

Physics-Informed Neural Networks for Corrosion-Fatigue Prognosis

Arinan Dourado¹ and Felipe A. C. Viana²

^{1,2}University of Central Florida, Orlando, FL, 12309, USA

arinandourado@knights.ucf.edu

viana@ucf.edu

ABSTRACT

In this paper, we present a novel physics-informed neural network modeling approach for corrosion-fatigue. The hybrid approach is designed to merge physics-informed and data-driven layers within deep neural networks. The result is a cumulative damage model where the physics-informed layers are used to model the relatively well-understood physics (crack growth through Paris law) and the data-driven layers account for the hard to model effects (damage bias due to corrosion). A numerical experiment is used to present the main features of the proposed physics-informed recurrent neural network for damage accumulation. The test problem consists of predicting corrosion-fatigue of an Al 2024-T3 alloy used on fuselage panels of aircraft wings. Besides cyclic loading, the panels are also subjected to saline corrosion (3.5% solution of sodium chloride, emulating coastal exposure). The physics-informed neural network is trained using full observation of inputs (far-field loads, stress ratio and a corrosion index for environment corrosivity defined by airport) and very limited observation of outputs (crack length at inspection for only a small portion of the fleet). We then address the following question: *Is the physics-informed neural network able to properly compensate corrosion effects on fatigue damage accumulation?* Results demonstrate that our proposed framework is able to accurately compensate for damage bias due to the lack of corrosion modeling in the mechanical fatigue model. Additionally, results indicate that corrosion plays a drastic role in crack propagation significantly reducing useful life.

1. INTRODUCTION

Aging aircraft fleets is currently a major issue. It was estimated that in the early 2000s about 60% of the US manufactured fleet consisted of aircrafts older than 20 years (Shi and Mahadevan, 2001). One of the main causes of failures in aged aircraft is corrosion-fatigue damage to aluminum alloys components (Larrosa et al., 2018). Reports

Arinan Dourado and Felipe A. C. Viana. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

of corrosion pits leading to cracks in operating aircrafts can be found in Chen et al. (1994), Wang et al. (2001) and DuQuesnay et al. (2003). Under usage life, these components are exposed to aggressive environments making it susceptible to corrosion. The synergistic nature of corrosion and fatigue leads to early life failures at lower stress levels than designed.

In published literature corrosion-fatigue is modeled following the basic stages of pit nucleation and growth; pit to crack transition; crack propagation. Most published models focus in either pit to crack transition (Lindley, 1982; Kondo, 1989; and Chen et al, 1996) or crack growth (Wang et al., 2001). To the best of our knowledge, no published model related to corrosion-fatigue addresses the aforementioned features for prognosis

This contribution aims to evaluate the use of a hybrid physics-informed neural network to model corrosion-fatigue damage accumulation in an operating fleet. The proposed framework is an extension of the cumulative damage recurrent neural network introduced by Nascimento and Viana (2019). While Nascimento and Viana (2019) focused on fatigue crack growth under mechanical loads, here we focus on corrosion-fatigue. **Additionally, the main novelty of this work is the use of a hybrid recurrent neural network to estimate the prediction bias of a physical model.** In our proposed framework, data-driven layers (artificial neural networks) are trained to act as a bias estimator, compensating the corrosion damage bias on a well-known mechanical fatigue model (physics-informed layers of the neural network).

Literature reports several applications of recurrent neural networks (RNNs) to model time-series (Connor et al., 1994; Sak et al., 2014; Chauban and Vig, 2015). Recently, RNNs architectures have been proposed to address physics-based applications. Trischler and D’Eleuterio (2016) present a review on current RNNs training approaches for dynamical systems response approximation along with a proposed feedforward neural network that is able to replicate the system dynamics. Yu et al. (2018) expand the framework known as deep residual recurrent network (DR-RNN) into a hybrid model in order to simulate aircraft dynamics. Nascimento and Viana (2019) proposed an RNN cell inspired

by cumulative damage models and applied for crack propagation.

Training RNNs is a daunting task. For the proposed framework, crack length information is only provided after several thousands of operational cycles, forcing the neural network to backtrack the entire crack propagation history of each aircraft until that point. It is also expected from the derived model the ability to forecast damage propagation and aid in the fleet prognosis. This work is also aligned with the currently developed trend of machine learning in prognosis model (Susto et al, 2015; Zhao et al, 2016; Khan and Yairi 2018; and Li et al, 2018).

The remaining of this paper is organized as follows: a brief discussion of the proposed damage accumulation model is presented in Section 2. The considered numerical experiment and its discussed Section 3. The main results regarding the proposed physics-informed recurrent neural network for fleet diagnosis and prognosis are presented and discussed in Section 4. Finally, some final remarks and future work are addressed in Section 5.

2. PHYSICS-INFORMED RECURRENT NEURAL NETWORK FOR CORROSION-FATIGUE DAMAGE ACCUMULATION

2.1. Recurrent Neural Networks and Cumulative Damage Models

A recurrent neural network (Goodfellow et al., 2016) repeatedly apply transformations to given states in a time-series sequence, as shown in Figure 1-(a).

$$\mathbf{a}_t = f(\mathbf{x}_t, \mathbf{a}_{t-1}) \quad (1)$$

where $t \in [0, \dots, T]$ represent the time discretization, $\mathbf{a} \in \mathbb{R}^{n_d}$ are the states representing the sequence, $\mathbf{x}_t \in \mathbb{R}^{n_x}$ are input (observable) variables, and $f(\cdot)$ is the transformation to the hidden state (in its simplest form, $f(\cdot)$ is a perceptron).

Cumulative damage models (Fatemi and Yang, 1998; and Frangopol et al, 2004) track progressive distress through a time series and can be formulated as RNNs. Nascimento and Viana (2019) introduced the cumulative damage cell for RNNs as illustrated in Figure 1-(b). In such cell design, the state represents the cumulative damage at time t and ‘‘MODEL’’ maps the inputs \mathbf{x}_t and previously accumulated damage \mathbf{a}_{t-1} into a damage increment $\Delta \mathbf{a}_t$:

$$\mathbf{a}_t = \mathbf{a}_{t-1} + \Delta \mathbf{a}_t, \quad (2)$$

‘‘MODEL’’ is the implementation of the physics of failure, which is highly application dependent. Nevertheless, as far as modeling approach, the ‘‘MODEL’’ block could be (a) a data-driven model, such as a multi-layer perceptron, (b) a physics-informed model, with appropriate physical fidelity to reflect the failure mechanism within the expected computational

efficiency, or more interestingly, (c) a hybrid model, where some parts are physics-based while others are data-driven.

