Mechanical and
Aerospace Engineering
UNIVERSITY OF CENTRAL FLORIDA

Prognosis and Health Management with Digital Twins and Hybrid Physics-Informed Neural Networks

Felipe A. C. Viana, PhD
PML Principal Investigator
Assistant Professor

E-mail: viana@ucf.edu

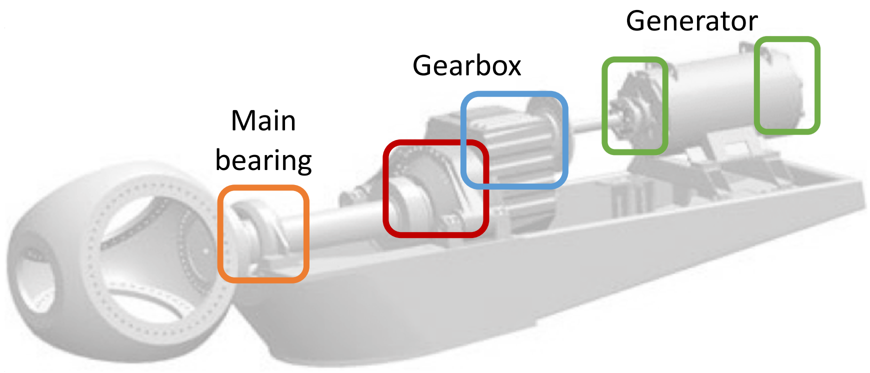
Probabilistic Mechanics Laboratory
pml-ucf.github.io

Mechanical and Aerospace Engineering
University of Central Florida

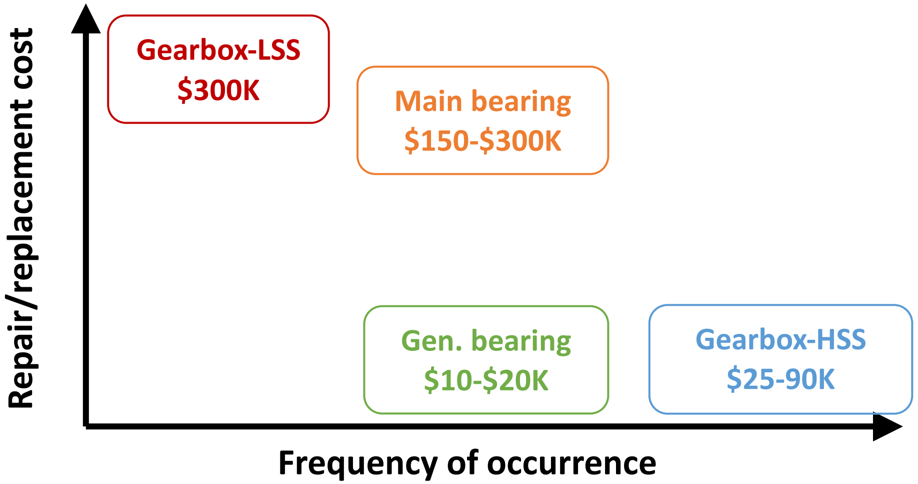
Prognosis and digital twins

Onshore wind energy example

(a) Drivetrain components



(b) Drivetrain components



Sethuraman, L., Guo, Y., & Sheng, S. (2015). Main bearing dynamics in three-point suspension drivetrains for wind turbines. American Wind Energy Association Conference & Exhibition, May 18–21, Orlando, FL.

Problem → challenge → solution → benefits

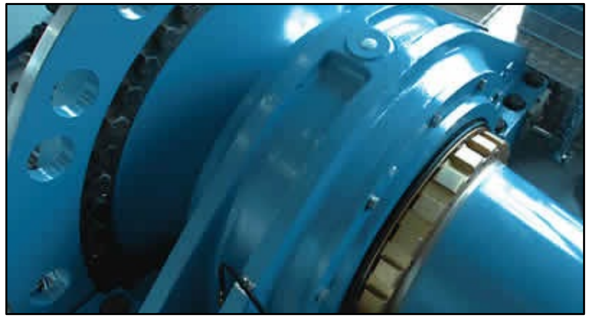
Major problem
- Maintenance and operation costs.

Challenges:
- Physics not fully understood
- Data is highly unstructured

Proposed solution
- Hybrid physics-informed neural networks

Benefit
- Predictive maintenance = reduced costs

Pristine



Unquantified internal damage (recorded flange wear)

Rolling element damage

Unquantified damage

Outline

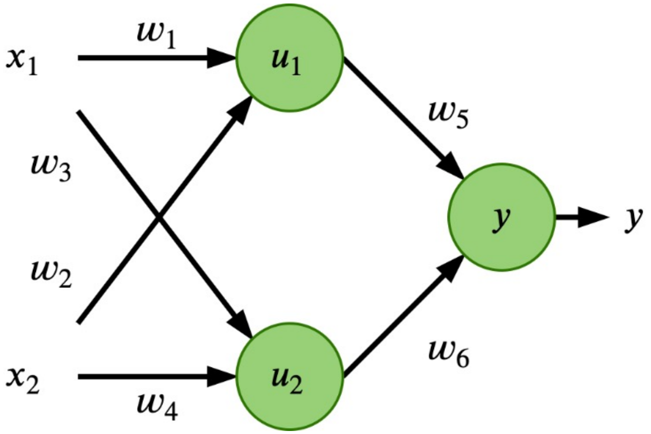
- Physics-informed neural networks?
- Hybrid models and predictive maintenance
- Application examples
- Summary and conclusions

Background: neural networks and backpropagation

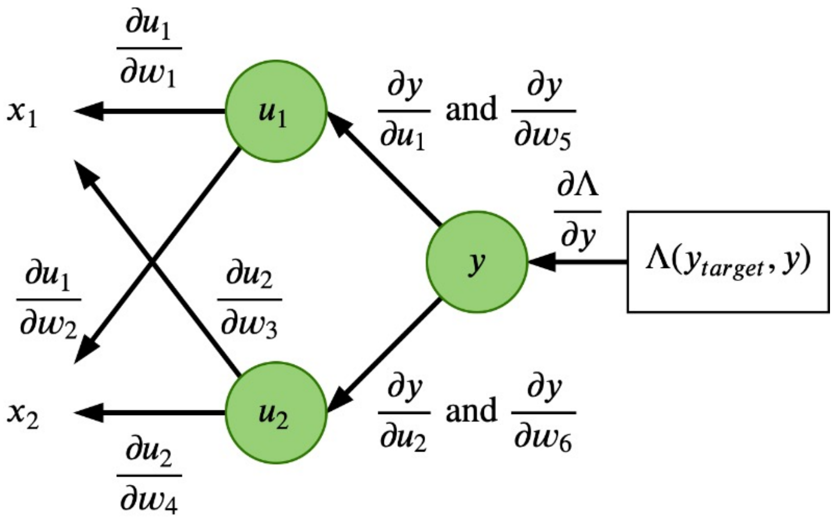
Main concern: Large number of parameters to be trained (depth of the neural networks)

Solution: Backpropagation of the gradients

(a) Forward pass



(b) Backward pass



Physics-informed neural networks (30,000 ft view)

Computational mechanics

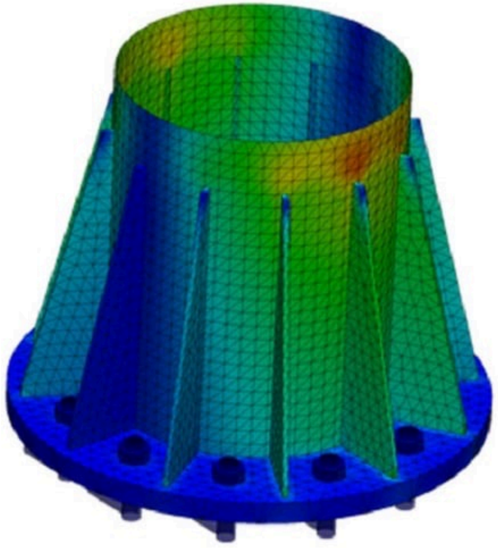
Elasticity:

$$\nabla . \boldsymbol{\sigma} + \boldsymbol{F} = \rho \ddot{\boldsymbol{u}}$$

$$\epsilon = \frac{1}{2}[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T]$$

