MASTER THESIS

An approach to some classical libration point orbit problems by means of artificial intelligence techniques

Sara Cecilia Abecia Hernanz

SUPERVISED BY

Josep Joaquim Masdemont Soler

Universitat Politècnica de Catalunya
Master in Aerospace Science & Technology
November 2021

Título: Aplicación de técnicas de Inteligencia Artificial a problemas clásicos de orbitas alrededor de puntos de libración
Autor: Sara Cecilia Abecia Hernanz

Director: Josep Joaquim Masdemont Soler

Fecha: 22 de noviembre de 2021

Resumen

Los rápidos avances de la última década en el campo de la Inteligencia Artificial (IA) han demostrado el potencial de estas nuevas técnicas como enfoque eficaz para resolver diferentes problemas clásicos en el área de la astrodinámica. Las estrategias de búsqueda de rutas en regiones dinámicas de sistemas multicuerpo en el espacio implican algunos desafíos que tradicionalmente se han basado en simplificaciones del modelo dinámico o cargas computacionales pesadas. Sin embargo, es probable que las técnicas modernas de aprendizaje automático proporcionen soluciones más ligeras que puedan aplicarse a los sistemas de navegación automatizada a bordo de futuras misiones que viajen a través de entornos complejos. Este trabajo tiene como objetivo aprovechar las técnicas de Redes Neuronales Artificiales (RNAs) para estudiar cómo estas se pueden aplicar a Orbitas de Puntos de Libración en el Problema Restringido de Tres Cuerpos. En particular, se proponen nuevos enfoques por medio de modelos de RNAs para calcular conexiones heteroclínicas entre órbitas planas de Lyapunov, y su rendimiento se compara con el de las metodologías clásicas. Para entrenar, validar y testear las RNAs utilizadas a lo largo del trabajo, se requieren conjuntos de datos suficientemente grandes que contengan información representativa y precisa sobre la dinámica del Problema Restringido de Tres Cuerpos. A tal efecto, los conjuntos de datos se generan mediante la resolución del problema utilizando las técnicas clásicas conocidas, ya que es la única forma disponible de adquirir un conjunto de observaciones precisas de lo que ocurre en modelo. Se siguen los siguientes pasos: primero, las conexiones heteroclínicas se calculan con una precisión arbitrariamente alta mediante la propagación numérica de las variedades inestables/estables de las órbitas de salida/llegada respectivamente, utilizando un integrador numérico de tipo Runge-Kutta 45; en segundo lugar, los datos de las propagaciones se tabulan y almacenan en conjuntos de datos; tercero, dichos conjuntos de datos se utilizan para entrenar y validar una serie de RNAs; y finalmente, se analiza el desempeño de los modelos de RNAs obtenidos y se compara con los resultados de referencia. Además, se estudia la influencia de la técnica de muestreo optada a la hora de generar los conjuntos de datos sobre el rendimiento del modelo, así como la robustez del método desarrollado. Como resultado de este trabajo, se puede concluir que los dos nuevos métodos de RNAs aquí desarrollados, pueden considerarse tan buenos como el método clásico en términos de su aptitud para ser utilizados como semilla en un modelo de efemérides realista, ya que en cualquier caso, el modelo Restringido de Tres Cuerpos utilizado aquí como referencia no es una representación precisa de las trayectorias realistas, aunque sí se considera una aproximación aceptable en las primeras etapas del diseño de misiones.

Title : An approach to some classical libration point orbit problems by means of artificial intelligence techniques
Author: Sara Cecilia Abecia Hernanz

Advisor: Josep Joaquim Masdemont Soler

Date: November 22, 2021

Abstract

Rapid developments in the last decade have demonstrated the potential of new techniques within the Artificial Intelligence (AI) field as an effective approach for different classical problems in the area of astrodynamics. Pathfinding strategies in multi-body dynamical regions of space implies some challenges which have traditionally relied on simplifications of the dynamical model or heavy computational loads. However, modern machine learning techniques are likely to provide more lightweight solutions that could be applied to automated on-board guidance systems in future missions traveling though complex environments. This work aims to leverage Artificial Neural Network techniques to study how they can be applied to Libration Point Orbits in the Circular Restricted Three-Body Problem. In particular, new ANN-approaches to compute heteroclinic connections between planar Lyapunov orbits are proposed, and their performance is compared to that of classic methodologies. In order to train, validate and test the ANNs used throughout the work, sufficiently large datasets containing information that accurately represents the dynamics of the Circular Restricted Three-Body Problem are required. To this effect, the datasets are generated by means of solving the problem using well-known classic techniques, since it is the only way to acquire an ensemble of accurate observations of the model. The following steps are taken: first, the heteroclinic connections are computed with arbitrarily high accuracy by numerical propagation of the unstable/stable invariant manifolds of the departure/arrival orbits respectively, using a Runge-Kutta 45 integrator; second, the data from the propagations are tabulated and stored in datasets; third, the datasets are used to train and validate a series of ANNs; and finally, the performance of the obtained ANN models is analysed and compared to the benchmark results. Also, the influence of the sampling technique when generating the datasets and the robustness of the developed method are studied. As a result of this work, it can be concluded that the two new ANN methods developed here can be considered as good as the classical method on terms their fitness to be used as a seed in a realistic ephemeris model, since the Restricted Three Body Problem model used here as a reference is also not an accurate representation of realistic trajectories anyways, but it is an acceptable approximation in early stages of mission design.

Nere eredu izan diren hiru emakumeei,
To my three female role models,

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The n-body problem consists of describing the temporal evolution of multiple bodies in space under the influence of their own gravitational forces. This work focuses on a particular model of this problem for the specific case of three bodies. Within this framework, the problem under study is the computation of heteroclinic connections between planar libration point orbits, which is a mathematical problem for which a closed solution exists if we take the simplifying hypotheses that one of the three bodies has negligible mass and that the trajectory of the other two bodies is restricted to circular motion. This simplified model is known as the Circular Restricted Three Body Problem (CRTBP), and it is used in this work as a benchmark to test the usefulness of new AI techniques in the field of celestial mechanics.

In nonlinear dynamical environments such as the CRTBP model, traditional trajectory design and optimization strategies rely on either simplifying assumptions in the dynamical model or on computationally-heavy processes. Thus, through the implementation of modern machine learning techniques a more lightweight solution is pursued, which has the potential to facilitate automated in-flight pathfinding and guidance in future onboard computers travelling within complex environments.

It is worth highligting that although the mathematical methods to solve the problems presented in this work already exist, it is not intended to replace them but to leverage the new tools that AI provides us with to complement these classic methods, since techniques such as ANNs have been proven to provide accurate solutions for other multi-body system problems at a fraction of the computational cost, and potentially 100 million times faster than other state-of-the-art solvers [27].

The main research publication that triggered the idea for this master thesis is the one carried by De Smet and Scheeres' about the identification of heteroclinic connections using ANNs [4], which inspired some of the approaches used in the present work. Regarding the method used to asses the feasibility of the new ANN technique as compared to the classical solution to the Lyapunov orbit problems, the workflow of the process has been inspired by the one presented in that work. However, the algorithm used to compute the heteroclinical connections has been adapted so that the Poincaré surface defined for that purpose is the same as the one used by Canalías in her work [3].

With regards to the mentioned workflow, it can be generally summarized as follows: first, the proposed problem is solved in the classical way, that is, propagating the state vector in the six-dimensional space (three coordinates in positions and three coordinates in velocities) by means of numerical integration. For this purpose, the Runge Kutta 45 algorithm is implemented throughout the entire work. Then, these solutions are saved in the form of tabulated datasets to be used for the training of the ANNs. Finally, in order to asses the performance of the obtained ANN models, a comparative analysis is carried out between the results predicted by the models and the actual expected responses that had been stored in the reference datasets. Furthermore, the influence of various factors on the predicting ability of the ANN is analysed, such as the sampling technique used to generate the training set or the random shuffle of the training and test data. Also, the robustness of the methods proposed here to develop the predictors is assessed by means of training an ensemble of equivalent models and verifying that the solutions are as well equivalent in terms of how large or small is the standard deviation of the performance indicators as compared to their mean.

The present work is divided into three chapters. The first one serves as a way of framing the problem by providing some concepts on the dynamics governing the libration regions. The second one is aimed at giving some insight on the Artificial Intelligence notions used in the development of this project. Lastly, this work ends with a summary of the conclusions drawn from the discussion of the results and a list of suggestions on how this research can be extended in future work.

CHAPTER 1. THEORETICAL FRAMEWORK OF LIBRATION ORBITS

1.1. The n-body problem

The well-known n-body problem in physics is the problem of solving the equations of motion of a group of celestial bodies under their own gravitational force in order to predict their individual trajectories. Rosenberg stated it as follows [21]:

> Given the quasi-steady orbital properties (instantaneous position, velocity and time) of a group of celestial bodies, predict their interactive forces; and consequently, predict their true orbital motions for all future times.

The n-body problem has no general solution due to its chaotic dynamics as discovered by Henri Poincaré in [19]. Nevertheless, there are two special cases for which the problem can be completely solved: the Two-Body Problem, and the Restricted Three Body Problem. The later forms the focus of this study and its formulation and dynamical equations are presented hereunder.

1.2. The Circular Restricted Three Body Problem

Multi-body dynamical regions in space can be challenging as the nonlinearity of the environment makes Keplerian-based approaches infeasible. For this work, the Circular Restricted Three Body Problem (often referred to in literature as CRTBP or just RTBP for short) will be employed to model the motion of a small mass moving under the influence of two main massive bodies that move in circular orbits around their center of mass. In this problem, the mass of the smallest body is reasonably neglected as its relative gravitational impact is assumed to be much less significant, and the equations are defined in a rotating frame of reference frame whose origin lies at the center of mass of the two large masses. This can be applied to different scenarios, e.g., a spacecraft traveling within the Cislunar space or the Earth-Sun system.

The equations of motion that describe the CRTBP are presented in the following section.

1.2.1. Equations of motion

According to Newton's law of universal gravitation and the second law of motion, for each particle of a system of n point-like masses, the gravitational force acting upon a particle of mass m_i is,

$$\mathbf{F} = \frac{\mathrm{d}}{\mathrm{d}t}\left(m_i\dot{\mathbf{r}}_i\right) = m_i\ddot{\mathbf{r}}_i = -G\sum_{j=1,j\neq i}^{n}\frac{m_i m_j}{\|\mathbf{r}_{ji}\|^3}\,\mathbf{r}_{ji} \tag{1.1}$$

where m_j is the mass of particle j (with $j = 1,...,n$) and G is the gravitational constant. Note that the notation used hereafter to represent a vector that points in the direction from particle j to particle i is $\mathbf{r}_{ji}$, meaning that $\mathbf{r}_{ji} = \mathbf{r}_i - \mathbf{r}_j$.

For the restricted three body problem, three objects are considered: two massive bodies called primaries and an object of negligible mass. The bigger primary is referred to as m_1 while the smaller one is m_2. Taking equation (1.1) and making that simplification, the system of equations that describes the problem results in,

$$\begin{cases} \ddot{\mathbf{r}}_1 = -Gm_2 \dfrac{\mathbf{r}_{21}}{\|\mathbf{r}_{21}\|^3} \\ \ddot{\mathbf{r}}_2 = -Gm_1 \dfrac{\mathbf{r}_{12}}{\|\mathbf{r}_{12}\|^3} \\ \ddot{\mathbf{r}}_3 = -Gm_1 \dfrac{\mathbf{r}_{13}}{\|\mathbf{r}_{13}\|^3} - Gm_2 \dfrac{\mathbf{r}_{23}}{\|\mathbf{r}_{23}\|^3} \end{cases} \tag{1.2}$$

where the two first equations in (1.3) are equivalent to the Keplerian equations of the primary masses and the third describes the restricted three body problem.

The equations are defined in the inertial frame and because they are independent of any particular frame, we chose the inertial frame fixed in space so that the orbital plane described by the motion of the primaries is contained in the $x_I y_I$ plane and the origin can be set to the barycenter, so that the distance from the barycenter to m_1 and m_2 are d_1 and d_2 respectively:

$$d_1 = \frac{\|\mathbf{r}_{12}\|}{1+\frac{m_1}{m_2}} \;;\; d_2 = \frac{\|\mathbf{r}_{12}\|}{1+\frac{m_2}{m_1}} \tag{1.3}$$

Now, supposing the two primaries describe circular orbits around the barycenter, a convenient change of frame that allows for the simplification of the equations for the CRTBP is to introduce a new frame of reference known as the synodic frame. This frame is rotating with constant angular velocity $\boldsymbol{\omega} = (0,0,\omega)^T$ with respect to the described inertial frame, so that its x axis is aligned with the line passing through the two primaries and pointing at the big primary, m_1. Taking the inertial acceleration in the third equation of (1.3) that describes the motion of the particle of negligible mass and referring to it as $\ddot{\mathbf{r}}_3^{\textcircled{I}}$, the expression that relates it with the new acceleration seen from the rotating synodic frame, $\ddot{\mathbf{r}}_3$ is,

$$\ddot{\mathbf{r}}_3^{\textcircled{I}} = \ddot{\mathbf{r}}_3 + 2\boldsymbol{\omega} \times \dot{\mathbf{r}}_3 + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_3) \tag{1.4}$$

In the above expression the Coriolis and the centripetal accelerations can be calculated as,

$$2\,\boldsymbol{\omega} \times \dot{\mathbf{r}}_3 = 2\,\omega \begin{bmatrix} \dot{y} \\ -\dot{x} \\ 0 \end{bmatrix} \tag{1.5}$$

$$\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_3) = \omega^2 \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \tag{1.6}$$

where x, y and z denote the position of m_3 in synodic coordinates. Thus, taking $\ddot{\mathbf{r}}_3 = (\dot{x},\dot{y},\dot{z})^T$ and substituting (1.4) in (1.3), one can arrive to,

$$\begin{cases} \ddot{x} - 2\,\omega\dot{y} - \omega^2 x = -Gm_1\,\dfrac{x-x_1}{\|\mathbf{r}_{13}\|^3} - Gm_2\,\dfrac{x-x_2}{\|\mathbf{r}_{23}\|^3} \\[2ex] \ddot{y} + 2\,\omega\dot{x} - \omega^2 y = -Gm_1\,\dfrac{y}{\|\mathbf{r}_{13}\|^3} - Gm_2\,\dfrac{y}{\|\mathbf{r}_{23}\|^3} \\[2ex] \ddot{z} = -Gm_1\,\dfrac{z}{\|\mathbf{r}_{13}\|^3} - Gm_2\,\dfrac{z}{\|\mathbf{r}_{23}\|^3} \end{cases} \tag{1.7}$$

where $(x_1,0,0)$ and $(x_2,0,0)$ correspond to the positions of the primaries in synodic coordinates.