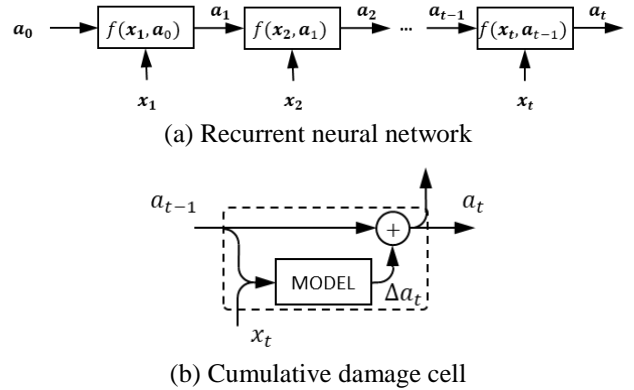


Figure 1. RNN cumulative damage model: (a) an unrolled recurrent neural network, and (b) cumulative damage cell.

2.2. Hybrid Recurrent Neural Networks for Bias Estimation

In this contribution, we extend the framework proposed by Nascimento and Viana (2019) to yield a **hybrid recurrent neural network** that acts as a **bias estimator** as illustrated in Figure 2. The proposed hybrid approach is designed to merge physics-informed and data-driven layers within deep neural networks. The result is a cumulative damage model where the physics-informed layers are used to model the relatively well-understood physics and the data-driven layers act as a bias compensator, accounting for effects of hard to model failure mechanisms.

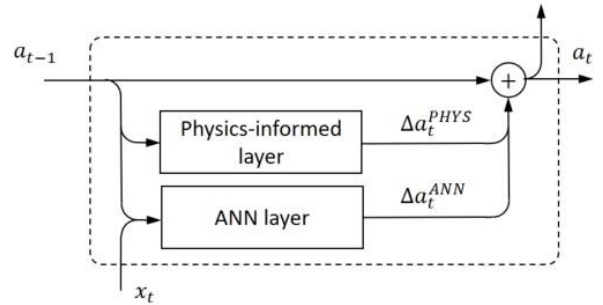


Figure 2. Bias estimation through an artificial neural network (ANN) layer.

2.3. Combined Damage Model: Corrosion-fatigue

Corrosion-fatigue is a very complex phenomenon involving pit nucleation, pit growth, fatigue crack nucleation, short crack growth, a transition from short crack to long crack, and long crack growth. Modeling corrosion-fatigue has proven to be a daunting task (Goswami and Hoepfner, 1995; Shi and Mahadevan, 2001) and it is not our objective to propose any

preferred method for it. Instead, we will focus on corrosion-fatigue crack growth to show how the framework proposed in the previous section can be designed and trained to act as a **bias estimator**, compensating the corrosion effects in the loading cycles subject to corrosion.

As shown in Menan and Henaft (2010) corrosion-fatigue crack propagation can be modeled through Paris' Law (Paris and Erdogan, 1963)

$$\frac{da}{dt} = C(\Delta K)^m, \quad (3)$$

where ΔK is the stress intensity range and C and m are material properties (dependent on the corrosive environment, as shown in Figure 3).

In its discrete form, Eq. (3) becomes

$$a_t = a_{t-1} + C(\Delta K_t)^m. \quad (4)$$

Assuming that the considered cracks are through the thickness center crack in an infinite plate with loads applied perpendicularly to the crack plane, the stress intensity can be expressed as shown in Eq. (5), where F is a geometry factor and ΔS_t is the far-field stress.

$$\Delta K_t = F\Delta S_t\sqrt{\pi a_{t-1}}. \quad (5)$$

The last effect we will consider in our model is the average load level through ratio between the minimum and maximum stress levels ($R = S_{min}/S_{max}$). A simple modification of Paris' model known as Walker equation (Dowling, 2012) is able to incorporate the stress ratio effects on the damage accumulation:

$$C = \frac{C_0}{(1-R)^{m(1-\gamma)}}, \quad (6)$$

where C_0 and m depend on the environmental conditions (see Figure 3) and γ depends on the material properties and loading conditions.

Using Eqs. (2) and (4), we can write the damage increment at Δa_t as a linear relationship in the log-space

$$\log_{10} \Delta a_t = \log_{10} C + m * \log_{10} \Delta K_t \quad (7)$$

where C and m are functions of the stress ration and environmental conditions (which functional form is not always known in real life).

In this paper, we assume that the corrosivity of an environment can be associated with an index C^{IDX} (much like a concentration of a particular corrosive agent in the air). Therefore, for modeling purposes, the material properties C and m can be expressed as functions of the stress ratio R and the proposed corrosion index C^{IDX} :

$$C = g(R, C^{IDX}) \text{ and } m = h(R, C^{IDX}) \quad (8)$$

where $g(\cdot)$ and $h(\cdot)$ are unknown input-output relationships.

Despite $g(\cdot)$ and $h(\cdot)$ being unknown functionals, their forms are bounded by the physical constraints illustrated in Figure 3 and Eq. (6). As shown in Figure 3, as the environment corrosivity becomes more aggressive C value increases while m value decreases. In this study, the corrosivity is expressed in terms of a corrosivity index C^{IDX} that essentially controls the interpolation between the two curves in Figure 3. From Eq. (6) it can be inferred that from stress ratio R within the range $0 < R < 1$, the C value will increase as R increases, if $0 < \gamma < 1$. Equation 6 also implies that m is not a function of the stress ratio R . Hence, regardless of its form, $g(\cdot)$ and $h(\cdot)$ must obey these constraints.

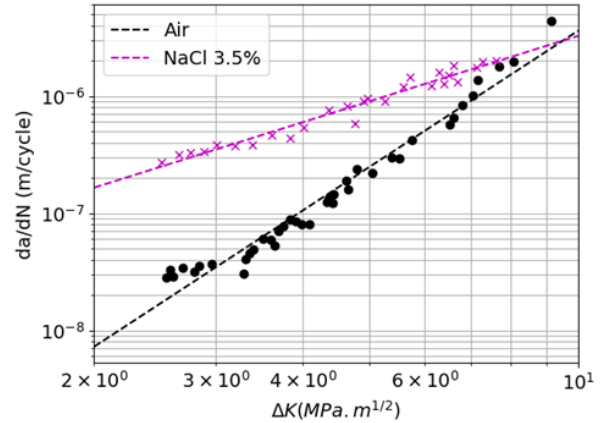


Figure 3. Paris' law coupon data (adapted from coupon tests presented by Menan and Henaft, 2010). For pure air $C = 5.008 \times 10^{-10}$ and $m = 3.859$. For sodium chloride (NaCl) 3.5% solution, $C = 4.577 \times 10^{-8}$ and $m = 1.853$.