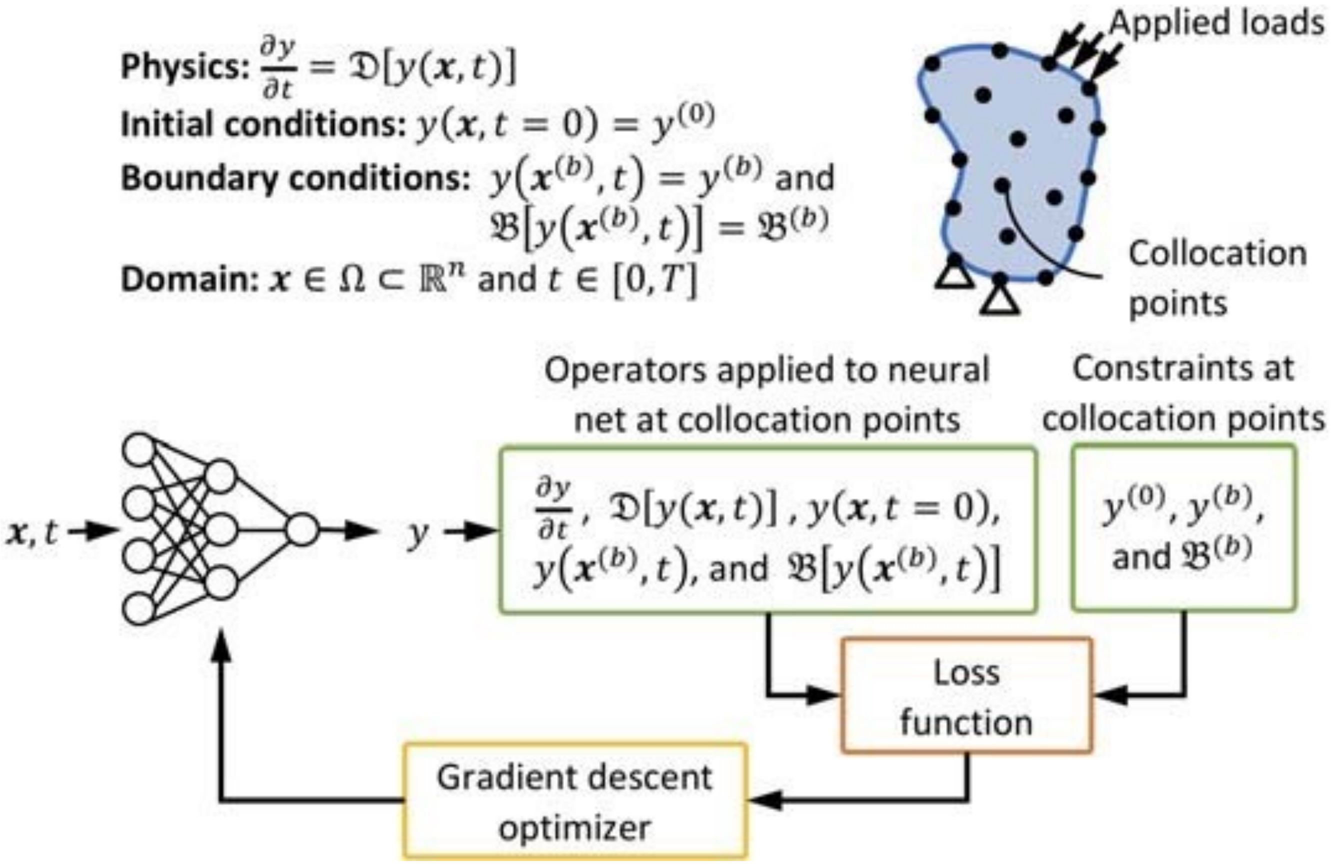
$$\boldsymbol{\sigma} = \boldsymbol{C} : \epsilon$$

Finite element modeling



https://developer.nvidia.com/simnet

Physics-informed neural networks



Literature is very abundant!

JOURNAL OF COMPUTATIONAL PHYSICS **91**, 110–131 (1990)

Neural Algorithm for Solving Differential Equations

HYUK LEE

Department of Electrical Engineering, Polytechnic Institute of New York, Brooklyn, New York 11201

AND

IN SEOK KANG

Department of Chemical Engineering, California Institute of Technology, Pasadena, California 91125

Received August 17, 1988; revised October 6, 1989

Finite difference equations are considered to solve differential equations numerically by utilizing minimization algorithms. Neural minimization algorithms for solving the finite difference equations are presented. Results of numerical simulation are described to demonstrate the method. Methods of implementing the algorithms are discussed. General features of the neural algorithms are discussed.

IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 16, NO. 6, NOVEMBER 2005 1381

Finite-Element Neural Networks for Solving Differential Equations

Pradeep Ramuhalli, *Member, IEEE*, Lalita Udpa, *Senior Member, IEEE*, and Satish S. Udpa, *Fellow, IEEE*

***Abstract*—The solution of partial differential equations (PDE) arises in a wide variety of engineering problems. Solutions to most practical problems use numerical analysis techniques such as finite-element or finite-difference methods. The drawbacks of these approaches include computational costs associated with the modeling of complex geometries. This paper proposes a finite-element neural network (FENN) obtained by embedding a finite-element model in a neural network architecture that enables fast and accurate solution of the forward problem. Results of applying the FENN to several simple electromagnetic forward and inverse problems are presented. Initial results indicate that the FENN performance as a forward model is comparable to that of the conventional finite-element method (FEM). The FENN can also be used in an iterative approach to solve inverse problems associated with the PDE. Results showing the ability of the FENN to solve the inverse problem given the measured signal are also presented. The parallel nature of the FENN also makes it an attractive solution for parallel implementation in hardware and software.**

***Index Terms*—Finite-element method (FEM), finite-element neural network (FENN), inverse problems.**

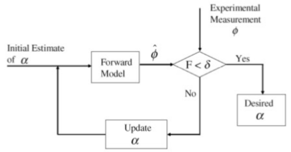


Fig. 1. Iterative inversion method for solving inverse problems.

resulting in the corresponding solution to the forward problem $(\hat{\phi})$. The model output is compared to the measurement (ϕ), using a cost function $F(\phi, \hat{\phi})$. If $F(\phi, \hat{\phi})$ is less than a tolerance δ, the estimate α is used as the desired solution. If not, α is updated to minimize the cost function.

SCIENCE ADVANCES | RESEARCH ARTICLE

APPLIED MATHEMATICS

Data-driven discovery of partial differential equations

Samuel H. Rudy,[1]* Steven L. Brunton,[2] Joshua L. Proctor,[3] J. Nathan Kutz[1]

We propose a sparse regression method capable of discovering the governing partial differential equation(s) of a given system by time series measurements in the spatial domain. The regression framework relies on sparsity-promoting techniques to select the nonlinear and partial derivative terms of the governing equations that most accurately represent the data, bypassing a combinatorially large search through all possible candidate models. The method balances model complexity and regression accuracy by selecting a parsimonious model via Pareto analysis. Time series measurements can be made in an Eulerian framework, where the sensors are fixed spatially, or in a Lagrangian framework, where the sensors move with the dynamics. The method is computationally efficient, robust, and demonstrated to work on a variety of canonical problems spanning a number of scientific domains including Navier-Stokes, the quantum harmonic oscillator, and the diffusion equation. Moreover, the method is capable of disambiguating between potentially nonunique dynamical terms by using multiple time series taken with different initial data. Thus, for a traveling wave, the method can distinguish between a linear wave equation and the Korteweg–de Vries equation, for instance. The method provides a promising new technique for discovering governing equations and physical laws in parameterized spatiotemporal systems, where first-principles derivations are intractable.

Neural Ordinary Differential Equations

Ricky T. Q. Chen*, Yulia Rubanova*, Jesse Bettencourt*, David Duvenaud
University of Toronto, Vector Institute
{rtqichen, rubanova, jessebett, duvenaud}@cs.toronto.edu

Abstract

We introduce a new family of deep neural network models. Instead of specifying a discrete sequence of hidden layers, we parameterize the derivative of the hidden state using a neural network. The output of the network is computed using a black-box differential equation solver. These continuous-depth models have constant memory cost, adapt their evaluation strategy to each input, and can explicitly trade numerical precision for speed. We demonstrate these properties in continuous-depth residual networks and continuous-time latent variable models. We also construct continuous normalizing flows, a generative model that can train by maximum likelihood, without partitioning or ordering the data dimensions. For training, we show how to scalably backpropagate through any ODE solver, without access to its internal operations. This allows end-to-end training of ODEs within larger models.

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

ELSEVIER

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations

M. Raissi [a], P. Perdikaris [b,*], G.E. Karniadakis [a]

[a] *Division of Applied Mathematics, Brown University, Providence, RI, 02912, USA*
[b] *Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA, 19104, USA*

ARTICLE INFO

Article history:
Received 13 June 2018
Received in revised form 26 October 2018
Accepted 28 October 2018
Available online 3 November 2018

Keywords:
Data-driven scientific computing
Machine learning
Predictive modeling
Runge–Kutta methods
Nonlinear dynamics

ABSTRACT

We introduce *physics-informed neural networks* – neural networks that are trained to solve supervised learning tasks while respecting any given laws of physics described by general nonlinear partial differential equations. In this work, we present our developments in the context of solving two main classes of problems: data-driven solution and data-driven discovery of partial differential equations. Depending on the nature and arrangement of the available data, we devise two distinct types of algorithms, namely continuous time and discrete time models. The first type of models forms a new family of *data-efficient* spatio-temporal function approximators, while the latter type allows the use of arbitrarily accurate implicit Runge–Kutta time stepping schemes with unlimited number of stages. The effectiveness of the proposed framework is demonstrated through a collection of classical problems in fluids, quantum mechanics, reaction–diffusion systems, and the propagation of nonlinear shallow-water waves.

Archives of Computational Methods in Engineering
https://doi.org/10.1007/s11831-021-09539-0

SURVEY ARTICLE

A Survey of Bayesian Calibration and Physics-informed Neural Networks in Scientific Modeling

Felipe A. C. Viana[1] · Arun K. Subramaniyan[2]

Received: 28 September 2020 / Accepted: 7 January 2021

Abstract

Computer simulations are used to model of complex physical systems. Often, these models represent the solutions (or at least approximations) to partial differential equations that are obtained through costly numerical integration. This paper presents a survey of two important statistical/machine learning approaches that have shaped the field of scientific modeling. Firstly we survey the developments on Bayesian calibration of computer models since the seminal work by Kennedy and O'Hagan. In their paper, the authors proposed an elegant way to use the Gaussian processes to extend calibration beyond parameter and observation uncertainty and include model-form and data size uncertainty. Secondly, we also survey physics-informed neural networks, a topic that has been receiving growing attention due to the potential reduction in computational cost and modeling flexibility. In addition, in order to help the interested reader to familiarize with these topics and venture into custom implementations, we present a summary of applications and software tools. Finally, we close the paper with suggestion for future research directions and a thought provoking call for action.

How about physics-informed neural networks for digital twins?