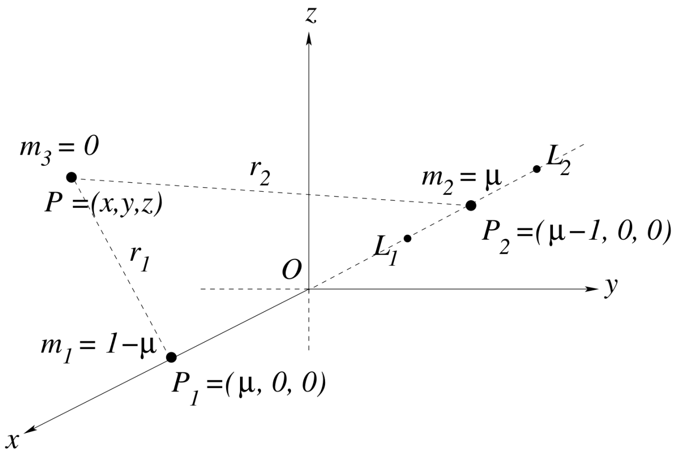


Figure 1.1: The synodic frame of reference

Taking adequate units of mass, length and time in (1.7) the equations of motion can be simplified in such a way that the distance between the primaries ($\|\mathbf{r}_{12}\|$), the angular velocity (ω) and the sum of the masses of the primaries (m_1+m_2) become unitary. With these changes, the gravity constant (G) equals one, while the mass of the primaries are μ for the smaller and $1-\mu$ for the bigger one. As a result the small and big primaries remain at (μ-1,0,0) and (μ,0) respectively (see figure 1.1). The resulting equations, which are often the starting point when the CR3BP is described in literature, are the following:

$$\begin{cases} \ddot{X} - 2\,\dot{Y} = \Omega_X \\ \ddot{Y} + 2\,\dot{X} = \Omega_Y \\ \ddot{Z} = \Omega_Z \end{cases} \tag{1.8}$$

where Ω_X, Ω_Y and Ω_Z denotes the derivative of the following function with respect to X, Y and Z respectively,

$$\Omega\,(X,Y,Z) = \frac{1}{2}\left(X^2+Y^2\right) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{1}{2}\mu\,(1-\mu) \tag{1.9}$$

and,

$$r_1{}^2 = \left(X-\mu\right)^2 + Y^2 + Z^2 \tag{1.10}$$

$$r_2{}^2 = \left(X-\mu+1\right)^2 + Y^2 + Z^2 \tag{1.11}$$

For the numerical implementation of the problem, it is convenient to have the vectorial form of the equations, so the procedure is explained here below. On the other hand, by doing a change of variables, it is possible to reduce the system of three second-order differential equations to a system of six first-order differential equations:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \rightarrow \dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix} \tag{1.12}$$

As a result the problem can be defined by six first-order differential equations.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \begin{cases} \dot{x_1} = x_4 \\ \dot{x_2} = x_5 \\ \dot{x_3} = x_6 \\ \dot{x_4} = \Omega_{x_1}(x_1, x_2, x_3) + 2x_5 \\ \dot{x_5} = \Omega_{x_2}(x_1, x_2, x_3) - 2x_4 \\ \dot{x_6} = \Omega_{x_3}(x_1, x_2, x_3) \end{cases} \tag{1.13}$$

1.2.2. First integral of the CRTBP and zero velocity curves

The first integral of the Circular Restricted Three Body Problem, which is known as the Jacobi constant and is noted as C_J hereunder, is derived from the previous equations by following the steps below:

- First, the first, second and third equations of the CRTBP system in (1.8) are multiplied by $\dot{X}$, $\dot{Y}$ and $\dot{Z}$ respectively, and adding all three results in,

$$\left(\ddot{X} - 2\dot{Y} - \Omega_X\right)\dot{X} + \left(\ddot{Y} + 2\dot{X} - \Omega_Y\right)\dot{Y} + \left(\ddot{Z} - \Omega_Z\right)\dot{Z} = 0$$

- Second, equation (1.2.2.) is simplified so that it yields,

$$\ddot{X}\dot{X} - 2\dot{Y}\dot{X} - \Omega_X\dot{X} + \ddot{Y}\dot{Y} + 2\dot{X}\dot{Y} - \Omega_Y\dot{Y} + \ddot{Z}\dot{Z} - \Omega_z\dot{z} = 0$$

$$\frac{\mathrm{d}\left(\frac{\dot{X}^2}{2}\right)}{\mathrm{d}t} - \cancel{2\dot{Y}\dot{X}} - \frac{\partial\Omega}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}t} + \frac{\mathrm{d}\left(\frac{\dot{Y}^2}{2}\right)}{\mathrm{d}t} + \cancel{2\dot{X}\dot{Y}} - \frac{\partial\Omega}{\partial Y}\frac{\mathrm{d}Y}{\mathrm{d}t} + \frac{\mathrm{d}\left(\frac{\dot{Z}^2}{2}\right)}{\mathrm{d}t} - \frac{\partial\Omega}{\partial Z}\frac{\mathrm{d}Z}{\mathrm{d}t} = 0$$

$$\frac{\mathrm{d}\left(\frac{\dot{X}^2}{2} + \frac{\dot{Y}^2}{2} + \frac{\dot{Z}^2}{2}\right)}{\mathrm{d}t} - \frac{\mathrm{d}\Omega}{\mathrm{d}t} = 0$$

- Third, (1.2.2.) is derived in order to obtain the first integral,

$$\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2 - 2\,\Omega = cte. \tag{1.14}$$

Rearanging the terms, the Jacobi constant, C_J, can be expressed as:

$$C_J = 2\,\Omega\,(X,Y,Z) - \dot{X}^2 - \dot{Y}^2 - \dot{Z}^2 \tag{1.15}$$

Although this definition is not unique and one might find different expressions for the Jacobi constant in literature ([4],[3]) by taking the convention in 1.15 the C_J equals the exact value of 3 at the two triangular libration points, L_4 and L_5.

1.2.3. Equilibrium points

We refer to the equilibrium points of the problem as the points where the velocity and the acceleration of a particle at that given position equals zero so that the particle would never change its state. Or equivalently, when the vectorial equations of the problem (1.13) are equal to zero.

$$\begin{aligned} \dot{X} &= \ddot{X} = 0 \\ \dot{Y} &= \ddot{Y} = 0 \\ \dot{Z} &= \ddot{Z} = 0 \end{aligned} \tag{1.16}$$

That implies that the derivatives of Ω must also equal zero at the equilibrium points; thus, one shall find the solution for the equilibrium points of the CR3BP by solving the following system of equations:

$$\begin{cases} \Omega_X = 0 = X - \dfrac{1-\mu}{{r_1}^3}(X-\mu) - \dfrac{\mu}{{r_2}^3}(X+1-\mu) \\ \Omega_Y = 0 = Y - \dfrac{1-\mu}{{r_1}^3}Y - \dfrac{\mu}{{r_2}^3}Y \\ \Omega_Z = 0 = -\dfrac{1-\mu}{{r_1}^3}Z - \dfrac{\mu}{{r_2}^3}Z \end{cases} \tag{1.17}$$

It is known that five solutions exist for system (1.17), which are known as libration points or Lagrange points (see figure 1.2). It is easy to see that all of them must be contained in the $Z=0$ plane in order to satisfy the third equation of the system. Three of these libration points are located in the synodic $Y=Z=0$ axis, which are known as the collinear points and are commonly denoted by L_1, L_2 and L_3, while the last two are known as triangular points and they are often referred to as L_4 and L_5. For the scope of this work, we focus on the collinear points of equilibrium, whose positions are calculated in a similar manner, as explained below. To this effect, the first equation in (1.17), which only depends on the X coordinate, is studied. Because of its non-linear nature, its changes in sign across the X axis are studied, which means there are three intersections of function $f(X)=\frac{\partial\Omega}{\partial X}$ with that axis, since the function is continuous for every value of X other than μ and $\mu-1$ (where the two primaries lie).

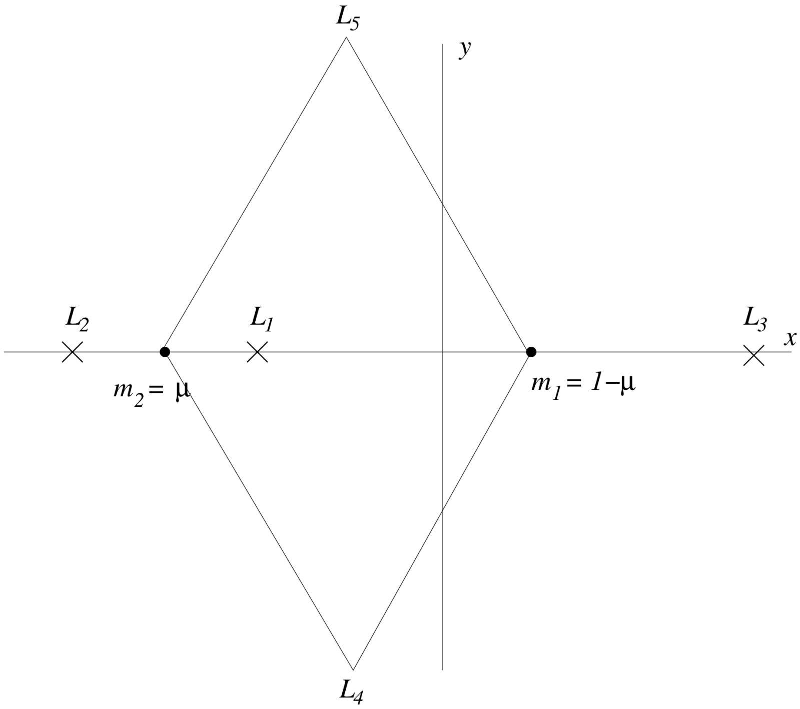


Figure 1.2: The five libration points of the Circular Restricted Three Body Problem

The equilibrium solution at $X < \mu - 1$, which corresponds to L_2, its position can be calculated by introducing a new variable for the distance between L_2 and the small primary: γ. By doing so, $X = \mu - 1 - \gamma$ and the first equation in (1.17) can be expressed as the following Euler's quintic equation (the quintic equations presented in this section as well as their solutions can be found in [24]):

$$\gamma^5 + (3-\mu)\gamma^4 + (3-2\mu)\gamma^3 - \mu\gamma^2 - 2\mu\gamma - \mu = 0 \tag{1.18}$$

whose solution can be written as,

$$\gamma = \left(\frac{\mu}{3}\right)^{\frac{1}{3}} + \frac{1}{3}\left(\frac{\mu}{3}\right)^{\frac{2}{3}} + \frac{1}{9}\left(\frac{\mu}{3}\right)^{\frac{3}{3}} + \ldots \tag{1.19}$$

As for the equilibrium solution at $\mu - 1 < X < \mu$, which corresponds to L_1, its position can be calculated by introducing the new variable γ, this time representing the distance between L_1 and the small primary. Now $X = \mu - 1 + \gamma$ and the first equation in (1.17) can be expressed as following Euler's quintic equation:

$$\gamma^5 - (3-\mu)\gamma^4 + (3+2\mu)\gamma^3 - \mu\gamma^2 + 2\mu\gamma - \mu = 0 \tag{1.20}$$

whose solution can be written as,

$$\gamma = \left(\frac{\mu}{3}\right)^{\frac{1}{3}} - \frac{1}{3}\left(\frac{\mu}{3}\right)^{\frac{2}{3}} - \frac{1}{9}\left(\frac{\mu}{3}\right)^{\frac{3}{3}} + \ldots \tag{1.21}$$

Lastly for L_3, which is located at $\mu < X$, it is calculated by introducing the variable γ so that the distance between the big primary and the equilibrium point is γ; that is, the point lies in $X = \mu + 1 + \gamma$. This way, the first equation in (1.17) can be expressed as the following Euler's quintic equation:

$$\gamma^5 - (7+\mu)\gamma^4 + (19+6\mu)\gamma^3 - (24+13\mu)\gamma^2 + (12+14\mu)\gamma - 7\mu = 0 \tag{1.22}$$

whose solution can be written as,

$$\gamma = \frac{7}{12}\mu + \frac{237^2}{12^4}\mu^3 + \ldots \tag{1.23}$$

For each collinear point, the only real root of Euler's quintic equation provides the distance to its closest primary. In the code, its value is numerically computed with high accuracy using Newton's method with the following seed values,

$$\begin{aligned} L_1, L_2 \rightarrow \gamma &= \left(\frac{\mu}{3}\right)^{\frac{1}{3}} \\ L_3 \rightarrow \gamma &= \frac{7\mu}{12} \end{aligned} \tag{1.24}$$

which correspond to the first term of the power series solutions (see equations (1.19), (1.21) and (1.23)).

1.3. Libration orbits

Extensive work has been done in the last decades in order to study the dynamics of the regions surrounding the collinear libration points (see for example [3], [14], [9] and [10]), resulting in effective methods to compute various types of libration point orbits near the collinear points, such as planar and vertical Lyapunov orbits, Lissajous, halo and quasi-halo orbits.

1.3.1. Stability of collinear libration points L_1 and L_2

The CR3BP is a Hamiltonian dynamical system, and although it is not linear, its equations can be linearised around the equilibrium points in order to study their stability, since given matrix B of the linearised problem $\dot{x} = Bx$, the eigenvalues of that matrix of constant values provide a means for analysing the stability.

As the first step to linearising the problem around the libration points that are of interest for this work, that is, L_1 and L_2, the following change of coordinates consisting of a translation with homothety, is applied to the equations in (1.7):

$$\begin{aligned} x &= -\frac{1}{\gamma}\left(X - \mu + 1 \mp \gamma\right) \\ y &= -\frac{1}{\gamma}Y \\ z &= \frac{1}{\gamma}Z \end{aligned} \tag{1.25}$$

With these new coordinates the equations in (1.7) become,

$$\begin{cases} \ddot{x} - 2\,\dot{y} = \dfrac{1}{\gamma^2}\,\Omega_x \\ \ddot{y} + 2\,\dot{x} = \dfrac{1}{\gamma^2}\,\Omega_y \\ \ddot{z} = \dfrac{1}{\gamma^2}\,\Omega_z \end{cases} \tag{1.26}$$

where the transformation in (1.25) is also applied to the $\Omega(X,Y,Z)$ function, which becomes a function of the new variables: $\Omega(x,y,x)$.

Recalling what was mentioned in the beginning of this section, with the aim of studying the stability of the system in the libration regions, the equations of the problem need to be linearised. Taking the linearised part of the equations one can arrive to the linearised equations around the libration points L_1 and L_2, which can be written as follows (see [14]),

$$\begin{cases} \ddot{x} - 2\dot{y} - (1+2c_2)x = 0 \\ \ddot{y} + 2\dot{x} - (c_2-1)y = 0 \\ \ddot{z} + c_2 z = 0 \end{cases} \tag{1.27}$$

where c_2 is a constant that can be computed as follows, taking the positive signs in the equation for L_1 and the negative signs for L_2,

$$c_2 = \frac{1}{\gamma^2}\left(\frac{\gamma^2\,(1-\mu)}{|-1 \pm \gamma|^3} + (\pm\,1)^2\,\frac{\mu}{\gamma}\right) \tag{1.28}$$

where γ is the distance from the libration point to its closest primary (see section 1.2.3.).

The solution of the linearised system (1.27) is then,

$$\begin{cases} x(t) = A_1\,e^{\lambda t} + A_2\,e^{-\lambda t} + A_3\,cos\,\omega t + A_4\,sin\,\omega t \\ y(t) = cA_1\,e^{\lambda t} - cA_2\,e^{-\lambda t} - \kappa A_4\,cos\,\omega t + \kappa A_3\,sin\,\omega t \\ z(t) = A_5\,cos\,\nu t + A_6\,sin\,\nu t \end{cases} \tag{1.29}$$

where the A_i are constants that can take any arbitrary value and λ ω, ν, c and κ are constants that depend on c_2 through the following relations (1.27):

$$\begin{aligned} \lambda &= \sqrt{\frac{c_2-2+\sqrt{9c_2^2-8c_2}}{2}} \\ \omega &= \sqrt{\frac{2-c_2+\sqrt{9c_2^2-8c_2}}{2}} \\ \nu &= \sqrt{c_2} \\ c &= \frac{\lambda^2-1-2c_2}{2\lambda} \\ \kappa &= \frac{-(\omega^2+1+2c_2)}{2\omega} \end{aligned} \tag{1.30}$$

From the first terms of the left-hand side of the first and second equations in (1.29), it is evident that points L_1 and L_2 are unstable points of the problem. This instability of the libration regions around L1 and L2 is precisely the property that justifies the existence of hyperbolic invariant manifolds, which we use in order to find orbits that asymptotically approach or leave the regions around these points: the heteroclinic connections.

1.3.2. Planar Lyapunov orbits

The type of libration orbits subject of study of the present work are the planar Lyapunov orbits. Since these orbits oscillate within the xy plane, we consider $z = \dot{z} = 0$ from this point onwards. Assuming we are arbitrarily close to libration points L_1 or L_2, an approximate solution of the planar Lyapunov orbits can be obtained from the solution of the linearised problem presented in (1.29). For we are interested in finding periodic orbits, coefficients A_1 and A_2 are set to zero so that the solution only contains the part resulting from purely imaginary eigenvalues. In doing so, the linearised solution becomes,

$$\begin{aligned} x(t) &= A_3 \, cos \, \omega t \, + \, A_4 \, sin \, \omega t \\ y(t) &= -\kappa A_4 \, cos \, \omega t \, + \, \kappa A_3 \, sin \, \omega t \end{aligned} \tag{1.31}$$

For every fixed value of the Jacobi constant, C_J, only one planar Lyapunov orbit exists around each libration point L_1 and L_2, respectively. In this work, we look for the connections that provide a free transfer between these two regions by leveraging the hyperbolic character of the libration orbits.

From the linearized solutions and continuation and numerical refinement methods, the planar Lyapunov orbits of the restricted problem can be computed (see section 3.2.1.1.).

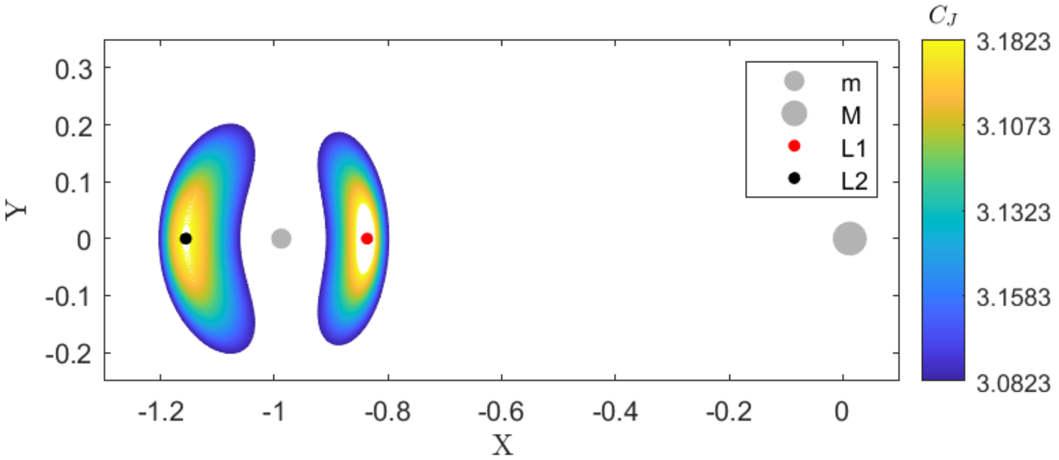


Figure 1.3: Planar Lyapunov orbits for different values of C_J. The big primary is denoted by M and the small primary by m.

