Gabriel Duarte da Silva

Artificial Neural Networks applied to Unnamed Aerial Vehicle Control

Universidade Estadual Paulista

Orientador: Douglas D. Bueno

Ilha Solteira, SP

15 de novembro de 2023

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Fig.   Area of the ith component 30

456   Isto éum número

123   Isto éoutro número

lauro cesar   este éo meu nome

SUMÁRIO

1 INTRODUCTION

The use of AI is very present nowadays (LEE, 2020; POOLA, 2017; RABUNAL; DORADO, 2006). This area of statistics neither is new nor started just now with the autonomous cars and voice assistants (MUTHUKRISHNAN et al., 2020), 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 (LEINER et al., 2009; COHEN-ALMAGOR, 2011), 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 (GODA; KITSUREGAWA, 2012). 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 (DUAN; EDWARDS; DWIVEDI, 2019).

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, You-Tube, Spotify, and so many others platforms (CHAN-OLMSTED, 2019), or to make predictions on the financial market and sports betting (MILANA; ASHTA, 2021; KOLLÁR, 2021; HUBÁČEK; ŠOUREK; ŽELEZNÝ, 2019), 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 (GOODFELLOW; BENGIO; COURVILLE, 2016; AURÉLIEN, 2022; RASCHKA, 2015; RASCHKA et al., 2022). 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 (BISWAS, 2023a,b; LUND; WANG, 2023; BAIDOO-ANU; OWUSU ANSAH, 2023).

Getting into the smart systems application, the use of AI is widely used to SHM, which is heavily used in the aerospace and civil fields, (AZIMI; ESLAMLOU; PEKCAN, 2020; YE; JIN; YUN, 2019). The level and the complexity of the AI to be applied to monitor the structure, whether is going to use DL and NN or simpler methods of ML like regressions, is determined by the problem itself and the results desired (FARRAR; WORDEN, 2012). 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 (SMARSLY; LEHNER; HARTMANN, 2007; SUN et al., 2020).

Still in this context, but in the field of UAV, the use of AI can be combined to integrate UAV through wireless communication networks (LAHMERI; KISHK; ALOUINI, 2021) what

can be useful in the agriculture sphere (AHIRWAR et al., 2019) with technologies like IOT (VERDOUW; WOLFERT; TEKINERDOGAN, 2016; TZOUNIS et al., 2017). 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 (GERONEL; BOTEZ; BUENO, 2023), 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 (GUO et al., 2021; ASSILIAN, 1974).

It is clear, therefore, that AI can transit into different fields, such as entertainment, business, health care, marketing, financial, agriculture, engineering, among others (RUIZ-REAL et al., 2020; YU; BEAM; KOHANE, 2018; DAVENPORT; KALAKOTA, 2019; VERMA et al., 2021; MHLANGA, 2020; PANNU, 2015; GHATREHSAMANI et al., 2023). The use of the Big Data can not only make it clear the scenario to be studied, but also to support making strategical decisions (JEBLE; KUMARI; PATIL, 2018; KOŚCIELNIAK; PUTO, 2015). The internet and hardware improvement (BAJI, 2018), 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 (MEINDL et al., 2021). 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 (PHAM; PHAM, 1998).

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 (KAHANDAWA et al., 2012; NAGAYAMA; SPENCER, 2007). 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 (LOYOLA-GONZÁLEZ, 2019; WU et al., 2016). 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 chapter 4. Finally, the way how the algorithms are going to be implemented and the tools necessary to achieve the desired NN.

2.1 Unnamed Aerial Vehicle Control

2.1.1 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 (RAMAKRISHNAN et al., 2023; MCPHILLIPS, 2022). Besides, risks are inherent to the transportation and in come countries, like the USA, UAV usage may be restricted (THIELS et al., 2015). 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 (DEL CERRO et al., 2021).

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 (SIVAKUMAR; TYJ, 2021).

2.1.2 Control Equations

Considering the UAV a quadcopter, as the fig. 1 shows, Geronel, Botez e Bueno (2023), based on the work of Fossen (1994), described the equation of motion for a quadcopter with a payload as being:

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

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 eq. (2.1) can be represented in the state space form as:

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

where $x_s = \left\{ \dot{\eta}_c \quad \eta_c \right\}^\top$ 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.

Figura 1 – 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.
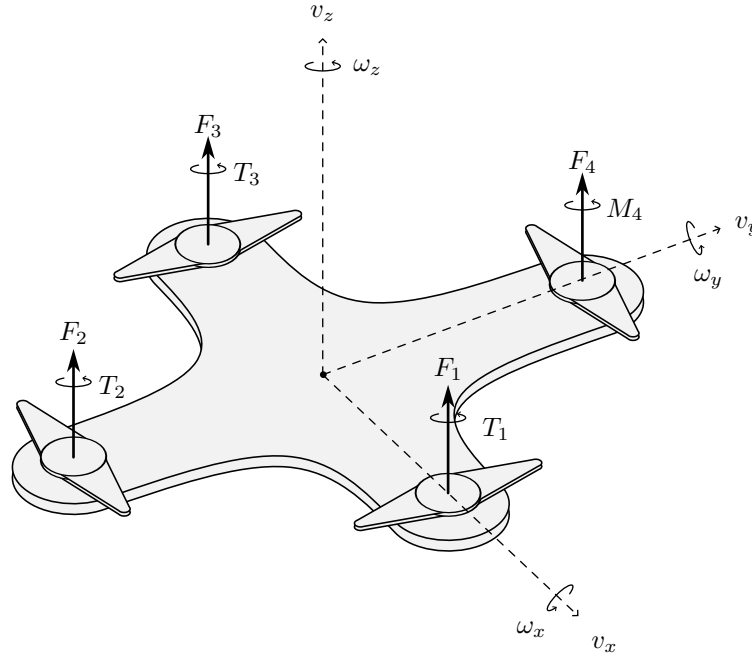
Source: prepared by the author.

All non-explicit matrices and the development of the equations are shown in the Geronel, Botez e Bueno (2023) work.