With that, we propose the repeating RNN cell illustrated in Figure 4 to model corrosion-fatigue crack growth. While we maintain the physics-based implementation of the Walker model, we allow for the deep neural network to learn the dependence between the inputs stress ratio and corrosivity index and the output coefficients of the Paris' law. Since the damage increment in non-corrosive cycles is purely mechanical, in practice, the neural network is working as a bias corrector compensating for the effect of corrosive cycles.

3. CASE STUDY

3.1. Synthetic Airplane Fleet Data

Consider a hypothetical control point on the underside of the airplane wings in Figure 5. For simplicity, we assumed the airplane was designed to fly the 10 flight types and 10 missions presented in Table 1 and Table 2, respectively.

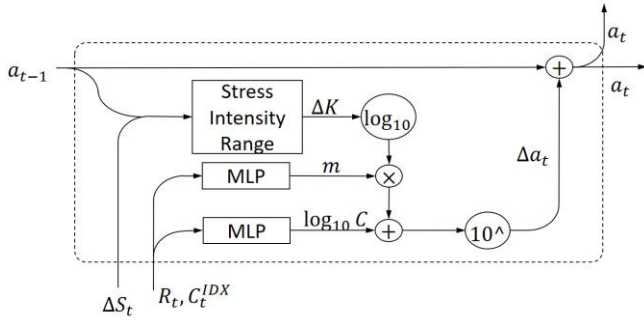


Figure 4. Proposed RNN cell for corrosion-fatigue crack propagation. MLP stands for multilayer perceptron.

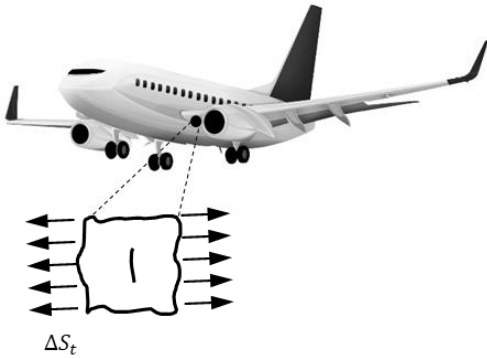


Figure 5. Control point on the airplane wing underside.

Now consider that:

- An original equipment manufacturer, airline company or service provider maintains a large aircraft fleet (hundreds to thousands). Here, we consider a fleet of 100 airplanes.
- Each airplane in the fleet flies, on average, 8 flights per day. Considering that each airplane is designed to endure 40,000 flights, the expected useful life of this panel is approximately 14 years.
- The panels are made of Al 2024-T3 alloy (see Figure 3 for material properties) with $a_0 = 0.5 \text{ mm}$ and $a_{max} = 20 \text{ mm}$.
- Inspection of control points is part of the scheduled maintenance activities. Here, we arbitrarily consider that the first inspection data is available for part of the fleet after 20,000 flights.

Figure 6 shows the evolution of fatigue crack growth under purely mechanical loading (i.e., disregarding the effect of corrosion). When corrosion is not considered, the 20,000 flight inspection is expected to return fatigue crack lengths below 1 mm.

3.2. Corrosion-fatigue

Let us assume that the original equipment manufacturer, airline companies, or service provider was not expecting to observe corrosion-fatigue failure in the fleet. However, due to unforeseen circumstances, the fleet is exposed to corrosive cycles at take-off and landing. This represents, on average, 5% of the total number of cycles in one mission (we considered that loads of type E and F are related to take-off and landing, respectively, and thus are subjected to corrosion). In this study, we penalize these cycles by accumulating damage using curves that are between air and sodium chloride (NaCl) 3.5%. The exact curve will depend on a corrosivity index C^{IDX} (which essentially controls the interpolation between the two curves). When $C^{IDX} = 0$, we use the curve for air. When $C^{IDX} = 1$, we use the curve for NaCl 3.5%.

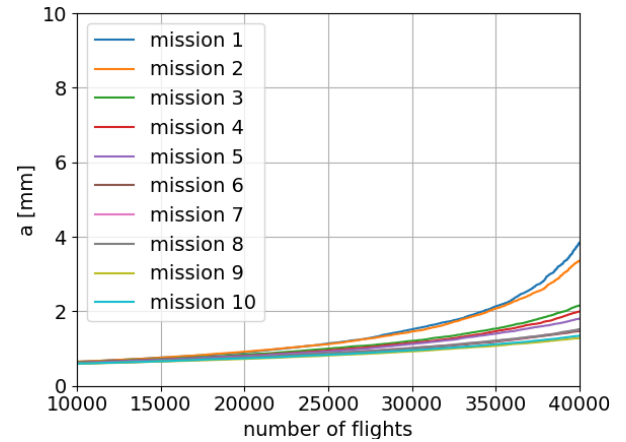


Figure 6. Fatigue crack growth over time (without the effect of corrosion).

Table 3 shows the C^{IDX} for the 10 distinct airports considered in this study. Figure 7 illustrates the crack length envelope, i.e., the history for the lowest and largest crack lengths in the fleet for each scenario (purely mechanical fatigue and corrosion-fatigue). It is clear how even the few corrosive take-offs and landing cycles can accelerate damage accumulation.

3.3. Physics-informed Neural Network Design

We consider the following information is available:

- For every airplane in the fleet: we consider the physics-informed neural network to have full visibility of its inputs on every cycle (in real-world applications, this is a considerable task by itself). This means that at each flown mission, far-field cyclic stresses, stress ratio, and airport corrosion index are known and treated as inputs.
- For part of the fleet: crack length observed after 20,000 flights.

Table 1. Flight type load matrix distribution and related normalized minimum and maximum stresses (adapted from De Jonge et al., 1973). The stresses are given in MPa.