Some planar Lyapunov orbits that were computed for a set of C_J values the interval of $C_J \in [3.1823, 3.1133]$ are shown in figure 1.3. a partir de las soluciones linealizadas y metodos de continuacion y refinado numerico, obtenemos las órbitas de lyapunov del problema restringido que se estan representadas en la figura x (ver seccion x)

1.4. Heteroclinic connections

As aforementioned, because of the inherent instability of the regions around L_1 and L_2, stable and unstable directions exist at each point of their surrounding orbits. The set of asymptotic orbits that either approach or leave the periodic orbits forms the hyperbolic invariant manifold of the orbit.

1.4.1. Hyperbolic invariant manifolds

The stable manifolds of an orbit are the ones that asymptotically approach the orbit, while the unstable manifolds asymptotically leave the obit. These can be easily computed using the monodromy matrix. The monodromy matrix of an orbit, which we denote as M, is defined as the state transition matrix of the orbit after one period (see [22]). While the eigenvector associated with the eigenvalue of the monodromy matrix of smallest modulus (referred to as $\mathbf{v}_u$) defines the unstable direction of the manifolds, the eigenvector associated with the eigenvalue of greatest modulus (referred to as $\mathbf{v}_s$) defines the stable direction. Thus, by computing the eigenvectors of the monodromy matrix, the initial states of the unstable and stable manifolds, x_u and x_s, can be computed from an initial point within the libration orbit, x_i [4], by using the following equations,

$$\begin{aligned} \mathbf{x}_u &= \mathbf{x}_i + \epsilon \mathbf{v}_u \\ \mathbf{x}_s &= \mathbf{x}_i + \epsilon \mathbf{v}_s \end{aligned} \tag{1.32}$$

where ϵ represents a small deviation from the orbit to the initial state of the manifold displacement along the stable or unstable direction. In our case where the Moon-Earth system is studied, the non-dimensional value of the selected ϵ is 10^{-5}.

As mentioned above, these hyperbolic invariant manifolds are useful since they provide a way of connecting the regions around L_1 and L_2 by means of heteroclinic connections, which can be seen as a trajectory between two orbits with the same Jacobi constant. Heteroclinic connections can be described asymptotic orbits that going forward in time, tend to a periodic orbit and going backward in time, tend to another periodic orbit. If the departure and arrival orbits are the same, then the connection is called homoclinic. These connecting trajectories can be computed by intersecting the hyperbolic invariant manifolds of the orbits: the unstable manifolds of the departure orbit and the stable manifolds of the arrival orbit. To find the intersections between the manifolds, the concept of the Poincaré section is introduced.

1.4.2. Poincaré sections

Poincaré sections are very useful to study dynamical systems since they provide a way to reduce the dimensionality of the problem, thereby making it easier to observe certain behaviours of the systems. The way they are used is by choosing a well defined surface and observing where the trajectories fall when crossing the surface. Because we will be studying the connections between L_1 and L_2 a convenient surface for this problem is taking a section that lies between the two libration points. Thereby, the Poincaré section that is used throughout this work is defined as,

$$X = 1 - \mu \tag{1.33}$$

that is, the vertical section at the X coordinate of the small primary is selected to study the crossings. As an example, figure 1.4 illustrates the manifolds arising from the planar Lyapunov orbit around L_2 (in green, stable) and L_1 (in red, unstable) stopping at their first crossing with the defined Poincaré section.

The cuts between the Poincaré section and the manifolds result in curves when looked at from the $Y\dot{Y}$ plane; that is why hyperbolic invariant manifolds can be referred to as tubes.

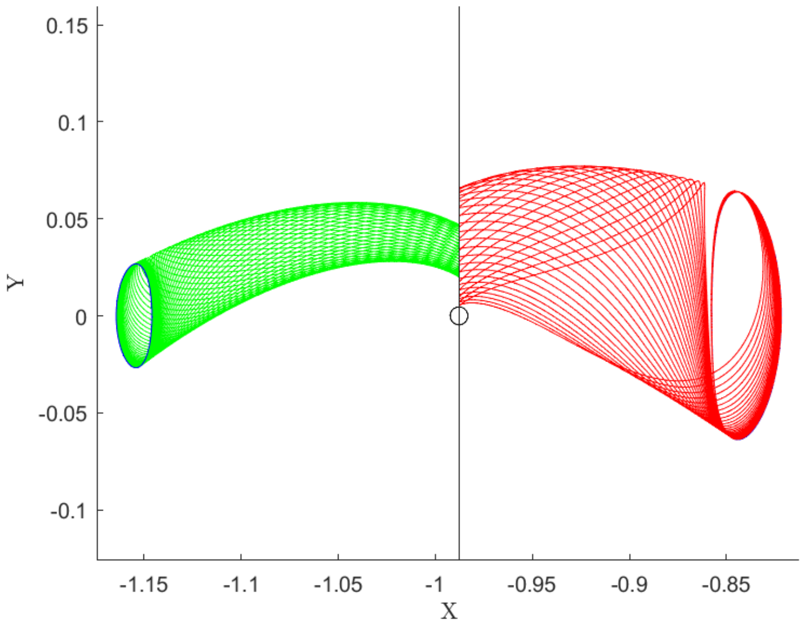


Figure 1.4: Stable (green) and unstable (red) manifolds of planar Lyapunov orbits intersecting with Poincaré section

Each point of these curves completely defines the state in $\mathbb{R}^4$ at that point (being X, Y, $\dot{X}$ and $\dot{Y}$ the state variables of the planar problem) since the X coordinate is given by the Poincaré surface and the Jacobi constant has been fixed to a certain value, which provides a relation among the state variables (see equation (1.2.)). Regarding the manifolds shown in figure 1.4, the curves in the $Y\dot{Y}$ plane formed by their corresponding crossings in the Poincaré section can be seen in figure 1.5.

As will be seen in detail further down, the curves of the stable and unstable manifolds might or may not intersect. When they do, each one of the intersections between the two curves represents an heteroclinic connection for the given value of the Jacobi energy. Some examples of intersecting and non-intersecting curves in the Poincaré section can be seen in figure 3.8 of Chapter 3.

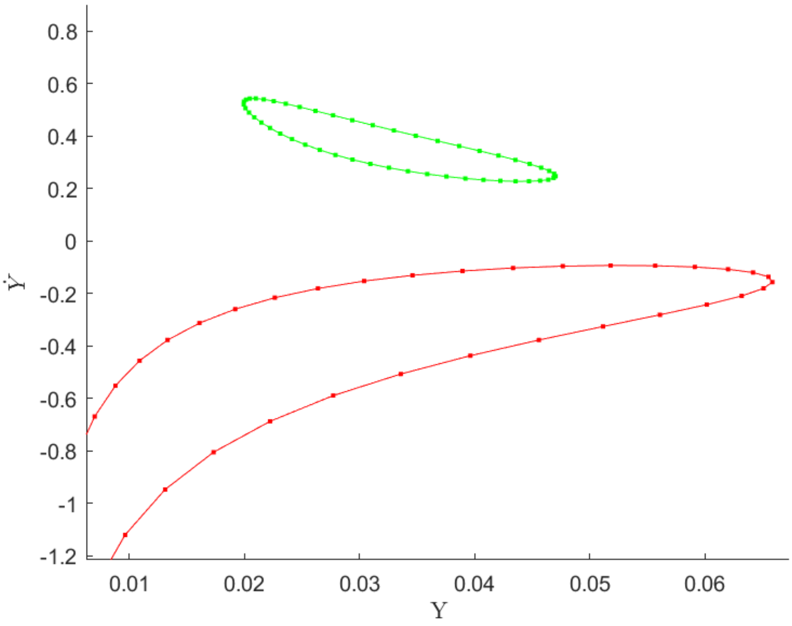


Figure 1.5: Crossings between the Poincaré section and the stable (green) and unstable (red) manifods of planar Lyapunov orbits around L_2 and L_1, respectively

CHAPTER 2. A BRIEF INTODUCTION TO ARTIFICIAL INTELLIGENCE

New advancements are proving AI to be a promising approach for solving different problems in astrodynamics, and because the capability to implement ANN on a flight computer is under active development, it has the capability to bring new opportunities for forthcoming sustained presence in Cislunar space. Among these techniques, recent investigations that focus in multi-body dynamical regimes with application to LPOs include heteroclinic transfer identification [4], trajectory optimization and guidance for low-thrust Electric Propulsion [23, 16] and station-keeping [11]. These investigations have produced desirable outcomes while arising new questions for future work, some of which will be explored in this work, such as the viability of AI techniques used to find low energy transfers between orbits.

The present chapter is used a brief introduction for the reader to Artificial Intelligence, particularly focusing on giving insight into some Artificial Neural Network concepts, the working principles behind them and some notions on how to use them.

2.1. Artificial Intelligence

Artificial Intelligence, also known as AI, can be defined as the ability of machines to perform a variety of tasks that imitate the use of reasoning in the human brain. That is, the theory and development of computer systems that perform using perception, classification and continuous learning logic to improve and later interact with reality [6].

Before continuing, a concept of great importance that emerges from what we define as intelligence is the notion of learning. It can be considered as the gain of knowledge by means of reasoning beginning with some data and based on previous experiences to predict information and make decisions about some particular task, which ends up developing wisdom [7].

Thanks to the new era of big data, rapidly increasing computational performance, and new machine learning algorithms, many new AI tools have emerged in recent years with the objective to better understand and interpret our environment. These new techniques that are constantly been created and perfected are part of everything that currently constitutes the branches of artificial intelligence [8]. These are:

- Neural Networks
- Machine Learning (ML)
- Deep Learning
- Computer vision
- Natural Language Processing (NLP)
- Natural Language Generation (NLG)
- Recommender Systems
- Predictive Analysis

On the other hand, AI is also often classified in a hierarchical way. As is illustrated in figure 2.1, the first layer in Artificial Intelligence is known as Machine Learning. This is the ability of machines and programs to learn by imitating human behaviour and it can be considered as a subset of AI techniques, that is composed of algorithms that process and classify data in order to find patterns. One of its biggest advantages as compared to traditional programming techniques is that it does not require being explicitly programmed for a particular problem. The next layer contained within Machine Learning, is Artificial Neuron Networks (ANNs) which is inspired in the human brain neurons and how these are able to process information [13]. Finally within ANN, we can find the set of techniques known as Deep Learning, which make use of complex multi-layer networks.

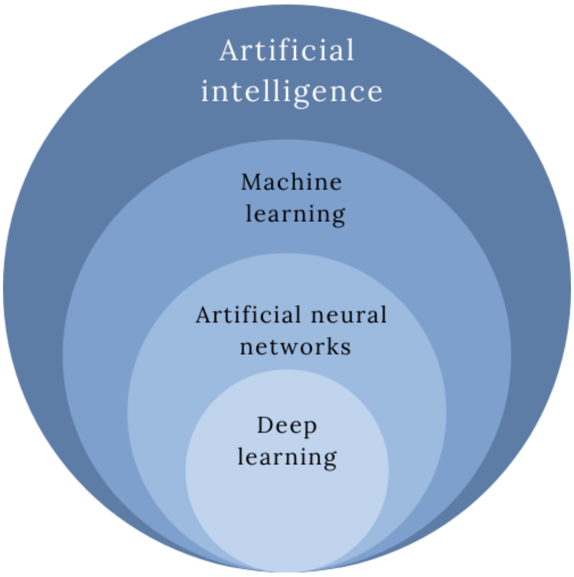


Figure 2.1: Taxonomy of artificial intelligence

It is commonly misunderstood that any of these AI techniques can be totally autonomous in a way that they could generate knowledge by themselves. The reality is, however, that they still depend on humans to a very large extent, on the one hand to be fed with new useful information and on the other hand, to optimize the algorithms for the target problem. For this reason, AI must be understood as a powerful tool that we can leverage to gain new insight on the problem, but it is not a self-contained alternative solution in most cases.

For this work, we focus on Artificial Neural Network techniques, which are explained in detain in the next section.

2.2. Artificial Neural Networks (ANN)

Haykin defines ANN as follows [13],

> A neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest; the network is usually implemented by using electronic components or is simulated in software on a digital computer.

The term Artificial Neural Network was originally motivated by the idea of modeling networks of real biological neurons in the brain. Nowadays, it is known that the artificial neurons used to construct artificial neural networks are very primitive approximations to those found in the brain. Therefore, the neural network architectures used are also extremely simple approximations of the brain structure. However, the remarkable progress has been made during the past decades inspired by this neuro-biological analogy. Biological neurons receive information from other neurons through the dendrites to then process the information in a cell body called *soma*. In order to send information, they have a connecting element called an axon. The point of connection between the axon branches and the dendrites of other neurons are called synapses. The information generated has the form of a spike in the electric potential (voltage) of the axon. Depending on the particular connections of the neurons, the information sent by one neuron can excite or inhibit other neurons.

All of this is possible because the brain contains billions of these information processing units that are connected each to other forming complex networks. Although the capacity to work with information is limited for only one neuron, their true power comes across when they are interconnected with each other.

2.2.1. Basic elements of the neural model

For a better understanding of ANNs, the components that constitute them are defined and explained in this section.

As we can see in figure 2.2, a basic neuron is formed by six main elements: the input signals, the synaptic weights, the bias, the activation function and the output signal. The n input signals are referred to as $x_1, x_2, ...x_n$ and to each one of them corresponds a synaptic weight, accordingly referred to as $w_1, w_2, ...w_n$ which can be qualitatively seen as the force or implicate grade of connection between two neurons.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.1}$$

The third factor is the summing junction (2.1), that calculates the inputs reading with the bias b_k to transmit the data to the activation function to finally give an output (2.3) as a result.

$$y_k = \varphi(u_k + b_k) \tag{2.2}$$

$$y_k = \varphi(v_k) \tag{2.3}$$

2.2.1.1. Synapses

The weights between neurons, the summing junction and the activation functions, altogether, model whether neurons excite or inhibit each other. As mentioned above, we described to the synapses in biological terms, which refers to a small structure in the neuron which has the main purpose of transmitting nervous impulses in an electrical or chemical

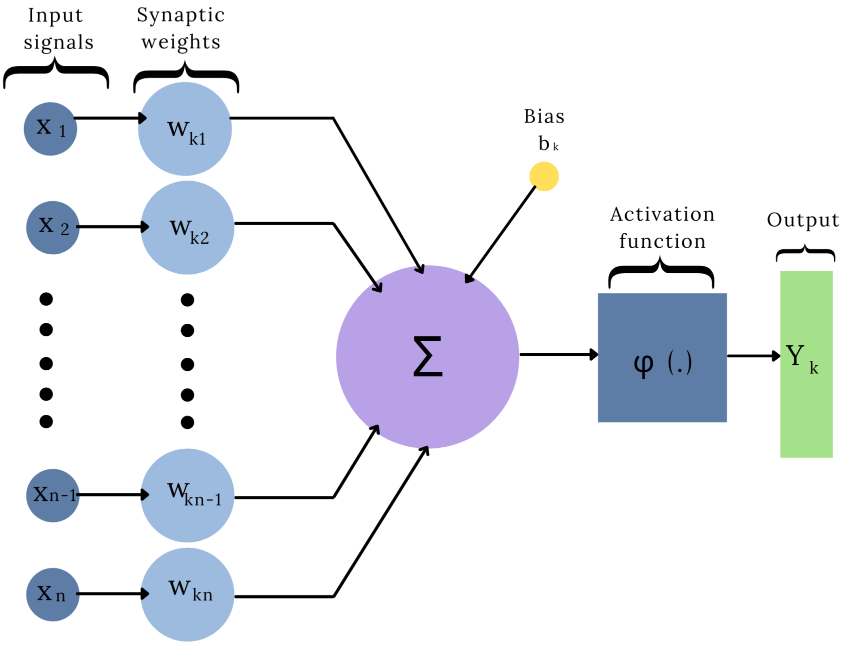


Figure 2.2: Schematic representation of the model of a neuron

way to other neurons. In other words, they constitute the bridge between every pair of connected neurons in the system. For that reason, synapses play a fundamental part in our neural systems and without them, we would not be able to communicate, process or memorize information.