2.1.3 Control Algorithm

Geronel, Botez e Bueno (2023) 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.

$$\tau = \left\{ U_1 \quad U_2 \quad U_3 \quad U_4 \right\}^{\top} \tag{2.3}$$

$$\mathbf{x}_s = \left\{ x \quad y \quad z \quad \phi \quad \theta \quad \psi \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi} \right\}^{\top} \tag{2.4}$$

2.2 Artificial Neural Networks

2.2.1 Deep Learning

The concepts of deep learning studied in this section is going to be based on the work of Goodfellow, Bengio e Courville (2016), Haykin (1999) and the documentation of PyTorch.

There are several definitions of AI (WINSTON, 1992), but the computer scientist McCarthy (2007) 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 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.

2.2.2 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 MLP, CNN, 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.

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 section 2.2.3). 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. fig. 2 represents a MLP.

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):

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

Or sigmoid function:

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

Figura 2 – 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.
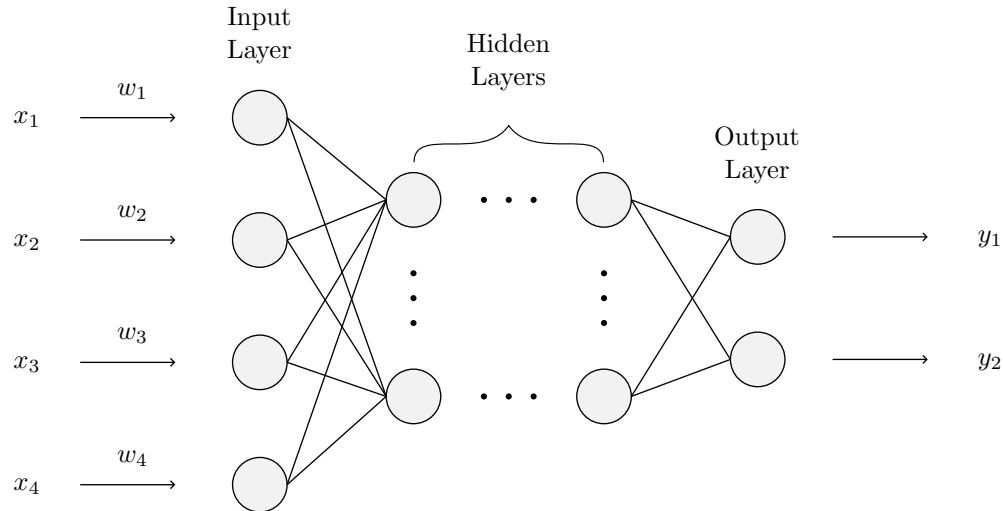
Source: prepared by the author.

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.

2.2.3 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 MAE and MSE (BUSSAB; MORETTIN, 2017):

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2.7}$$

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

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

fig. 3 shows a linear data and how the domain of the loss function is obtained for a linear regression.

Figura 3 – 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.
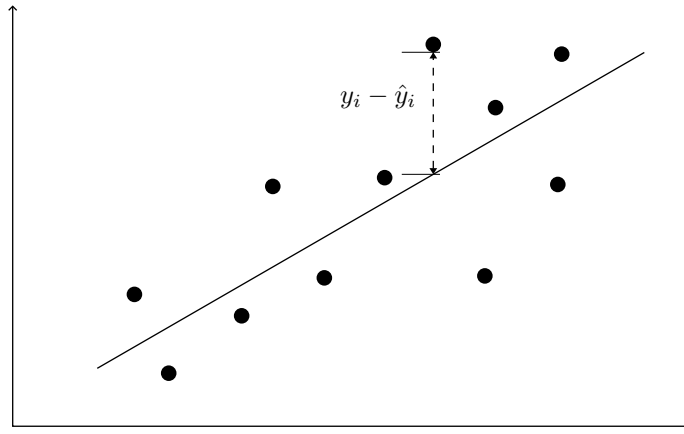
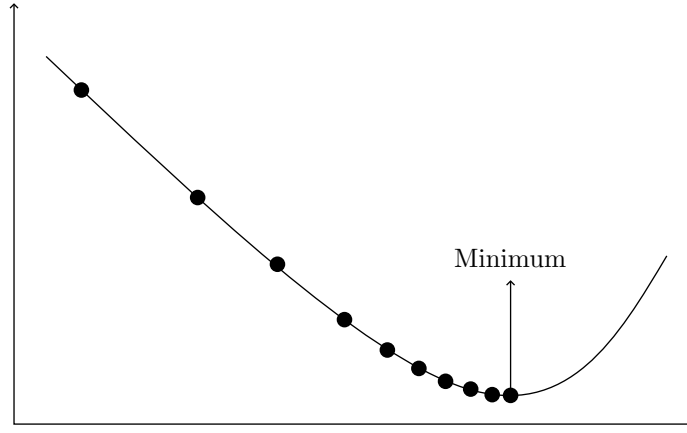Source: prepared by the author.

2.2.4 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 (NESTEROV, 2004) that optimizes the model and many important ones are based on it, like the 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.

Figura 4 – 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 (STEWART, 2016). 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 (NIELSEN, 2015). 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:

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

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:

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

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 ith 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 (DARKEN; MOODY, 1991) and the momentum technique to improve it (SUTSKEVER et al., 2013).

Adam

Adam is an algorithm developed by Kingma e Ba (2017) for first-order gradient-based optimization of stochastic objective functions, like the loss function, as seen in the section 2.2.3. 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 (KINGMA; BA, 2017).

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 (WINSTON, 1992) 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 (DEMIRKAYA; ASYALI; SAHOO, 2009), like image processing, until the most complex problems in engineering (BANSAL; GOEL; SHARMA, s.d.), that involve matrix operation and control simulation, MATLAB has lots of tools that aid to solve problems.

