout the sequence, the long-term components can grow or decrease exponentially (exploding/vanishing gradient). This fast saturation makes it very hard for the model to learn the correlation between the observations. The interested reader is referred to the work of Pascanu et al. [64] and Sutskever [65] for further discussion on the difficulties of training recurrent neural networks under such circumstances.

A. Fleet of Aircraft and Fuselage Panel Control Point

When airline companies operate a fleet of a particular aircraft model, they usually adopt a series of operation and maintenance procedures that maximize the use of their fleet under economical and safety considerations. For example, these companies rotate their aircraft fleet through different routes following specific mission mixes. This way, no single aircraft is always exposed to the most aggressive (or mildest) routes, which helps in managing useful lives of critical components. Here, we assume an aircraft model designed to fly the four hypothetical missions shown in Fig. 4a and consider the mission mixes detailed in Table 1. We assume the airline company has 300 aircraft allocated to each mission mix. Each aircraft is assigned a fixed percentage of flights for each mission that composes the mission mix. These percentages vary uniformly from 0 to 100%. Therefore, within the fleet flying mission mix A, for example, there is one aircraft flying 0% of mission 0 and 100% of mission 3; there is another aircraft flying 1% of mission 0 and 99% of mission 3; there is yet another aircraft flying 2% of mission 0 and 98% of mission 3; and so forth. The same logic applies to the other mission mixes. We chose these four missions as an effort to illustrate the different conditions the fleet of aircraft usually experience in commercial aviation. In reality, though, the number of missions and the number of mission mixes depend on how operators decide to manage their fleets. This way, we synthetically created data for a fleet of 300 aircraft.

We consider the control point on an aircraft fuselage illustrated in Fig. 4b (crack in infinite plate) and assume that fatigue damage accumulates throughout the useful life, as illustrated in Fig. 4c. For the sake of this example, we assumed that the initial and the maximum allowable crack lengths are $a_0 = 0.005$ m and $a_{\max} = 0.05$ m, respectively. The metal alloy is characterized by the following Paris law constants: $C = 1.5 \times 10^{-11}$ and $m = 3.8$.

Table 1 Missions mix details[a]

	Mission (load in kilopascals)			
Mission mix	0 (92.5)	1 (100)	2 (110)	3 (130)
A	✓		——	✓
B	——	✓	✓	——
C	——	✓	——	✓

[a] Percentage of flights for each mission varies from 0 to 100%.

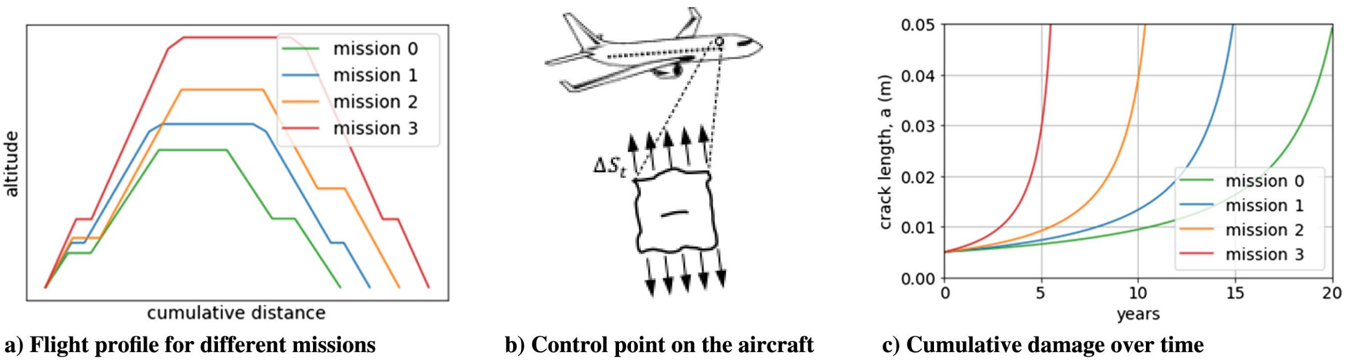
Figure 5 illustrates part of the data used here. Figure 5a shows the complete mission history in terms of far-field stresses for the aircraft flying the most aggressive and most mild mission mixes of the fleet (for the entire fleet, the far-field stress time histories follow the mixes described in Table 1). Figure 5b shows how the crack length time histories can be different across the entire fleet and highlights the two extreme cases (most aggressive and most mild mission mixes of the fleet) as well as the fleetwide crack length distribution at the fifth year. For the sake of this study, we consider that the history of cyclic loads is known throughout the operation of the fleet (i.e., the data shown in Fig. 5a are observed). On the other hand, the crack length history is only partially known (i.e., data shown in Fig. 5b) and the availability of the information depends on the adopted monitoring/inspection strategy.

B. Scenario I: Continuous Monitoring of Control Point

As previously mentioned, in the first scenario, we assume that the control point is instrumented and continuously monitored through dedicated structural health monitoring sensors (e.g., comparative vacuum monitoring and fiber Bragg grating sensors, etc.). In practical applications, airline companies could limit the number of monitored aircraft due to cost associated with the structural health monitoring system (including its own maintenance). In such cases, data gathered on part of the fleet are used to build predictive models for the entire fleet.

We arbitrarily assume the dedicated structural health monitoring sensors were installed on 60 aircraft of the fleet. This is equivalent to 20% of the fleet and represents a scenario in which the airline companies are willing to instrument a significant portion of the fleet. It should also allow us to study continuous monitoring without having to instrument the entire fleet or having only very few instrumented aircraft. As detailed in Table 2, we studied different rates in which sensor data are acquired (from data collection as sparse as once a year to as refined as once a day). As mentioned before, we also considered the case of weekly and daily crack length observation.

First, we evaluated the performance of purely data-driven recurrent neural networks. We used the long short-term memory and gated recurrent unit (Fig. 2b) architectures. Multiple configurations were tested, varying the number of layers (stacked cells) and units (with the same number of neurons per layer), as detailed in Table 3. Every one of the nine configurations was used for each inspection period (yearly, monthly, weekly, and daily), totaling 72 distinct models

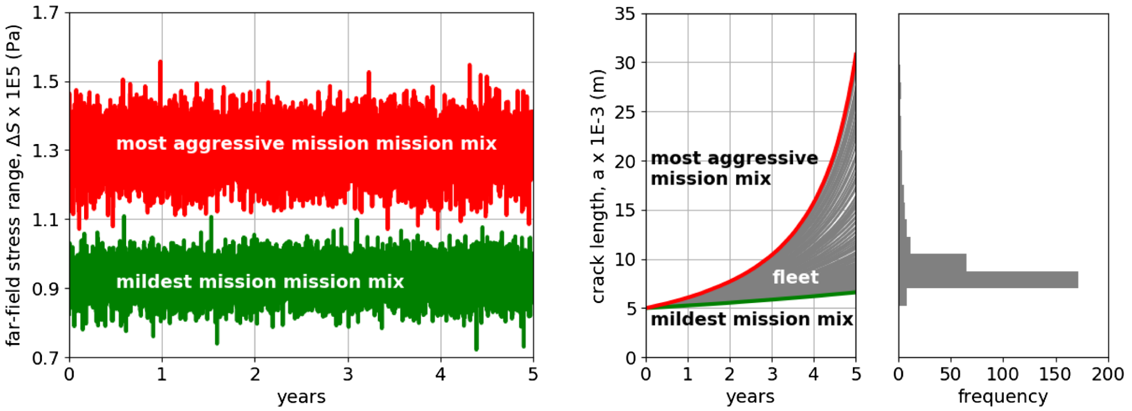


a) Flight profile for different missions

b) Control point on the aircraft fuselage

c) Cumulative damage over time

Fig. 4 Cumulative damage over time for control point on aircraft fuselage as a function of mission profile (assuming aircraft flies four missions per day).



a) Missions histories for two different aircraft

b) Fatigue crack length history for the fleet (300 aircraft)

Fig. 5 Snapshot of synthetic data.

