

**SÃO PAULO STATE UNIVERSITY
SCHOOL OF ENGINEERING OF ILHA SOLTEIRA**

**NEURAL NETWORKS: ENHANCING INTELLIGENT SYSTEMS WITH DEEP
LEARNING**

**Gabriel D. Silva
Student
Douglas D. Bueno
Advisor**

**UNESP
Ilha Solteira – SP
2023**

RESEARCH REPORT

The present report approaches a way to improve smart systems. Through artificial intelligence applied in the mechanical engineering field, it provides a consistent algorithm that can reads data, trains the machine and provides results about the situation and what to do with it. It will be studied two cases, one of them using machine learning classical techniques to determine the forces applied to a unnamed aerial vehicle and other using deep learning techniques like neural networks in the structural health monitoring area.

Complete after the research is done.

Keywords: machine learning, structural health monitoring, unnamed aerial vehicle

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYM

This document is incomplete. The external file associated with the glossary ‘main’ (which should be called `deep_learning.gls`) hasn’t been created.

Check the contents of the file `deep_learning.glo`. If it’s empty, that means you haven’t indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can’t be generated. If the file isn’t empty, the document build process hasn’t been completed.

If you don’t want this glossary, add `nomain` to your package option list when you load `glossaries-extra.sty`. For example:

```
\usepackage[nomain]{glossaries-extra}
```

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

```
\usepackage[automake]{glossaries-extra}
```

- Run the external (Lua) application:

```
makeglossaries-lite.lua "deep_learning"
```

- Run the external (Perl) application:

```
makeglossaries "deep_learning"
```

Then rerun \LaTeX on this document.

This message will be removed once the problem has been fixed.

CONTENTS

1 INTRODUCTION

The use of Artificial Intelligence (AI) is very present nowadays [? ? ?]. This area of statistics neither is new nor started just now with the autonomous cars and voice assistants [?], but it is clear that in the last years it has been increasingly gaining more popularity. This happens mainly because of the advances that the World Wide Web has been had over the years [? ?], since dial-up internet connection, back in the eighties, until now, with broadband internet and smartphones equipped with 5G connection. Another factor is that in the past, the cost to get a large capacity of storage memory was significantly more expensive than it is now, what makes today cheaper and easy to get memory to store information [?]. With the amount of data available, the evolution of internet and storage capacity, now it is not difficult to obtain, keep and analyze databases to make decisions [?].

AI application is everywhere and today, more than ever, it is easy to realize that. Either to get multimedia recommendations on streaming platforms, like occurs at Netflix, YouTube, Spotify, and so many others platforms [?], or to make predictions on the financial market and sports betting [? ? ?], AI is there behind the scenes making all the magic happen. Evidently there is nothing really magical about them, it is pure mathematics combined with a programming language that produces the algorithm capable of doing those things [? ? ? ?]. The launch of ChatGPT-3, and shortly thereafter ChatGPT-4, has shown the power of those technologies and how they can change the way people do things [? ? ? ?].

Getting into the smart systems application, the use of AI is widely used to Structural Health Monitoring (SHM), which is heavily used in the aerospace and civil fields, [? ?]. The level and the complexity of the AI to be applied to monitor the structure, whether is going to use Deep Learning (DL) and Neural Network (NN) or simpler methods of Machine Learning (ML) like regressions, is determined by the problem itself and the results desired [?]. In some cases, the standards methods use numerical techniques and they may not be feasible, especially when there is a huge data to be analyzed. Thus, taking the AI road is an alternative to get the needed results for the monitoring in a more practical way [? ?].

Still in this context, but in the field of Unmanned Aerial Vehicle (UAV), the use of AI can be combined to integrate UAV through wireless communication networks [?] what can be useful in the agriculture sphere [?] with technologies like Internet of Things (IoT) [? ?]. Also, the use of the AI can be subtle, such as the use of a built-in MATLAB® function to make a simple NN to determine the final pose of a UAV based on the initial pose and the forces applied on it [?], or can be more sophisticated, like the use of ML and DL algorithms to predict materials properties, design new materials, discover new mechanisms and control real dynamic systems [? ?].

It is clear, therefore, that AI can transit into different fields, such as entertainment, business, health care, marketing, financial, agriculture, engineering, among others [? ? ? ? ? ?]. The use of the Big Data (BD) can not only make it clear the scenario to be studied, but also to support making strategical decisions [? ?]. The internet and hardware improvement [?], alongside the facility to storage data with accessible costs, encourages the AI use due to the benefits it can provide.

1.1 Motivation

With the 4.0 industry, the engineering evolution is growing bigger every year [?]. Solving engineering problems the traditional way may not be the best solution due to its non-triviality to complex systems and their mathematical modeling. With the amount of available data and power processing, many tasks may now be done with the AI aid [?].

In the SHM field, detecting failures and monitoring the structure is vital to prevent damages. Installing sensors in the structure and send signals to a central processing is a common way to provide predictive maintenance for structures [? ?]. Accidents with wagons can be avoided by detecting cracks in railways. AI can be used to determine the failures and possible damages to determine if the railway is able to keep working.

For the control engineering area, a control system to UAVs are necessary to get them to do their tasks properly. A traditional *white box* method is certainly a rock-solid but not a trivial way to do that task. With a *black box* approach, it is possible to determine the control forces by having only the initial pose and the desired trajectory, instead of modeling mathematically from the scratch [? ?]. This way, for determined situations, all the mathematical modeling complexity can be replaced to an AI system that can predict the control forces by giving easy-to-get information.