Geronel, Botez e Bueno (2023) 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 (ABADI et al., 2016) and PyTorch (PASZKE et al., 2019). 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 (GOODMAN, 2017) and Tesla (PYTORCH, 2019), use PyTorch in their AI team, while companies like Coca-Cola uses TensorFlow (TENSORFLOW, 2018). 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   Data Generation

Since the script of Geronel, Botez e Bueno (2023) 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 $200\,\mathrm{s}$ 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:

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

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

Circular trajectory was arbitrary selected as starting point. By a loop, it was generated 1000 different trajectories changing the position by adding random noise. Input and output vector were stored in a MATLAB variable and exported through the `.mat` extension to be used inside the Python environment.

3.3   Algorithm 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 fig. 5 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 fig. 6 scheme.

Figura 5 – Data and neural networks relation. The training data in the extremes are the ones generated by the white box parametric model.
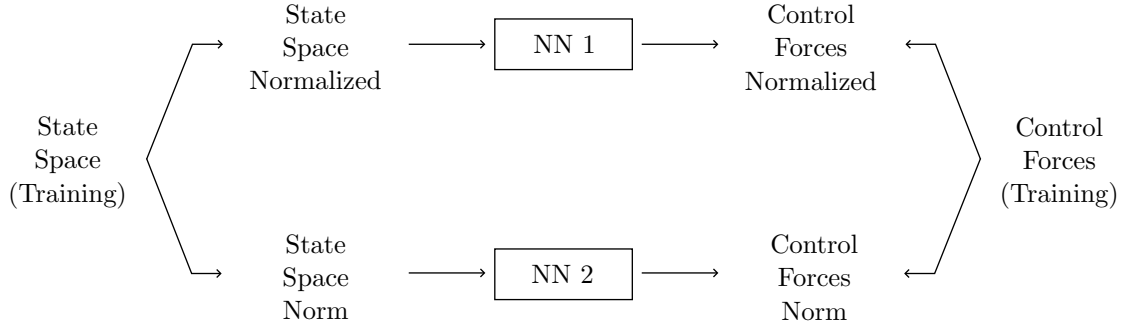
Source: prepared by the author.

Figura 6 – Model in production. The scheme shows how the process returns the control forces from the state space.
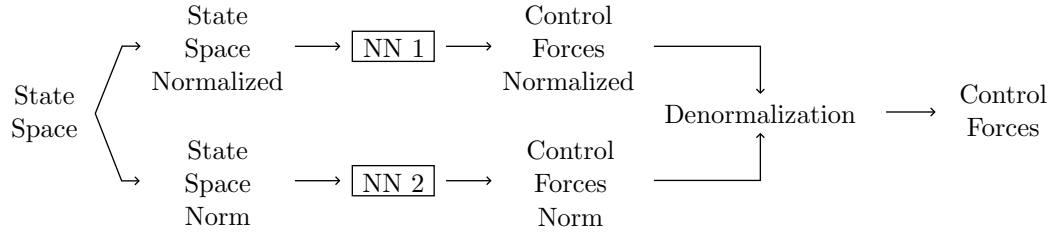
Source: prepared by the author.

3.4 Neural Network Modeling

The NN 1 is responsible for, from the normalized state space, to return the normalized control forces. The problem is considered as a regression problem, as the eq. (3.3) shows. Input and output data are all matrices.

$$f\left(x, y, \ldots, \dot{\theta}, \dot{\psi}\right) = \langle U_1, U_2, U_3, U_4 \rangle \tag{3.3a}$$

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

The NN 2 is responsible for, from the raw data, to get the norms from each input and output data of the NN 1. The problem is considered as a regression problem. Input and output data are all vectors.

Both NNs have similar parameters, being the main difference the input and output layer size. Characteristics of them are provided in the table 1.

Tabela 1 – Parameters of the neural networks and their training

Parameters	NN 1	NN 2
Input layer neurons	12	1
Output layer neurons	4	1
Hidden layers neurons	128	128
Hidden layers	8	8
Activation function	ReLU	ReLU
Loss function	MSE	MSE
Batch size	1	1
Train/test split	0.8/0.2	0.8/0.2

4 RESULTS

This chapter introduces and shows the results obtained with the NN modeling. Discussion about them are made in forward chapters.

4.1 Loss functions

Since there are two NN, two loss functions were obtained with the training. They are shown in the fig. 7.

Figura 7 – Loss function versus number of epochs for the neural networks. The number of epochs in the figure does not represent the total number of epochs in the training.

(a) Loss function for NN 1   **(b)** Loss function for NN 2
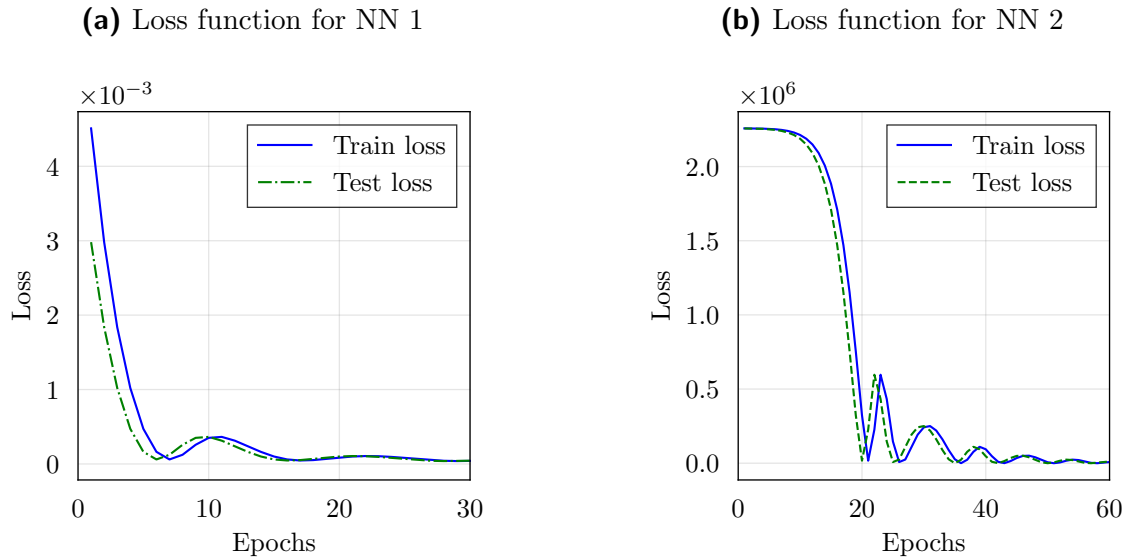
Source: prepared by the author.