Table 2 Inspection periods with its number of time steps and total data points used for training

Periodicity	Yearly	Monthly	Weekly	Daily
Number of time steps (per aircraft)	5	60	260	1,825
Number of data points (60 aircraft)	300	3,600	15,600	109,500

Table 3 Designs based for LSTM and GRU networks

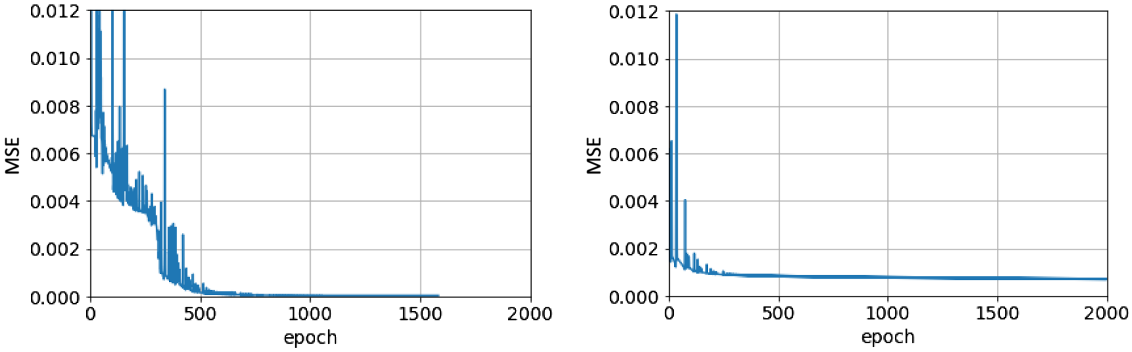
	Configuration								
Layer	1	1	1	2	2	2	4	4	4
Neurons	16	32	64	16	32	64	16	32	64
LSTM parameters	1,168	4,384	16,960	3,280	12,704	49,984	7,504	29,344	116,032
GRU parameters	928	3,392	12,928	2,560	9,728	37,888	5,824	22,400	87,808

(36 LSTM and 36 GRU). The training of these neural networks was performed using the mean square error loss function and the Adam optimizer [66]. The different model architectures converged at slightly different epochs, with all returning the best results before 1000 epochs. Nevertheless, the training was carried over to 2000 epochs to all models to ensure no further improvement would occur.

As illustrated in Fig. 6, the loss function of most of the models converged to small values (although their rate of convergence can greatly differ). Since we are using the Adam optimizer, we see high oscillation early in the optimization process is a manifestation of the learning rate adjustment, which is followed by a rapid convergence midway, and then stagnation toward the end the optimization. After training with the subfleet of 60 aircraft, all models were validated against the entire fleet (300 aircraft). Figure 7 shows the percent prediction errors of the recurrent neural network models at the end of the fifth year for all 300 aircraft when crack length is observed yearly, monthly, weekly, and daily. This detailed look at the prediction accuracy reveals that the number of observations strongly contributes to overall prediction accuracy. Not surprisingly, the more observations used in training, the more accurate the model predictions are. Although the box plots show that some architectures would outperform others, there is no clear trend with regard to complexity (i.e., more complex models outperforming simpler models up to a point). This indicates that the result is likely to be dependent on the specific training of the neural network (through a combination of random initialization of weights, and optimization parameters such as optimization algorithm, learning rate, number of epochs, etc.).

Figure 8 shows how the predictions at the end of the fifth year for all 300 aircraft compared with actual crack lengths. Interestingly, Figs. 8a and 8b show that both the LSTM- and the GRU-based recurrent neural networks have similarly poor performance when trained with yearly observations. The performance improves when these models are trained with daily observations, although there is still considerable prediction error across the crack length range (the GRU-based seems to exhibit a bias toward the high crack length).

Figure 9 illustrates the model predictions up to the end of the fifth year for all 300 aircraft. Figure 9a shows the time histories for the actual crack length and the model predictions coming from one



a) LSTM, 4 × 32, daily

b) GRU, 4 × 32, daily

Fig. 6 Example of loss function history throughout training of LSTM and GRU networks.

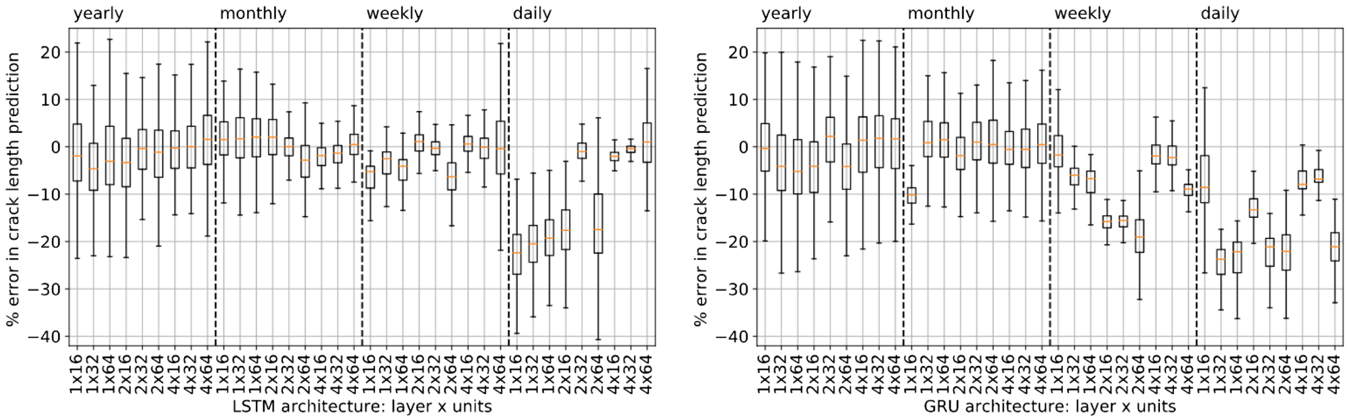
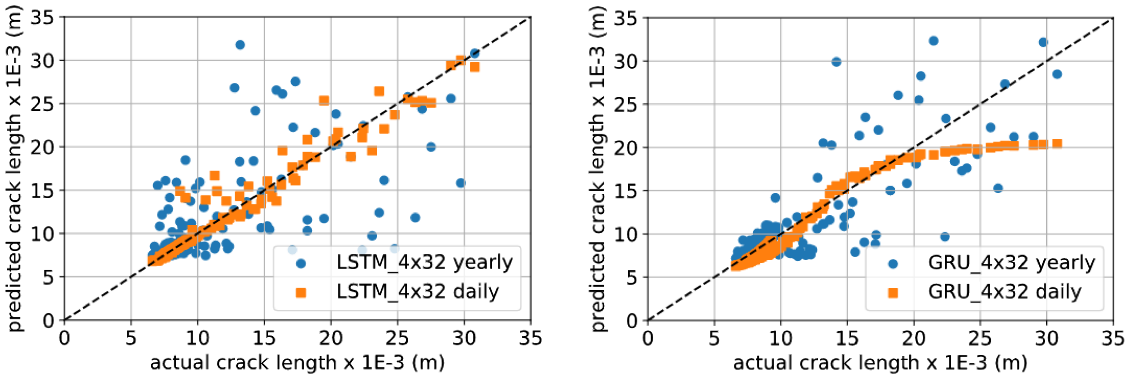