Hybrid models can reduce prediction error

Fatigue crack growth

$$\frac{da}{dN} = C\Delta K^m$$

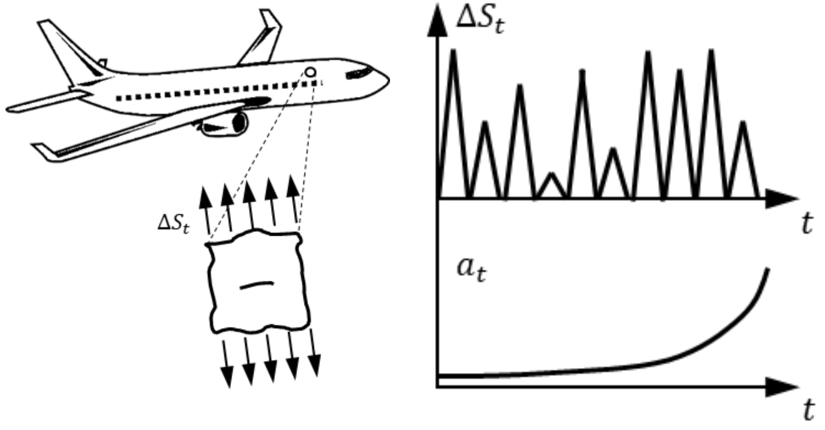
where:

- N: number of cycles
- C and m: material properties (coupon tests)
- $\Delta K = F\Delta S\sqrt{\pi a}$
- ΔS: cyclic stresses (e.g., from finite element models)

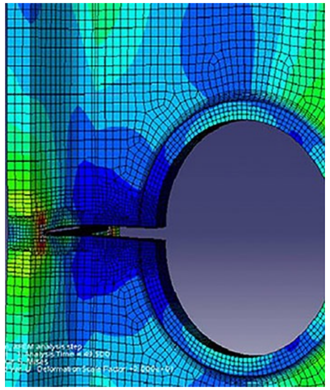
What if ΔK or ΔS are not accurate?

We will show how hybrid models can account for missing physics

(a) Fatigue crack growth at fuselage panel

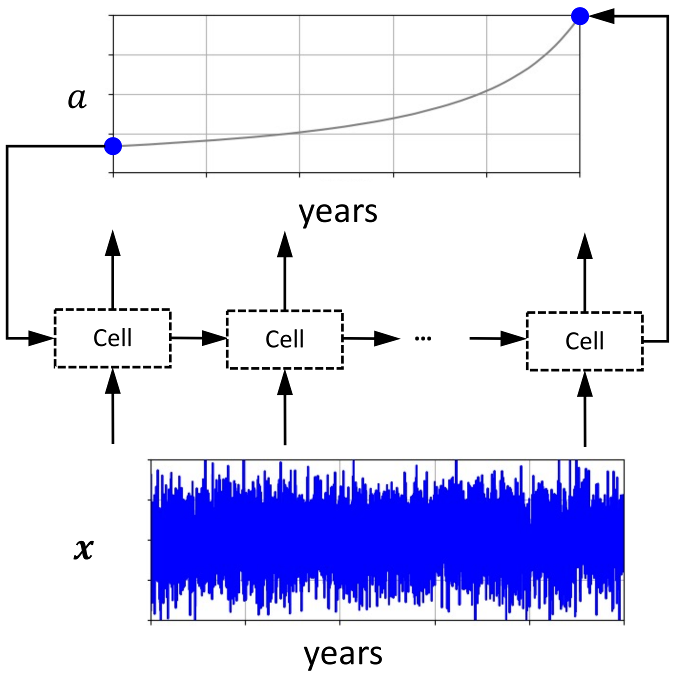


(b) Finite element modeling

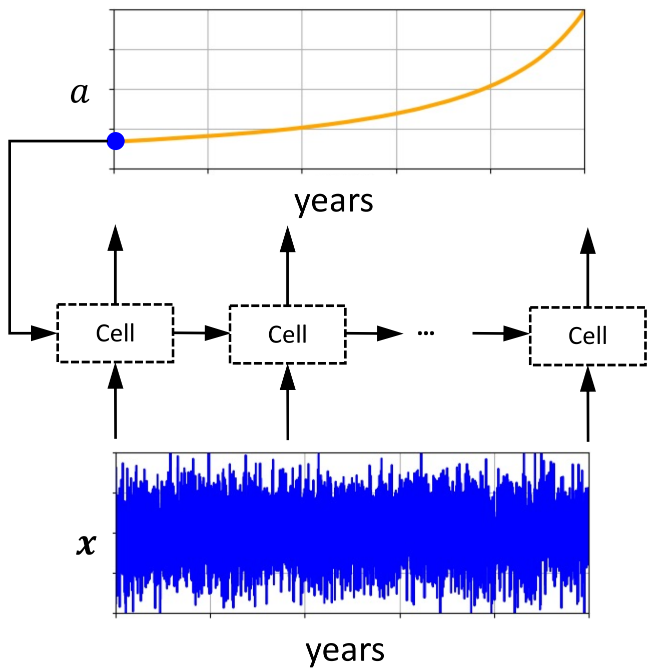


In prognosis, data is very unbalanced

(a) Typical training



(b) Typical prediction

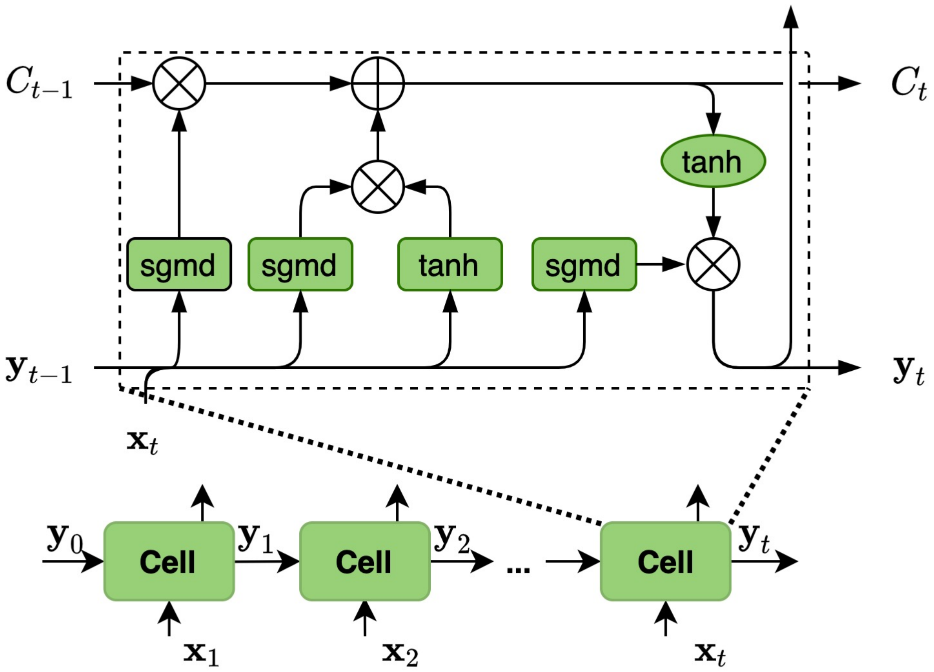


Very hard (impossible) without physics

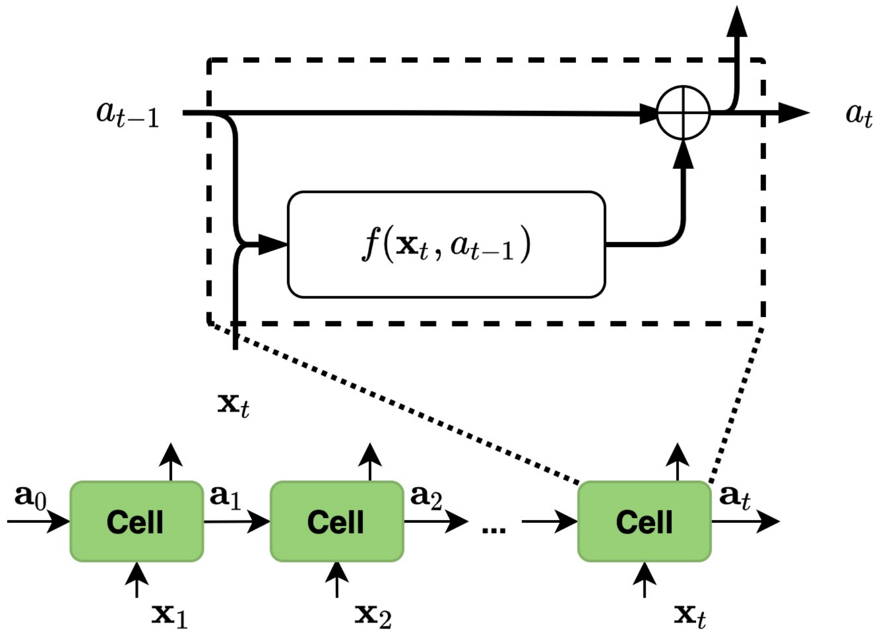
Blue: observed data
Gray: desired output (never fully observed)
Orange: Recurrent neural network prediction

Hybrid physics-informed neural networks?

(a) Long short-term memory (LSTM) cell



(b) Euler integrator cell (cumulative damage)



$$\frac{da}{dN} = C\Delta K^m \quad \Longrightarrow \quad f(\boldsymbol{x}_t, a_{t-1}) = C\Delta K^m$$

F. A. C. Viana, R. G. Nascimento, A. Dourado, and Y. A. Yucesan, "Estimating model inadequacy in ordinary differential equations with physics-informed neural networks," Computers and Structures, Vol. 245, pp. 106458, 2021.