4.2 UAV Forces

The control forces are generated to make the UAV travel in a circular trajectory. From Geronel, Botez e Bueno (2023) script, the correspondent trajectory for all the input and output data generated is shown in the fig. 8. This trajectory is arbitrary chosen and the other ones generated are very similar with noised added.

      The normalized forces from neural network NN 1 go through a detailed comparison with the white box script within the corresponding state space. This analysis is crucial because it reveals the consistancy of the model, aiming to show insights into the accuracy of NN 1

predictions. The results are presented in fig. 9. The denormalized forces obtained by joining the results of both NN are also presented in fig. 10, with the same goal.

Figura 8 – Trajectory of the UAV. Circular trajectory was chosen and the thousand other ones generated are only variations with noise added.
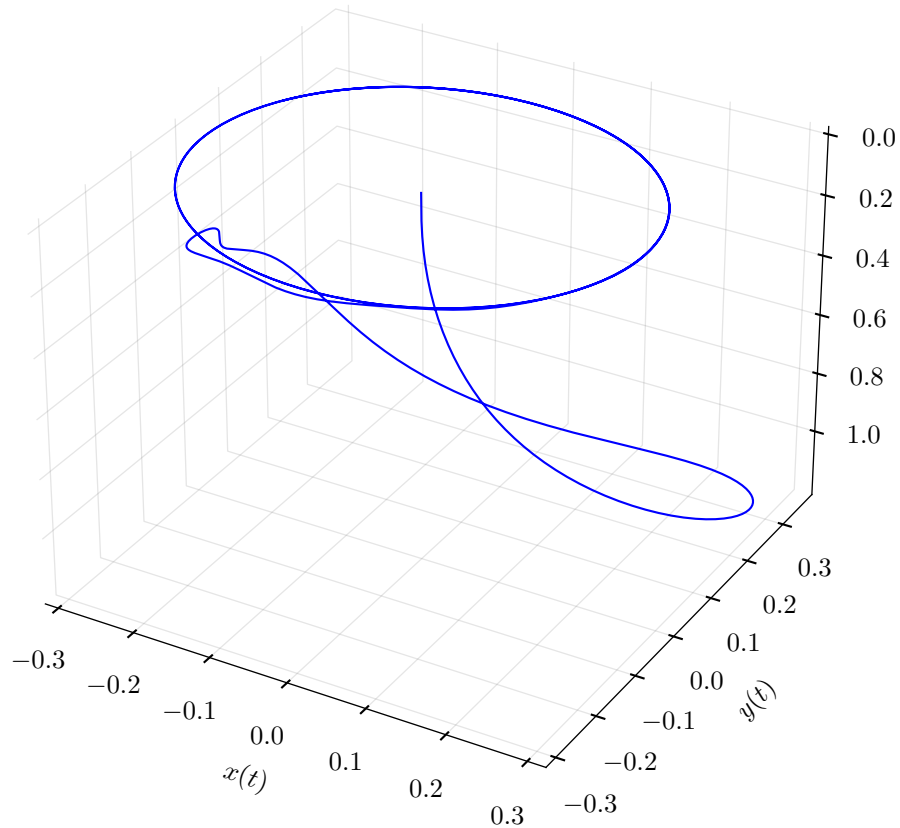

Source: adapted from Geronel, Botez e Bueno (2023)

Figura 9 – 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.
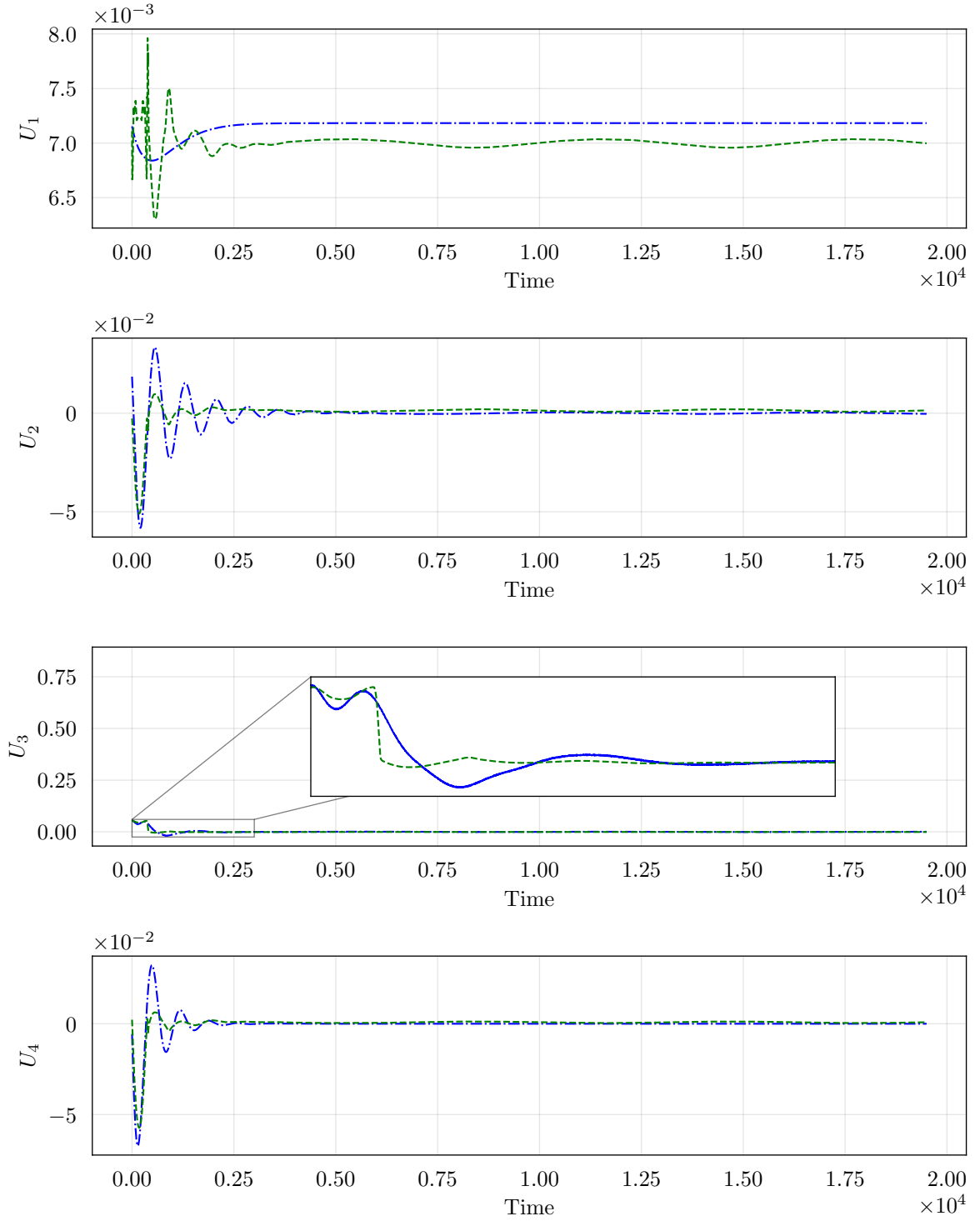
Source: prepared by the author.