In a similar way, in Artificial Neural Networks the synapses acts as a connecting link characterized by a weight or strength [13] that controls the relevance of the signals coming from the inputs of the system (or from the outputs of other neurons). The synaptic weight of neuron k related to input x_n is denoted as w_{kn} and it can take a range of positive or negative values that depend on its degree of connection within the neural network.

2.2.1.2. Summing junction

This is the element of the neuron that performs the weighed sum of all the input signals. If the sum results in a value equal to zero, the bias can be added in order to make the output non-zero or to scale up for a different response in the system.

2.2.1.3. Activation function

The activation function provides a way to control the desired output of a neuron. This means that the activation function enables us to limit the amplitude of the outputs to convenience.

The activation function of a neuron k, is denoted by $\varphi(v_k)$ (see equation (2.3)). Also according to [13] activation functions can be classified into two different types:

- **Types of Activation Function**

- **Threshold Function:** As can appreciated in figure 2.3, a threshold function acts like a logical Boolean, that is, it can take one of two values: either one or zero. The value of v that appears in (2.4) is known as the threshold gate.

$$\varphi(x) = \begin{cases} 0 & if \quad v > x \\ 1 & if \quad v \geq 0 \end{cases} \tag{2.4}$$

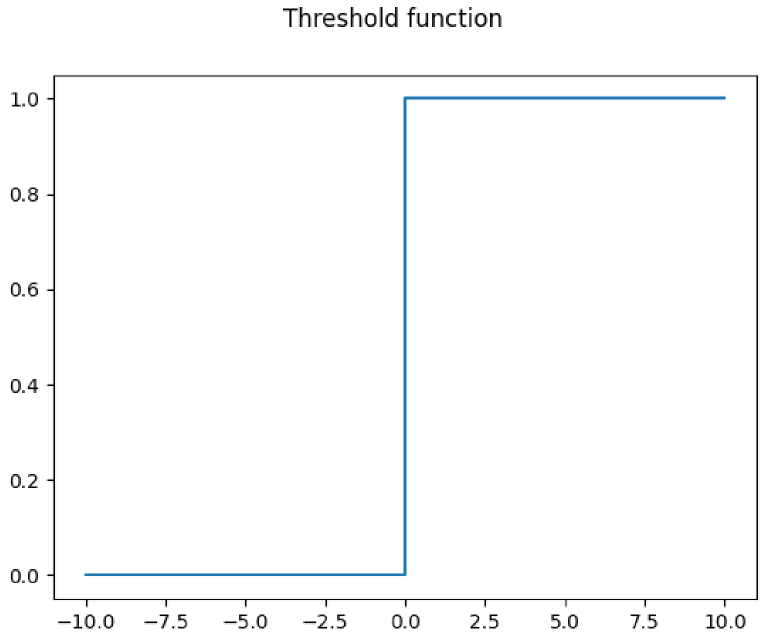


Figure 2.3: Threshold Function

- **Sigmoid Function:** A sigmoid function has a non-negative derivative at every point and is also known as an S-shape function. The main purpose of this function is to convert real values into probabilities. It can written as follows [13],

$$\varphi = \frac{1}{1+e^{(-ax)}} \tag{2.5}$$

  where a is the slope parameter of the sigmoid function and x is the the synapse.

  Out of the many functions of this type, one of the most popular ones for its use in ANNs is the logistic function. However, depending on the problem, another type of function might be more appropriate in order to find the perfect balance between linear and nonlinear behavior in our neural network system.

  The first derivative of all sigmoid functions is bell-shaped and monotone. There are several sigmoid functions, and some of the most well-known are included here.

- **Some examples of sigmoid activation functions:**
  - Logistic function

$$\varphi(x) = \frac{1}{1+e^{-x}} \tag{2.6}$$

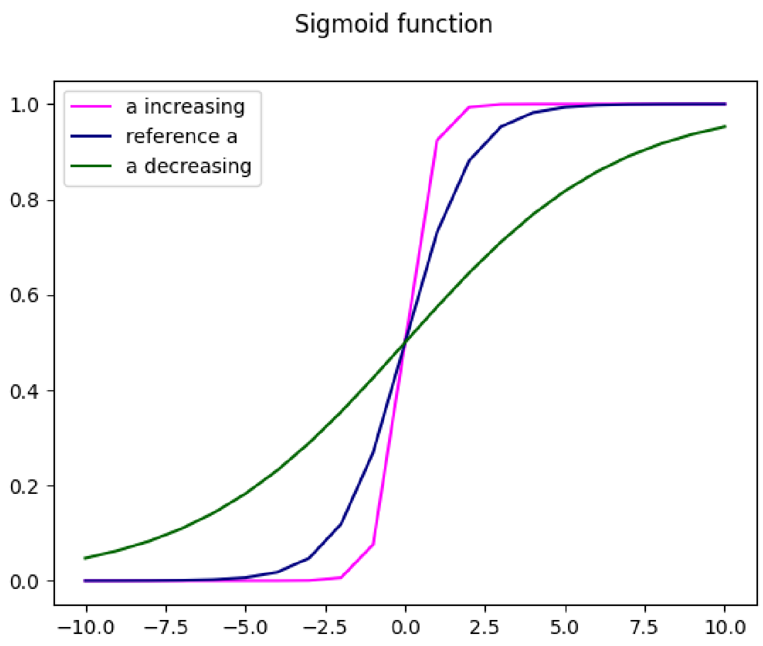


Figure 2.4: Sigmoid Function

- ⋆ Hyperbolic tangent function

$$\varphi(x) = tanh(x) \tag{2.7}$$

- ⋆ Arctangent function

$$\varphi(x) = arctan(x) \tag{2.8}$$

2.2.1.4. Bias

Depending on whether it is positive or negative, this parameter provides a way to increase or decrease the net input of the activation function.

2.2.2. Arquitecture of ANNs

Once the elements that conform a single neural network have been presented and explained, it is interesting to know the different ways in which they can interact with each other, since their true potential lies when they work in the form of networks. For this reason, some structures or architectures are shown in this section that allow the user to have a certain level of performance according to the level of complexity, size and organization of the network.

The connection between neurons through nodes and edges, describes the topology of a neural network. Each layer can be connected to different activation functions, thus increasing the complexity of the network. The layers can be stacked together in order for each layer to perform sequential processing. These types of networks are referred to as *deep* neural networks.

As explained in [13], different types of networks exist, which can be classified in the sections below.

Feedforward networks

On the one hand, feedforward networks are named in such manner because the information from the input units is transmitted to the next layer units. Then, new output is calculated for the next layer and the output is again passed to the next layer units. This is repeated again and again until the information reaches the output.

2.2.2.1. Single-Layer feedforward networks

If we have a network that is arranged with all the connections going in the same direction and without any cycles, it is known as a feed-forward neural network.

2.2.2.2. Multilayer Feedforward Networks

With this kind of NN we can observe that their architecture is composed by an input layer that receives the inputs to transform and prepare them to enter in the next layer.

Second, there are what are called the Hidden layers, that are the intermediate layers which do all the computations and extract features form the data.

Finally, there is an output layer that takes all the information processed by thee previous hidden layers and computes a final prediction or classification based in the learning models.

Recurrent Networks

The recurrent neural networks are distinct from the other networks because they have the presence of feedback loops and that produces a significant impact on the learning capability and performance in the network. As a result of the loops, this architecture has a nonlinear dynamic behavior in all the system.

2.2.3. Feedback

From the observation of how biological neural networks work, it is understood that biological ANNs are also dynamic systems, and for that reason there may be feedback in these models. As explained in [13], due to the feedback, the system allows one or more closed paths for the transmission of the signal around the system, which results in a concept known as short-term memory.

2.3. Data preparation process

For most cases, preparing and applying certain transformations to the raw data can be a critical step to obtain good results in our model.

2.3.1. Generation of the data

Depending on the nature of the problem to be studied, the dataset can be formed by collecting data by means of a field study or, if an appropriate model is available, by generating data through numerical methods. For the present work, all the necessary datasets have been created through the implementation of an algorithm that is able to compute heteroclinic transfers between periodic orbits with arbitrarily high accuracy by numerically integrating a series of initial conditions of the problem (see section 3.2.1. for an extended explanation of the algorithm and other computational details of the implementation).

2.3.2. Data preprocessing

In data preprocessing the first step is always to manipulate the dataset to clean and remove unnecessary information, and standardize the values in numerical or categorical format. This is an important task because it allows for big improvements in the performance and the accuracy of the model. In many cases, this step is one of the most important tasks of the process. We call data preprocessing to a series of tasks, which are summarized as follows:

- **Formatting:** The format in which we work with our data sets can be numerical and categorical, and deciding which is the best will depend on the problem and the goal to be achieved.
- **Cleaning:** Before entering the information in our neural model it is important to clean the datasets of errors, otherwise, we can only can expect errors in the computations. For that reason it is highly recommended to find all the 'NaN' values in the datasets and replace or remove them. If we have many errors but the dataset is not very big, it is better to replace these errors with some equivalence that would not alter the other features, rather than eliminating them completely; that way, a more convenient approach would be to simply remove the instances for those values.

2.3.3. Dataset splitting

The evaluation of the learning model is given for a simple methodology or strategy, where we divide our dataset into three parts, data test, data train, and validation data. This is to know well the performance in our model.

2.3.3.1. Training set

This dataset has the main to help our model to learn the parameters and how can adapt them to solve the problem. In the neural networks for each iteration, the training data provide some feedback that is assimilated for the neurons, and in this way, the ANN can continue learning.

2.3.3.2. Validation set

The validation set is used to avert over-fitting the model, and that validation is realized simultaneously in the model trained performed on the validation set after every epoch.

2.3.3.3. Test set

At the moment that we finish training our model it is necessary to test if it is working, and for that, we use some data that the model never use previously, for all of this is denominated as data test or model evaluation. Basically, this dataset computes the performance of the implemented method.

An important aspect to take into consideration is the fact that when the data is split, the proportion to train should be enough to learn but not to copy the patterns so closely that we can obtain over-fitting in our model. For that reason, a good compromise between having enough training data to ensure the accuracy of the model and having enough test data as a reliable source to evaluate the performance is required. As a rule-of-thumb while working with ANNs, the recommended percentage to split the data is 70 % for training and 30% for testing. Also, when we use K-Fold cross-validation as explained in section 2.4.1. all data is used to make the validation.

2.3.4. Data transformation

2.3.4.1. Reescaling of the data

A crucial step before feeding the training data to the network for it to start the learning process is to scale the variables for a better performance of the model. Scaling a variable means that its values are shifted by a certain constant and then divided by another constant. The most common scaling methods used in statistics and AI are *normalization* and *standardization*. Although these two terms are mathematically defined differently, in the field of AI the two words are used non-strictly and they are often used interchangeably. Generally speaking, if we consider a single variable of the set given as a vector $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, its normalized value **x'** can be defined according to the mentioned scaling methods [1]:

- **Normalization**: the division of a vector by its norm (most often the Euclidean norm).

$$\mathbf{x'} = \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{2.9}$$

  Also, it is not uncommon in ANN bibliography to see the term used as a synonym for min-max scaling, that is, reescaling a variable in order for it to fall within the range from 0 to 1.

$$\mathbf{x'} = \frac{\mathbf{x} - min(x)}{max(x) - min(x)} \tag{2.10}$$

- **Standardization**: scaling of the variable so that the resulting data has zero mean and unit standard deviation.

$$\mathbf{x'} = \frac{\mathbf{x} - \mu}{\sigma} \tag{2.11}$$

where μ is the average of the variable and σ is its standard deviation.

In this work the data is standardized following equation (2.11).

2.4. Validation of the model

Many techniques are available for cross-validation. Among the most common are the following [18]:

- K-fold
- Holdout
- Leaveout
- Repeated random sub-sampling
- Stratify
- Resubstitution

The technique used in this work is the K-fold cross-validation, which is described below.

2.4.1. K-Fold cross validation

This popular cross-validation technique consists of partitioning the data into K nearly equal-sized groups called folds, at random. The model is then trained and validated once for each partition of the data. This step is repeated K times to ensure that each subset is validated exactly once. At the end of the training/validation process, the mean error among the different partitions is provided as a measurement of performance.

This provides a way to asses the model performance without the need of devoting a certain percentage of the available data to validation. That is, it offers the advantage that all the data (except for the test dataset) can be used for training, but at the cost of multiplying the training process time by the number of folds (K).

2.5. Statistical error indicators for the evaluation of the models

In order to make an exhaustive analysis of the performance of the ANN predictive engines developed throughout the present work, it is necessary to make use of error indicators, understanding as error the difference between the target and output values measured by means of different formulas. This section presents different indicators of error that are popular in the field of AI due to their ability to show different aspects related to such errors (such as their absolute and relative magnitude or their distribution).

While the misclassification rate and the classification loss are indicators of how well the multiclass classification is performed by the ANNs, other indicators such as MAE, MAPE, RMSE and RMSRE reflect the capability of the regression ANNs to match the output values with the exact values comprehended in the test sets.

2.5.1. Mean absolute error (MAE)

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{2.12}$$

- MAE : Mean absolute error
- yi : Prediction
- xi : True value
- n : Total number of data points

2.5.2. Mean absolute percentage error (MAPE)

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - f_t}{A_t} \right| \tag{2.13}$$

- $MAPE$: Mean absolute percentage error
- n : Number of times the summation iteration happens
- A_t : Actual value
- f_t : Forecast value

2.5.3. Root mean square error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \widehat{y}(i)\|^2}{N}} \tag{2.14}$$

- $RMSE$: Root mean square error or root mean square deviation
- n : Number of data points
- $y(i)$: $i-th$ measurement
- $\widehat{y}(i)$: Corresponding prediction

2.5.4. Root mean square relative error (RMSRE)

$$RMSRE = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \Delta X_{rel,i}^2} \tag{2.15}$$

$$\Delta X_{rel,i}^2 = \frac{X_i}{T_i} - 1 \tag{2.16}$$

where,

- $RMSE$: Relative root mean square error
- T_i : Desired value
- X_i : Actual value

2.5.5. Misclassification rate

A simple way to evaluate and compare many classification models, whether binary or multiclass, is to calculate the number of misclassified samples to number of total samples ratio [25]:

$$M_r = \frac{n\{E\}}{n\{T\}} \tag{2.17}$$

where,

- $n\{A\}$: Cardinality of set A
- E : Set of misclassified instances
- T : Set of total instances

2.5.6. Classification loss

Different classification loss functions can be used as a way to measure the inaccuracy of a classification model, which is useful when comparing different models. For this work the loss function that was used was the following [17]:

$$L = \sum_{i=1}^{n} w_i I_A\{\hat{y}_i - y_i\} \tag{2.18}$$

where a lower value of L indicates a better performance. As for the other variables,

- n : Sample size
- w_i : Weight for observation i
- $I_A\{\cdot\}$: Indicator function
- $\hat{y}_i$: Class label corresponding to the maximal score
- $\hat{y}_i$: Observed class label

2.5.7. Coefficient of variation

The coefficient of variation, c_v, is defined as the ratio of the standard deviation [5].

$$c_v = \frac{\sigma}{\mu} \tag{2.19}$$

- σ : Standard deviation
- μ : Sample mean

2.6. Tools for the numerical implementation

Throughout this work, all the code has been developed in MATLAB and run on a single core of a 2.40 GHz 11th Generation Intel® Core™ i5 processor.

For the design of the neural networks, for the purpose of focusing on leveraging the existing functions to solve the proposed problems, rather than developing new AI-based algorithms, the following MATLAB toolboxes were employed:

- *Statistics and Machine Learning Toolbox*
- *Deep Learning Toolbox*

CHAPTER 3. COMPUTATION OF HETEROCLINIC CONNECTIONS BETWEEN PLANAR LYAPUNOV ORBITS

3.1. Introduction

In the present chapter, two new ANN approaches are developed and tested to solve the problem of computing heteroclinic connections in the Planar Restricted Three Body Problem. Then, the results are contrasted with those from the classic resolution of the same problem, for which a well defined solution exists.

The study case for this work is the Earth-Moon system, for which the parameters used are listed in table 3.1.

Table 3.1: Parameters of the Planar Restricted Three Body Problem for the Earth-Moon system (astronomical data for the length scale and the Moon radius retrieved from [12] and [28] respectively)

Parameter	Value	Unit
μ	1.215058191870689E-02	
Length scale (mean Earth-Moon distance)	3.84401E+05	km
Moon radius	1.7381E+03	km

The length scale corresponds to the distance between the Earth and the Moon and it is required to nondimensionalize the Restricted Three Body Problem; thus, the obtained results of magnitudes with length units, such as position, must be multiplied by that parameter to return them to real scale. On the other hand, the considered radius for the small primary is the equatorial as it is a conservative value when considering collisions. Like the rest of magnitudes with length units, the Moon radius needs to be nondimensionalized with the length scale for its use in the implementation of the problem.