Load magnitude						
	A	B	C	D	E	F
S_{min}	0.10	4.25	8.25	12.25	16.25	20.25
S_{max}	52.15	48.10	44.0	40.0	36	32
Load frequency						
Flight type	A	B	C	D	E	F
I	2	2	3	5	13	30
II	2	2	2	2	12	29
III	1	2	2	2	12	29
IV	1	1	2	2	12	29
V	1	1	1	2	12	29
VI	0	1	1	2	12	29
VII	0	0	1	2	12	29
VIII	0	0	0	2	12	29
IX	0	0	0	0	1	7
X	0	0	0	0	0	3

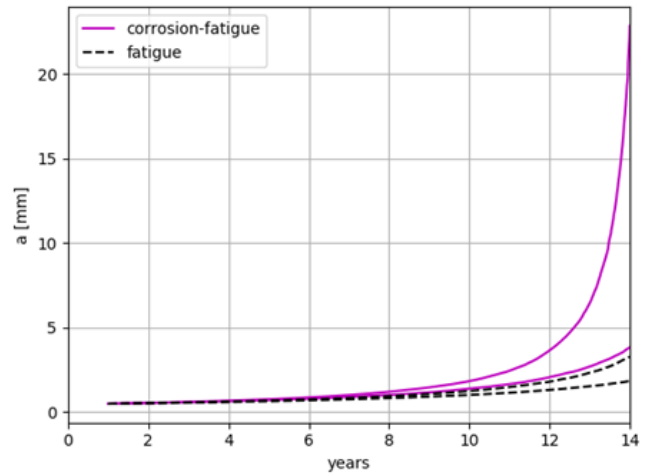
Table 2. Mission mix configuration (flight type probability per mission).

Flight type					
Mission	I	II	III	IV	V
1	$5.10 \cdot 10^{-5}$	$7.5 \cdot 10^{-5}$	$1.10 \cdot 10^{-3}$	$2.4 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$
2	$2.5 \cdot 10^{-5}$	$2.5 \cdot 10^{-5}$	$1.2 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$	$6.75 \cdot 10^{-3}$
3	0	0	$1.3 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$1.6 \cdot 10^{-2}$
4	0	$5.10 \cdot 10^{-5}$	$9.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$	$8.5 \cdot 10^{-3}$
5	0	0	$1.2 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$6.10 \cdot 10^{-3}$
6	$1.10 \cdot 10^{-4}$	$2.5 \cdot 10^{-5}$	$7.5 \cdot 10^{-4}$	$2.3 \cdot 10^{-3}$	$6.13 \cdot 10^{-3}$
7	0	0	0	$2.3 \cdot 10^{-3}$	$7.3 \cdot 10^{-3}$
8	$2.5 \cdot 10^{-5}$	0	$1.10 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$	$1.10 \cdot 10^{-2}$
9	0	$7.5 \cdot 10^{-5}$	$7.5 \cdot 10^{-4}$	$7.7 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$
10	$7.5 \cdot 10^{-5}$	0	$7.5 \cdot 10^{-4}$	$2.4 \cdot 10^{-3}$	$6.25 \cdot 10^{-3}$
Flight type					
Mission	VI	VII	VIII	IX	X
1	0.01	0.095	0.15	0.248	0.4824
2	0.025	0.035	0.155	0.273	0.5025
3	0.025	0.025	0.105	0.273	0.5530
4	0.013	0.095	0.055	0.323	0.5030
5	0.015	0.045	0.105	0.2975	0.5278
6	0.015	0.045	0.08	0.3975	0.4528
7	0.015	0.070	0.055	0.3225	0.5278
8	0.02	0.02	0.12	0.285	0.5528
9	0.025	0.025	0.055	0.2225	0.6528
10	0.005	0.05	0.08	0.2725	0.5828

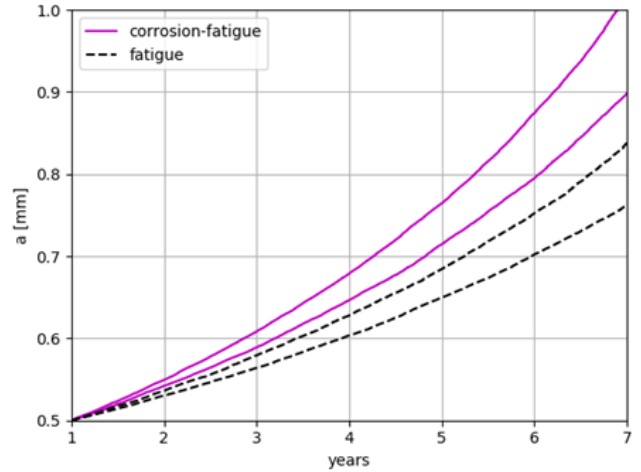
Table 3. Airport corrosion index values.

		Airport				
		A	B	C	D	E
C^{IDX}		0.517	0.389	0.361	0.545	0.781
		F	G	H	I	J
C^{IDX}		1	0.125	0.58	0.608	0.844

With that information, we proceed to build a hybrid physics-informed neural network model for corrosion-fatigue. In this model, there are two multilayer perceptrons (MLPs), as shown in Figure 4. Table 4 and Table 5 detail the multilayer perceptrons used in this work. We decided to use these architectures to illustrate the ability to fit a neural network with a large number of trainable parameters. No attempt was made to further simplify the multi-layer perceptron. In practical applications, we believe reducing the model is worth pursuing, as it could potentially lead to a more manageable number of trainable parameters without sacrificing accuracy.



(a) Fleet crack propagation envelope at 40,000 flights.



(b) Fleet crack envelope at 7th year.

Figure 7. Fleet exposed only to mechanical fatigue vs fleet exposed to the described corrosion-fatigue conditions, a) crack envelope of the fleet considering 5% of corrosive loads in take-off and landing, and 14 years of operation, b) crack envelope in the 7th year, when the inspection campaign is carried.

Table 4. Paris' law exponent m layer details. The first column addresses the type of layer used in the multilayer perceptron. A “dense” layer is a fully connected perceptron (i.e., all the inputs are connected to each neuron). The second column refers to the number of neurons (i.e., outputs) on each layer. The third column shows the number of parameters (weights and biases) yielded by each layer. Finally, the fourth column specifies if those parameters are tuned in the training stage. The last column specifies the activation function of each layer. Dense #0 is responsible to scale the layer inputs. Dense #1 to #3 are regular perceptrons. There are 91 trainable parameters and 6 non-trainable parameters.