Figura 10 – 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.
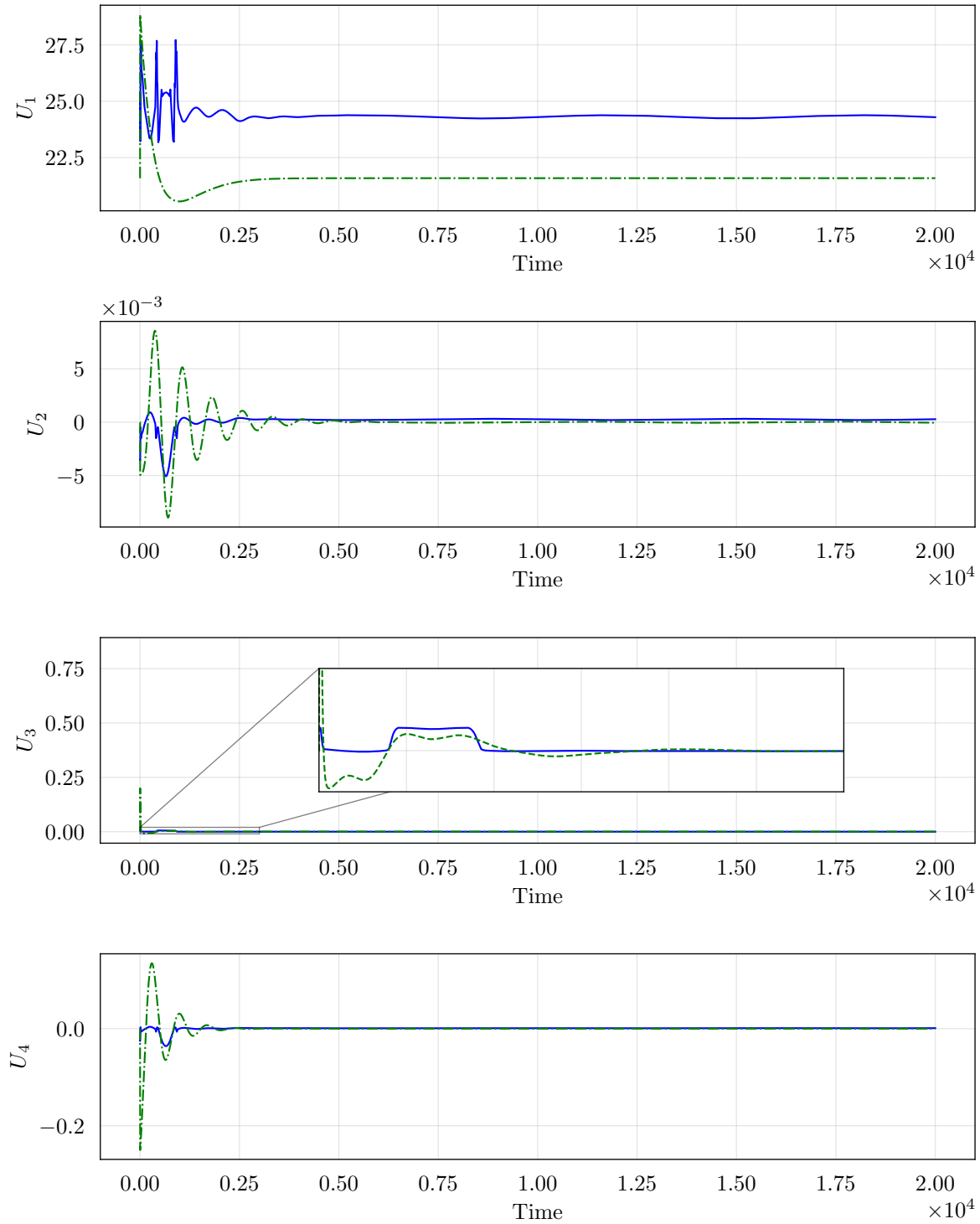
REFERÊNCIAS

ABADI, Martın et al. TensorFlow: A System for Large-Scale Machine Learning. **12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)**, p. 265–283, 2016.

AHIRWAR, S. et al. Application of Drone in Agriculture. **International Journal of Current Microbiology and Applied Sciences**, v. 8, n. 01, p. 2500–2505, jan. 2019. ISSN 23197692, 23197706. DOI: `10.20546/ijcmas.2019.801.264`. Acesso em: 4 abr. 2023.