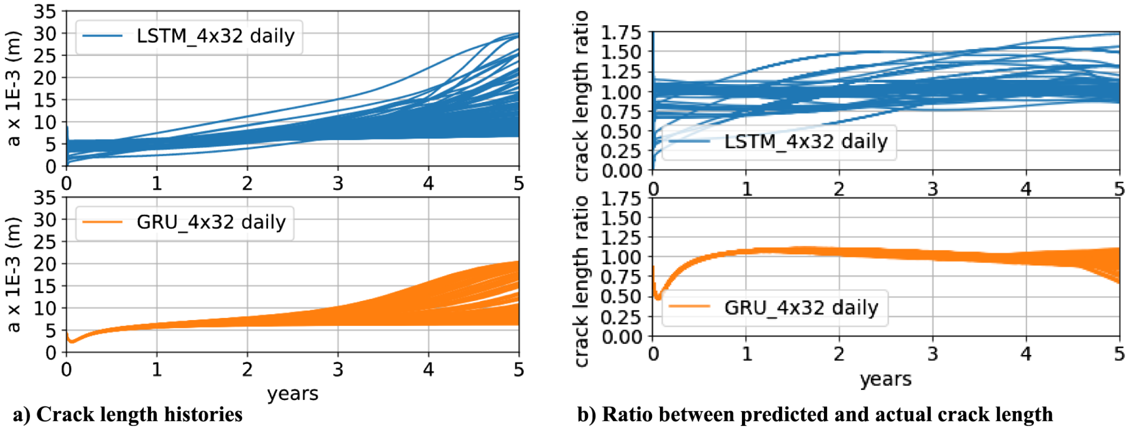
Fig. 7 Box plot of percent error in crack length prediction for LSTM and GRU networks at fifth year for entire fleet (300 aircraft): $\%\text{error} = 100 \times (a_{\text{PRED}} - a_{\text{actual}}/a_{\text{actual}})$.



a) Long-short term memory (LSTM)

b) Gated recurrent unit (GRU)

Fig. 8 LSTM and GRU predictions versus actual crack length at the fifth year for entire fleet (300 aircraft).



a) Crack length histories

b) Ratio between predicted and actual crack length

Fig. 9 Actual and LSTM- and GRU-predicted crack lengths over time for entire fleet (300 aircraft).

LSTM- and one GRU-based neural network. Figures 8 and 9a clearly show that the crack length trends might be captured but the shape of the curve is poorly approximated. Figure 9b illustrates the ratio between the predicted and actual crack lengths. The ratio being close to one is a good indication of prediction accuracy. For both LSTM- and GRU-based neural networks, the predictions are mostly within ±25% (except for some predictions out of the LSTM-based neural network, which can overestimate the final crack lengths by as much as 75%).

We also tested the performance of the proposed hybrid physics-informed neural network when there is continuous monitoring of the control point. We built the model using multilayer perceptrons as the stress intensity layer in series with a physics-based Paris law layer (as illustrated in Fig. 3c). Table 4 details the multilayer perceptron designs considered in this study. We varied the number of layers and neurons within each layer as well as the activation functions. We used the linear, hyperbolic tangent (tanh) and the exponential linear unit (elu) activation functions, given as follows:

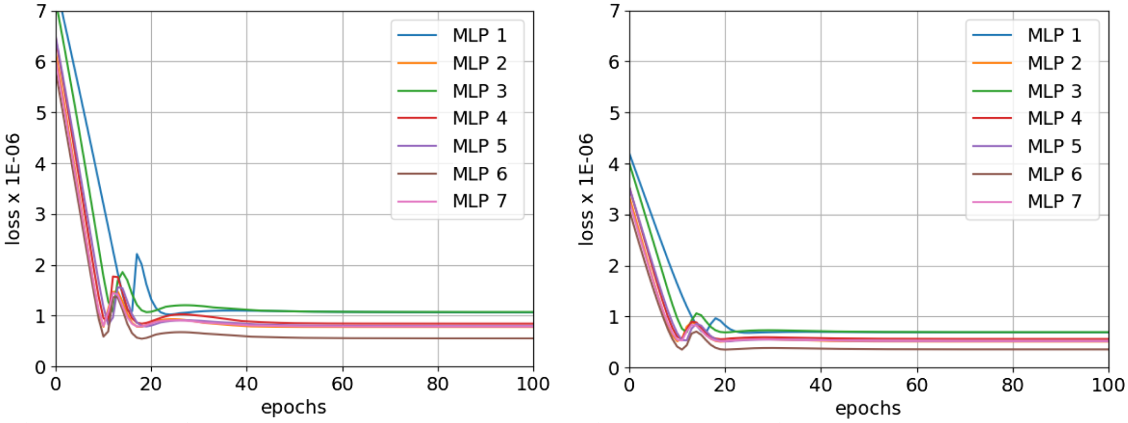
$$\text{linear}(z) = z, \qquad \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}},$$
$$\text{and} \quad \text{elu}(z) = \begin{cases} z & \text{when } z > 0 \\ e^z - 1 & \text{otherwise} \end{cases} \tag{7}$$

Table 4 Multilayer perceptron configurations used to model the stress intensity range

	Number of neurons/activation function						
Layer no.	MLP 1	MLP 2	MLP 3	MLP 4	MLP 5	MLP 6	MLP 7
0	5/ tanh	10/elu	10/ tanh	10/ tanh	20/ tanh	20/ tanh	40/ tanh
1	1/linear	5/elu	5/elu	5/ tanh	10/elu	10/elu	20/elu
2	— —	1/linear	1/linear	1/linear	5/elu	5/ tanh	10/elu
3	— —	— —	— —	— —	1/linear	1/linear	5/ tanh
4	— —	— —	— —	— —	— —	— —	1/linear
Parameters	21	91	91	91	331	331	1,211

The training of these neural networks was performed using the mean square error loss function and the Adamax optimizer [66]. Given that the hybrid physics-informed neural network converged much faster (all models converged before 50 epochs), the training was carried over to 100 epochs to ensure no further improvement would still occur.

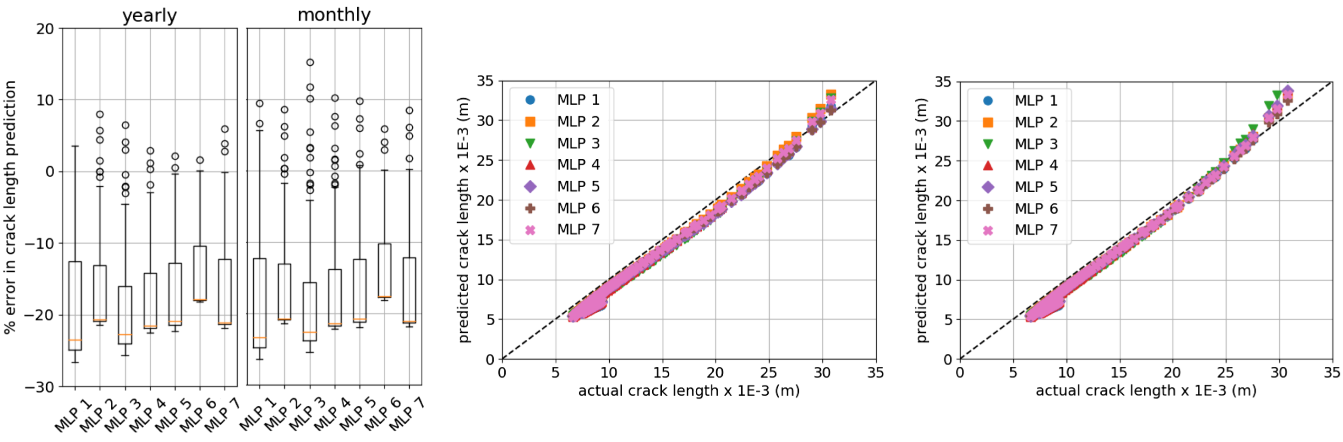
Similarly to what happened with the data-driven architectures (see Fig. 6), Fig. 10 illustrates the loss function oscillation early in the optimization process followed by convergence and stagnation toward the end the optimization. Figure 10 also shows that the different multilayer perceptrons converged to roughly the same loss function values at the end of the training process. Figure 11a shows the percent prediction errors of the recurrent neural network models at the end of the fifth year for all 300 aircraft when crack length is observed yearly and monthly. The main advantage of using physics-informed neural networks is reducing the need for training points. In this case, the costly part is the acquisition of crack length observations. One can argue that a monitoring system that offloads data yearly or monthly is cheaper to operate and maintain when compared to one that is expected to produce data on a daily basis, for example. Therefore, we will only show results for yearly and monthly crack length observations. Visual comparison shows that all the seven physics-informed neural networks had percent errors within the same order of magnitude. This is true even when models are trained with yearly observations, which confirms that the physics-informed layer compensates for the reduced number of observations. From an airline company perspective, the motivation behind the reduction in the number of training points is the cost associated with continuously monitoring the aircraft in the fleet. In this case study, the costly part is the acquisition of crack length observations. Therefore, it is expected



a) Yearly observations

b) Monthly observations

Fig. 10 Example of loss function history throughout training of physics-informed neural networks with continuous monitoring of control point.