Layer	Output shape	#params	Trainable	Activation
Dense #0	(None,2)	6	N	None
Dense #1	(None,10)	30	Y	None
Dense #2	(None,5)	55	Y	None
Dense #3	(None,1)	6	Y	None

Table 5. $\log_{10} C$ layer details. The first column addresses the type of layer used in the multilayer perceptron. A “dense” layer is a fully connected perceptron (i.e., all the inputs are connected to each neuron). The second column refers to the number of neurons (i.e., outputs) on each layer. The third column shows the number of parameters (weights and biases) yielded by each layer. Finally, the fourth column specifies if those parameters are tuned in the training stage. The last column specifies the activation function of each layer. Dense #0 is responsible to scale the layer inputs. Dense #1 to #3 are regular perceptrons. There are 1205 trainable parameters and 12 non-trainable parameters

Layer	Output shape	#params	Trainable	Activation
Dense #0	(None,2)	6	N	None
Dense #1	(None,40)	120	Y	elu
Dense #2	(None,20)	820	Y	elu
Dense #3	(None,10)	210	Y	elu
Dense #4	(None,5)	55	Y	elu
Dense #5	(None,1)	6	Y	elu

The constructed MLPs take two inputs (stress ratio and corrosivity index) and provide one output each. Unfortunately, optimizing all the deep neural network parameters can be a challenging task. Randomly initialized

*

www.tensorflow.org/api_docs/python/tf/keras/optimizers/RMSprop

weights can be far away from the “optimum” values that would produce an accurate model (or to say the least, make the training process difficult to converge). Therefore, we propose a systematic approach to initialize the weights and biases of this neural network model. We propose constructing a simple linear plane representation of the input-output relationship:

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 \quad (9)$$

where, x_i are normalized inputs, and y is the output.

The coefficients are initialized using engineering judgment. For example, we can assume that C increases with stress ratio and corrosivity index. For illustration purpose, one random plane is plotted against the actual input-output relationship in Figure 8.

The RNN is fitted with observations for inputs throughout the time series and observations for crack length only at the inspection (besides the initial crack length). Note that the output of neither MLPs is observed (ever). Instead, it is the crack length that is used in the loss function:

$$L = \frac{(\mathbf{a}_{OBS} - \mathbf{a}_{PRED})^T (\mathbf{a}_{OBS} - \mathbf{a}_{PRED})}{N_{airplanes}} \quad (10)$$

where $N_{airplanes}$ is the number of airplanes with observed crack length (inspection data).

Table 6 describes the selected training parameters for the MLPs and for the RNN. We use the mean squared error (MSE) as the loss function used for the pretraining (i.e., initialization) of the MLPs and Eq. (10) for the training of the RNN. In all three cases, we used the RMSprop* optimizer. Learning rates and number of epochs vary with each case (the final values were obtained after we repeated the fitting process several times).

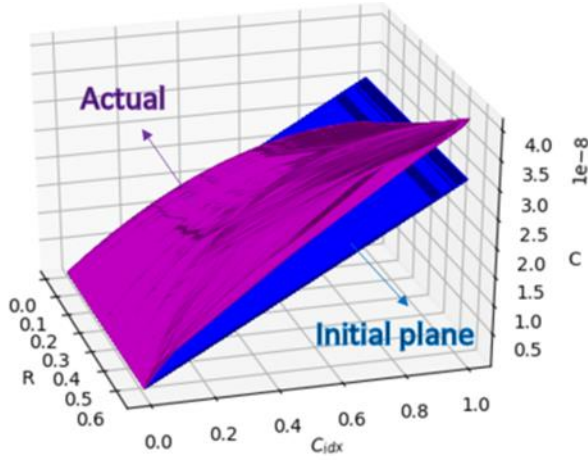
3.4. Replication of results

Our implementation is all done in TensorFlow[†] (version 2.0.0-beta1) using the Python application programming interface. In order to replicate the results presented here, the interested reader can download the codes and data. First, install the PINN python package (base package for physics-informed neural networks used in this work) available at Viana et al (2019). The required datasets are uploaded in a Harvard Dataverse repository available at Dourado and Viana, (2019a). Python scripts demonstrating the proposed framework (with scripts implemented the MLP and RNN training as well as the RNN prediction processes) can be found in Dourado and Viana (2019b).

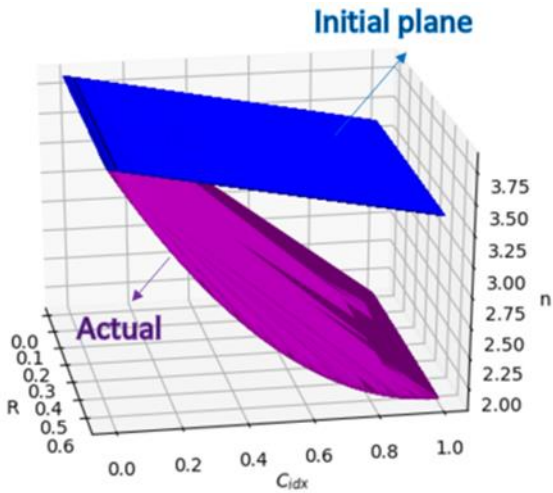
[†] www.tensorflow.org

Table 6. Training parameters description for $\log_{10} C$ MLP layer, n MLP layer and overall RNN.

	Loss function	Optimizer	Learning Rate	Epochs
MLP - $\log_{10} C$	MSE	RMSprop	10^{-3}	50
MLP - n	MSE	RMSprop	10^{-3}	10
RNN	Eq. (10)	RMSprop	10^{-6}	250



(a) $\log_{10} C$ surfaces



(b) m surface.

Figure 8. Initial and actual surfaces for both MLPs.

4. RESULTS AND DISCUSSION

4.1. Recurrent Neural Network (RNN) training.

Using Eq (9) we derived initial guesses for the MLPs described in Table 4 and Table 5, by the planes shown in Figure 8. With this initial guesses, the MLPs yields predictions for $\log_{10} C$ and n as illustrated in Figure 9. As we explained in Section 3.3, this is done to avoid random initialization of the MLP, which we found to be detrimental to the RNN training. It is worth mentioning that Figure 9 predictions are just illustrative since at this point the RNN training has not been performed yet. As previously addressed, this procedure is performed with the sole purpose of generating an initial estimate for the RNN weights and bias values.

Based on this starting condition, the RNN training is performed considering the available crack length data from the synthetic fleet inspection campaign. Different configurations could be considered in the synthetic inspection campaign, but for simplicity we considered that 20% of the fleet (20 aircrafts randomly selected) undergoing inspection after 20.000 flights (approximately at the 7th year of operation). As previously mentioned, we conceived the proposed RNN as having full visibility of its inputs and partial visibility of outputs. Hence, in the training stage the RNN receives information regarding load stresses, stress ratio and corrosion index of every cycle on each flight of the selected 20 aircrafts, and only the initial and final crack lengths. The data regarding the other 80% of the fleet is used for testing the RNN response. These results are discussed in the next section.