The use of AI techniques as a different approach to the traditional engineering problems can decrease efforts to complex systems and facilitate problems resolution, providing low-costs and faster solutions.

1.2 Objective

To develop two AI algorithms based on NN to apply in smart systems. The main goals are: (i) to determine the control forces used to move an UAV based on its initial pose and the desired trajectory; and (ii) to detect railways cracks through piezoelectric signal for SHM.

2 LITERATURE REVIEW

This chapter deals with the history, the main concepts and some practical cases of SHM inside the industry and academic area, besides showing how it may be used in the railway crack detection context. Next, in the dynamic field, it will be studied the main mechanical concepts to get the necessary understanding to an UAV motion as well the basics to know how an UAV can be controlled. Then, it will be shown the mathematics behind the algorithms of deep learning that will be implemented in the ???. Finally, the way how the algorithms are going to be implemented and the tools necessary to achieve the desired NN.

2.1 Structural Health Monitoring

2.1.1 Definition

According to [?], the SHM main purpose is to provide, during the life of a structure, a diagnosis of: the state of the constituent material; the different parts of the structure; and the full assembly of each part that makes the structure as a whole. It is an improved way to make non-destructive evaluation. It can be applied in several areas such as civil infrastructure, like bridges and buildings; aerospace, like airplanes and spaceships; and mechanical, like machines.

Figure 2.1. SHM and Human Nervous System Analogy. The nerve endings are responsible to pass the information to the brain; as the sensors embedded in the structure play the role of the nerve endings, the central processing does the brain role.

Source: prepared by the author.

Furthermore, it also can be associated as an analogy to the human nervous system. Just like the sensors send a signal to the central processor, the human senses send a signal to the brain to make the recognition of what is happening, as shown in the ???.

2.1.2 Brief History

The SHM development began back in the 20th century and it has been coupled with the evolution of the digital computing hardware, what allowed the costs of the applied techniques less expensive over time.

It all starts back in the early 1970s and 1980s. The oil industry tried to develop vibration-based damage identification methods for offshore platforms by simulating damage scenarios, examining the changes in the resonant frequencies and correlating them with those measured on a platform. In the same period, the aerospace community studied vibration-based damage identification along with the development of the space shuttle. From that, it was developed the shuttle modal inspection system which aimed to identify fatigue damage in components like fuselage panels and control surfaces. The system was so successful that all orbiter vehicles had been periodically subjected to this test. Also, the civil engineering community studied vibration-based damage evaluation of bridge structures and buildings in the late 1980s [?].

From the late 1990s to the early 2000s, [?] showed the evolution of the techniques used in SHM, analyzing mainly the following factors: the operational evaluation; data acquisition and cleansing; feature extraction; and statistical modeling for feature discrimination. He also verified

that the statistical pattern recognition had not been embraced by the researchers to be more often used in such matter.

Nowadays, in order to contour inherent issues of SHM methods, as large computational effort and hand-crafted work that results in poor classification performance, many deep learning techniques have been used, such as Convolutional Neural Network (CNN) [?].

2.1.3 Main Techniques

Accelerometers

The use of accelerometers is consolidated in the engineering community to be used in several areas and it is present in the people daily life in things like game consoles, smartphones, and tablets.

Micro Electromechanical Systems (MEMS) sensor have several applications in measuring linear acceleration or angular motion along axis as an input to control a system. MEMS accelerometer sensors often measure the movement of a mass with a position measuring interface circuit that is converted into a digital electrical signal by an analog-to-digital converter for digital processing [?].

In SHM situation, the accelerometers are in the MEMS. The MEMS are, then, embedded in the structure and can provide information about the structure by detecting low-amplitude and low-frequency vibrations that are not always viable with the conventional low-cost sensor boards [?].

There are many others sensors used in vibration-base techniques like velocity and displacement sensors, however the accelerometers are widely used for this purpose.

Graphical Inspection

The use of digital cameras to detect any kind of irregularity in the surface is also a way to monitor the structure, mainly in the surface areas. The camera itself may be static in a strategical position that allow it to provide good images to be analyzed or can be embedded in the structure itself or in an UAV that will surround it.

To automate and improve the accuracy of the damage detection, image processing techniques are employed, that being a non-conventional approach [?]. In the civil engineering context, it is commonly utilized computer vision to damage detection [?] and also UAV integrated in the same local as the structures for SHM [?].

Many of the images obtained can have their not only the images improved by AI, but also the analyses can take advantage of it.

Piezoelectric Materials

When dealing with acoustic-based techniques, the use of piezoelectric materials as sensor is a great choice due to its ability to respond to stimuli, incorporation, and compatibility with construction materials. Beyond that, these materials are relatively cost-efficient and can sense vibrations in the structures they are installed [?].

Piezoelectric materials and their main property were discovered back in 1880 [?]. The phenomenon is that by the application of pressure in those kinds of materials in the correct direction, it is observed the production of a potential difference and consequently an electrical

charge. Examples of materials that are piezoelectric are quartz, zinc, sodium chlorate, tourmaline, calamine, topaz, tartaric acid, cane sugar, and others [?].