3.2. Methodologies

To model the Earth-Moon system here studied, the Circular Restricted Three Body Problem is used, for which a detailed description, along with its equations of motion, can be found in Chapter 1. Nevertheless, some further ideas on the algorithms used in the process of determining the heteroclinic connections and its numerical implementation are included in the following sections.

In order to assess the performance of the new ANN approaches to identify the heteroclinic connections for a given Jacobi constant, three complementary methods are implemented. For the sake of making a distinction between the three they shall be conveniently referred to as follows (in order of appearance): *classic method*, *ANN refinement* and *ANN curve fitting*. The first consists of the foundation for the database generation for the computation of the two ANN-based methods. Therefore, the latter methods are necessarily dependent on the classic method.

3.2.1. Classic methodology to compute heteroclinic connections

The computation of heteroclinic transfer trajectories between planar Lyapunov orbits is a closed problem. The method summarized in this section for finding these trajectories, henceforth referred to as the classic method, follows similar steps as the ones described by Canalias and Masdemont in [2], with some variations that are explained here.

3.2.1.1. Computation of planar Lyapunov orbits

This work is focused on the planar Lyapunov orbits in the vicinity of L_1 and L_2 of the Earth-Moon system. Each planar Lyapunov orbit that exists around each libration point for a given Jacobi constant is unique, according to the Poincaré theorem [2], and it can be completely defined by finding an initial condition in the $\mathbb{R}^4$ state space, that is, finding a state vector of the form $(X, Y, \dot{X}, \dot{Y})$, that belongs to the orbit. We refer to that $\mathbb{R}^4$ state space as the initial condition of the orbit, since the trajectory of the orbit can be propagated in time for a period.

With the purpose of computing the heteroclinic connections between L_1 and L_2, the first step is to define the two libration orbits around these points for a given energy level, i.e., for a fixed value of the Jacobi constant. Therefore, a function capable of computing the two orbits corresponding to a given energy level is developed. In order to do so, these steps are followed:

- Generate a table containing a set of several initial conditions that correspond to the planar Lyapunov orbits around each libration point for several values of C_J within a range slightly wider than the range of interest shown in 3.5.. This is done using a third-order Richardson approximation to feed a differential correction procedure based on Newton's method, by starting with small-amplitude orbits and using the computed solutions of the initial conditions for the next iteration.

- The data contained in the previous table is interpolated with a cubic spline to obtain a continuous function of the Jacobi constant, which gives an initial guess of the Lyapunov orbits around each libration point. These initial guesses are valid to be used in the range of interest as a seed for the next step.

- For a Jacobi constant, the initial guesses given by the interpolation functions are refined by means of Newton's method, until accurate initial conditions are obtained for the two planar Lyapunov orbits.

Note that the continuation method explained above is necessary since algorithms that provide a seed for the initial condition such as the one used here, can often show convergence issues for larger amplitudes of the orbits.

3.2.1.2. Propagation of an initial state

Given an initial state of the planar R3BP in $\mathbb{R}^4$, it is possible to approximately propagate that state forward or backwards in time by applying a numerical integrator to the vectorial equations of the model (see (1.13)). Throughout this work, such propagator is used within different functions for different purposes, e.g., propagation of the initial condition that defines a planar Lyapunov orbit to obtain the locus that conforms it, or integration of the manifolds until a crossing with the Poincaré section occurs.

For this purpose, a Runge–Kutta–Fehlberg 45 numeric integrator is implemented with arbitrarily high precision. This integrator is used to obtain the manifold crossings for the classic method, and is capable of generating multiple instances of the solution of the problem. This is a crucial step, since it is the only way to acquire an ensemble of accurate observations that can be used to train the ANNs [27]. Therefore, the results obtained from its numerical implementation are used as a benchmark (see section 3.3.) to which the predictions from the other two methods are contrasted against.

3.2.1.3. Integration of unstable and stable manifolds

In order to compute the manifolds of a periodic orbit, a new parameter is introduced: τ. This provides a way to parametrize the invariant manifolds, thus enabling us to obtain a variety of crossing points of the manifolds at the Poincaré section, one for every integrated manifold corresponding to a unique τ value. This parameter can be interpreted as a number analogous to an angle, where $\tau = 0$ refers to the manifold that exits the orbit at its starting point (as given by the initial conditions which are computed as explained above), while $\tau = 360$ refers to the exit point after one period. The idea of taking this angle-like interpretation of the parameter is taken from De Smet and Scheeres recent paper [4] that inspired the idea for this work, as mentioned in the introduction.

3.2.1.4. Refinement of the Poincaré section intersections

A modified version of the approach described by Canalias in her work [3] is used so as to compute the intersections in the Poincaré section that lead to heteroclinic connections with the states of the stable and unstable manifolds matching up to an arbitrarily high accuracy.

As explained in section 1.4.2. of Chapter 1, the state at the cut of a manifold with the Poincaré surface is completely defined by its values of Y and $\dot{Y}$, since the X coordinate is restricted by the Poincaré surface and $\dot{X}$ is inferred from the fixed value of C_J which relates the coordinates through equation 1.2..

As a first step to computing these intersections, the points in the Y-$\dot{Y}$ plane obtained as a result of integrating the stable manifolds backward in time and the unstable manifolds forward in time are connected with polygonal segments. An algorithm is then implemented to identify if there is any pair of unstable and stable segments that intersect. If an intersection is detected, initial guesses for the τ values of the stable and unstable manifolds that lead to an heteroclinic connection (referred to as τ^{s*} and τ^{u*} respectively) are calculated from the values of the τ parameter at each endpoint of the intersecting segment (see figure 3.1) through the following expressions,

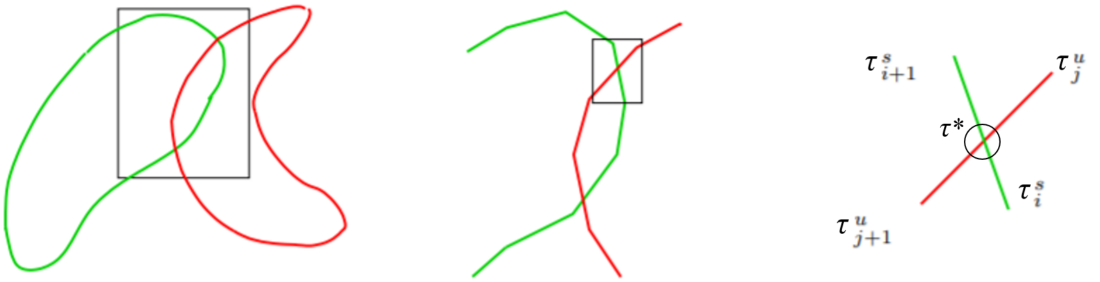


Figure 3.1: Schematic representation of the procedure used to compute the intersections in the Poincaré section (step 1)

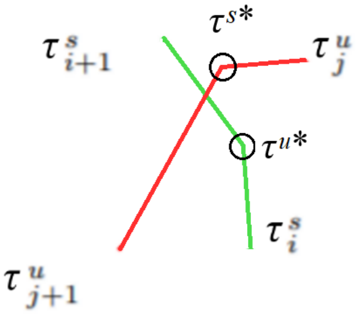


Figure 3.2: Schematic representation of the procedure used to compute the intersections in the Poincaré section (step 2)

$$\tau^{s*} = \tau_i{}^s + \left(\tau_{i+1}{}^s - \tau_i{}^s\right) \frac{Y_{cut} - Y_i{}^s}{Y^s_{i+1} - Y_i{}^s} \tag{3.1}$$

where Y_{cut} is determined by the intersection between the straight segments, while $Y_i{}^s$ and $Y_{i+1}{}^s$ are the Y coordinates of the endpoints of the stable segment. Equivalently for the unstable manifold, we can write that,

$$\tau^{u*} = \tau_j{}^u + \left(\tau_{j+1}{}^u - \tau_j{}^u\right) \frac{Y_{cut} - Y_j{}^u}{Y^u_{j+1} - Y_j{}^u} \tag{3.2}$$

With the obtained values of τ^{s*} and τ^{u*}, we can now integrate the stable and unstable manifolds that correspond to these values of their respective τ parameter. In doing that, we find a new pair of crossings of the manifolds in the Poincaré section, thus obtaining four new segments as outlined in figure 3.2.

Among those four segments, the two segments that intersect are identified and the procedure explained above is iterated for these by applying equations (3.1) and (3.2) again, until sufficient level of accuracy is obtained for the solution.

When a solution is reached, the values of τ_s and τ_u are used to build a representation of the heteroclinic trajectory by propagating the unstable and stable manifolds forward and backward in time.

3.2.2. ANN approaches to predict heteroclinic connections

3.2.2.1. Prediction of the manifold cuts with the Poincaré section

The first of the two ANN approaches used to predict the heteroclinic connections, which is from this point forward referred to as the *ANN refinement* method, consists of training multiple ANNs to develop a prediction engine that is able to find the cuts of the manifolds with the Poincaré section. Said prediction engine consists of a sequence of ANNs that work in two steps to infer the results:

- First, a classification-type ANN assigns one of the three classes to each instance of the input data: positive, negative or collision (see the schematic representation in figure 3.3).
- Once an instance has been predicted to belong to a certain class, it enters a regression-type ANN, where the coordinates of its corresponding manifold cut are predicted. The specific ANN model that an input instance enters depends on its assigned class. This means that an instance that has been classified into the positive group will enter the regression-ANN that models the manifold cuts that belong to the positive group. Similarly for an instance that has been classified into the negative group, it will enter the ANN that models the cuts of the negative group. As for the input that are predicted to fall into the collisions class, they do not enter any ANN, since their manifold cuts are not computed.

Thus, given the C_J of a trajectory and the parameter τ that defines the exiting point of the unstable/stable manifold from the departure/arrival orbit as the two input values, the prediction engine shall output the Y and $\dot{Y}$ coordinates that define the cut of the said manifold with the Poincaré section.

Regarding the hyper-parameter fitting of these ANNs, the use of different types of activation functions were compared and it was concluded that with a hyperbolic tangent type function a better fit was obtained, according to the error measurements provided by the k-fold cross validation data.

Once this prediction engine is built, it can be used as a black-box that given a C_J-τ input, it outputs the Y and $\dot{Y}$ coordinates at the Poincaré section almost instantaneously, since evaluating the ANN functions requires very little computational cost. This black-box can be used as a means to generate an approximation of the curves in the Poincaré section for a fixed Jacobi constant. It can also be used to refine the intersections between the curves, when integrated into the refinement procedure explained in 3.2.1.4.. Similarly to the classic method, when such an intersection is found, the τ_s and τ_u are used to integrate backwards and forward in time respectively, so as to obtain the heteroclinic trajectory.

3.2.2.2. Curve fitting to predict heteroclinic connections

The method presented here will be hereinafter referred to as "ANN curve fitting". As will be seen in the benchmark results, the pairs of C_J-τ values that correspond to heteroclinic connections form continuous curves. The basis of the *ANN curve fitting* method lies on the generation of a dataset that contains a series of points of the heteroclinic connections curve. Then, the dataset is fed to a neural network as for it to fit the data and create a model of the curve. Because the heteroclinic connections considered here are from L_1 to L_2, it is necessary to generate the (C_J, τ_u) and (C_J, τ_s) pairs corresponding to the manifolds that form the connections.

For the design of the neural networks, different numbers of fully connected layers (between one and three) and different numbers of neurons for each of the layers (between 1 and 300 per layer) were tested by means of Bayesian optimization with 30 iterations, and a 44-node single-layer network was found to give good results for all the regressions.

As the performance of these neural networks is going to be tested comparing them with the cubic interpolation shown in section 3.3., and the validation to prevent overfitting is performed through k-fold cross validation (thus, there is no need to set aside part of the data for the validation as explained in section 2.4.1.), we have chosen to use all the available data that had been calculated for obtaining the benchmark results as a training set (100 pairs of points in total corresponding to the heteroclinic connections).

As an alternative to this ANN curve fitting, it would also be possible to adjust the same points used for training by means of a cubic interpolation producing very good results. However, because of the shape of the heteroclinic connections curves (which can map two different values of τ to a single value of C_J and vice versa and therefore, it cannot be fitted with a single continuous function as it is) this requires the additional step of converting the curve from the Cartesian coordinates C_J and τ, to auxiliary polar coordinates. By contrast, ANN curve fitting in this context has the advantage of being more straightforward once the points that represent the heteroclinic connections have been properly sorted in the datasets (since the ANNs for fitting the upper τ values and the ANN for fitting the lower τ values are independent, they must be sorted into separate training sets). The cubic interpolation alternative, which is the same as the one presented in section 3.3. as a benchmark of the heteroclinic connections curve, will be used to compare the results of the ANN curve fitting method.

3.2.3. Generation of the datasets

In order to train an ANN on a chaotic problem, a set of solutions obtained from multiple initializations of the input variables of interest is required. For the ANN to be able to provide sufficiently accurate predictions of the solutions, numerous and varied observations must be fed to the network [27]. As aforementioned, this set of solutions is generated by finding heteroclinic connections using the scheme described for the classic method (section 3.2.1.). The data is stored in tables that are then used as the learning material for the ANNs. Each table contains the variables that correspond to the inputs and the outputs of the networks. Thereby, the networks try to match the correspondence among them.

As explained in the Methodology section, multiple neural networks are trained independently. That means that a unique dataset is given to each one of them. The interval of C_J

where the heteroclinic connections are searched, is the same for the three methods and corresponds to:

$$C_J \in [3.1823, 3.1133] \tag{3.3}$$

These values were chosen so as to stay within the range where there are no tangencies of the manifolds with the Poincaré section. Although this is a small range of the Jacobi, but it allows for considerable variation of the amplitudes of the Lyapunov orbits (see examples of multiple planar Lyapunov orbits for different values of the Jacobi constant in figure 1.3).

As for the parameter τ, because of the periodic nature of the orbits, the inverval considered is:

$$\tau \in [0, 360] \tag{3.4}$$

3.3. Benchmark results from the classic method

The results here presented are the outcome of applying the classic method for finding heteroclinic connections explained in section 3.2.1.. Throughout this work, the considered direction of the connecting trajectories is from the planar Lyapunov orbits around the lunar point L_1 to orbits around lunar point L_2. Therefore, the unstable manifolds are computed for L_1 and the stable manifolds for L_2.

3.3.1. Manifold cuts in the Poincaré section

Given a pair of values of C_J and τ that determines a single manifold of the departure or arrival orbit, the cut of that manifold on the Poincaré section (S surface as defined in equation 1.33) can be classified according to the value of the Y coordinate that it takes at the crossing. This section shows how the input variable space formed by C_J and τ is divided according to where the manifold corresponding to a point in that space falls.

See for example in figure 3.8 for the case where $C_J = 3.1140$, the cuts of the unstable manifolds on S (in red) that have a positive coordinate of Y form a continuous curve on the $Y > 0$ half-space, while the cuts of the unstable manifolds with a negative coordinate of Y form a separate continuous curve on the $Y < 0$ half-space. This is also true for the stable manifolds. In the other two cases of the same figure, however (where $C_J = 3.1823$ and $C_J = 3.1600$), none on the manifold cuts fall into the $Y < 0$ half-space, resulting in a single continuous curve on the $Y > 0$ half-space for L_1 and for L_2.

Doing this division is important because depending on whether a manifold falls into the positive region or the negative region, the patterns that are intended to be grasped with the ANNs differ (as can be clearly appreciated in the bottom left plot of figure 3.6) Therefore, the positive and the negative regions are modelled separately, each with a ANN that is trained with data within its own region only.

On the other hand, being able to detect collisions is also critical, because given that the position of the small primary is a singularity of the problem, the numerical propagation of the trajectory might pose some problems when getting too close to said point. This problem is easily solved by discarding the trajectories that intersect with a sphere of a radius slightly greater than the radius of the small mass. The value of the Moon's radius used to define these collisions is the one shown in the table 3.1 applying a conservative margin of 1%.

Regarding the terminology used in following figure legends, the three possible cases are referred to as:

- Positive: Y coordinate of the manifold cut on the S section is positive.
- Negative: Y coordinate of the manifold cut on the S section is negative.
- Collision: Y coordinate of the manifold cut on the S section is within the collision radius.

This classification of the manifold cuts on the S section is schematically depicted in figure 3.3 (the coloring scheme of the different regions is the same as the one that is being used to show the results in the figures below: blue for the *positive* group, red for the *collision* group and yellow for the *negative* group).