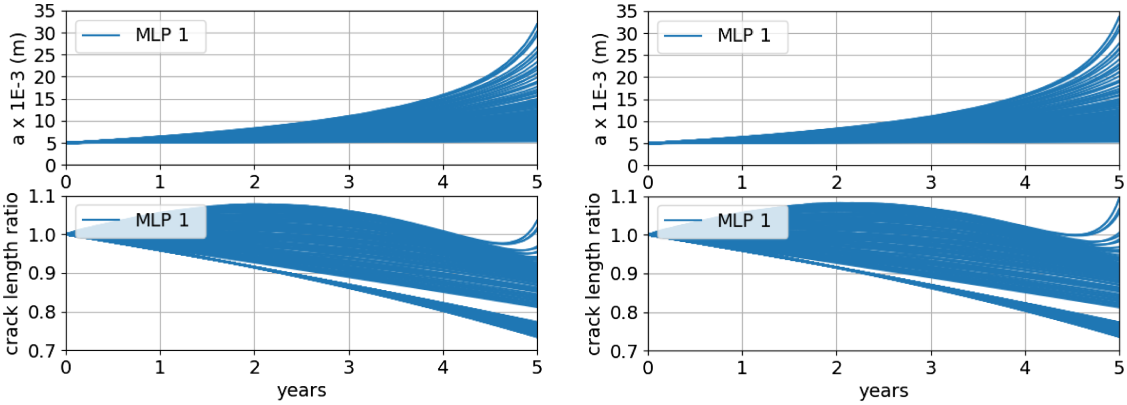


a) Boxplot of percent error in crack length prediction

b) Yearly observations

c) Monthly observations

Fig. 11 Physics-informed neural network predictions versus actual crack length at fifth year for entire fleet (300 aircraft): $\%\text{error} = 100 \times (a_{\text{PRED}} - a_{\text{actual}}/a_{\text{actual}})$.



a) Yearly observations

b) Monthly observations

Fig. 12 Actual and physics-informed neural network predicted crack length over time.

that a monitoring system that offloads data yearly or monthly is more economical to operate and maintain when compared to one that is expected to produce data on a daily basis, for example.

Figure 11 shows how the predictions at the end of the fifth year for all 300 aircraft compare with actual crack lengths. Interestingly, Figs. 11b and 11c confirm the robustness of the physics-informed neural networks when trained with yearly observations. We also see that the relatively large percent errors (let us say above 10%) are likely coming from aircraft exhibiting small crack lengths at the end of the fifth year. On the basis of the number of training points and prediction errors, the physics-informed neural networks outperform the data-driven models for this application.

Finally, Fig. 12b illustrates the model predictions up to the end of the fifth year for all 300 aircraft for multilayer perceptron 1 (detailed in Table 4). As opposed to Fig. 9, Fig. 12b shows that the crack length trends as well as the curve over time are well approximated (that is attributed to the physics-informed layer). The models are better suited for tracking crack length than the data-driven counterparts (i.e., the LSTM- and the GRU-based neural networks).

C. Scenario II: Scheduled Inspection of Control Point

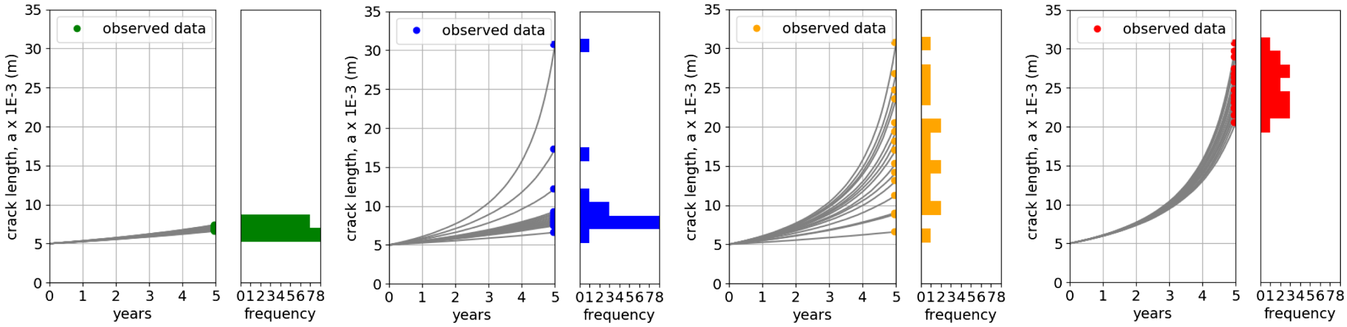
As previously mentioned, in the second scenario, we also consider the case in which the control point in the fuselage panel is inspected in regular intervals through nondestructive evaluation approaches (e.g., eddy current, ultrasound, dye penetrant inspection, etc.). Due to cost associated with inspection (mainly, downtime, parts, and labor), it is customary to perform inspection in predefined intervals (which might vary from control point to control point). Usually, aircraft are inspected in batches to avoid grounding the entire fleet. In such a case, data gathered on part of the fleet are used to build predictive models for the entire fleet. These predictive models can be used to guide the decision of which aircraft should be inspected next.

Table 5 Scenarios for inspection data

	Inspected aircraft			
Distribution of observed crack length	5	15	30	60
Case 1: biased toward small crack lengths	——	✓	——	——
Case 2: following true distribution of crack length	——	✓	——	——
Case 3: uniform coverage of crack lengths	✓	✓	✓	✓
Case 4: biased toward large crack lengths	——	✓	——	——

Table 5 details the hypothetical cases for selecting the aircraft out of fleet for inspection. Figure 13 highlights the 15 observed crack lengths at the end of the fifth year for cases 1 to 3 of Table 5. In the training of the recurrent neural networks, the inputs are always observed (i.e., the full far-field stress range time history is observed). However, the output is only observed at inspection. At a rate of four flights per day, in a period of five years, this means that we observed 5 to 60 time histories of 7300 data points each (total of 36,500 to 438,000 input points) and only 5 to 60 output observations. Here, the intent is to study the influence of number and distribution of inspections (output observations) in the training of the neural network. In all cases, inspection is assumed to take place at the fifth year of operation. Similarly to what would happen in real life, the first inspection results are used to train the models, which will be used to make crack length prediction across the entire fleet. Based on the performance results of data-driven and physics-informed neural networks in scenario I, we decided to focus this study only on the physics-informed neural networks. The poor performance of the purely data-driven neural networks as the number of training points gets reduced indicates they are not suitable for scenario II.