The application of these materials in SHM is basically to install the piezoelectric sensor in the structure intend to be monitored and through the tension or compression in it done, a sign will be sent to the central system by the potential differential. The signal indicates that something not usual is happening in the structure. Of course there are levels of the signals and each case must be evaluated in its context. In the last years the use of piezoelectric materials has been capable to identify failures, like the presence of delamination damage, as long as the piezoelectric sensors are close to the damage [?].

2.1.4 Railway Cracks

Train is one of the most used means of transportation around the world, either to transport people or groceries, therefore, there are inherent problems in the attached to it. One of the most common problems is the crack on the railway track, mainly due to the expansion and contraction caused by the heat and to constant pressure because of the wagon.

The crack in a railway is considerable problem because it may cause fatal accidents since the wagon is able to leave the railway. In this scenario, many methods are used to detect the crack or to foresee it before any misfortune happen. [?] proposed a system to identify the cracks and prevent the accidents. One of its advantages is that if some crack is detected on the track, the train starts to slow and stop before it passes by there. Other method includes the use of sensor coupled in the track that allow to detect the crack and send a signal to the command center through IoT [?].

The use of piezoelectric materials for SHM is very common, as seen in the [?]. [?] presents a system where piezoelectric transducer are installed along the railway track. They receive an electrical wave and send it, then, a signal to the receiver, making it possible, also through Finite Element Method (FEM), to detect any inconsistency that should not be there.

There are, hence, lots of methods that can be used to detect and prevent accidents in railway tracks. Putting them together and optimize them with SHM techniques are an efficient way of improve the railway ecosystem.

2.2 Unnamed Aerial Vehicle Control

2.2.1 Usage

An UAV has several applications, going from the simplest to the most sophisticated. It can be used since for entertainment, like toys; commercially, to record big shows in arenas; surveillance, to monitor places; and also in engineering, aiding in various context to improve some processing.

Due to its portability and autonomy, it can be used to facilitate the delivery o medicines. In this sense, UAV can be used for transportation of medical goods in critical times, where other means of transportation may not be feasible. In the final of 2019, COVID-19 pandemics spread throughout the world, making it difficult to deliver patients their needed medicines [? ?]. Besides, risks are inherent to the transportation and in come countries, like the USA, UAV usage may be restricted [?]. A strategical way to use them is also welcome.

In the agriculture context, in order to boost the productivity, UAV can be used to remotely

sense the farming, obtaining information on the state of the fields with non-contact procedures, like nutrient evaluation and soil monitoring; or even for aerial spraying, using pesticide to prevent damages in the plantation [?].

The main reason for its adoptions is the mobility, low maintenance costs, hovering capacity, ease of deployment, etc. It is widely used for the civil infrastructure, gathering photographs faster than satellite imagery and with better quality. Combining those benefits with AI can be a powerful tool for the future [?].

2.2.2 Control Equations

Considering the UAV a quadcopter, as the ?? shows, ?], based on the work of ?], described the equation of motion for a quadcopter with a payload as being:

$$\mathbf{M}_{\eta_c}(\eta_c)\ddot{\eta}_c + \mathbf{C}_{\eta_c}(\nu, \eta_c)\dot{\eta}_c + \mathbf{g}_{\eta_c}(\eta_c) + \mathbf{K}_{\eta_c}(\eta_c)\eta_c = \tau_{\eta_c}(\eta_c) + \mathbf{F}_d \quad (2.1)$$

where $\mathbf{M}_{\eta_c}(\eta_c)$ is the inertial matrix; $\mathbf{C}_{\eta_c}(\nu, \eta_c)$ is the Coriolis matrix; $\mathbf{g}_{\eta_c}(\eta_c)$ is the gravitational vector; $\mathbf{K}_{\eta_c}(\eta_c)$ is the stiffness matrix; $\tau_{\eta_c}(\eta_c)$ is the control torque; \mathbf{F}_d is the gust vector; and ν is the velocity generalized coordinate in the body-frame. The ?? can be represented in the state space form as:

$$\dot{x}_s = \mathbf{A}_c x_s(t) + \mathbf{B}(u)(t) + \mathbf{X} \quad (2.2)$$

where $x_s = \{\dot{\eta}_c \quad \eta_c\}^\top$ is the state vector; $\mathbf{B}(u)(t)$ is the input vector; \mathbf{X} is the state vector of gravity; and \mathbf{A}_c and \mathbf{B} are the dynamic and input matrices, respectively.

Figure 2.2. Quadcopter dynamic scheme. F_i and T_i , ($i = 1, 2, 3, 4$), are the forces and the torque applied in the propeller, respectively. ω_j and v_j , ($j = x, y, z$), are the momentum and the velocities applied in the UAV, respectively. The payload is not represented in the figure.

Source: prepared by the author.

All non-explicit matrices and the development of the equations are shown in the ?] work.

2.2.3 Control Algorithm

?] developed a MATLAB® algorithm to control the quadrotor, as a *white box* method. It controls the UAV in three different trajectories: rectangular, circular and linear. Given τ as the input vector, which represents the position controller $U_1(t)$ and the attitude controller $U_2(t)$, $U_3(t)$, $U_4(t)$, it is able to give a complete overview of the quadrotor's motion. The algorithm provides the state space vector \mathbf{x}_s with the quadrotor position and angles, as their derivatives.

$$\tau = \{U_1 \quad U_2 \quad U_3 \quad U_4\}^\top \quad (2.3)$$

$$\mathbf{x}_s = \{x \quad y \quad z \quad \phi \quad \theta \quad \psi \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi}\}^\top \quad (2.4)$$

2.3 Artificial Neural Networks

2.3.1 Deep Learning

The concepts of deep learning studied in this section is going to be based on the work of [?], [?] and the documentation of PyTorch¹, TensorFlow² and MATLAB®³.