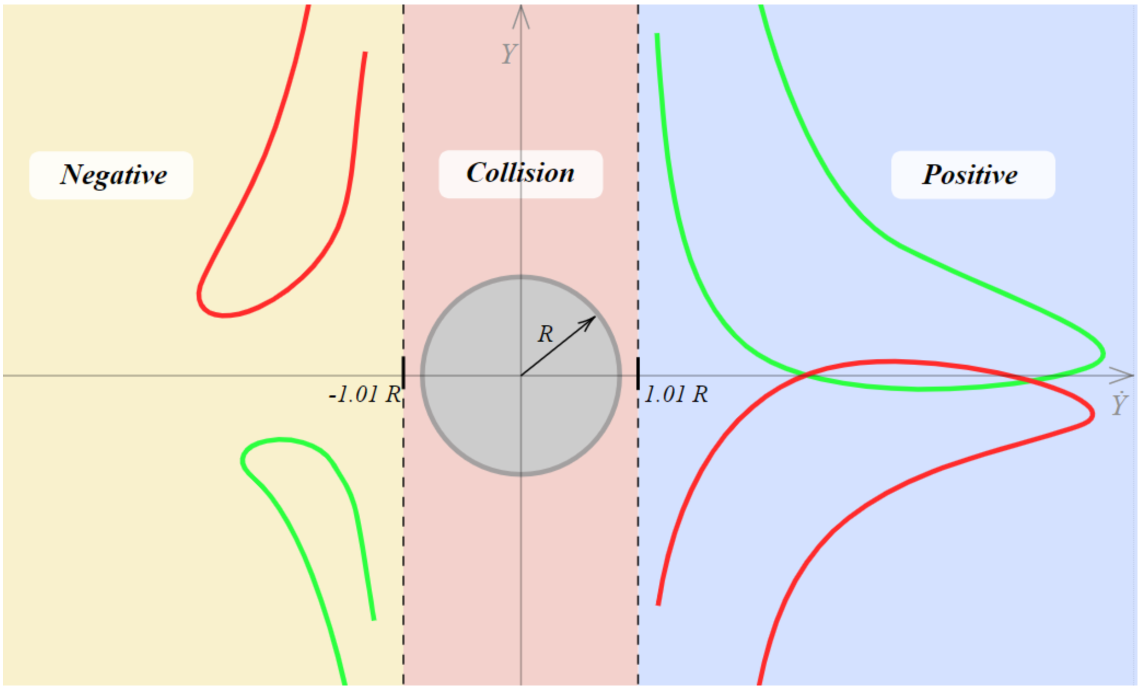


Figure 3.3: Schematic representation of the classification of the manifold cuts on the Poincaré section according to the Y coordinate where they fall

Once the three categories have been defined, the benchmark datasets can be sorted to define the different areas in which the $C_J - \tau$ plane is divided. Each one of these datasets contains 39700 points (that is, almost four times the number of points used for the training and test of the ANNs) for the sake of higher resolution of the reference solution. Figures 3.4 and 3.5 show the classification into the three regions for the case of unstable manifolds of L_1 and stable manifolds of L_2, respectively.

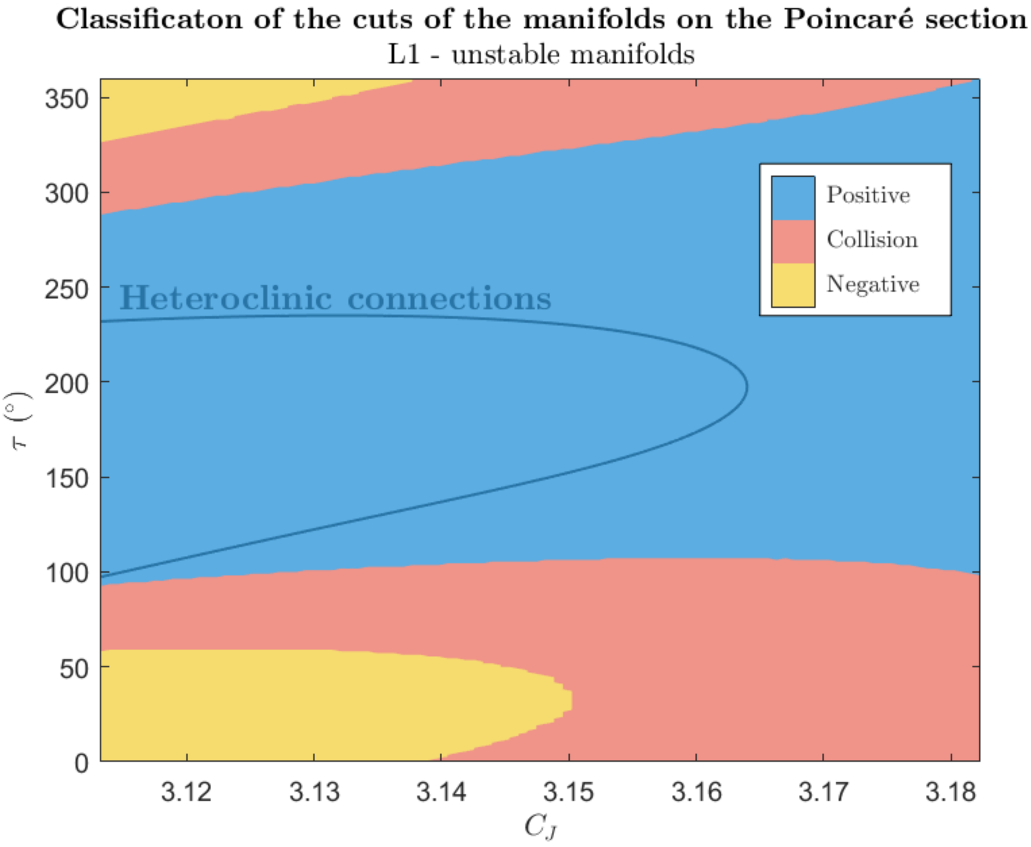


Figure 3.4: Classification of the cuts of the unstable manifolds of L1 according to the sign of the Y component. Manifolds that approach the small primary within a given radius (see table 3.1) belong to the group named as "Collision".

Additionally, in these figures the curves that give the C_J and τ pairs corresponding to the heteroclinic connections are drawn from the interpolation of the data points. From the curves, it is inferred that for each fixed value of the Jacobi constant, two heteroclinic connections can be found up to $C_J \approx 3.164$. For greater values of the Jacobi constant (which correspond to orbits of smaller amplitudes around L_1 and L_2), no heteroclinic connection is found.

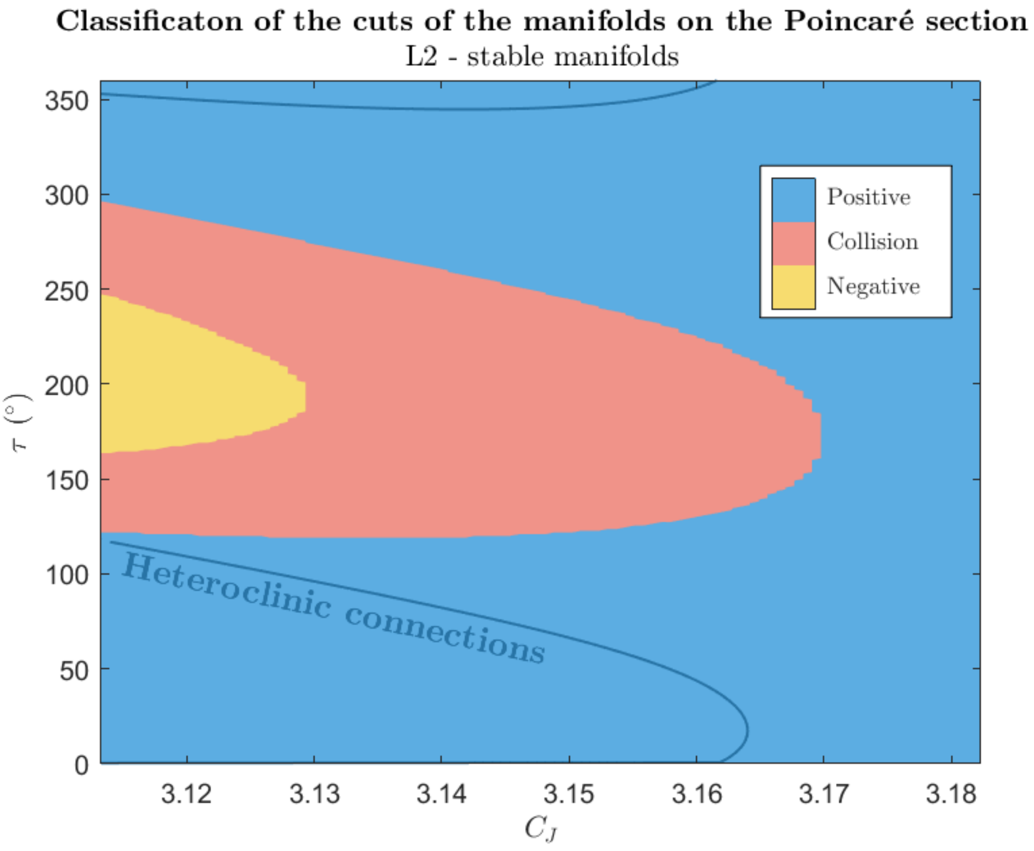


Figure 3.5: Classification of the cuts of the stable manifolds of L2 according to the sign of the Y component. Manifolds that approach the small primary within a given radius (see table 3.1) belong to the group named as "Collision".

The periodic nature of the problem is made clear in both figures, meaning that the results show continuity from values of $\tau = 0°$ to $\tau = 360°$.

Furthermore, the values of the Y and $\dot{Y}$ coordinates of the aforementioned manifold cuts are plotted and shown in figures 3.6 and 3.7. These coordinates can be seen to form surfaces that are continuous within the boundaries of each group (considering the aforementioned periodicity in τ).

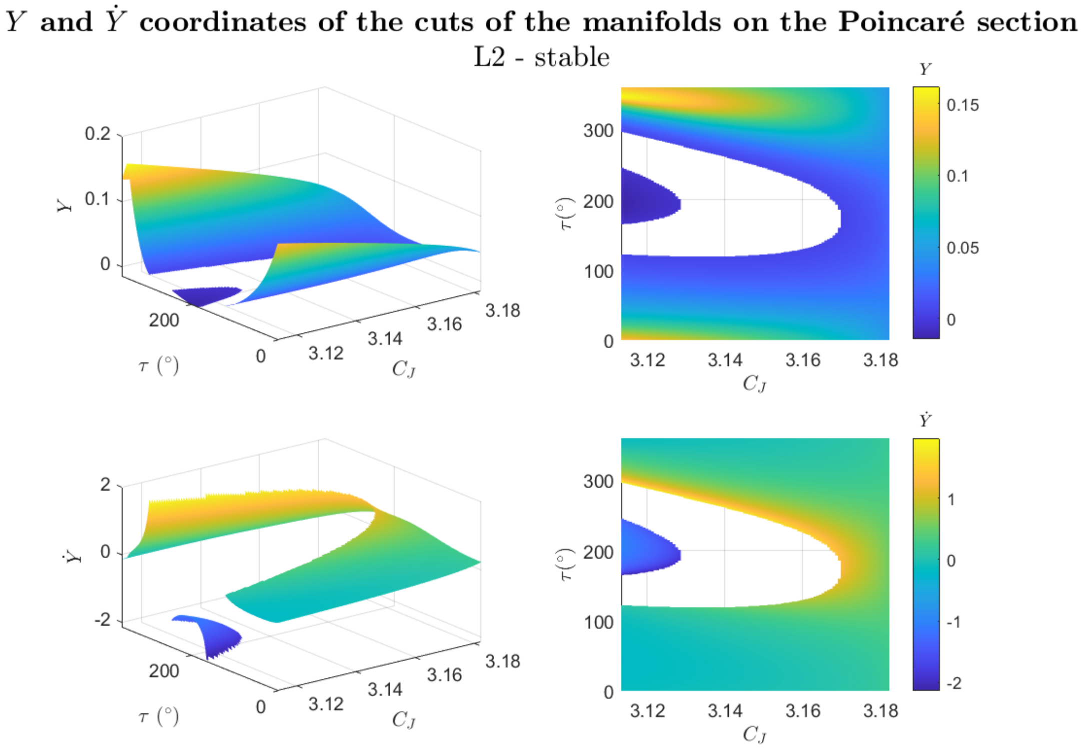


Figure 3.6: $\dot{Y}$ coordinate of the cuts of the unstable manifolds (L_1) with the Poincaré section

For both libration orbits, comparing the surfaces corresponding to the coordinate Y with the surfaces corresponding to $\dot{Y}$, a remarkable observation is that there is a difference of one order of magnitude between the two, being the values of the velocity coordinate greater than the position coordinate.

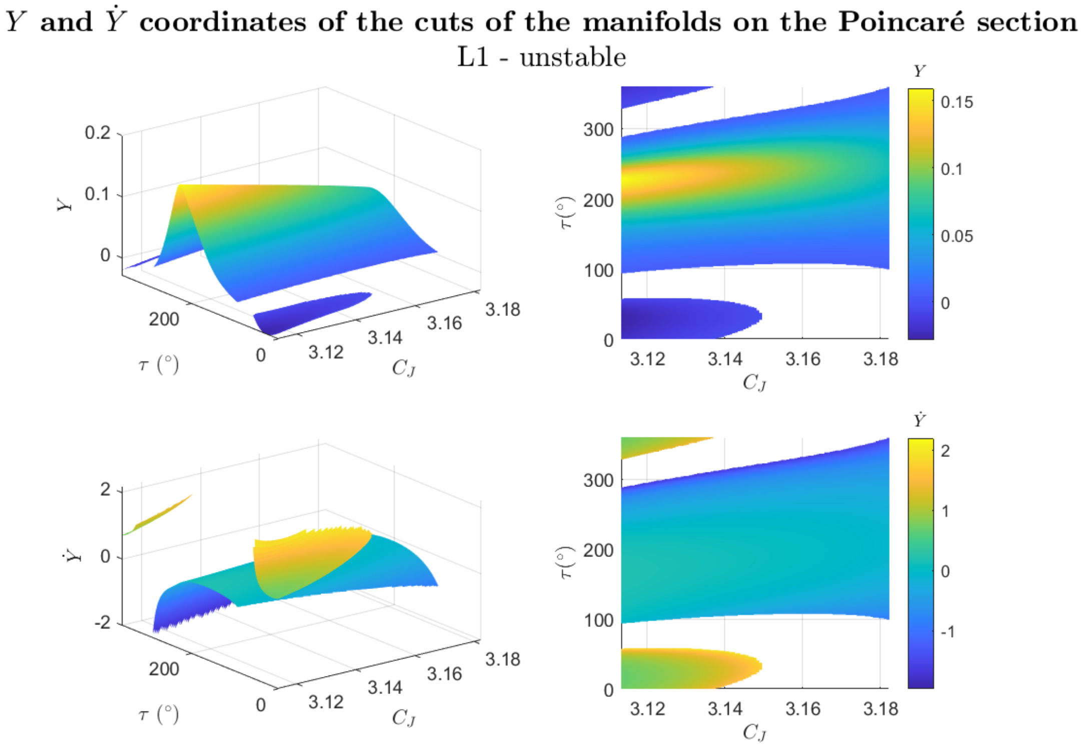


Figure 3.7: $\dot{Y}$ coordinate of the cuts of the unstable manifolds (L_2) with the Poincaré section

Taking a section with $C_J = C_J^*$ in the surfaces depicted in the previous figures, so that C_J^* is a constant value within the studied interval, it is possible to construct the Poincaré maps. The curves formed in some of these sections are shown in the following section.

3.3.2. Some examples of heteroclinic connections

In order to compute the heteroclinic connections, the Poincaré section defined by (1.33) is used as explained in section 1.4.. Recalling that for an heteroclinic connection to exist for a given constant value of $C_J = C_J^*$, the curves formed by the manifold crossings in the Poincaré section must intersect in at least one point. In figure 3.8, three example cases of these curves are shown. It can be observed that while the Poincaré section for $C_J = 3.1823$ does not show any intersection between the curves of the stable and unstable manifolds and thus, a pair of heteroclinic connections can be found for these cases (one pair for each Jacobi constant).