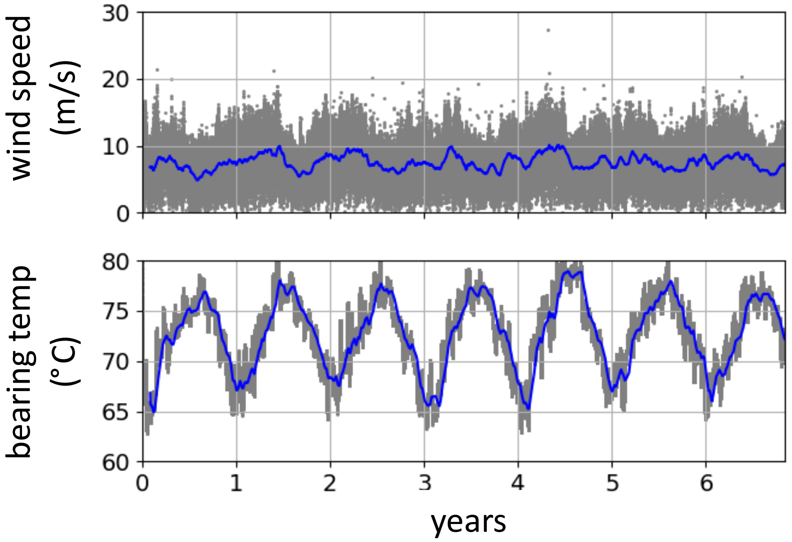
```python
from tensorflow.keras.layers import RNN, Dense, Layer
```

```python
class EulerIntegratorCell(Layer):
    def __init__(self, C, m, dKlayer, a0=None, units=1, **kwargs):
        super(EulerIntegratorCell, self).__init__(**kwargs)
        self.units = units
        self.C      = C
        self.m      = m
        self.a0     = a0
        self.dKlayer      = dKlayer

     def call(self, inputs, states):
         inputs  = convert_to_tensor(inputs)
         a_tm1   = convert_to_tensor(states)
         x_d_tm1 = concat((inputs, a_tm1[0, :]), axis=1)
         dk_t    = self.dKlayer(x_d_tm1)
         da_t    = self.C * (dk_t ** self.m)
         a       = da_t + a_tm1[0, :]
         return a, [a]

def create_model(C, m, a0, dKlayer, batch_input_shape, return_sequences=False, return_state=False):
    euler = EulerIntegratorCell(C=C, m=m, dKlayer=dKlayer, a0=a0, batch_input_shape=batch_input_shape)
    PINN  = RNN(cell=euler, batch_input_shape=batch_input_shape, return_sequences=return_sequences, return_state=return_state)
    model = Sequential()
    model.add(PINN)
    model.compile(loss='mse', optimizer=RMSprop(1e-2))
    return model
```

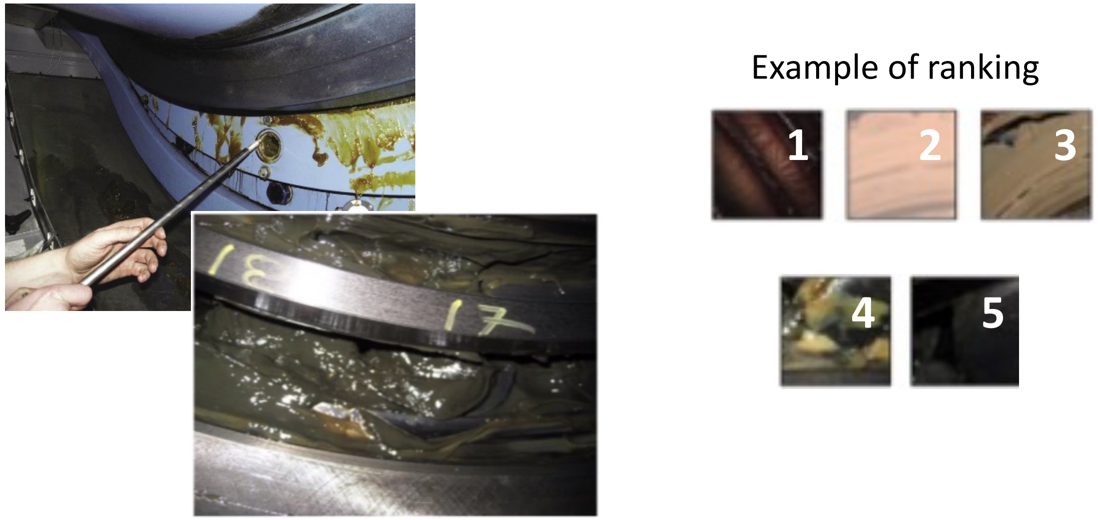
R. G. Nascimento, K. Fricke, and F. A. C. Viana, “A tutorial on solving ordinary differential equations using Python and hybrid physics-informed neural network,” Engineering Applications of Artificial Intelligence, Vol. 96, 2020, 103996.

Case study #1: wind turbine main bearing fatigue

(a) Input data



(b) Visual grease inspection ranking



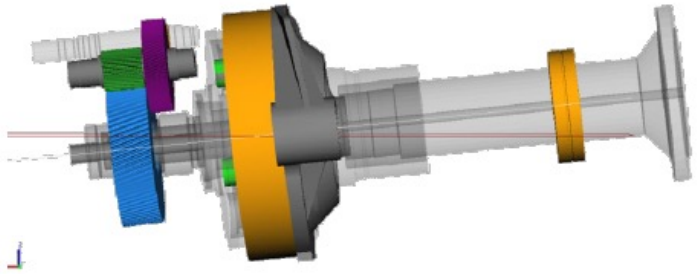
Y. A. Yucesan and F. A. C. Viana, "Hybrid physics-informed neural networks for main bearing fatigue prognosis with visual grease inspection," Computers in Industry, Vol. 125, pp. 103386, 2021.

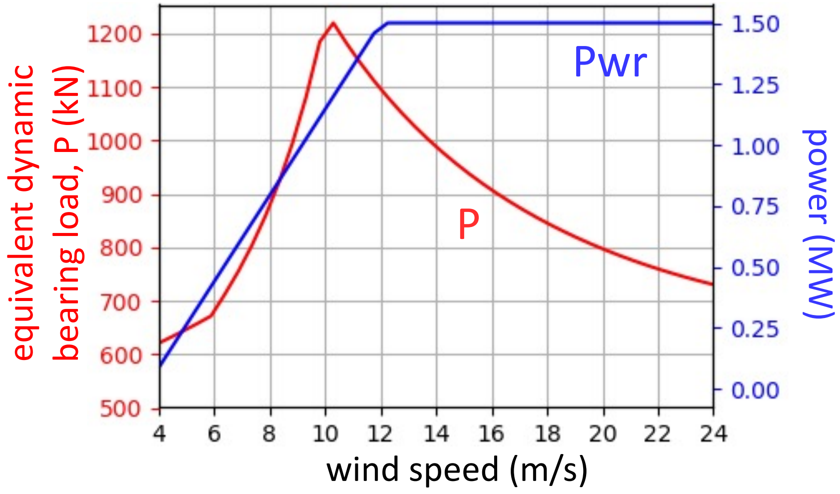
1.5MW / 80m hub-height turbine

Large installed basis: very popular between 2005 and 2010.

Available simulations and data (National Renewable Energy Laboratory)

Multi-body physics model





Sethuraman, L., Guo, Y., & Sheng, S., "Main bearing dynamics in three-point suspension drivetrains for wind turbines," American Wind Energy Association Conference & Exhibition, Orlando, USA, May 18–21, 2015.

Physics-based cumulative damage model

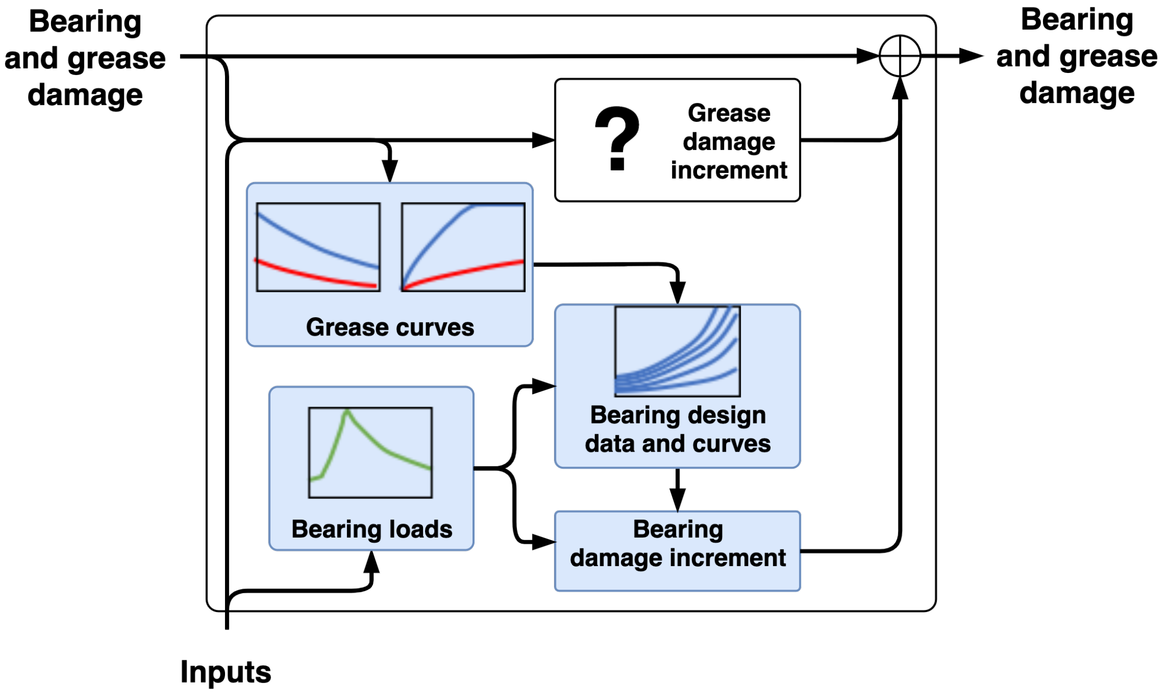
Bearing life is calculated using formula (ISO 281):

$$L = \alpha (C/P)^{10/3}$$

- α grease related life adjustment factor,
- C/P dynamic load ratio.