There are several definitions of AI [?], but the computer scientist [?] defines it as “the science and engineering of making intelligent machines, especially intelligent computer programs”. He also states that “it is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable”.

The big area of study is the AI and it includes several branches like fuzzy logics, robotics and machine learning. The later one, in turn, is another field with also some branches and one of them is the deep learning. However, all the three terms can be interchangeable in the major context.

The deep learning history goes back to the 1940s and it had several names over the years. It was called by *cybernetics* (1940s–1960s), *connectionism* (1980s–1990s), and from 2006 until now is known as *deep learning*. The DL models were engineered systems inspired by the biological brain and they were denominated Artificial Neural Network (ANN). One of the motivations of the neural perspective was to understand that the brain provides a proof by example that intelligent behavior is possible and try to reverse engineer the computation principals behind the brain, duplicating its functionality. Today it goes beyond the neuroscientist perspective and it is more of general principle of learning multiple levels of composition.

DL dwells in the programming sphere. The approach, however, it is not like the traditional programming scripts and models. To automate stuff, there are three main parts: (i) the input data, (ii) the rule (function) and (iii) the output data. In oth types there are two of three parts available, but different ones for each other. In the traditional programming, there is the input data and the rule, for the algorithm output the data. For deep learning, there is the input data and the output data, for the algorithm provides the rule. A good analogy is cooking: in the traditional programming context, one has the ingredients and the recipe to make the main course; in the deep learning context, one has the ingredients and the main course to discover the recipe.

2.3.2 Neural Networks Models

A ANN is machine learning a model that simulate a biological NN to make a machine learns as the human being learns. ANN are the heart of DL and there are several models of them, each one most suitable for different kind of problems. Some of them are Multilayer Perceptron (MLP), CNN, Recurrent Neural Network (RNN), among others.

Multi-layer Perceptron

A MLP is a important class of NN. It consists of a set of sensorial units that compose the *input layer*; one or more *hidden layers*; and an *output layer*. The input signal propagates forward through the network, layer by layer. They are used to solving complex problems, with the supervised training with the *error back-propagation* algorithm.

¹<https://pytorch.org/docs/stable/index.html>

²https://www.tensorflow.org/api_docs

³<https://www.mathworks.com/help/matlab/>

The learning by back-propagation consists of two steps through the layers of the perceptron: a forward pass (propagation) and a backward pass (back-propagation). In the forward pass, an input vector is applied to the sensorial nodes of the network and it propagates through the network, layer by layer; in this step the weights are fixed. During the backward pass, the weights are fit accordingly through a loss function (see ??). This error signal is propagated through the network in the opposite direction of the synaptic connections. The weights are adjusted to make that the network output gets closer of the wanted output. ?? represents a MLP.

Figure 2.3. Visual Representation of a MLP. The input vector x_i is given in the input layer with i assuming any integer, just as the output data y_i . The weights w_{x_i} are specific for each input data. Input and output data can have multiple entries.

Source: prepared by the author.

The three main features of the MLP are:

- Non-linear activation function. It is commonly used a smooth non-linear activation function, like rectifier function (ReLU):

$$y_j = \begin{cases} x, & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2.5)$$

Or sigmoid function:

$$y_j = \frac{1}{1 + \exp(-v_j)} \quad (2.6)$$

where v_j is the weighted sum of all input layers with their respective weights of the j neuron; and y_j is the output of the neuron.

- Hidden layers. They allow the network to learn complex tasks, extracting progressively the most significantly features of the input vector.
- Connectivity. High level of connectivity, determined by the network synapses.

These features, plus the ability to learn from the experience of the training, that makes the MLP so powerful, however, they are also responsible for its deficiency. First, the non-linearity and the high connectivity makes hard the theoretical analysis of an MLP; second, the hidden layers make it more difficult to visualize the learning processing. The learning process is harder because the search must be conducted in a much bigger space of possible functions.

2.3.3 Loss Function

The *loss function*, also called *cost function* or *error function*, is the one used measure the error between the predicted output of an algorithm and the real target output. There are several loss functions suitable to different kind of situation. For each distributed data there is one that fits better. Many kinds of them are available and must be analyzed the most proper one to each case. The choice of what loss function should be picked will depend on not only the data and its pattern, but also the computational processing and the cost attached to it.

Regression

Although regression problems do not require DL to create a satisfactory model, naturally it is possible to do so. For regression problems, common loss functions adopted are the Mean Absolute Error (MAE) and Mean Squared Error (MSE) [?]:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.7)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.8)$$

where n is the sample size; y_i is the predicted output; and \hat{y}_i is the real target.

?? shows a linear data and how the domain of the loss function is obtained for a linear regression.

Figure 2.4. Loss Function for Linear Regression. The loss function take all the distances between the predicted and the target value to verify if the model is in the right path. The lower the distance, the better the model. The y -axis represents the output data and the x -axis represents the input data.

Source: prepared by the author.