We first study the impact of the multilayer perceptron design in the training and validation of the physics-informed recurrent neural networks. We use data from 15 aircraft in which there is uniform coverage of the crack lengths (case 3 from Table 5, shown in Fig. 13c). The mean square error has fast convergence throughout training, as shown in Fig. 14a. Figure 14b shows the predictions at the end of the



a) Case 1

b) Case 2

c) Case 3

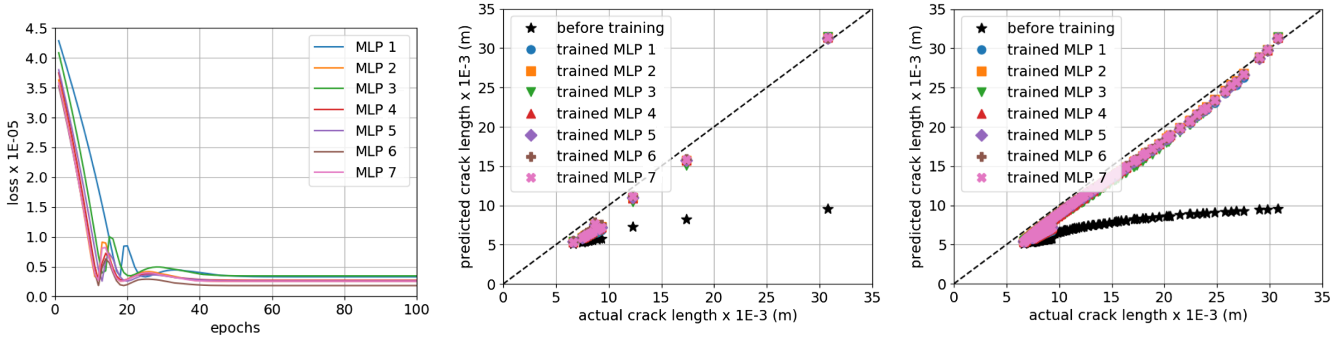
d) Case 4

Fig. 13 Fatigue crack length history and observations (15 aircraft) at the fifth year. Details about each case are found in Table 5.

fifth year for the training set (15 inspected aircraft) before and after training. Figure 14c shows the predictions at the end of the fifth year for the entire fleet (300 aircraft) before and after training. In both cases, the model initially tends to underpredict the large crack lengths. After the recurrent neural network is trained, the predictions are in good agreement with the actual values. There are only marginal differences in performance of the various multilayer perceptron configurations (confirming that the stress intensity range is relatively simple function of current crack length and far-field stress).

Figure 15 illustrates the crack length predictions over time for multilayer perceptron (MLP) 1 (see Table 4 for details) before and after training and how they compare with the actual crack length histories. As seen in Fig. 15a, there is good agreement between predicted and actual crack length histories. Figure 15b shows the ratio between the predicted and actual crack growths over time for the entire fleet before and after training. Initially, the model grossly underpredicts large crack lengths and predictions are within 35 and 85% of the actual crack length. After the recurrent neural network is trained, predictions stay within ±15% of the actual crack length, for the most part.

Figure 16 shows how the number of inspected aircraft affects the prediction accuracy of the physics-informed neural network.

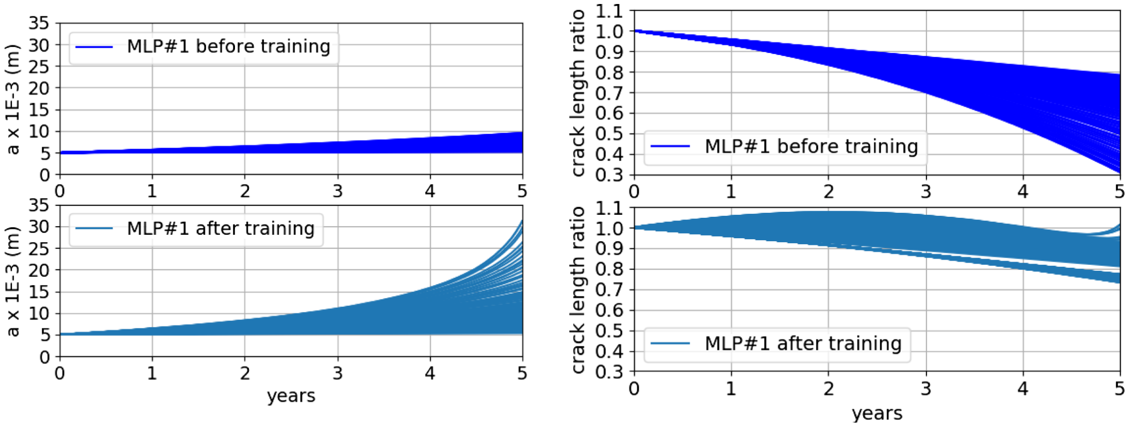


a) Loss function history

b) Predictions at the fifth year for the training data (15 aircraft)

c) Predictions at the fifth year for the entire fleet (300 aircraft)

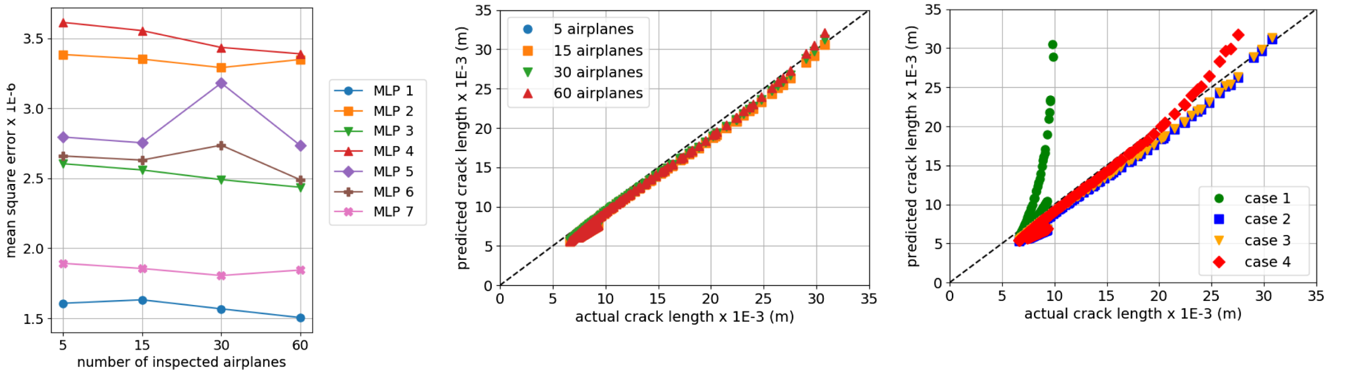
Fig. 14 Loss function history and predictions before and after training.



a) Crack length histories

b) Ratio between predicted and actual crack length

Fig. 15 Actual and predicted crack lengths over time for the entire fleet (300 aircraft).



a) Mean squared error versus number of training points

b) MLP 1 predictions versus actual crack length at the fifth year for entire fleet (300 aircraft) for different number of training points

c) Predictions versus actual crack length at the fifth year for entire fleet (300 aircraft) when MLP#1 is trained with different datasets (see Table 5 and Fig 13 for further details)

Fig. 16 Effect of training points in crack length predictions. Table 4 details the MLP configurations.

As mentioned before, besides the time series for loads (inputs), only observations for crack at the fifth year are used for training the model. Figure 16a shows the mean squared error (i.e., loss function at the end of training) as a function of the number of training points. Figure 16b shows the predictions versus actual crack lengths for the entire fleet at the end of the fifth year for MLP 1. Even with as few as five inspected aircraft (entire load histories and crack length at the fifth year), the model is capable of producing accurate predictions. This might look counterintuitive at first because one would expect that the more crack length observations used for training, the more accurate the models would be. Nevertheless, we found that, even for the case of five inspected aircraft, the number of input observations (36,500) is large enough compared to the number of trainable parameters (as shown in Table 4, MLP 1 has only 21 trainable parameters). On top of that, the physics-informed layer (Paris law) greatly influences the shape of the output versus time (monotonically and exponentially increasing). Therefore, the resulting physics-informed neural networks are relatively accurate, even with few observed outputs.