Figure 10 illustrates the results of the RNN training considering the available inspection data. As can be seen, the initial planes provided a relatively good initial guess for the RNN training. It can be noticed that after the training the RNN crack length predictions match with the crack length data (actual in Figure 10a). This is indicative of the RNN optimization convergence. Another indicative of convergence is the crack propagation history shown in Figure 10b. For the considered sample, the RNN was able to accurately backtrack the damage accumulation of the aircraft yielding the presented crack histories. However, an important feature can also be noticed in Figure 10. The provided training data has an inherent bias highlighted by the observable clusters seem in Figure 10a. This feature is due to the mission mix distribution in the training data. The considered data were mainly formed by aircrafts operating with a predominant mission. This bias might not accurately represent the fleet actual behavior and could lead to unexpected predictions by the RNN.

4.2. Recurrent Neural Network (RNN) damage diagnosis and prognosis.

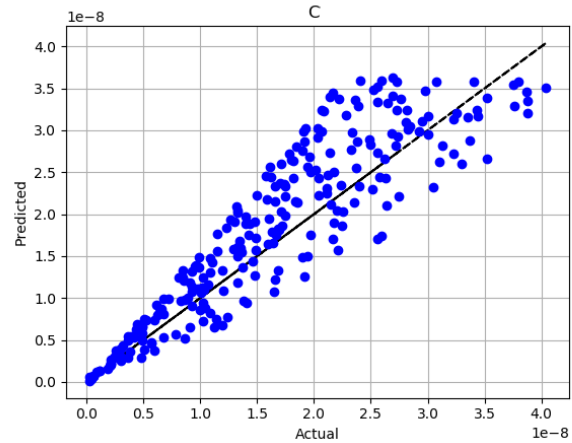
After the RNN is properly trained the resulting model is used to perform the fleet diagnosis and prognosis. For fleet diagnosis, we mean to ascertain if the RNN can accurately act as a bias estimator and properly handle the corrosion effects in the crack length values of the entire fleet. We carried this analysis in the same year that the fleet was inspected (20,000 flights). Since in this contribution we are considering a synthetic fleet, the crack length values of each aircraft can be obtained at any instant. Figure 11 presents the results of the fleet-wide diagnosis analysis.

If the fleet was performing as designed, i.e. if the aircrafts were operating in a non-corrosive environment subjected only to mechanical fatigue, crack lengths in the range of 0.80 and 0.90 mm would be expected (blue squares in Figure 11). Considering the corrosion-fatigue mechanism, the crack length values revolve around 0.95 and 1.10 mm (magenta dots in Figure 11), an increment of approximately 20%. As can be seen in Figure 11, the RNN was able to act as a bias estimator, compensating the lack of corrosion modeling in the damage estimation. It is noticeable that the RNN predictions capture the overall trend of the corrosion-fatigue crack length. Despite being underestimating the crack length value for a few aircrafts, the RNN is still predicting values far from the mechanical fatigue results. This is a clear indication that even in the worst scenarios the RNN is still acting as bias estimator and compensating the corrosion effects.

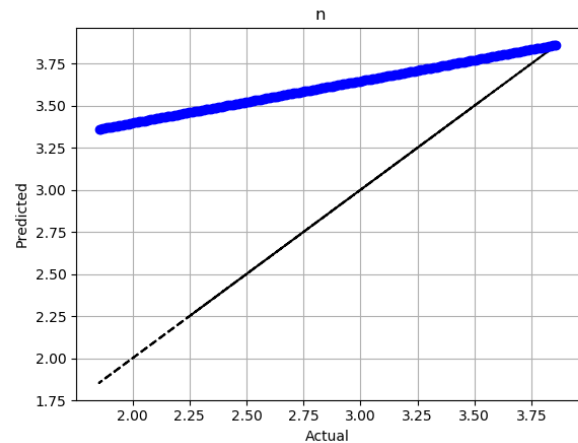
Our goal in this contribution was to evaluate the use of a **hybrid RNN as a bias estimator** compensating the lack of corrosion modeling in a fatigue damage accumulation problem. The obtained results for the diagnosis analysis indicates that the research goal was achieved. The RNN is clearly compensating the lack of corrosion damage information and even in the worst predictions is far from the mechanical fatigue responses.

Another important feature that we evaluated was the RNN ability to accurately forecast damage accumulation. For this end, the proposed PINN has to provide a prognosis of future fleet crack propagation. At first, it might seem a simple task, but it is important to point out the daunting challenge behind this analysis. The PINN model is recursive accumulating damage at each cycle with MLPs compensating the corrosion bias. The crack propagation itself is a non-linear problem, with the crack length growing exponentially with each new cycle. The MLPs are calibrated with a single observation (crack length data after 20,000 flights). At that instant, the exponential growth of the crack is not as significant as in the aircraft final missions (missions closer to the intended MTTF). Basically, as the fleet ages, the crack propagation becomes more aggressive and the corrosion effects have a higher impact. Hence, a small deviation can lead to very far-

off estimates of future crack lengths. A representation of the prognosis task is illustrated in Figure 12.



(a) $\log C$ prediction vs actual.

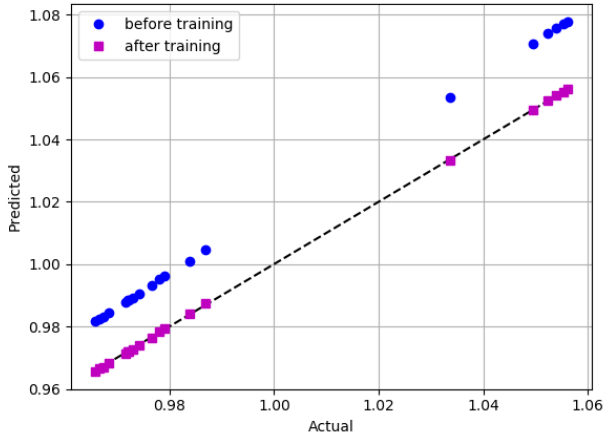


(b) n prediction vs actual.

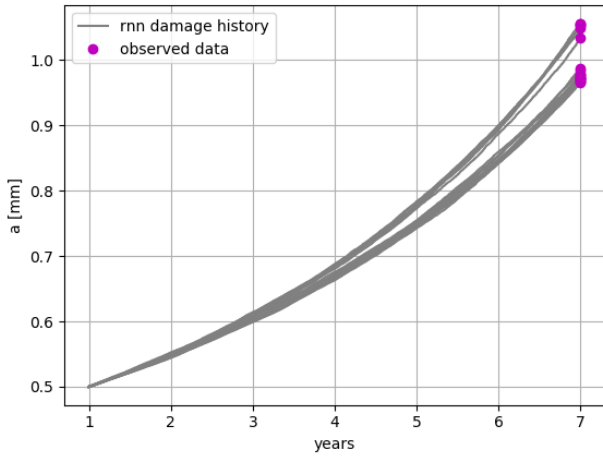
Figure 9. MLP prediction vs actual spread before RNN training, meaning that only MLP were trained with the referred initial planes, but no crack length data was used, (a) $\log C$, (b) n .