Palmgren-Miner's rule (different load levels):

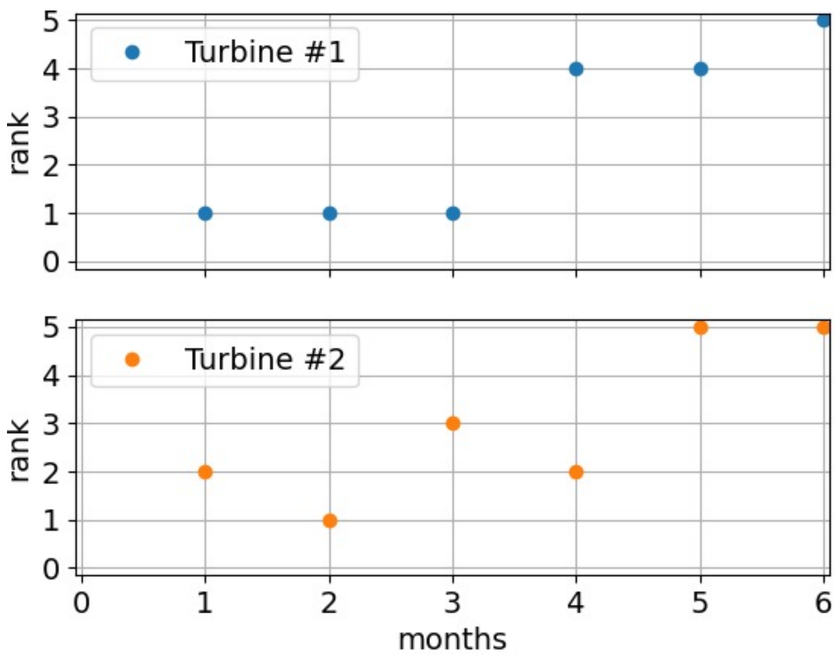
$$\Delta\,\text{damage} = \frac{1}{L}$$



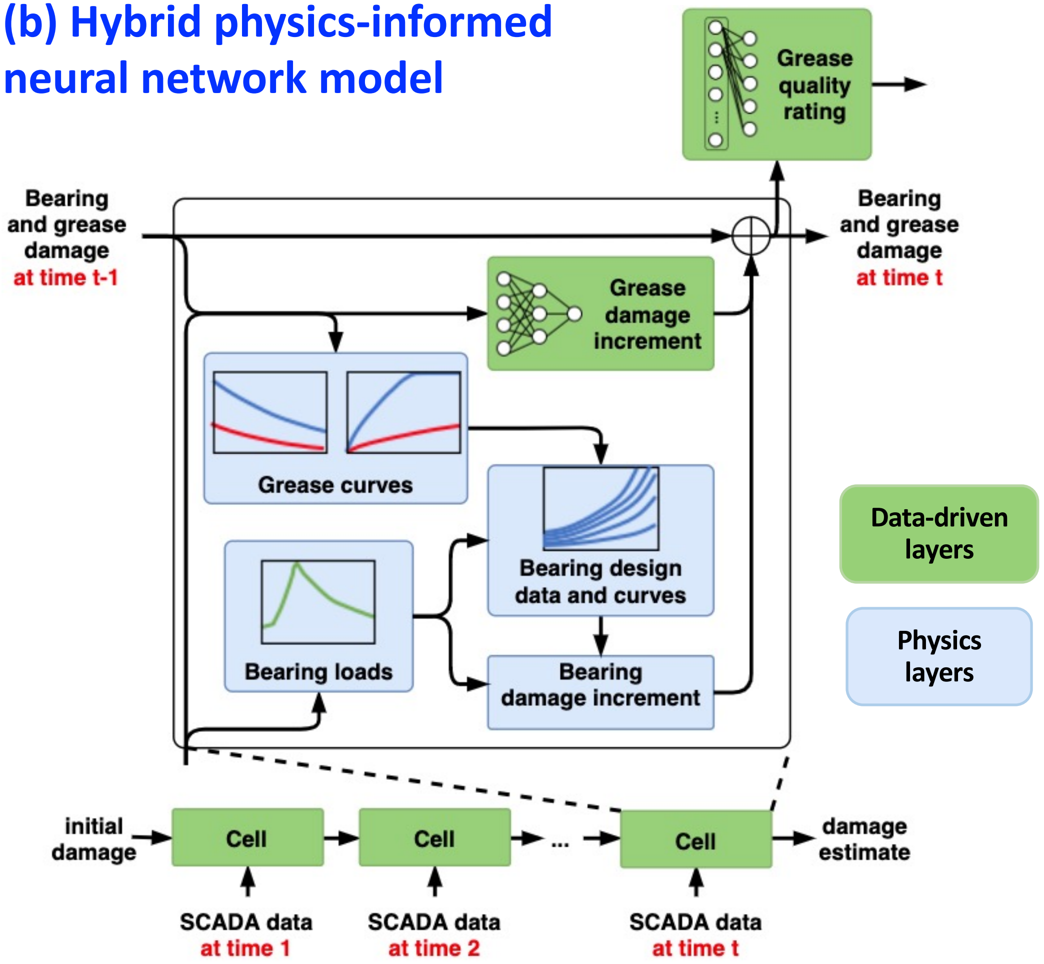
Y. A. Yucesan and F. A. C. Viana, "A physics-informed neural network for wind turbine main bearing fatigue," International Journal of Prognostics and Health Management, Vol. 11 (1), 2020.

Grease inspection and hybrid model

(a) Example of grease visual inspection data



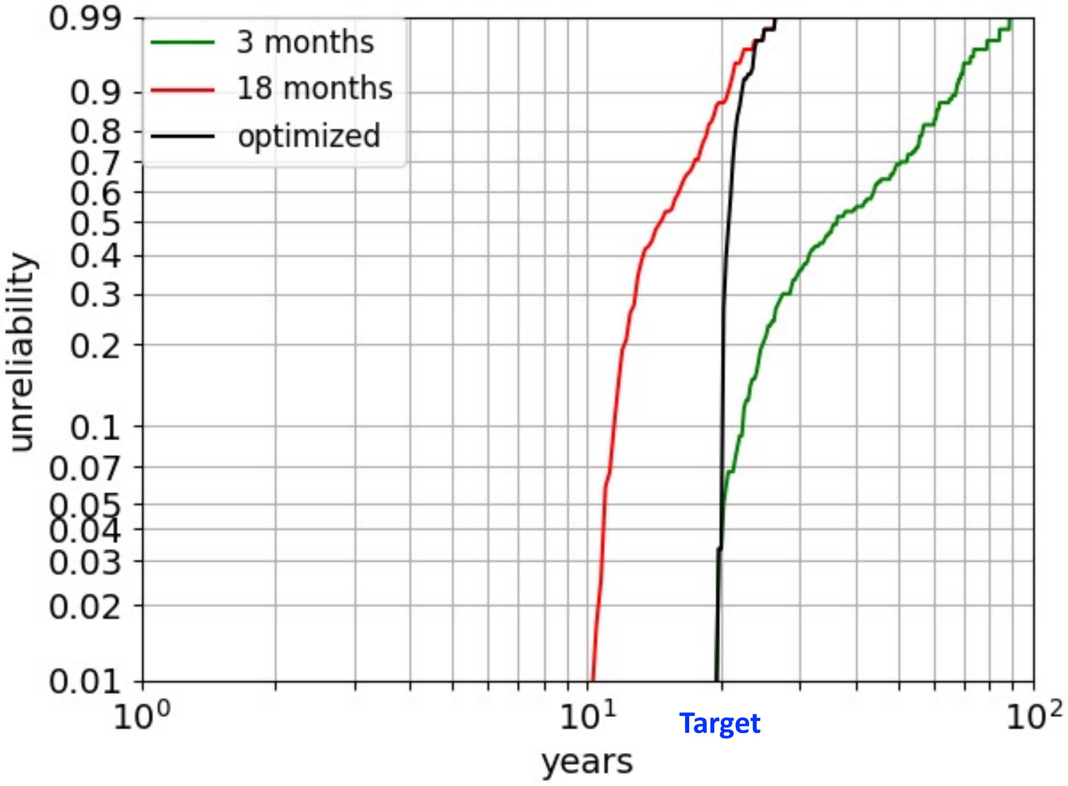
(b) Hybrid physics-informed neural network model



Turbine-level service optimization

Hybrid model can be used for turbine-specific regreasing optimization.

Benefits are cumulated throughout a park → reduced maintenance costs.



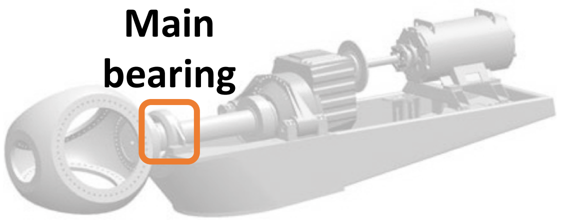
Y. A. Yucesan and F. A. C. Viana, "Hybrid physics-informed neural networks for main bearing fatigue prognosis with visual grease inspection," Computers in Industry, Vol. 125, pp. 103386, 2021.