Last but not least, we also studied the effect of the distribution of crack length observations used for training the recursive neural network. For the sake of illustration, consider that the training set consists of observations for crack lengths and far-field cyclic stress at 15 different aircraft. One might be interested in looking at how well the resulting model is when the crack length observation is biased toward the low values (or toward high values) or, maybe, the distribution of crack length observations does not reflect the fleet distribution. In real life, in the absence of estimators for crack length, the first planes to be inspected are chosen based on rudimentary models based on the history of flown missions. Table 5 and Fig. 13 detail the distribution of observed crack lengths considered here. Figure 16c shows the summary of results for this part of the study. Interestingly, except for case 2, the trained physics-informed neural network was able to predict crack length (there are only minor differences among cases 1, 3, and 4). This indicates that as long as the range of observed crack length covers the plausible crack length range at the fleet level, the resulting model tends to be accurate, and the distribution of observed crack length has minor effects on the resulting network.

D. Notes About Computational Cost

Our implementation is done in TensorFlow (version 2.0.0-beta1) using the Python application programming interface. To replicate our results, the interested reader can download codes and data, as well as install the PINN package (base package for physics-informed neural networks used in this work) available at Ref. [67]. In light of the discussed application, the computational cost associated with the neural networks is considered small (training done in a few minutes using a laptop computer and predictions for the five years of operation at a small fraction of a second per aircraft). The intended use of the models is such that after data are collected, models will be trained and prediction is performed across the fleet. The model predictions are used to decide which aircraft should be inspected next and potentially monitor the fleet with predictions done after each flight. In other words, the few minutes and fractions of seconds needed to run the training and prediction are negligible compared to the time between flights and between scheduled inspections (given the moderate crack growth rates for some aircraft, it is conceivable that the model is exercised between large periods of time, such as weekly runs).

E. Intended Usage and Limitations of the Proposed Algorithm

We believe our proposed approach can be used in several practical applications. For example, engineers and scientists could leverage physics-informed kernels that have been previously developed and proved to be able to model certain failure modes. Then, neural networks layers can be used to compensate limitations of such physics-informed kernels by modeling parts of the system that are not fully understood. This is a straightforward and practical way to characterize model-form uncertainty. Although we illustrated the case in which physics-informed and data-driven layers are connected in series, we believe the final design of the neural network architecture depends on the problem. The implementation of MODEL in Fig. 3a can combine physics-informed and data-driven layers forming complex architectures (mixing series, parallel, bridge, and other arrangements).

We expect the hybrid models should perform very well in cases for which inputs $\boldsymbol{x}_t$ are observed throughout all the time stamps but outputs $\boldsymbol{a}_t$ are observed only at few time stamps. The physics-informed kernels are expected to compensate for the lack of output observations. In cases where both inputs and outputs are observed throughout the time stamps, we acknowledge that purely data-driven approaches could perform as well as the hybrid models.

We advocate toward the hybrid implementation of MODEL, combining physics-informed and data-driven layers. As we will demonstrate with the numerical experiments, we have observed that the hybrid model requires very few output observations to be trained. Unfortunately, one might also argue that the hybrid nature of the model is its main limitation. In fact, we believe there are at least two cases that can complicate the implementation of our proposed algorithm. First, it could be that the understanding of the physics is so limited that no physics-informed approximations are available. In this case, one might have to use a purely data-driven implementation of MODEL (or maybe other recurrent neural network architecture, such as the long short-term memory). Even though this is a valid approach, we believe it could limit the benefits in terms of reduction in required training data. Second, the physics-informed kernels need to be fast to compute (i.e., computational cost comparable to matrix algebra found in multilayer perceptrons). Complex models, such as those involving iterative solvers, could make the computational cost of training and prediction prohibitive and/or make it difficult to fit the neural network.

V. Conclusions

In this paper, a novel physics-informed recurrent neural network was proposed for cumulative damage modeling. The case of monitoring fatigue crack growth in a fleet of aircraft was specifically investigated. As inputs for the cumulative damage model, the current damage level (crack length) and far-field stresses (however, the recurrent neural networks can take other inputs, depending on the problem) were considered. Well-known recurrent neural network cells were tested, such as the long short-term memory and the gated recurrent unit; and they were compared with the proposed novel physics-informed cell for cumulative damage modeling. This proposed cell is designed such that models can be built using purely data-driven or physics-informed layers or, more interestingly, hybrids of physics-informed and data-driven layers (as the model discussed in this paper). Two numerical experiments were designed in which a fleet of 300 aircraft is to be monitored. In the first scenario, onboard structural and health monitoring sensors provide crack length observations for a portion of the fleet. In the second scenario, crack length observation is obtained through scheduled inspection.

With the help of the numerical studies, it was demonstrated that recurrent neural networks can be used to model cumulative damage (here, exemplified by fatigue crack growth). For the case in which onboard structural and health monitoring sensors are installed, it was learned that 1) architectures such as the long short-term memory and the gated recurrent unit tend to require frequent aircraft inspection data so that predictions of trained models can start tracking damage over time, and 2) the proposed recurrent neural network cell (hybrid of data-driven and physics-informed layers) can track damage over time, even when trained with a limited number of inspection data. As expected, it was confirmed that the performance of purely data-driven architectures is highly dependent on the amount of data; unfortunately, for the studied scenario, their predictions tend to be poor. For the case in which crack length data are obtained through scheduled inspection, it was learned that the proposed recurrent neural network cell successfully models fatigue crack growth. The effect of the distribution and number of training data was studied in the accuracy of the crack length predictions at the fleet level after five years worth of operation. It was learned that even with a reduced number of data points, the proposed physics-informed neural network can approximate fatigue crack growth.

The results motivate extending the study in several aspects. For example, studying the effect of improved physics of failure models (e.g., by including both initiation and propagation in the cumulative damage) is suggested. It is seen that considering uncertainty is important for this class of problems. For example, one can study the effect of the scatter in material properties as well as uncertainty in damage detection and quantification. Finally, one can also study how this physics-informed neural networks could be used to help decision making in operations and maintenance of industrial assets (e.g., by guiding decisions regarding repair and replacement of components).

Acknowledgments

This work was supported by the University of Central Florida (UCF). Nevertheless, any view and conclusions expressed in this material are those of the authors alone. Therefore, the UCF does not accept any liability in regard thereto.

References

[1] Tsang, A. H. C., "Condition-based Maintenance: Tools and Decision Making," *Journal of Quality in Maintenance Engineering*, Vol. 1, No. 3, 1995, pp. 3–17.
https://doi.org/10.1108/13552519510096350

[2] Jardine, A. K. S., Lin, D., and Banjevic, D., "A Review on Machinery Diagnostics and Prognostics Implementing Condition-Based Maintenance," *Mechanical Systems and Signal Processing*, Vol. 20, No. 7, 2006, pp. 1483–1510.
https://doi.org/10.1016/j.ymssp.2005.09.012

[3] Peng, Y., Dong, M., and Zuo, M. J., "Current Status of Machine Prognostics in Condition-based Maintenance: A Review," *International Journal of Advanced Manufacturing Technology*, Vol. 50, Nos. 1–4, 2010, pp. 297–313.
https://doi.org/10.1007/s00170-009-2482-0

[4] Alaswad, S., and Xiang, Y., "A Review on Condition-based Maintenance Optimization Models for Stochastically Deteriorating System," *Reliability Engineering and System Safety*, Vol. 157, Jan. 2017, pp. 54–63.
https://doi.org/10.1016/j.ress.2016.08.009

[5] Martin, J. D., Finke, D. A., and Ligetti, C. B., "On the Estimation of Operations and Maintenance Costs for Defense Systems," *Proceedings of the 2011 Winter Simulation Conference (WSC)*, Phoenix, AZ, Dec. 2011, pp. 2478–2489.
https://doi.org/10.1109/WSC.2011.6147957

[6] "Department of the Navy Total Ownership Cost (TOC) Guidebook," U.S. Dept. of the Navy, 2012, https://www.ncca.navy.mil/references/DON_TOC_Guidebook.pdf [retrieved 14 Feb. 2019].