Figure 13 presents the results of the fleet-wide prognosis analysis. As in the diagnosis analysis, the proposed PINN model is compensating the corrosion bias in the damage accumulation, accurately capturing the trend of the fleet crack length values after 40,000 flights. It seems that for smaller cracks values (cracks of around 5mm) the RNN was able to forecast the damage accumulation process with high accuracy. For cracks higher than 10 mm the RNN is either under or over predicting the crack final value. However, this is not an alarming issue. For cracks of this range, crack propagation is already in a highly exponential behavior. In these stages, a few cycles can lead to very significant crack growth hindering the RNN ability to determine the crack

actual value. Nonetheless, it is safe to assume that with the proposed PINN model an operation engineer would be able to satisfactorily track the crack propagation in the fleet.



(a) RNN training response considering 20 aircrafts. Actual values are related to the crack length data



(b) RNN training history response given the considered 20 aircrafts.

Figure 10. RNN training responses considering crack length data of 20 aircrafts inspected at 7th year of operation: (a) predicted vs actual response for the considered sample; (b) RNN crack propagation history for the given sample.

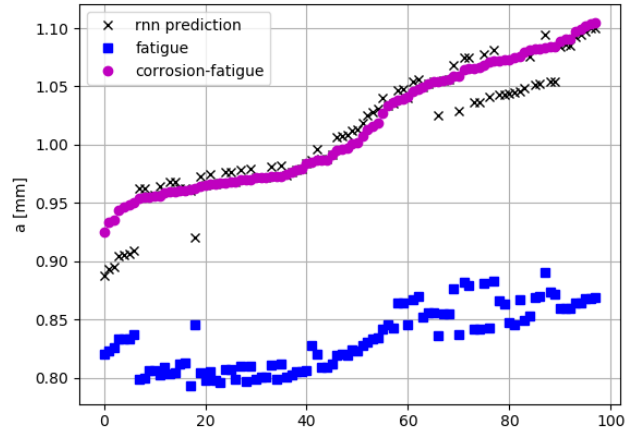


Figure 11. Fleet-wide diagnosis, where the black crosses illustrate the RNN prediction for every aircraft on the fleet; the blue squares illustrates crack length after 20,000 flights if the fleet is only being affected by mechanical fatigue; and the magenta dots shows crack length value of each aircraft of the fleet under the combined effects of corrosion-fatigue.

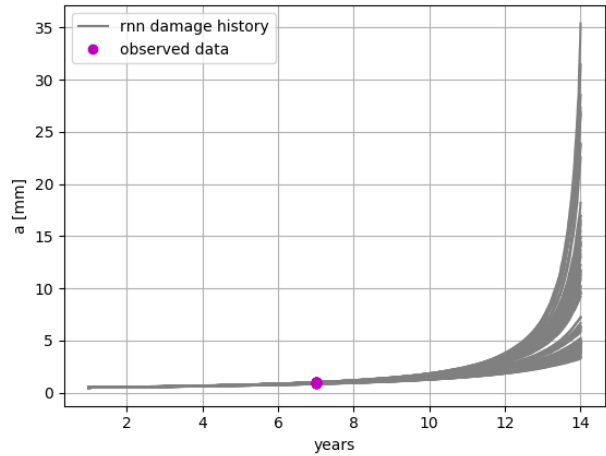


Figure 12. RNN prognosis task, where the RNN as to estimate the fleet crack propagation being calibrated only once with the inspection data after 20,000 flights.

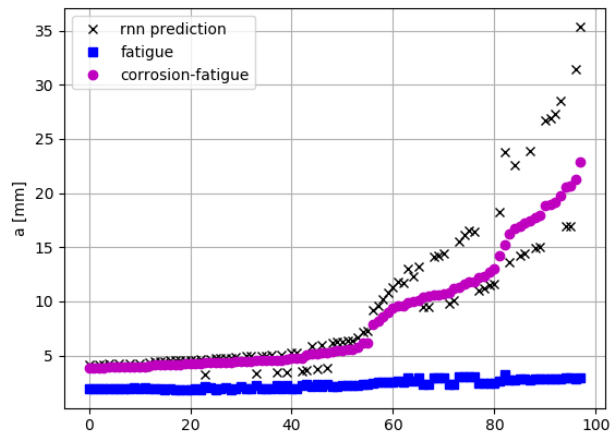


Figure 13. Fleet-wide prognosis. Black crosses illustrate the RNN prediction for every aircraft on the fleet; the blue squares illustrate the expected crack length after 40,000 flights if the fleet is only being affected by mechanical fatigue, and the magenta dots shows the actual crack length value of each aircraft of the fleet under the combined effects of corrosion-fatigue.

5. CONCLUSIONS AND FUTURE WORK

In this contribution, we proposed a novel physics-informed neural network model (illustrated in Figure 4) to be used to compensate for a lack of information in corrosion-fatigue damage accumulation. The proposed framework was derived such as a purely mechanical fatigue model (for instance Paris' law) can be used to assess crack propagation in corrosive environments. Two artificial neural networks are derived and trained to account for the corrosion effects on the fatigue model constants. A simple numerical experiment was designed to evaluate the effectiveness of the proposed methodology, in which:

- A fleet of 100 aircrafts operating between ten different missions and ten distinct airports is considered;
- The different missions introduce load variability while the different airports emulate distinct corrosive environments;
- A physics-informed recurrent neural network is formulated with two multi-layer perceptrons (MLP) modeling the corrosion effects on the fatigue model exponents;
- The MLPs act as bias estimators compensating the corrosion effect on the fatigue model estimated damage.

The presented results demonstrate that:

- The proposed physics-informed recursive neural network can accurately model corrosion-fatigue damage accumulation;
- The proposed model can successfully perform a fleet-wide diagnosis;
- As defined, the proposed framework can satisfactorily forecast damage accumulation and be used for fleet-wide prognosis.

The results obtained were promising and future work will focus on:

- Improve physics of failure by including pit growth in the cumulative damage;
- Ascertain the effects of different training sets in the model performance;
- Evaluate the effects of material properties and other sources of uncertainty in the model prediction;
- Evaluate possible risk mitigation procedures such as fleet recommissioning and assets repair, replacement.