Deploying it at scale

Main bearing fatigue example:
- Training (10 turbines/6 months)
  - Input data: ~260K points
  - Labeled data: 60 points
  - Small GPU cluster: overnight

- Inference (120 turbines/30 years)
  - Input data: ~190M points
  - Small GPU cluster: few minutes

At scale
- 10s to 100s digital twin models per turbine
- 100s to 1000s turbines
- Decent GPU cluster
  - Training: weeks
  - Inference: hours
  - Optimization: days



Lithium-ion battery aging modeling

Challenges:

- Prognosis models depend on several empirically adjusted factors
- Hard to account for aging

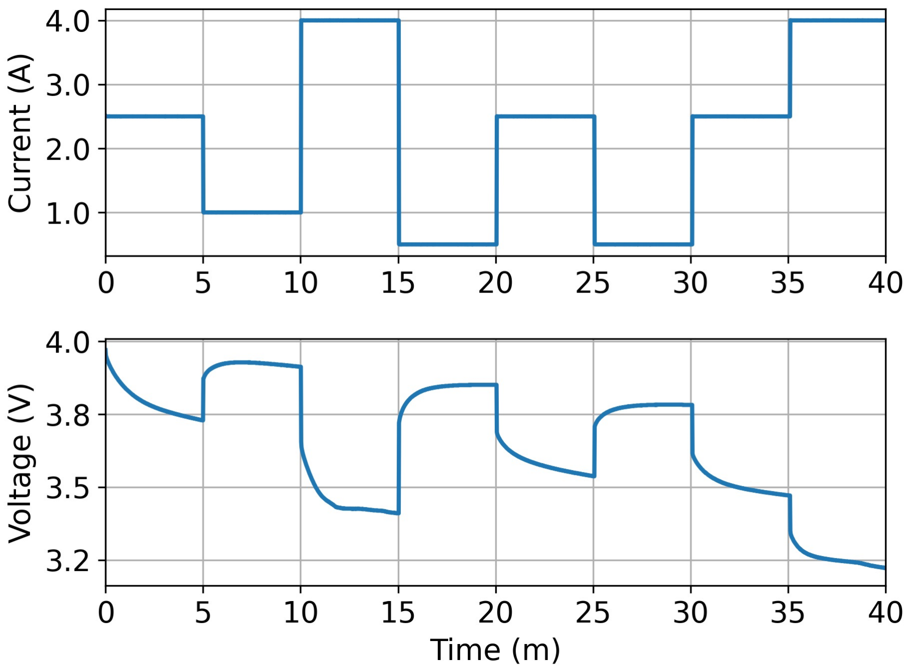
Collaboration with

- Diagnosis and Prognosis Group @ NASA Ames Research Center

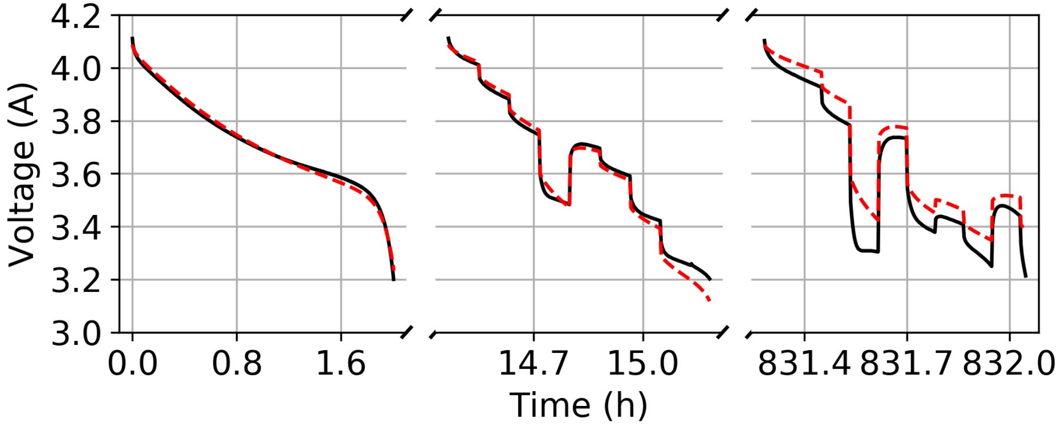
R. G. Nascimento et al., “Hybrid Physics-Informed Neural Networks for Lithium-Ion Battery Modeling and Prognosis,” to be published.

Typical duty cycle and drifting of models

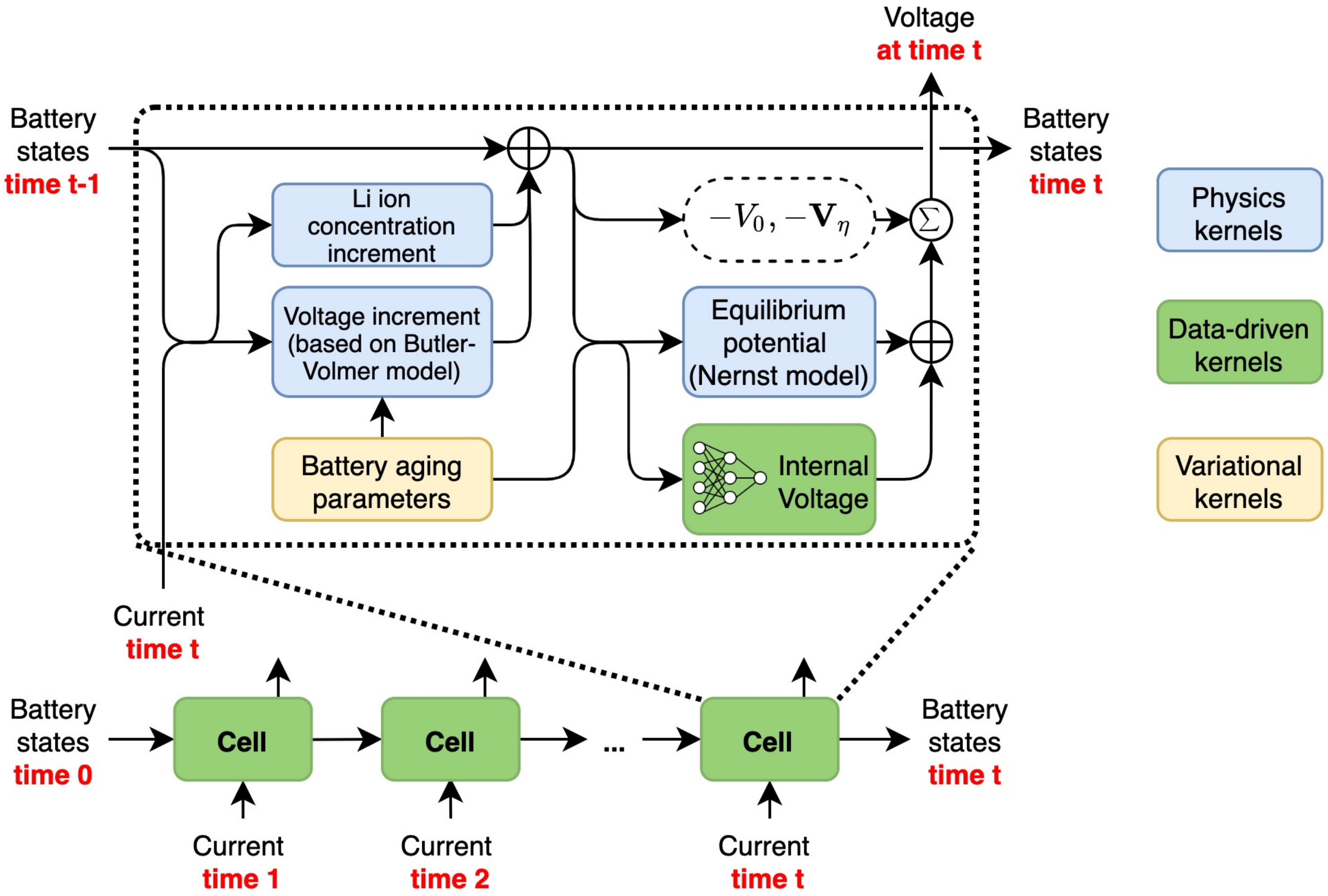
(a) Example of random loading conditions



(b) Aging can cause models to diverge from observations



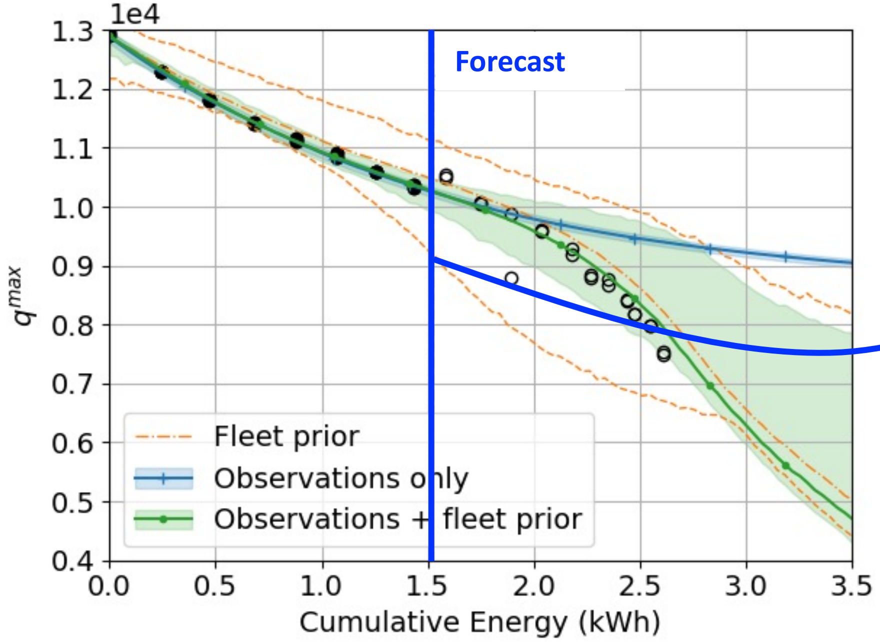
Hybrid physics-informed neural network



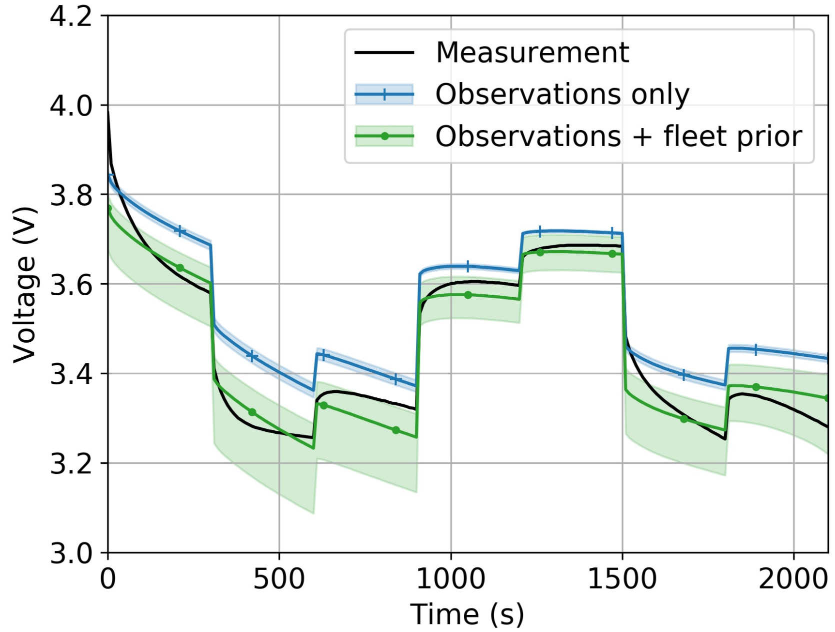
R. G. Nascimento et al., "Hybrid Physics-Informed Neural Networks for Lithium-Ion Battery Modeling and Prognosis," to be published.