[7] Si, X.-S., Wang, W., Hu, C.-H., and Zhou, D.-H., "Remaining Useful Life Estimation—A Review on the Statistical Data Driven Approaches," *European Journal of Operational Research*, Vol. 213, No. 1, 2011, pp. 1–14.
https://doi.org/10.1016/j.ejor.2010.11.018

[8] Tamilselvan, P., and Wang, P., "Failure Diagnosis Using Deep Belief Learning Based Health State Classification," *Reliability Engineering and System Safety*, Vol. 115, July 2013, pp. 124–135.
https://doi.org/10.1016/j.ress.2013.02.022

[9] Son, K. L., Fouladirad, M., Barros, A., Levrat, E., and Iung, B., "Remaining Useful Life Estimation Based on Stochastic Deterioration Models: A Comparative Study," *Reliability Engineering and System Safety*, Vol. 112, April 2013, pp. 165–175.
https://doi.org/10.1016/j.ress.2012.11.022

[10] Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., and Beghi, A., "Machine Learning for Predictive Maintenance: A Multiple Classifier Approach," *IEEE Transactions on Industrial Informatics*, Vol. 11, No. 3, 2015, pp. 812–820.
https://doi.org/10.1109/TII.2014.2349359

[11] Khan, S., and Yairi, T., "A Review on the Application of Deep Learning in System Health Management," *Mechanical Systems and Signal Processing*, Vol. 107, July 2018, pp. 241–265.
https://doi.org/10.1016/j.ymssp.2017.11.024

[12] Stadelmann, T., Tolkachev, V., Sick, B., Stampfli, J., and Dürr, O., "Beyond ImageNet-Deep Learning in Industrial Practice," *Applied Data Science-Lessons Learned for the Data-Driven Business*, Springer, New York, 2018, pp. 205–232.
https://doi.org/10.1007/978-3-030-11821-1_12

[13] Daigle, M. J., and Goebel, K., "Model-Based Prognostics with Concurrent Damage Progression Processes," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 43, No. 3, 2013, pp. 535–546.
https://doi.org/10.1109/TSMCA.2012.2207109

[14] Li, C., Mahadevan, S., Ling, Y., Choze, S., and Wang, L., "Dynamic Bayesian Network for Aircraft Wing Health Monitoring Digital Twin," *AIAA Journal*, Vol. 55, No. 3, 2017, pp. 930–941.
https://doi.org/10.2514/1.J055201

[15] Ling, Y., Asher, I., Wang, L., Viana, F. A. C., and Khan, G., "Information Gain-Based Inspection Scheduling for Fatigued Aircraft Components," *AIAA Scitech 2017 Forum*, AIAA Paper 2017-1565, 2017.
https://doi.org/10.2514/6.2017-1565

[16] Yucesan, Y. A., and Viana, F. A. C., "Onshore Wind Turbine Main Bearing Reliability and Its Implications in Fleet Management," *AIAA Scitech 2019 Forum*, AIAA Paper 2019-1225, 2019.
https://doi.org/10.2514/6.2019-1225

[17] Berri, P. C. C., Vedova, M. D. L. D., and Mainini, L., "Real-time Fault Detection and Prognostics for Aircraft Actuation Systems," *AIAA Scitech 2019 Forum*, AIAA Paper 2019-2210, 2019.
https://doi.org/10.2514/6.2019-2210

[18] Byington, C. S., Watson, M., and Edwards, D., "Data-driven Neural Network Methodology to Remaining Life Predictions for Aircraft Actuator Components," *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No. 04TH8720)*, Vol. 6, IEEE, New York, 2004, pp. 3581–3589.

[19] Jacazio, A. D. M. G., and Sorli, M., "Enhanced Particle Filter Framework for Improved Prognosis of Electro-Mechanical Flight Controls Actuators," *Fourth European Conference of the Prognosis and Health Management Society*, PHM Society, Utrecht, The Netherlands, 2018, Paper 10

[20] Lee, H., and Kang, I. S., "Neural Algorithm for Solving Differential Equations," *Journal of Computational Physics*, Vol. 91, No. 1, 1990, pp. 110–131.
https://doi.org/10.1016/0021-9991(90)90007-N

[21] Meade, A. J., Jr., and Fernandez, A. A., "Solution of Nonlinear Ordinary Differential Equations by Feedforward Neural Networks," *Mathematical and Computer Modelling*, Vol. 20, No. 9, 1994, pp. 19–44.
https://doi.org/10.1016/0895-7177(94)00160-X

[22] Takeuchi, J., and Kosugi, Y., "Neural Network Representation of Finite Element Method," *Neural Networks*, Vol. 7, No. 2, 1994, pp. 389–395.
https://doi.org/10.1016/0893-6080(94)90031-0

[23] Wang, Y.-J., and Lin, C.-T., "Runge-Kutta Neural Network for Identification of Dynamical Systems in High Accuracy," *IEEE Transactions on Neural Networks*, Vol. 9, No. 2, 1998, pp. 294–307.
https://doi.org/10.1109/72.661124

[24] Sankararaman, S., McLemore, K., Mahadevan, S., Bradford, S. C., and Peterson, L. D., "Test Resource Allocation in Hierarchical Systems Using Bayesian Networks," *AIAA Journal*, Vol. 51, No. 3, 2013, pp. 537–550.
https://doi.org/10.2514/1.J051542

[25] Viana, F. A. C., Simpson, T. W., Balabanov, V., and Toropov, V., "Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come?" *AIAA Journal*, Vol. 52, No. 4, 2014, pp. 670–690.
https://doi.org/10.2514/1.J052375

[26] Xu, H., Liu, R., Choudhary, A., and Chen, W., "A Machine Learning-Based Design Representation Method for Designing Heterogeneous Microstructures," *Journal of Mechanical Design*, Vol. 137, No. 5, 2015, Paper 051403.
https://doi.org/10.1115/1.4029768

[27] Singh, A. P., Medida, S., and Duraisamy, K., "Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils," *AIAA Journal*, Vol. 55, No. 7, 2017, pp. 2215–2227.
https://doi.org/10.2514/1.J055595

[28] Raissi, M., and Karniadakis, G. E., "Hidden Physics Models: Machine Learning of Nonlinear Partial Differential Equations," *Journal of Computational Physics*, Vol. 357, March 2018, pp. 125–141.
https://doi.org/10.1016/j.jcp.2017.11.039

[29] Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K., "Neural Ordinary Differential Equations," *31st Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Curran Assoc., Red Hook, 2018, pp. 6572–6583.

[30] Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Numerical Gaussian Processes for Time-dependent and Nonlinear Partial Differential Equations," *SIAM Journal on Scientific Computing*, Vol. 40, No. 1, 2018, pp. A172–A198.
https://doi.org/10.1137/17M1120762

[31] Kani, J. N., and Elsheikh, A. H., "Reduced-Order Modeling of Subsurface Multi-Phase Flow Models Using Deep Residual Recurrent Neural Networks," *Transport in Porous Media*, Vol. 126, No. 3, 2018, pp. 713–741.
https://doi.org/10.1007/s11242-018-1170-7

[32] Raissi, M., "Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations," *Journal of Machine Learning Research*, Vol. 19, No. 25, 2018, pp. 1–24.