2.3.4 Optimizer

The optimizer is an algorithm that updates the model in response to the output of the loss function, that is, it aids to minimize the loss function. As the loss function minimizes, the model is getting closer to the target values and, hence, closer to the real pattern.

Gradient Descent

The *gradient descent* is one of the main algorithm [?] that optimizes the model and many important ones are based on it, like the Stochastic Gradient Descent (SGD). The goal is to get the minimum, as the error (loss) between the predicted and the target data is null. This would mean that the model fits to the pattern of the data.

Figure 2.5. Gradient Descent Process. In this case, the loss function (yellow curve) can be represented in a two-axes plan. Depending on the data, it is not possible to represent graphically due to its multi dimension. Each point represents the learning step. When the gradient descent reaches the minimum of the loss function, it means that the model may be accurate. Note that the gradient descent can reach a local minimum of the function and not the global minimum necessarily. The y -axis represents the loss function and the x -axis represents the weight values.

Source: prepared by the author.

The gradient descent is a powerful algorithm that reduces the loss function, minimizing the error between the predicted value and the target value.

Since the gradient of a function gives the direction of the steepest ascent of a function and it is orthogonal to the surface at a determined point, it seems reasonable that moving in the perpendicular direction gives the maximum increase of the function [?]. On the other hand, the negative of the gradient may be used to find the opposite, that is, the steepest descent of the

function, or the minimum decrease. If the steps given to the direction of the negative gradient of the function are small, there is a good chance to get minimum value of the function. However, if the steps are too long, the chance to pass by the minimum value is high [?]. These steps are called *learning rate* and should be chosen wisely.

This way, let \mathbf{x} be the entry vector with the predicted data and L the loss function adopted for some deep learning model, and ϵ the learning rate, the gradient descent is:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \epsilon \nabla L(\mathbf{x}_t) \quad (2.9)$$

In determined cases, it is possible to avoid running the iterative algorithm and just go directly to the critical point by solving $\nabla L(\mathbf{x}_t) = 0$ for \mathbf{x} .

Stochastic Gradient Descent

As seen, gradient descent is a powerful tool to minimize the loss function, however, for large data, the cost of operation is very high and its use is not feasible. The main idea of SGD is that the gradient is an expectation. Later, the data is divided in subsets, also called *mini-batch* and then the gradient is performed over them. The mini-batch size is chosen to be a relatively small numbers of examples. The data inside each subset may be considered redundant, that is why it uses one single value of the subset to compute the gradient descent. This way, the process is considerable better for computational resources.

The SGD can be written as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\epsilon}{m} \sum_{i=1}^m \nabla L(\mathbf{x}_t; p^{(i)}, q^{(i)}) \quad (2.10)$$

where m is the mini-batch size; and $\nabla L(\mathbf{x}; p^{(i)}, q^{(i)})$ is the gradient of the loss function with respect to the parameter vector \mathbf{x} for the i^{th} example $(p^{(i)}, q^{(i)})$ in the mini-batch.

Yet, nowadays, with the amount of data, many techniques are still applied in SGD as creating an automatic adaptive learning rates which achieve the optimal rate of convergence [?] and the momentum technique to improve it [?].

Adam

Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, like the loss function, as seen in the ???. It is based on adaptive estimates of low-order moments and computationally efficient, requiring little computational memory. Adam is a strategical choice when using large data or parameters and with very noisy/sparse gradients [?].

[?] showed that the algorithm can be implemented as it follows:

Algorithm 1 Adam Algorithm. Good default setting are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. Operations on vectors are element-wise.

Require: α : stepsize

Require: $\beta_1, \beta_2 \in [0, 1]$: exponential decay rates for the moment estimates

Require: $f(\theta)$: loss function

Require: θ_0 : initial parameter

$m_0 \leftarrow 0$

$v_0 \leftarrow 0$

$t \leftarrow 0$

while θ_t not converged **do**

$t \leftarrow t + 1$

$g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$

$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$

$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$

$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$

$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$

$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$

end while

return θ_t

3 METHODOLOGY

The code implementation will be pragmatical and the lines of the code will not be fully explained. The frameworks methods will not be explained either, but their documentation are reasonably comprehensive with previous programming knowledge, especially in Python and MATLAB®, and they are going to be linked whenever possible.

While the engineering goal of AI is to solve real-world problems using it as an equipment, the scientific goal is to determine which ideas explain the various sorts of intelligence [?] and the current objective is to use AI from the engineering perspective.

3.1 Softwares

3.1.1 MATLAB®

The standard in the engineering industry, MATLAB® is a powerful toolbox that can be used to several activities. Since applications in fields like medicine and biology [?], like image processing, until the most complex problems in engineering [?], that involve matrix operation and control simulation, MATLAB® has lots of tools that aid to solve problems.

[?] used MATLAB® to develop their algorithm, hence the data generation will be done with it.

3.1.2 PyTorch

A framework is a group of libraries for a programming language that implements a lot of tools to facilitate some tasks. There is a lot of deep learning ones available and the most popular ones are TensorFlow [?] and PyTorch [?]. While the first one was developed by Google and released in 2015 the second one was developed by Meta (Facebook), although it is now under the Linux Foundation umbrella, and released in 2016, being both open-source. Many companies, like Uber [?] and Tesla [?], use PyTorch in their AI team, while companies like Coca-Cola uses TensorFlow [?]. This means that both are trustful frameworks to rely upon their built-in functions. PyTorch will be used to develop all the NN.