Forecasting with hybrid digital twin

(a) Aging model



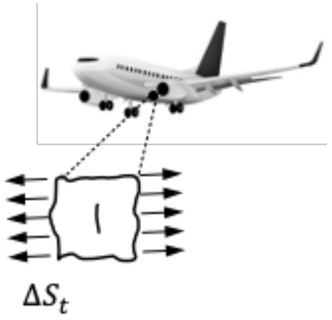
(b) Probabilistic forecast data



Model-form uncertainty in corrosion fatigue

Challenge

- Assumed: pure mechanical fatigue
- After 5 years: corrosion-fatigue

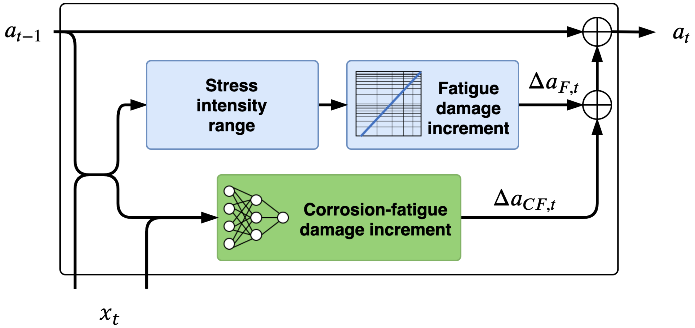


Data

- Load history of 5 years: 150 aircraft
- Crack length: 15 aircraft at end of 5th year.

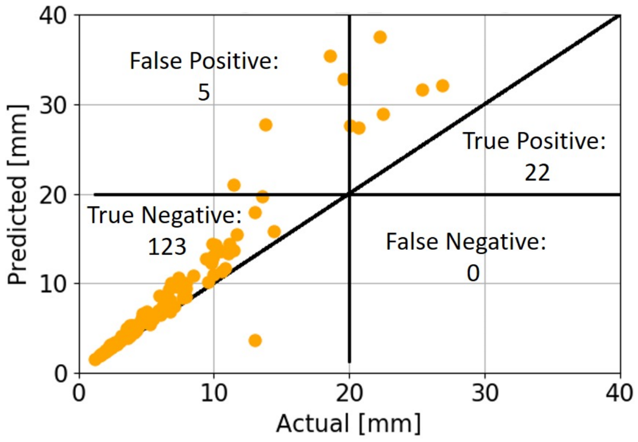
Damage accumulation grossly underestimated!!!

(a) Hybrid physics-informed neural network cell

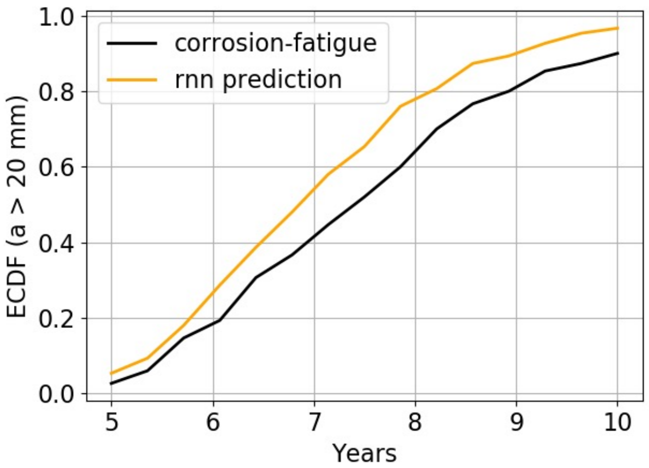


A. Dourado and F. A. C. Viana, “Physics-informed neural networks for missing physics estimation in cumulative damage models: a case study in corrosion fatigue,” ASME Journal of Computing and Information Science in Engineering, Vol. 20 (6), 10 pages, 2020.

(b) Fleet prediction at the end of 5th year.



(c) Probability of failure forecast



Integration with NVIDIA SimNet

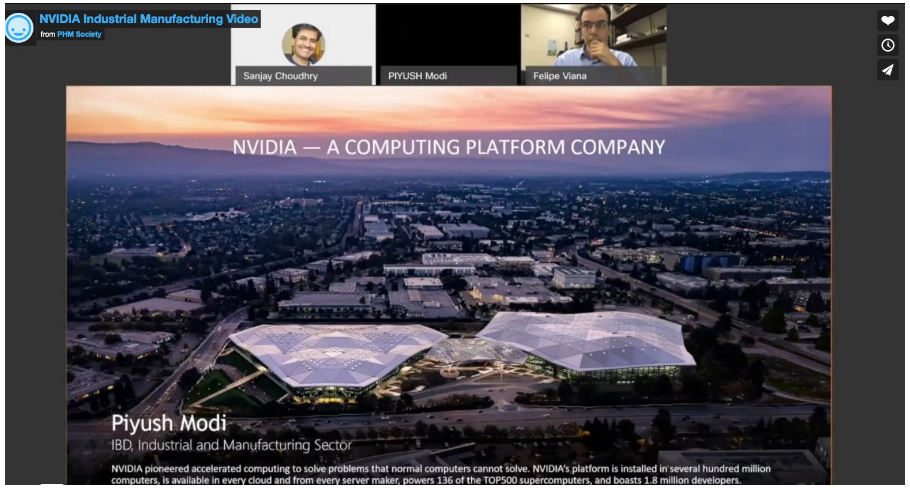
Proposed digital twin:

NVIDIA SimNet:

- Stress as a function of pressure differential
- Linear elastic analysis

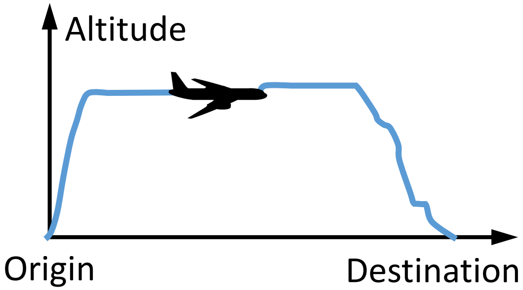
UCF hybrid physics-informed neural networks:

- Perform damage accumulation
- Adjust local stresses



https://player.vimeo.com/video/474830082

(a) Engineering analysis



$\sigma_{HOOP} \propto \Delta P$

ΔP: pressure differential (function of altitude)

Crack growth:

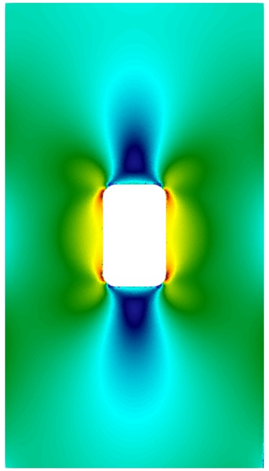
$\frac{da}{dN} = C\Delta K^m$ and $\Delta K = F\Delta S\sqrt{\pi a}$