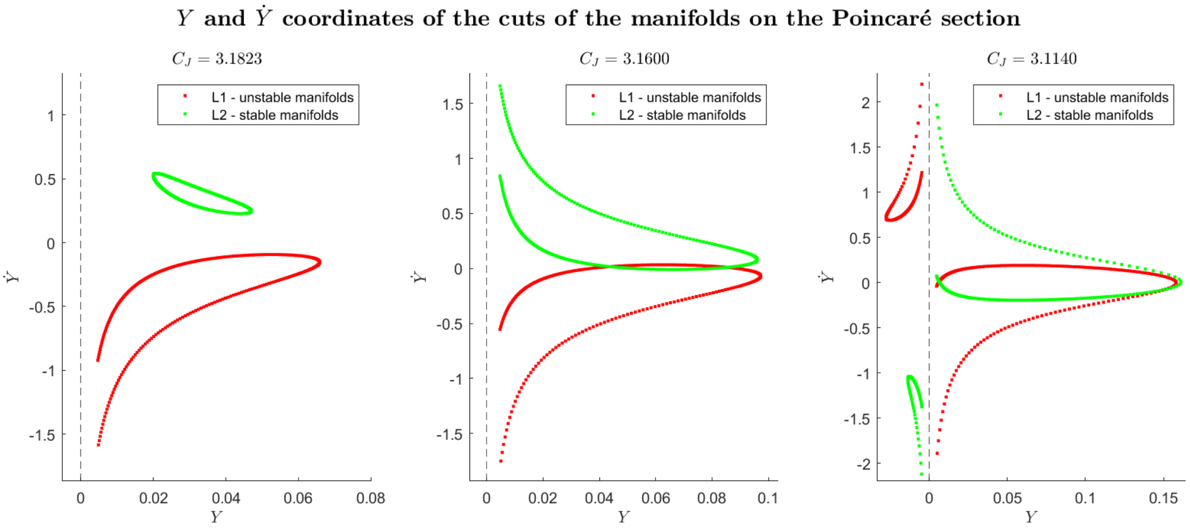


Figure 3.8: Examples of different curves of the manifold cuts in the Poincaré section computed for different values of the Jacobi constant

The corresponding pairs of heteroclinic connections computed for these two cases are depicted in figure 3.9. As can be appreciated, the states of the stable and unstable manifolds closely match at the X coordinate given by the Poincaré section, due to the refinement of the intersections, which gives the τ value that corresponds to these solutions as an output.

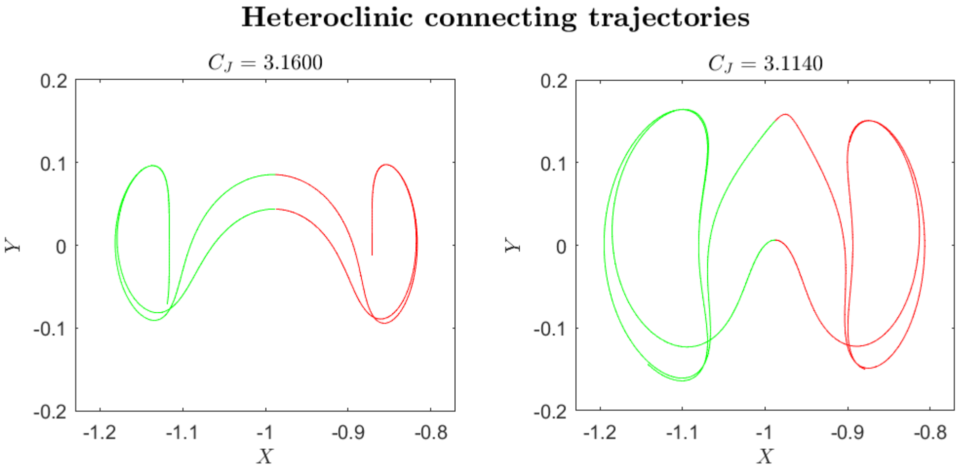


Figure 3.9: Examples of heteroclinic trajectories connecting the libration regions of L_1 and L_2 computed for different values of the Jacobi constant

These are the results that are intended to be found by means of the ANN methods presented in this work. Therefore, the ultimate goal is to develop different methodologies that can compute these heteroclinic at a fraction of the computational cost of the classic refinement method.

3.4. Results from the ANN refinement method

The results from the implementation of the so-called ANN refinement method detailed in section 3.2.2.1. are presented here.

3.4.1. Training with the gridded vs randomly scattered data

Some studies show good predictive results training ANNs with datasets that were generated by initializing the problem with input values taken at some set of grid points [4], while others opt for the generation of the datasets by random initialization of the input values within certain range [27]. To analyse the influence of the input data distribution on the ability of the ANN to accurately reproduce the model under study, two different collections of datasets were created, each with a different distribution of the input variables: gridded and randomly scattered among the input space. Both dataset collections are composed of 10,000 simulations of the unstable manifolds (of orbits around L_1) and 10,000 of the stable manifolds (of orbits around L_2). The split between the training-validation set and the test set is done by randomly partitioning the datasets in a 70/30 proportion. This ratio to divide the datasets into the training and testing datasets is common among AI applications since it provides a generally good compromise between having enough data for the ANNs to recognise the patterns present in the problem and having sufficient data to reliably test and compare the performance of the trained models. As for the intervals used for the two input variables, C_J and τ, they are the same for both dataset collections and are defined in section 3.2.3.. In this section, the results obtained from the ANN prediction engine trained with the gridded data and the one trained with the randomly scattered data are compared.

First, the performances of the trained classification ANNs are compared by testing the networks with the data that had never seen by the network before. It can be observed in figures 3.10 and 3.11 that both ANNs behave well when classifying the new data, given that the missclassification rates are less than 1%. Also, we see comparable behaviour in that the points that the few misclassified instances were in all four cases located near the limits of the different regions as could be expected.

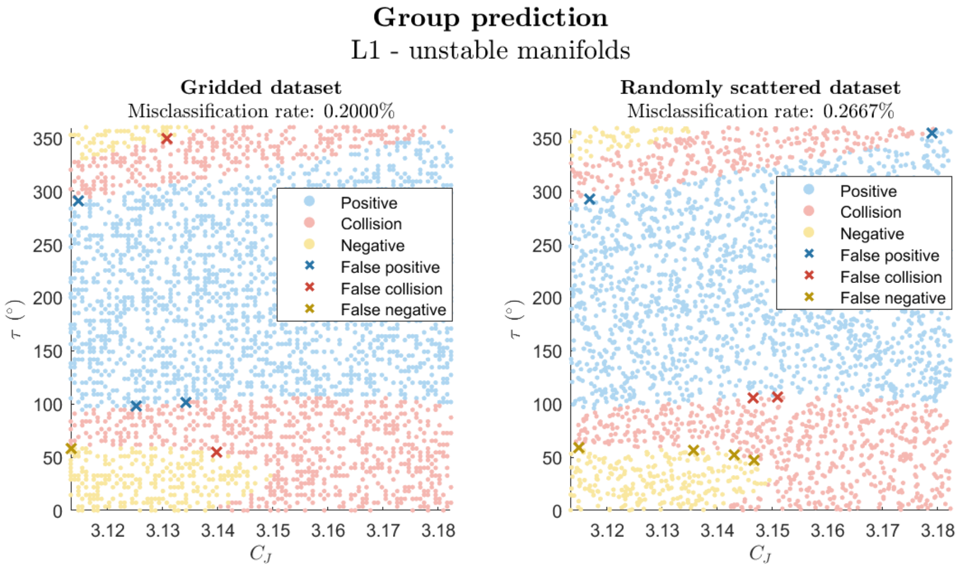


Figure 3.10: Classification performance comparison when training with gridded vs. randomly scattered data for the unstable manifolds of L1

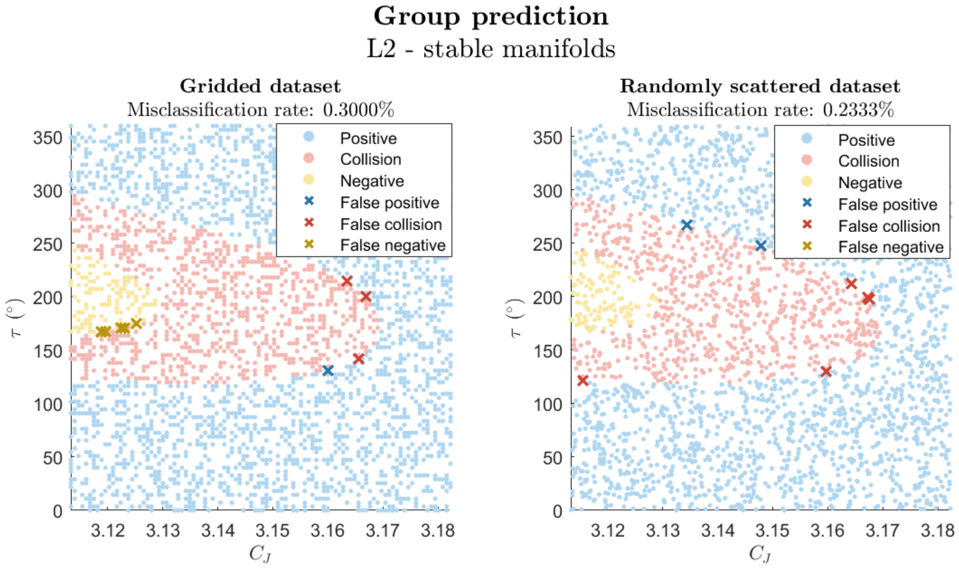


Figure 3.11: Classification performance comparison when training with gridded vs. randomly scattered data for the stable manifolds of L2

Looking at the results obtained for the Y and $\dot{Y}$ predictions by the regression ANNs for the unstable manifolds of L_1 (figure 3.12) and the stable manifolds of L_2 (figure 3.13), they appear to be essentially the same. When compared to the benchmark results, we could say that at a first glance, the networks seem to have learned to reproduce the surfaces well. To be able to compare the two into more detail, the statistical differences between the predicted values and the target are computed by means of multiple error indicators.

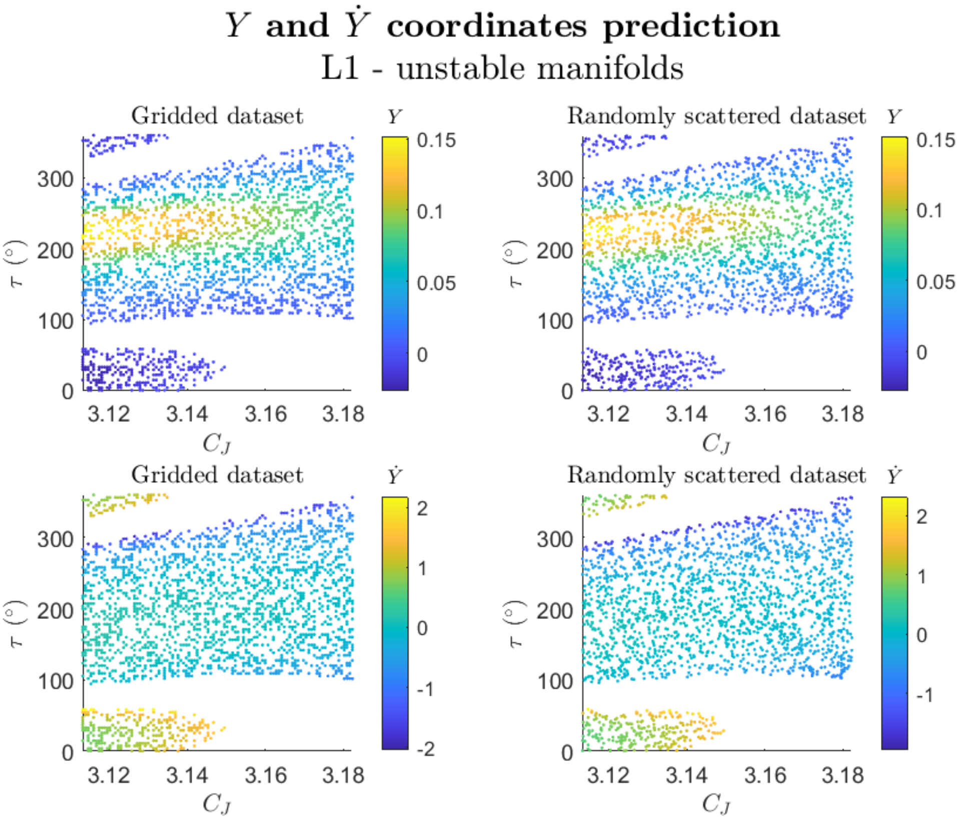


Figure 3.12: Predictions of the Y and $\dot{Y}$ coordinates of the manifold cuts when training with gridded vs. randomly scattered data for the stable manifolds of L2

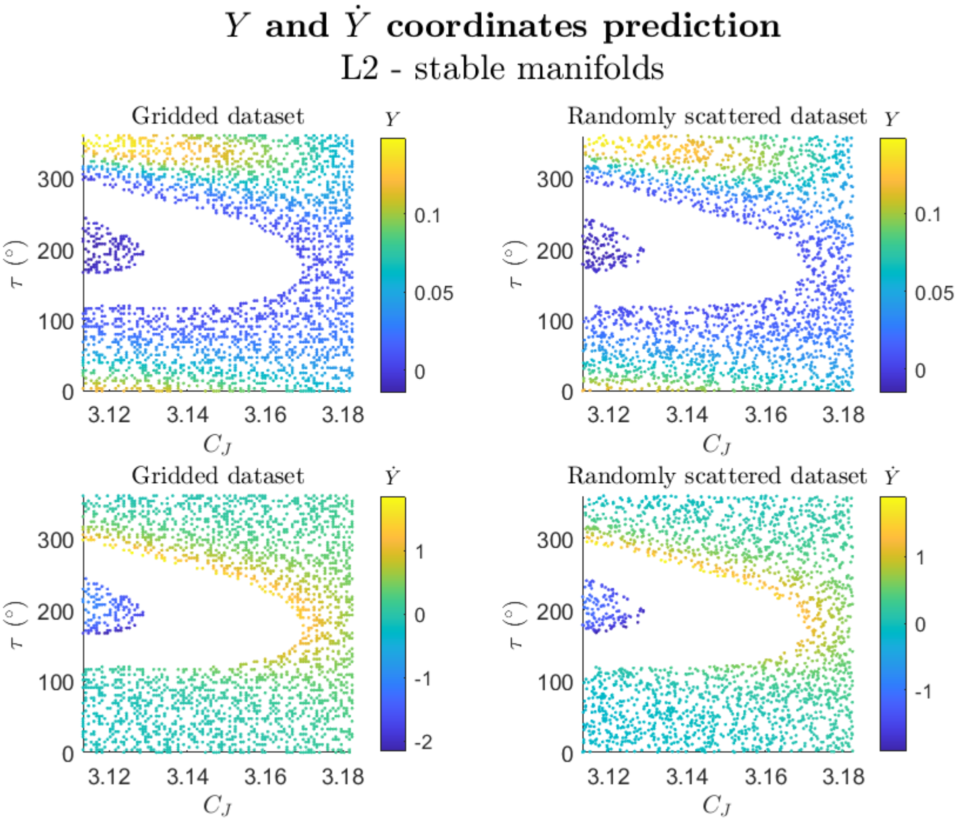


Figure 3.13: Predictions of the Y and $\dot{Y}$ coordinates of the manifold cuts when training with gridded vs. randomly scattered data for the unstable manifolds of L1

For the figures where the regression errors are plotted (both for the absolute and relative error plots), error values represented in the plots go from their minimum absolute value to the value corresponding to the 95th percentile, P_{95}, hence the difference in the scales for each case. This was done to avoid outlier values of the error to hide the results from smaller values of the errors.

The distribution of the absolute errors accross the input space is shown in figures 3.14 and 3.14. It is interesting to see that some high-error regions appear in all cases (yellow-orange areas), and although errors seem to be more concentrated in the limits of the interval where the training data is contained, it is not always the case; some areas show significant rates of error even far from the limits of the data (notice the yellow-like region at the top right plot in figure 3.15).