3.2 Neural Network for Unmanned Aerial Vehicle Control

3.2.1 Data Generation

Since the script of [?] provides the control torque τ as input and the state-space \mathbf{x}_s as output vector through dynamic and control equations, the NN goal developed is to go in the opposite direction, as a inverse function: take \mathbf{x}_s as the input vector and predict the τ_η vector as output. Modifications in the script are minimal. The time is a discrete vector with 200s and step 0.01, therefore the time vector has 1×20001 dimension. The “extra” value of time is the zero value.

The output vector \mathbf{T} has 20001×4 dimension the and the input vector \mathbf{X}_s has 20001×12

dimension:

$$\mathbf{T} = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (3.1)$$

$$\mathbf{X}_s = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & \dot{\phi} & \dot{\theta} & \dot{\psi} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (3.2)$$

Circular trajectory was arbitrary selected as starting point. By a loop, it was generated 1000 different trajectories changing the position by increasing 1/600 for each loop.

This way, it was generated one thousand input and output vector. Both were stored in a MATLAB® variable and exported through the .mat extension to be used with TensorFlow inside Python environment.

3.2.2 Neural Networks Overview

The first approach for the NN is to use the raw data, both for input and output and do the training. Even though it works, it does not give the proper result. Therefore, the preprocessing of the data is mandatory to get the best results. This way, all input and output data were normalized in order to get them all standardized.

The normalization is in L2 form, from the *sklearn.preprocessing.normalize* function, applied in each matrix column. From the trained NN, all input data should be normalized and naturally the output also will be normalized. However, the control forces (output data) can not be normalized to be useful, but there is no “denormalized” correspondent matrix to the output data from the NN.

To solve this problem, a second NN was created to be able to denormalize the output data. When preprocessing the data, as the normalization is done, both norms of the input and output data are stored and the second NN is made from them. The ?? show how the data and the NNs are related for the training. When the training and the validation is done, the ready-to-use model will perform as shown in the ?? scheme.

Figure 3.1. Data and neural networks relation. The training data in the extremes are the ones generated by the white box parametric model.

Source: prepared by the author.

Figure 3.2. Model in production. The scheme shows how the process returns the control forces from the state space.

Source: prepared by the author.

3.2.3 Neural Network Modeling

The NN 1 is responsible for, from the normalized state space, to return the normalized control forces, as shown in the ??. The problem is considered as a regression problem, as the ??

shows. Input and output data are all matrices.

$$f(x, y, \dots, \dot{\theta}, \dot{\psi}) = \langle U_1, U_2, U_3, U_4 \rangle \quad (3.3a)$$

$$f(\mathbf{X}_s) = \mathbf{T} \quad (3.3b)$$

Figure 3.3. Schematic model of the neural network for the normalized data. The input layer receives every element of the \mathbf{x}_s and returns every element of τ . There are two hidden layers, each one with 64 neurons.

Source: prepared by the author.

Characteristics of the NN 1 are provided in the ??.

TABELA AQUÍ

Figure 3.4. Schematic model of the neural network for the norms. The input data receives the state space norm and return the correspondent force control norm. There are two hidden layers, each one with 64 neurons.

Source: prepared by the author.