ASSILIAN, Sedrak. **Artificial Intelligence in the Controle of Real Dynamic Systems**. 1974. Tese (Doutorado) – Queen Mary University of London.

AURÉLIEN, Géron. **Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow**. [S.l.]: O'Reilly Media, Inc., 2022.

AZIMI, Mohsen; ESLAMLOU, Armin; PEKCAN, Gokhan. Data-Driven Structural Health Monitoring and Damage Detection through Deep Learning: State-of-the-Art Review. **Sensors**, v. 20, n. 10, p. 2778, mai. 2020. ISSN 1424-8220. DOI: `10.3390/s20102778`. Acesso em: 4 abr. 2023.

BAIDOO-ANU, David; OWUSU ANSAH, Leticia. Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning. **SSRN Electronic Journal**, 2023. ISSN 1556-5068. DOI: `10.2139/ssrn.4337484`. Acesso em: 4 abr. 2023.

BAJI, Toru. Evolution of the GPU Device Widely Used in AI and Massive Parallel Processing. In: 2018 IEEE 2nd Electron Devices Technology and Manufacturing Conference (EDTM). Kobe: IEEE, mar. 2018. P. 7–9. ISBN 978-1-5386-3712-8. DOI: `10.1109/EDTM.2018.8421507`. Acesso em: 4 abr. 2023.

BANSAL, Raj Kumar; GOEL, Ashok K.; SHARMA, Manoj Kumar. **MATLAB and Its Applications in Engineering (Based on MATLAB 7.5 (R2007b)). - Description Based on Print Version Record**. Delhi: Dorling Kindersley. ISBN 978-81-317-4207-5.

BISWAS, Som S. Potential Use of Chat GPT in Global Warming. **Annals of Biomedical Engineering**, mar. 2023. ISSN 0090-6964, 1573-9686. DOI: `10.1007/s10439-023-03171-8`. Acesso em: 4 abr. 2023.

_____. Role of Chat GPT in Public Health. **Annals of Biomedical Engineering**, mar. 2023. ISSN 0090-6964, 1573-9686. DOI: `10.1007/s10439-023-03172-7`. Acesso em: 4 abr. 2023.

BUSSAB, Wilton de O; MORETTIN, Pedro A. **Estatística Básica**. [S.l.]: Saraiva Uni, 2017.

CHAN-OLMSTED, Sylvia M. A Review of Artificial Intelligence Adoptions in the Media Industry. **International Journal on Media Management**, Routledge, v. 21, n. 3-4, p. 193–215, out. 2019. ISSN 1424-1277. DOI: 10.1080/14241277.2019.1695619.

COHEN-ALMAGOR, Raphael. Internet History: **International Journal of Technoethics**, v. 2, n. 2, p. 45–64, abr. 2011. ISSN 1947-3451, 1947-346X. DOI: 10.4018/jte.2011040104. Acesso em: 4 abr. 2023.

DARKEN, Christian; MOODY, John E. Towards Faster Stochastic Gradient Search. v. 4, 1991.

DAVENPORT, Thomas; KALAKOTA, Ravi. The Potential for Artificial Intelligence in Healthcare. **Future Healthcare Journal**, v. 6, n. 2, p. 94–98, jun. 2019. ISSN 2514-6645, 2514-6653. DOI: 10.7861/futurehosp.6-2-94. Acesso em: 5 abr. 2023.

DEL CERRO, Jaime et al. Unmanned Aerial Vehicles in Agriculture: A Survey. **Agronomy**, v. 11, n. 2, p. 203, jan. 2021. ISSN 2073-4395. DOI: 10.3390/agronomy11020203. Acesso em: 19 abr. 2023.

DEMIRKAYA, Omer; ASYALI, Musa Hakan; SAHOO, Prasanna. **Image Processing with MATLAB: Applications in Medicine and Biology**. Boca Raton: CRC Press, 2009. ISBN 978-0-8493-9246-7.

DUAN, Yanqing; EDWARDS, John S.; DWIVEDI, Yogesh K. Artificial Intelligence for Decision Making in the Era of Big Data – Evolution, Challenges and Research Agenda. **International Journal of Information Management**, v. 48, p. 63–71, out. 2019. ISSN 02684012. DOI: 10.1016/j.ijinfomgt.2019.01.021. Acesso em: 4 abr. 2023.

FARRAR, Charles R; WORDEN, Keith. **Structural Health Monitoring: A Machine Learning Perspective**. [S.l.]: John Wiley & Sons, 2012.

FOSSEN, Thor I. **Guidance and Control of Ocean Vehicles**. Chichester ; New York: Wiley, 1994. ISBN 978-0-471-94113-2.

GERONEL, R. S.; BOTEZ, R. M.; BUENO, D. D. Dynamic Responses Due to the Dryden Gust of an Autonomous Quadrotor UAV Carrying a Payload. **The Aeronautical Journal**, v. 127, n. 1307, p. 116–138, jan. 2023. ISSN 0001-9240, 2059-6464. DOI: 10.1017/aer.2022.35. Acesso em: 4 abr. 2023.

GHATREHSAMANI, Shirin et al. Artificial Intelligence Tools and Techniques to Combat Herbicide Resistant Weeds—A Review. **Sustainability**, v. 15, n. 3, p. 1843, jan. 2023. ISSN 2071-1050. DOI: 10.3390/su15031843. Acesso em: 5 abr. 2023.

GODA, K.; KITSUREGAWA, M. The History of Storage Systems. **Proceedings of the IEEE**, v. 100, Special Centennial Issue, p. 1433–1440, mai. 2012. ISSN 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2012.2189787. Acesso em: 4 abr. 2023.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016.

GOODMAN. **Uber AI Labs Open Sources Pyro, a Deep Probabilistic Programming Language**. [S.l.: s.n.], nov. 2017. https://www.uber.com/en-GR/blog/pyro/. Acesso em: 3 mai. 2023.