$\boldsymbol{F = 1.122}$ **(assumed)**

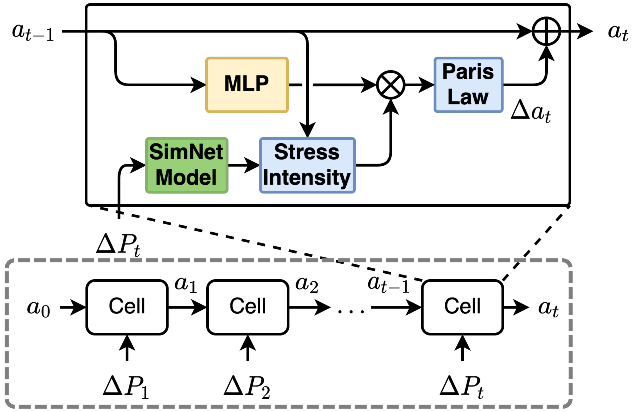
$\boldsymbol{F = f(a)}$ **(reality)**

(b) Digital twin model

SimNet simulations



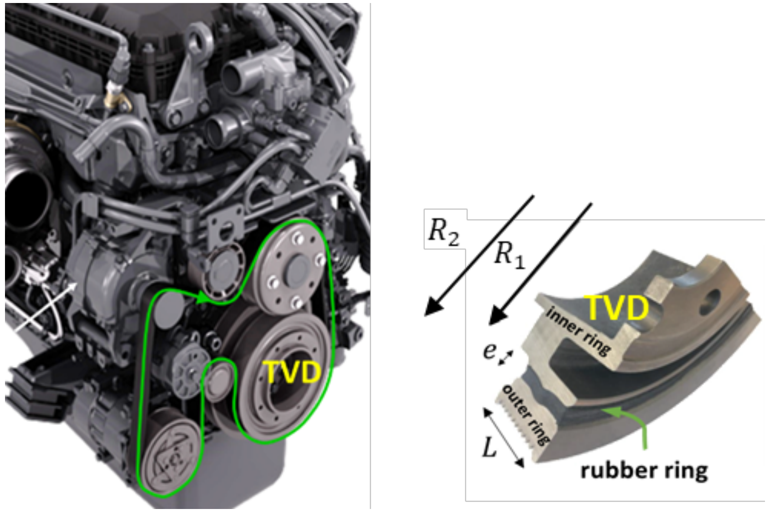
Cumulative damage cell



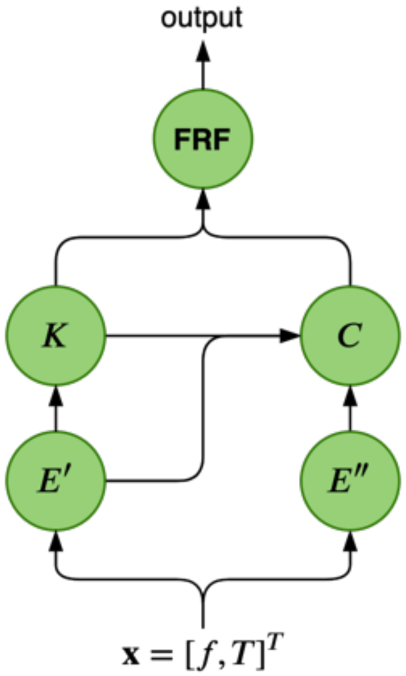
Recurrent neural network

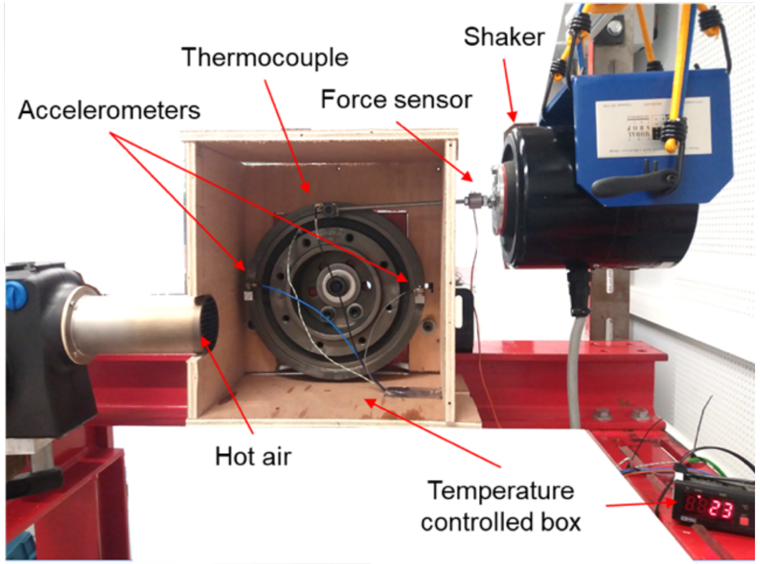
Application outside prognosis: torsional vibration damper

(a) Front engine accessory drive



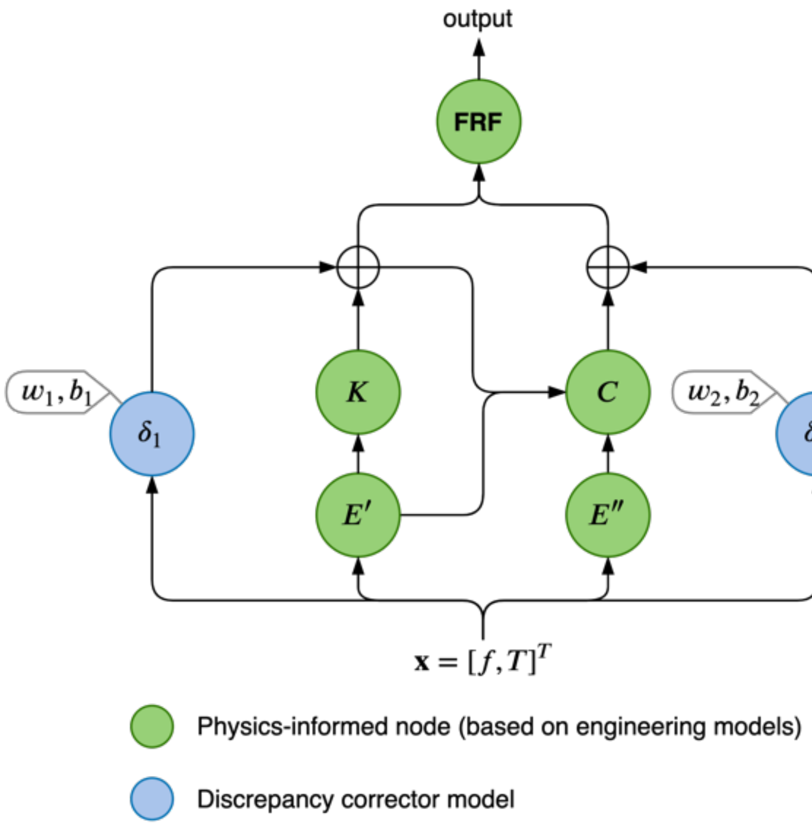
(b) Model and experimental testing

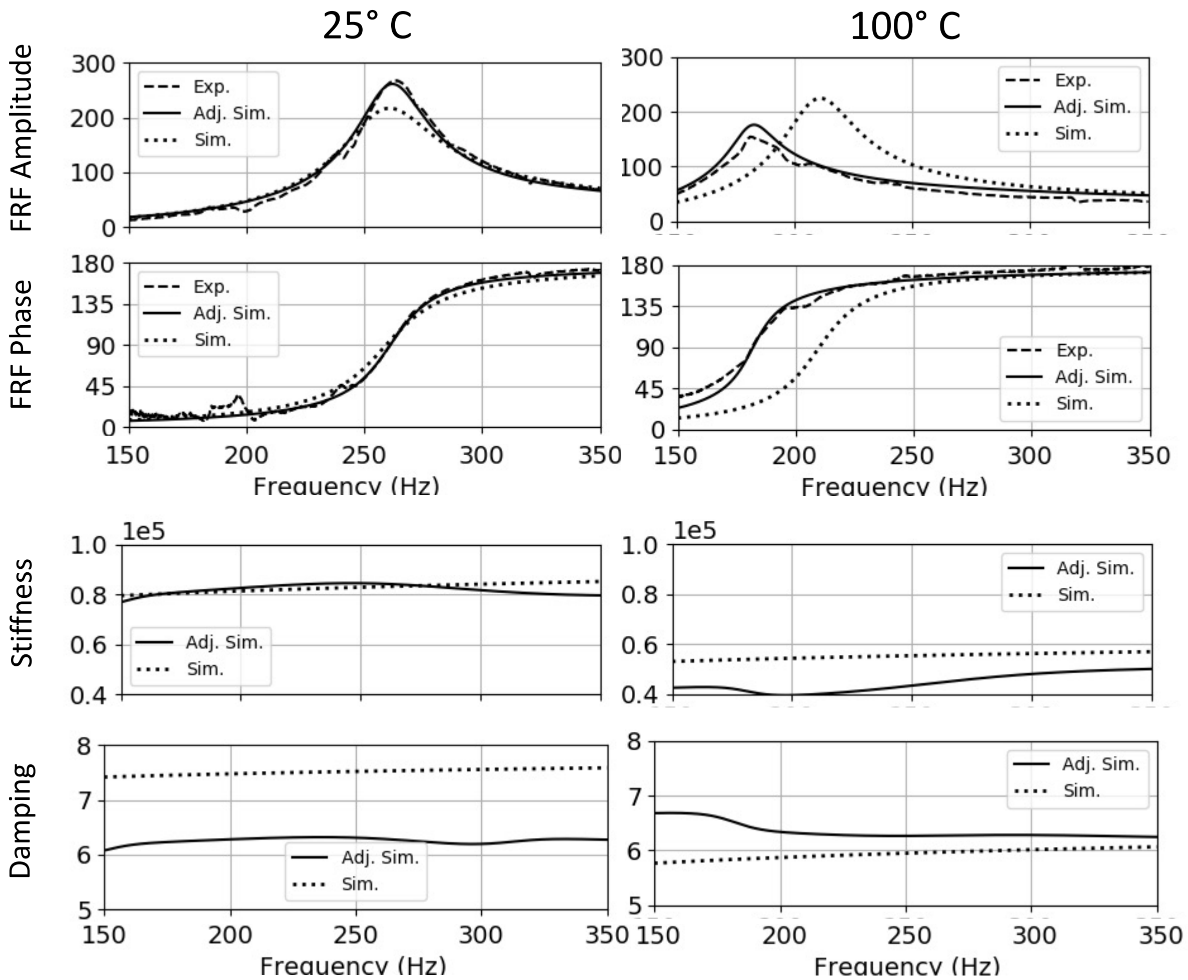




Y. A. Yucesan, F. A. C. Viana, L. Manin, and J. Mahfoud, "Adjusting a torsional vibration damper model with physics-informed neural networks," Mechanical Systems and Signal Processing, Vol. 154, pp. 107552, 2021. (DOI: 10.1016/j. ymssp.2020.107552).

Hybrid model and results





Probabilistic Mechanics Laboratory

Publications:

pml-ucf.github.io/publications

GitHub

Physics-informed neural networks package
github.com/PML-UCF/pinn

Ordinary differential equation solver:
https://github.com/PML-UCF/pinn_ode_tutorial

Wind turbine main bearing fatigue
github.com/PML-UCF/pinn_wind_bearing

Corrosion-fatigue prognosis
github.com/PML-UCF/pinn_corrosion_fatigue

Credit really goes to my PhD students


Andre Von Zuben


Arinan Dourado


Kajetan Fricke


Renato Nascimento


Yigit Yucesan

Sponsors and Collaborators

BakerHughes

NASA

NREL
NATIONAL RENEWABLE ENERGY LABORATORY

NVIDIA

INSA
INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
LYON

Probabilistic Mechanics Laboratory