ACKNOWLEDGMENT

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

REFERENCES

- Chauhan, S., & Vig, L. (2015). Anomaly detection in ECG time signals via deep long short-term memory networks. *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, (1-7).
- Chen, G. S., Gao, M., Harlow, D. G., & Wei, R. P., (1994). *Corrosion and corrosion fatigue of airframe aluminum alloys*. Lehigh University, Bethlehem, PA (N95-14465).
- Chen, G. S., Wan, K. C., Gao, M., Wei, R. P., & Flournoy, T.H., (1996). Transition from pitting to fatigue crack growth – modeling of corrosion fatigue crack nucleation in a 2024-T3 aluminum alloy. *Material Science Engineering*. pp. 126-132.
- Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, vol. 5(2), pp. 240-254.
- De Jonge, J. B., Schutz, D., Lowak, H., & Schijve J., (1973) *A standardized load sequence for flight simulation tests on transport aircraft wing structure*, report from National Aerospace laboratory, Amsterdam (NLR TR 73029U).

- Dourado, A. & Viana, F. A. C. (2019a), Replication data for PINN corrosion-fatigue: PHM 2019. <https://doi.org/10.7910/DVN/PZ7DRJ>.
- Dourado, A. & Viana, F. A. C., (2019b). Python Scripts for Physics-Informed Neural Networks for Corrosion-Fatigue Prognosis, v0.0.1, Zenodo, https://github.com/PML-UCF/pinn_corrosion_fatigue [retrieved on Aug/2019], doi:10.5281/zenodo.3355729.
- Dowling, N. E., (2012). *Mechanical behavior of materials: engineering methods for deformation, fracture and fatigue*. Pearson.
- DuQuesnay, D. L., Underhill, P. R., & Britt, H. J., (2003). Fatigue crack growth from corrosion damage in 7075-t6511 aluminium alloy under aircraft loading. *International Journal of Fatigue*. vol. 25(5), pp.371–377.
- Fatemi, A., & Yang, L., (1998). Cumulative fatigue damage and life prediction theories: a survey of the state of the art for homogeneous materials. *International Journal of Fatigue*, vol. 20(1), pp. 9-34.
- Frangopol, D. M., Kallen, M.-J., & Van Noortwijk, J. M., (2004). Probabilistic models for life-cycle performance of deteriorating structures: review and future directions. *Progress in Structural Engineering and Materials*, vol. 6(4), pp. 197-212.
- Goodfellow, I., Bengio, Y., & Courville, A., (2016). *Deep Learning*, MIT press.
- Goswami, T. K., & Hoepfner, D. W., (1995). Pitting corrosion fatigue of structural materials. *Structural integrity in aging aircraft*. New York: ASME p.47.
- Khan, S., & Yairi, T., (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, vol. 107, pp. 241-265.
- Kondo, Y., (1989). Prediction of fatigue crack initiation life based on pit growth. *Corrosion*, vol. 45(1), pp. 7-11.
- Li, X., Ding, Q., & Sun, J.-Q., (2018) Remaining useful life estimation in prognosis using deep convolution neural networks, *Reliability Engineering and System Safety*, vol. 172, pp. 1-11.
- Lindley, T. C., McIntyre, P., & Trant, P.J, (1982). Fatigue crack initiation at corrosion pits. *Material Technology*, vol. 9, pp.135–142.
- Lishchuk, S., & Akid, R., (2009). A review of corrosion fatigue modelling methodology. Report from Sheffield Hallam University.
- Menan, F., & Henaft, G., (2010). Synergistic action of fatigue and corrosion during crack growth in the 2024 aluminum alloy. *Procedia Engineering*, vol. 2, pp. 1441-1450.
- Larossa, N.O., Akid, R., & Ainsworth, R.A., (2018). Corrosion-fatigue: a review of damage tolerance models. *International Materials Reviews*, vol. 63(5), pp. 283-308.
- Nascimento, R. G., & Viana, F. A. C., (2019). Fleet prognosis with physics-informed recurrent neural networks. arxiv.org/abs/1901.05512
- Paris, P., & Erdogan, F., (1963). A critical analysis of crack propagation laws. *Journal of Basic Engineering*, vol. 85(4), pp. 528-533.
- Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Fifteenth annual conference of the international speech communication association*.
- Shi, P., & Mahadevan, S., (2001). Damage tolerance approach for probabilistic pitting corrosion fatigue life prediction. *Engineering Fracture Mechanics*, vol. 68, pp. 1493-1507.
- Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A., (2015). Machine learning for predictive maintenance: a multiple classifier approach. *IEEE Transactions on Industrial Informatics*, vol. 11(3), pp. 812-820.
- Trischler, A. P., & D’Eleuterio, G. (2016). Synthesis of recurrent neural networks for dynamical system simulation. *Neural Networks*, vol. 80, pp. 67-78.
- Viana, F. A. C., Nascimento, R. G., Yucesan, Y. & Dourado, A. Physics-informed neural networks package, v0.0.2, Zenodo, <https://github.com/PML-UCF/pinn>, doi:10.5281/zenodo.3356877.
- Wang, Q. Y., Pidaparti, R. M., & Palakal, M.J., (2001). Comparative study of corrosion-fatigue in aircraft materials. *AIAA Journal*, vol. 39(2), pp.325–330.
- Yu, Y., Yao, H., & Liu, Y. (2018). Physics-based Learning for Aircraft Dynamics Simulation. *Proceedings of the Annual Conference of the PHM Society*, vol. 10 (1), September 24-27, Philadelphia, USA. <https://doi.org/10.1234/phmconf.2018.v10i1.513>.
- Zhao, Z., Liang, B., Wang, X., & Lu, W., (2017) Remaining useful life prediction of aircraft engine based on degradation pattern learning, *Reliability Engineering and System Safety*, vol. 164, pp. 74-83.

BIOGRAPHIES

Arinan Dourado is a graduate research assistant at UCF with Bachelor and Master’s degrees in chemical engineering. He also has a PhD degree in mechanical engineering from the Federal University of Uberlandia. He is currently working towards a second PhD at UCF. His research topics involve uncertainty quantification and propagation, and physics-informed neural networks.

Felipe A. C. Viana is an assistant professor at the University of Central Florida. The vast majority of Dr. Viana’s work has been applied to new designs and improvement of fielded products with a focus on aircraft propulsion, power generation, and oil and gas systems. Before joining UCF, Dr. Viana was a Sr. Scientist at GE Renewable Energy, where he led the development of state-of-the-art computational methods for improving wind energy asset performance and reliability. Prior to moving to that role at GE, he spent five years at GE Global Research, where he leads and conducted research on design and optimization under uncertainty,

probabilistic analysis of engineering systems, and services
engineering.