GUO, Kai et al. Artificial Intelligence and Machine Learning in Design of Mechanical Materials. **Materials Horizons**, v. 8, n. 4, p. 1153–1172, 2021. ISSN 2051-6347, 2051-6355. DOI: 10.1039/D0MH01451F. Acesso em: 4 abr. 2023.

HAYKIN, Simon S. **Neural Networks: A Comprehensive Foundation**. 2nd ed. Upper Saddle River, N.J: Prentice Hall, 1999. ISBN 978-0-13-273350-2.

HUBÁČEK, Ondřej; ŠOUREK, Gustav; ŽELEZNÝ, Filip. Exploiting Sports-Betting Market Using Machine Learning. **International Journal of Forecasting**, v. 35, n. 2, p. 783–796, abr. 2019. ISSN 01692070. DOI: 10.1016/j.ijforecast.2019.01.001. Acesso em: 4 abr. 2023.

JEBLE, Shirish; KUMARI, Sneha; PATIL, Yogesh. Role of Big Data in Decision Making. **Operations and Supply Chain Management: An International Journal**, p. 36–44, jan. 2018. ISSN 2579-9363. DOI: 10.31387/oscm0300198. Acesso em: 5 abr. 2023.

KAHANDAWA, Gayan C. et al. Use of FBG Sensors for SHM in Aerospace Structures. **Photonic Sensors**, v. 2, n. 3, p. 203–214, set. 2012. ISSN 1674-9251, 2190-7439. DOI: 10.1007/s13320-012-0065-4. Acesso em: 4 set. 2023.

KINGMA, Diederik P.; BA, Jimmy. **Adam: A Method for Stochastic Optimization**. [S.l.]: arXiv, jan. 2017. arXiv: 1412.6980 [cs]. Acesso em: 20 abr. 2023.

KOLLÁR, Aladár. **Betting Models Using AI: A Review on ANN, SVM, and Markov Chain**. [S.l.], mar. 2021. DOI: 10.31219/osf.io/mr2v3. Acesso em: 4 abr. 2023.

KOŚCIELNIAK, Helena; PUTO, Agnieszka. BIG DATA in Decision Making Processes of Enterprises. **Procedia Computer Science**, v. 65, p. 1052–1058, 2015. ISSN 18770509. DOI: 10.1016/j.procs.2015.09.053. Acesso em: 5 abr. 2023.

LAHMERI, Mohamed-Amine; KISHK, Mustafa A.; ALOUINI, Mohamed-Slim. Artificial Intelligence for UAV-Enabled Wireless Networks: A Survey. **IEEE Open Journal of the Communications Society**, v. 2, p. 1015–1040, 2021. ISSN 2644-125X. DOI: 10.1109/OJCOMS.2021.3075201. Acesso em: 4 abr. 2023.

LEE, Raymond S. T. **Artificial Intelligence in Daily Life**. Singapore: Springer Singapore, 2020. ISBN 9789811576942 9789811576959. DOI: 10.1007/978-981-15-7695-9. Acesso em: 5 abr. 2023.

LEINER, Barry M. et al. A Brief History of the Internet. **ACM SIGCOMM Computer Communication Review**, v. 39, n. 5, p. 22–31, out. 2009. ISSN 0146-4833. DOI: 10.1145/1629607.1629613. Acesso em: 4 abr. 2023.

LOYOLA-GONZÁLEZ, Octavio. Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View. **IEEE Access**, v. 7, p. 154096–154113, 2019. ISSN 2169-3536. DOI: 10.1109/ACCESS.2019.2949286.

LUND, Brady D.; WANG, Ting. Chatting about ChatGPT: How May AI and GPT Impact Academia and Libraries? **Library Hi Tech News**, fev. 2023. ISSN 0741-9058, 0741-9058. DOI: 10.1108/LHTN-01-2023-0009. Acesso em: 4 abr. 2023.

MCCARTHY, John. What Is Artificial Intelligence? **Stanford University**, 2007.

MCPHILLIPS, Deidre. Home Delivery of Medications Can Help Improve Access, Especially When Time Is Tight. **CNN Health**, dez. 2022. Acesso em: 19 abr. 2023.

MEINDL, Benjamin et al. The Four Smarts of Industry 4.0: Evolution of Ten Years of Research and Future Perspectives. **Technological Forecasting and Social Change**, v. 168, p. 120784, jul. 2021. ISSN 00401625. DOI: 10.1016/j.techfore.2021.120784. Acesso em: 2 set. 2023.

MHLANGA, David. Industry 4.0 in Finance: The Impact of Artificial Intelligence (AI) on Digital Financial Inclusion. **International Journal of Financial Studies**, v. 8, n. 3, p. 45, jul. 2020. ISSN 2227-7072. DOI: 10.3390/ijfs8030045. Acesso em: 5 abr. 2023.

MILANA, Carlo; ASHTA, Arvind. Artificial Intelligence Techniques in Finance and Financial Markets: A Survey of the Literature. **Strategic Change**, v. 30, n. 3, p. 189–209, mai. 2021. ISSN 1086-1718, 1099-1697. DOI: 10.1002/jsc.2403. Acesso em: 4 abr. 2023.

MUTHUKRISHNAN, Nikesh et al. Brief History of Artificial Intelligence. **Neuroimaging Clinics of North America**, v. 30, n. 4, p. 393–399, nov. 2020. ISSN 10525149. DOI: 10.1016/j.nic.2020.07.004. Acesso em: 4 abr. 2023.

NAGAYAMA, Tomonori; SPENCER, Billie F. Structural Health Monitoring Using Smart Sensors. **Newmark Structural Engineering Laboratory Report Series 001**, Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign., nov. 2007. ISSN 1940-9826. Acesso em: 4 set. 2023.

NESTEROV, IU E. **Introductory Lectures on Convex Optimization: A Basic Course**. Boston: Kluwer Academic Publishers, 2004. (Applied Optimization, v. 87). ISBN 978-1-4020-7553-7.