3.3 Neural Network for SHM

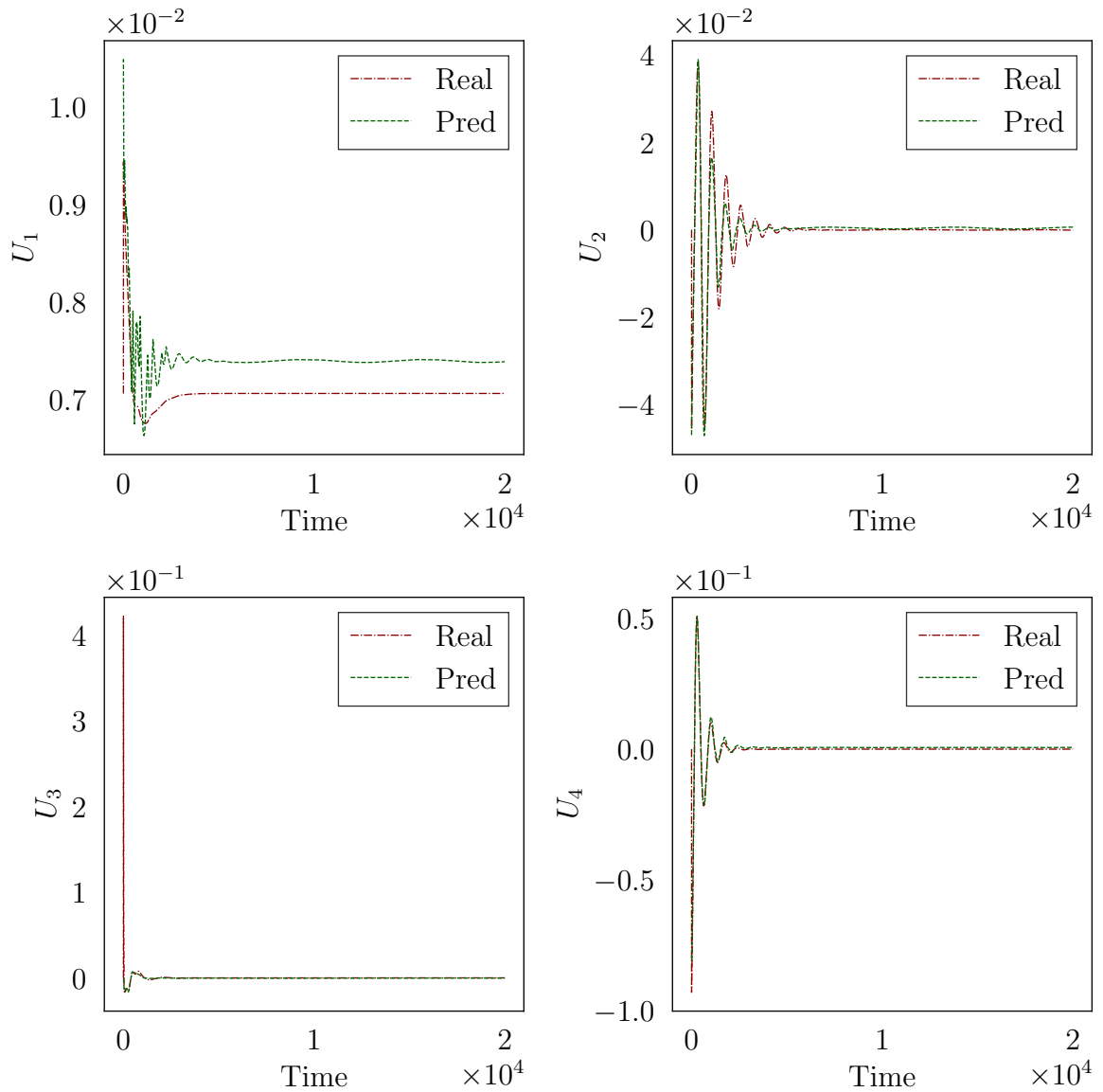
4 RESULTS AND DISCUSSION

This chapter shows the results obtained with the NN modeling. For both cases (SHM and UAV) the first approach is the model itself, showing the metrics and evaluating them. The second one is a comparison from the true label and the predict label for a random sample. Finally, a discussion is made based on the obtained results.

4.1 Case Study 1: Unnamed Aerial Vehicle

4.1.1 Model A

Figure 4.1. Forces normalized obtained from the neural network 1. Comparison is made from the results predicted from the neural network and the normalized preprocessing data, both for input and output data.



Source: prepared by the author.

4.2 Case Study 2: Structural Health Monitoring

Figure 4.2. Forces denormalized obtained from the combination of neural network 1 and 2. Comparison is made from the results predicted from the combination of both neural network and the raw data got from the white box parametric model.