[33] Wu, J.-L., Xiao, H., and Paterson, E., "Physics-informed Machine Learning Approach for Augmenting Turbulence Models: A Comprehensive Framework," *Physical Review Fluids*, Vol. 3, No. 7, 2018, Paper 074602. https://doi.org/10.1103/PhysRevFluids.3.074602

[34] Hesthaven, J. S., and Ubbiali, S., "Non-Intrusive Reduced Order Modeling of Nonlinear Problems Using Neural Networks," *Journal of Computational Physics*, Vol. 363, June 2018, pp. 55–78. https://doi.org/10.1016/j.jcp.2018.02.037

[35] Swischuk, R., Mainini, L., Peherstorfer, B., and Willcox, K., "Projection-Based Model Reduction: Formulations for Physics-Based Machine Learning," *Computers and Fluids*, Vol. 179, Jan. 2019, pp. 704–717. https://doi.org/10.1016/j.compfluid.2018.07.021

[36] Schober, M., Duvenaud, D. K., and Hennig, P., "Probabilistic ODE Solvers with Runge-Kutta Means," *27th Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Curran Assoc., Red Hook, NY, 2014, pp. 739–747.

[37] Kani, J. N., and Elsheikh, A. H., "DR-RNN: A Deep Residual Recurrent Neural Network for Model Reduction," Preprint, submitted 4 Sept. 2017, http://arxiv.org/abs/1709.00939.

[38] Fatemi, A., and Yang, L., "Cumulative Fatigue Damage and Life Prediction Theories: A Survey of the State of the Art for Homogeneous Materials," *International Journal of Fatigue*, Vol. 20, No. 1, 1998, pp. 9–34. https://doi.org/10.1016/S0142-1123(97)00081-9

[39] Frangopol, D. M., Kallen, M.-J., and van Noortwijk, J. M., "Probabilistic Models for Life-cycle Performance of Deteriorating Structures: Review and Future Directions," *Progress in Structural Engineering and Materials*, Vol. 6, No. 4, 2004, pp. 197–212. https://doi.org/10.1002/pse.180

[40] Bogdanoff, J. L., and Kozin, F., *Probabilistic Models of Cumulative Damage*, Wiley, New York, 1985.

[41] Nakagawa, T., and Kijima, M., "Replacement Policies for a Cumulative Damage Model with Minimal Repair at Failure," *IEEE Transactions on Reliability*, Vol. 38, No. 5, 1989, pp. 581–584. https://doi.org/10.1109/24.46485

[42] Dowling, N. E., *Mechanical Behavior of Materials: Engineering Methods for Deformation, Fracture, and Fatigue*, Pearson, Upper Saddle River, NJ, 2012.

[43] Paris, P., and Erdogan, F., "A Critical Analysis of Crack Propagation Laws," *Journal of Basic Engineering*, Vol. 85, No. 4, 1963, pp. 528–533. https://doi.org/10.1115/1.3656900

[44] "MMPDS–12: Metallic Materials Properties Development and Standardization," MMPDS Collaborators, Battelle Memorial Institute, 2017, https://www.mmpds.org [retrieved 14 Feb. 2019].

[45] Boresi, A. P., and Schmidt, R. J., *Advanced Mechanics of Materials*, 6th ed., Wiley, Hoboken, NJ, 2003.

[46] Murakami, Y., *Stress Intensity Factors Handbook*, Oxford, New York, 1987.

[47] Tada, H., Paris, P. C., and Irwin, G. R., *The Stress Analysis of Cracks Handbook*, 3rd ed., ASME Press, New York, 2000.

[48] "Standard Practices for Cycle Counting in Fatigue Analysis," ASTM STD E1049-85(2017), West Conshohocken, PA, 2017. https://doi.org/10.1520/E1049-85R17

[49] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, Cambridge, MA, 2016, http://www.deeplearningbook.org [retrieved 14 Feb. 2019].

[50] Connor, J. T., Martin, R. D., and Atlas, L. E., "Recurrent Neural Networks and Robust Time Series Prediction," *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, 1994, pp. 240–254. https://doi.org/10.1109/72.279188

[51] Sak, H., Senior, A., and Beaufays, F., "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," *Fifteenth Annual Conference of the International Speech Communication Association*, Singapore, Sept. 2014, pp. 338–342, https://www.isca-speech.org/archive/interspeech_2014/i14_0338.html [retrieved 14 Feb. 2019].

[52] Chauhan, S., and Vig, L., "Anomaly Detection in ECG Time Signals via Deep Long Short-Term Memory Networks," *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE Publ., Piscataway, NJ, 2015, pp. 1–7. https://doi.org/10.1109/DSAA.2015.7344872

[53] Graves, A., Mohamed, A., and Hinton, G., "Speech Recognition with Deep Recurrent Neural Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE Publ., Piscataway, NJ, 2013, pp. 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947

[54] Sutskever, I., Martens, J., and Hinton, G., "Generating Text with Recurrent Neural Networks," *28th International Conference on Machine Learning*, edited by L. Getoor, and T. Scheffer, ACM, Bellevue, WA, 2011, pp. 1017–1024, https://icml.cc/2011/papers/524_icmlpaper.pdf [retrieved 14 Feb. 2019].

[55] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[56] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., "Gated Feedback Recurrent Neural Networks," *Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research*, Vol. 37, edited by F. Bach, and D. Blei, PMLR, Lille, France, 2015, pp. 2067–2075, http://proceedings.mlr.press/v37/chung15.html [retrieved 14 Feb. 2019].

[57] Amsallem, D., and Farhat, C., "Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity," *AIAA Journal*, Vol. 46, No. 7, 2008, pp. 1803–1813. https://doi.org/10.2514/1.3537410.2514/1.35374

[58] Benner, P., Cohen, A., Ohlberger, M., and Willcox, K., *Model Reduction and Approximation: Theory and Algorithms*, Vol. 15, SIAM, Philadelphia, 2017.

[59] Roach, D., "Real Time Crack Detection Using Mountable Comparative Vacuum Monitoring Sensors," *Smart Structures and Systems*, Vol. 5, No. 4, 2009, pp. 317–328. https://doi.org/10.12989/sss.2009.5.4.317

[60] Hill, K. O., and Meltz, G., "Fiber Bragg Grating Technology Fundamentals and Overview," *Journal of Lightwave Technology*, Vol. 15, No. 8, 1997, pp. 1263–1276. https://doi.org/10.1109/50.618320

[61] Aldrin, J. C., Knopp, J. S., Lindgren, E. A., and Jata, K. V., "Model-assisted Probability of Detection Evaluation for Eddy Current Inspection of Fastener Sites," *AIP Conference Proceedings*, Vol. 1096, AIP Publ., Melville, NY, 2009, pp. 1784–1791. https://doi.org/10.1063/1.3114175

[62] Drinkwater, B. W., and Wilcox, P. D., "Ultrasonic Arrays for Non-Destructive Evaluation: A Review," *NDT & E International*, Vol. 39, No. 7, 2006, pp. 525–541. https://doi.org/10.1016/j.ndteint.2006.03.006

[63] Hoye, M. V., "Fluorescent Penetrant Crack Detection," U.S. Patent 4621193A, Nov. 1986.

[64] Pascanu, R., Mikolov, T., and Bengio, Y., "On the Difficulty of Training Recurrent Neural Networks," *Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research*, Vol. 28, edited by S. Dasgupta, and D. McAllester, PMLR, Atlanta, 2013, pp. 1310–1318, http://proceedings.mlr.press/v28/pascanu13.html [retrieved 14 Feb. 2019].

[65] Sutskever, I., "Training Recurrent Neural Networks," Ph.D. Thesis, Univ. of Toronto, Toronto, ON, Canada, 2013.

[66] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," Preprint, submitted 22 Dec. 2014, http://arxiv.org/abs/1412.6980.

[67] Viana, F. A. C., Nascimento, R. G., Yucesan, Y., and Dourado, A., "Physics-Informed Neural Networks Package," *GitHub* (online database), Aug. 2019, https://github.com/PML-UCF/pinn [retrieved 14 Feb. 2019].

K. E. Willcox
Associate Editor