NIELSEN, Michael. **Neural Networks and Deep Learning**. [S.l.]: Determination press San Francisco, CA, USA, 2015. v. 25.

PANNU, Avneet. Artificial Intelligence and Its Application in Different Areas. v. 4, n. 10, 2015.

PASZKE, Adam et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: ADVANCES in Neural Information Processing Systems. [S.l.]: Curran Associates, Inc., 2019. v. 32. Acesso em: 3 mai. 2023.

PHAM, D T; PHAM, P T N. Artificial Intelligence in Engineering, 1998.

POOLA, Indrasen. How Artificial Intelligence in Impacting Real Life Everyday. **International Journal for Advance Research and Development**, IJARnD, v. 2, n. 10, p. 96–100, 2017.

PYTORCH. **PyTorch at Tesla - Andrej Karpathy, Tesla**. [S.l.: s.n.], nov. 2019. Acesso em: 3 mai. 2023.

RABUNAL, Juan Ramon; DORADO, Julian (Ed.). **Artificial Neural Networks in Real-Life Applications**. Hershey PA: Idea Group Pub, 2006. ISBN 978-1-59140-902-1 978-1-59140-903-8 978-1-59140-904-5.

RAMAKRISHNAN, Manasvini et al. Impact of COVID-19 Pandemic on Medicine Supply Chain for Patients with Chronic Diseases: Experiences of the Community Pharmacists. **Clinical Epidemiology and Global Health**, v. 20, p. 101243, mar. 2023. ISSN 22133984. DOI: 10.1016/j.cegh.2023.101243. Acesso em: 19 abr. 2023.

RASCHKA, Sebastian. **Python Machine Learning**. [S.l.]: Packt Publishing Ltd, 2015.

RASCHKA, Sebastian et al. **Machine Learning with PyTorch and Scikit-Learn: Develop Machine Learning and Deep Learning Models with Python**. [S.l.]: Packt Publishing Ltd, 2022.

RUIZ-REAL, José Luis et al. Artificial Intelligence in Business and Economics Research: Trends and Future. **Journal of Business Economics and Management**, v. 22, n. 1, p. 98–117, out. 2020. ISSN 1611-1699, 2029-4433. DOI: 10.3846/jbem.2020.13641. Acesso em: 5 abr. 2023.

SIVAKUMAR, Mithra; TYJ, Naga Malleswari. A Literature Survey of Unmanned Aerial Vehicle Usage for Civil Applications. **Journal of Aerospace Technology and Management**, v. 13, e4021, 2021. ISSN 2175-9146. DOI: 10.1590/jatm.v13.1233. Acesso em: 19 abr. 2023.

SMARSLY, Kay; LEHNER, Karlheinz; HARTMANN, Dietrich. Structural Health Monitoring Based on Artificial Intelligence Techniques. In: COMPUTING in Civil Engineering (2007). [S.l.: s.n.], 2007. P. 111–118.

STEWART, James. **Calculus**. EIghth edition. Boston, MA, USA: Cengage Learning, 2016. ISBN 978-1-285-74062-1 978-1-305-27176-0.

SUN, Limin et al. Review of Bridge Structural Health Monitoring Aided by Big Data and Artificial Intelligence: From Condition Assessment to Damage Detection. **Journal of Structural Engineering**, v. 146, 2020.

SUTSKEVER, Ilya et al. On the Importance of Initialization and Momentum in Deep Learning, p. 1139–1147, 2013.

TENSORFLOW. **The Coca-Cola Company Using TensorFlow for Digital Marketing Campaigns (TensorFlow Meets)**. [S.l.: s.n.], ago. 2018. Acesso em: 26 mai. 2023.

THIELS, Cornelius A. et al. Use of Unmanned Aerial Vehicles for Medical Product Transport. **Air Medical Journal**, v. 34, n. 2, p. 104–108, mar. 2015. ISSN 1067991X. DOI: 10.1016/j.amj.2014.10.011. Acesso em: 19 abr. 2023.

TZOUNIS, Antonis et al. Internet of Things in Agriculture, Recent Advances and Future Challenges. **Biosystems Engineering**, v. 164, p. 31–48, dez. 2017. ISSN 15375110. DOI: 10.1016/j.biosystemseng.2017.09.007. Acesso em: 4 abr. 2023.

VERDOUW, C.; WOLFERT, S.; TEKINERDOGAN, B. Internet of Things in Agriculture. **CABI Reviews**, v. 2016, p. 1–12, jan. 2016. ISSN 1749-8848. DOI: 10.1079/PAVSNNR201611035. Acesso em: 4 abr. 2023.

VERMA, Sanjeev et al. Artificial Intelligence in Marketing: Systematic Review and Future Research Direction. **International Journal of Information Management Data Insights**, v. 1, n. 1, p. 100002, abr. 2021. ISSN 26670968. DOI: 10.1016/j.jjimei.2020.100002. Acesso em: 5 abr. 2023.

WINSTON, Patrick Henry. **Artificial Intelligence**. 3rd ed. Reading, Mass: Addison-Wesley Pub. Co, 1992. ISBN 978-0-201-53377-4.

WU, Z. F. et al. On Membership of Black-Box or White-Box of Artificial Neural Network Models. In: 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA). [S.l.: s.n.], jun. 2016. P. 1400–1404. DOI: 10.1109/ICIEA.2016.7603804.

YE, X.W.; JIN, T.; YUN, C.B. A Review on Deep Learning-Based Structural Health Monitoring of Civil Infrastructures. **Smart Structures and Systems**, v. 24, n. 5, p. 567–585, nov. 2019. DOI: 10.12989/SSS.2019.24.5.567. Acesso em: 4 abr. 2023.

YU, Kun-Hsing; BEAM, Andrew L.; KOHANE, Isaac S. Artificial Intelligence in Healthcare. **Nature Biomedical Engineering**, v. 2, n. 10, p. 719–731, out. 2018. ISSN 2157-846X. DOI: 10.1038/s41551-018-0305-z. Acesso em: 5 abr. 2023.