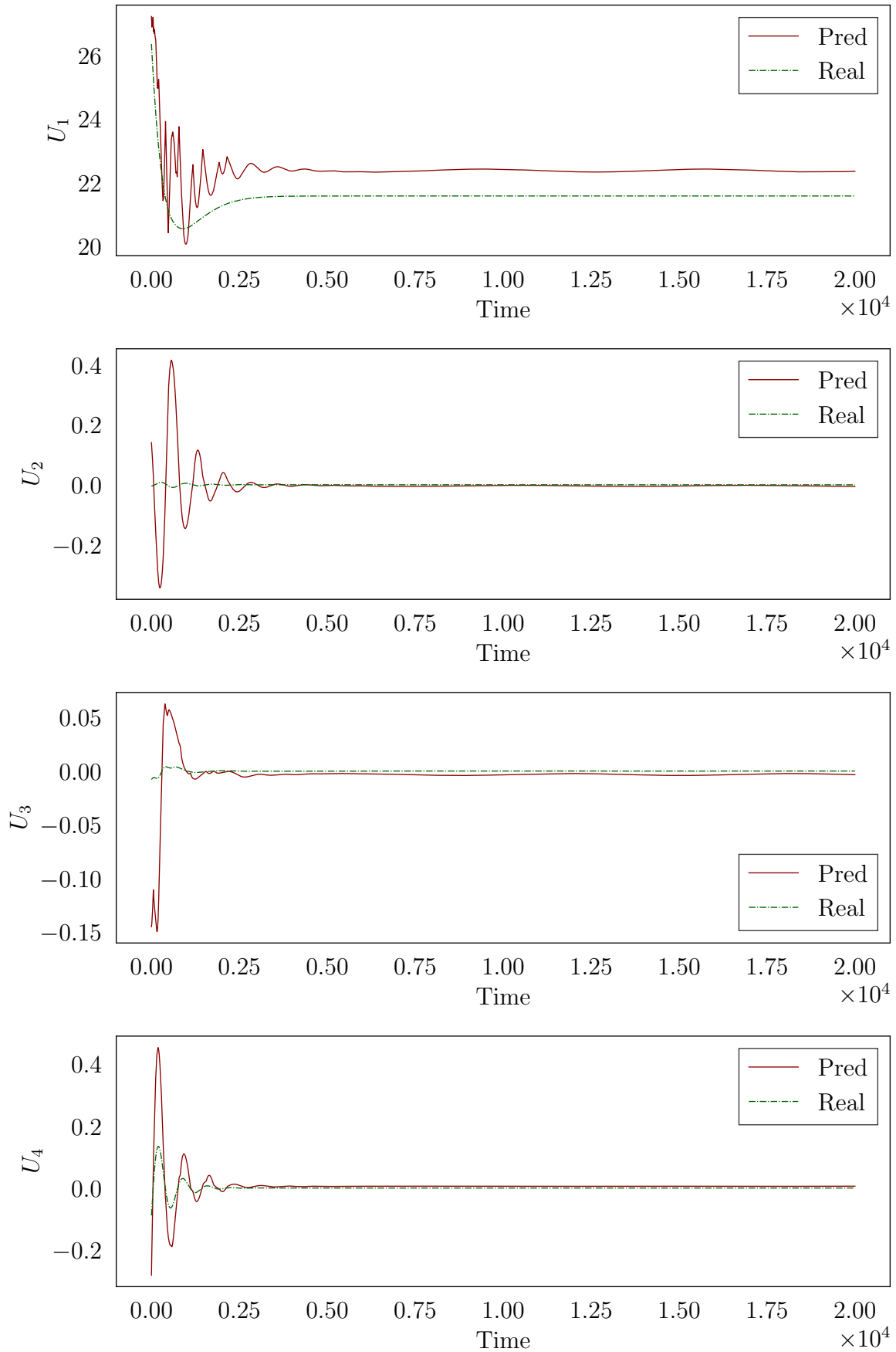
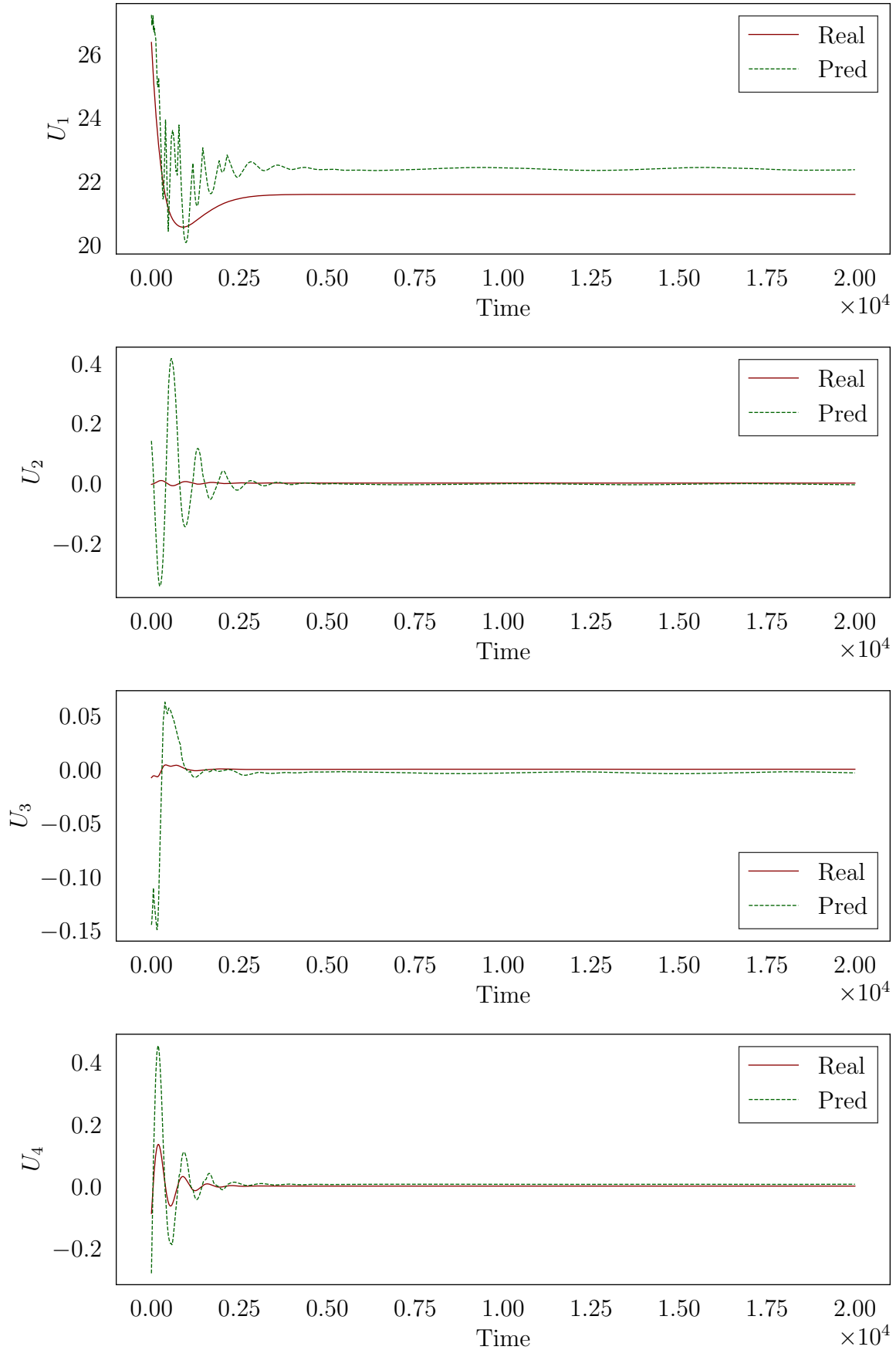


Figure 4.3. denormalized obtained from the combination of neural network 1 and 2 plus correction. Comparison is made from the results predicted from the combination of both neural network and the raw data got from the white box parametric model with a correction for U_1 .



5 CONCLUSION