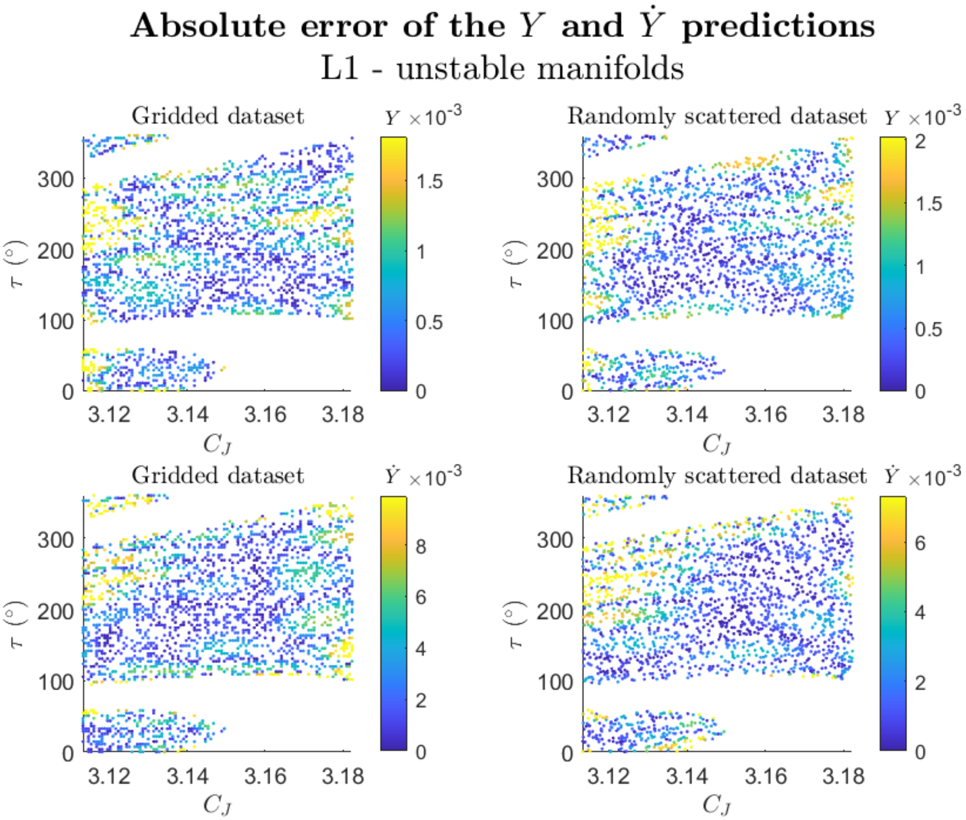


Figure 3.14: Absolute error of the predictions of the L1 manifold cuts' coordinates

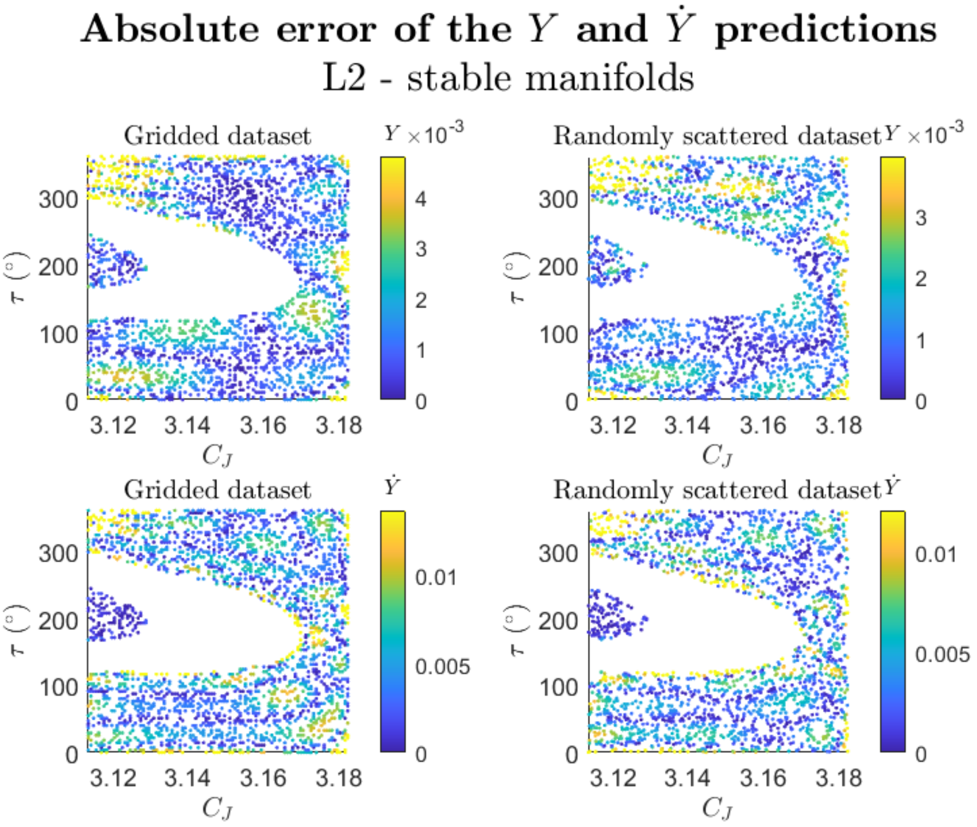


Figure 3.15: Absolute error of the predictions of the L2 manifold cuts' coordinates

To gain a better understanding of the spacial distribution of the errors, the relative error is considered as well, as shown in figures 3.16 and 3.17. As compared to the absolute errors, it can be observed that the relative error is distributed in a less chaotic manner through the $C_J - \tau$ space. Regarding the relative errors of the computations by the ANNs as compared to the benchmark, for the position coordinate of the cuts Y, greater errors are accumulated near the limits between the different regions (between *collision* and *positive* and between *collision* and *negative*). Meanwhile, for the velocity coordinate $\dot{Y}$, the greatest relative errors appear near the regions where the heteroclinic connections are found (see figures 3.4 and 3.5).

As for the effect of training the ANNs with the gridded or the randomly scattered datasets, no significant difference between the two is apparent from these figures.

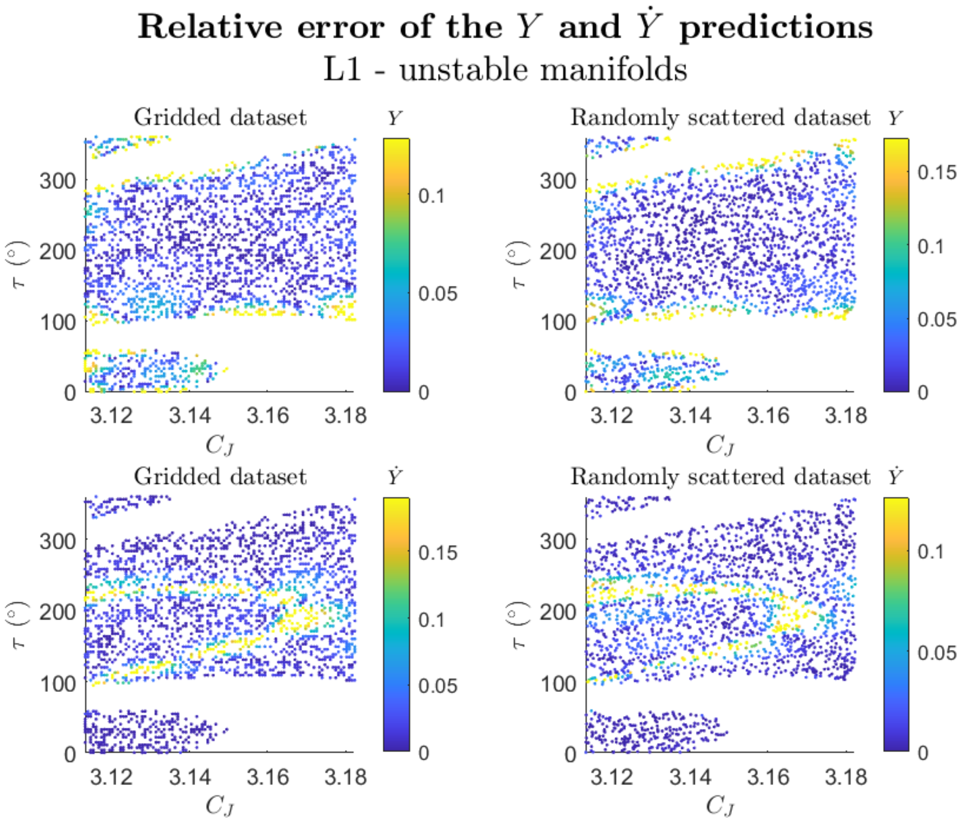


Figure 3.16: Relative error of the predictions of the L1 manifold cuts' coordinates

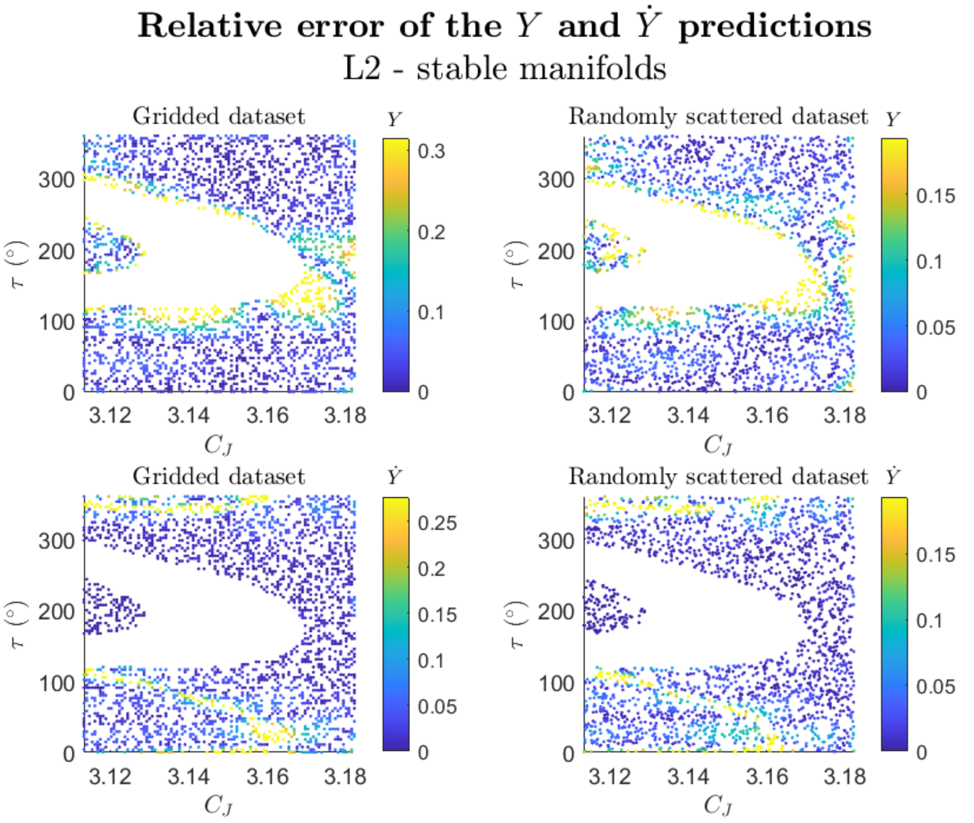


Figure 3.17: Relative error of the predictions of the L2 manifold cuts' coordinates

This can be further discussed by looking at the statistical error measurements given in table 3.2 for the regression ANNs: MAE, MAPE, RMSE and RMSRE (see for the formulas on how the statistical indicators are calculated). By comparing the values of the error-indicators between the ANNs trained with different distributions of input data, it is clear that the networks that had been trained with randomly scattered data have smaller error rates than the ones that had been trained with gridded datasets in all the studied cases, for the predictions of both coordinates and for both libration points. Although not very significant (the difference does not always reach one order of magnitude), there is a slight difference in the value of the error rates between the two data distribution types, nor big differences are observed in the regions where larger errors are concentrated. Thus, it is concluded from this observation that randomly scattered data can provide a more accurate match of the results, and is therefore considered hereunder as the default distribution of the inputs when generating data for such ANNs.

Table 3.2: Statistical error measurements. Comparison between random and gridded datasets.

	MAE			
	L1		**L2**	
	Y	**$\dot{Y}$**	**Y**	**$\dot{Y}$**
Uniformly gridded	1.63E-003	5.23E-003	1.63E-003	5.23E-003
Randomly scattered	1.48E-003	4.52E-003	1.48E-003	4.52E-003

	MAPE			
	L1		**L2**	
	Y	**$\dot{Y}$**	**Y**	**$\dot{Y}$**
Uniformly gridded	7.72E+000	3.22E+001	7.72E+000	3.22E+001
Randomly scattered	5.67E+000	7.57E+000	5.67E+000	7.57E+000

	RMSE			
	L1		**L2**	
	Y	**$\dot{Y}$**	**Y**	**$\dot{Y}$**
Uniformly gridded	2.42E-003	7.27E-003	2.42E-003	7.27E-003
Randomly scattered	2.22E-003	6.31E-003	2.22E-003	6.31E-003

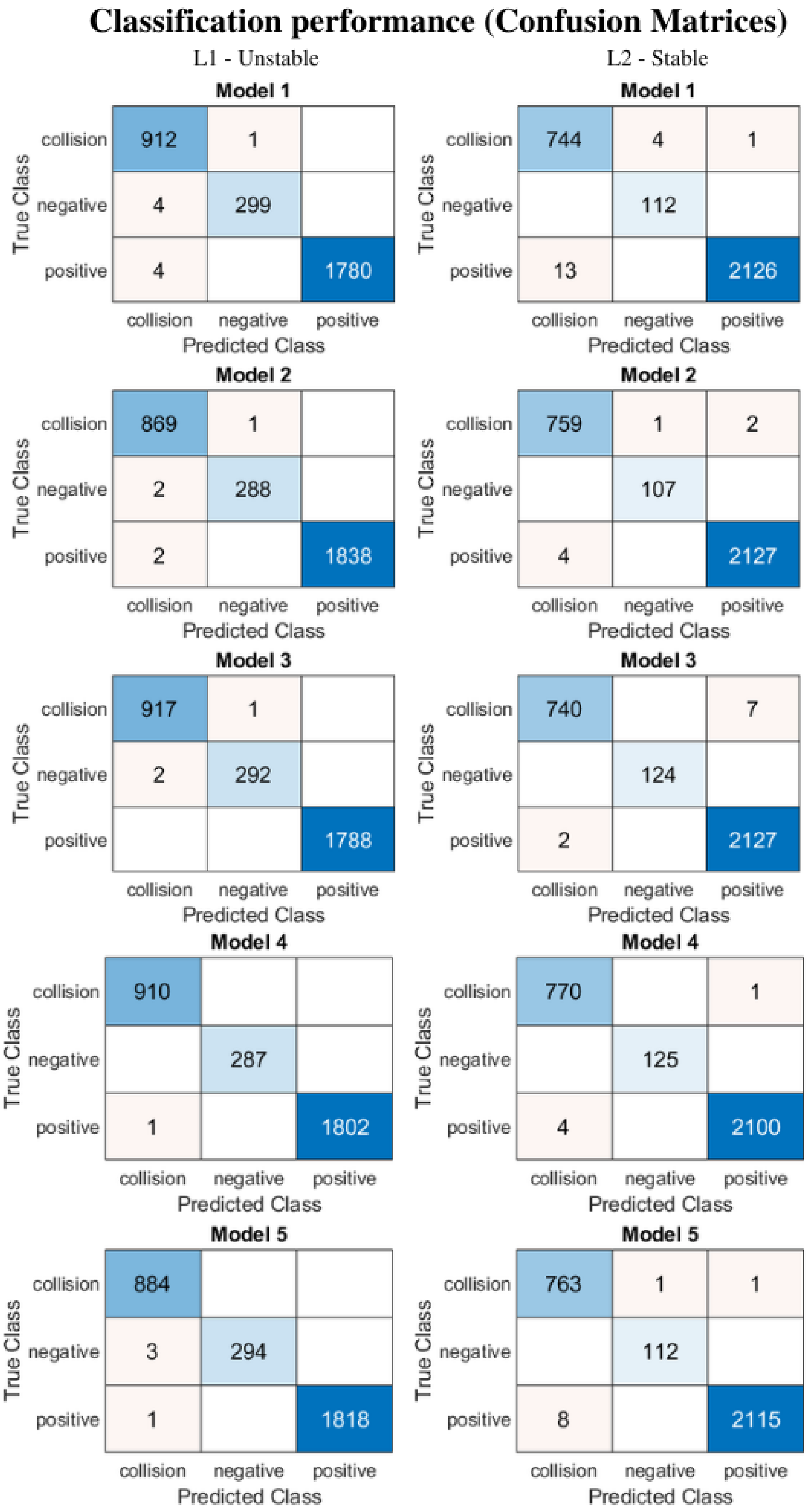
	RMSRE			
	L1		**L2**	
	Y	**$\dot{Y}$**	**Y**	**$\dot{Y}$**
Uniformly gridded	1.41E-001	5.63E+000	1.41E-001	5.63E+000
Randomly scattered	9.39E-002	4.71E-001	9.39E-002	4.71E-001

The reason behind the predictions showing better overall accuracy for the randomly scattered data despite having the same number of entries as the gridded data, seems to lay behind the fact that by randomly initializing the input variables, we are introducing 10,000 unique values of C_J and 10,000 unique values of τ, instead of having 100 different values respectively, as in the gridded set. To summarize, it has been concluded from the analysis of the statistical errors that using randomly scattered dataset results in better predictions. Thus, datasets of this type are recommended for training such ANNs.

3.4.2. Robustness analysis of the ANN

By studying a series of different but similar models of the same problem, the robustness of the method can be analysed. In the case of AI, a method is said to be robust if the model trained with a unique dataset has a similar performance when trained over a range of similar but unique datasets. This analysis is useful to reveal when the obtained results depend on the hyper parameters of the network that were chosen [26]. Therefore, because the good results obtained in the previous section do not guarantee equally good performance for other similar ANNs trained with different random partitions of the datasets, 5 similar ANN models of the prediction engines are created, each one trained with a different 70/30 partition of the training/test data. Then, a statistical analysis of the results is performed in order to evaluate if similar rates of error are achieved or not.

First, a comparison among the performances of equivalent classification models is carried out. As apparent from the confusion matrices of each model as shown in figure 3.18, good consistent classification results are obtained for the different models. With a total of 13 *positive* instances mistakenly classified as *collisions*, the worst results are shown for the classification of L_2 manifolds with Model 1. However, even the worst model behaves adequately, and that good overall performance seems to remain regardless of the particular shuffle of data used for training.



L1 - Unstable, Model 1

True Class \ Predicted Class	collision	negative	positive
collision	912	1	
negative	4	299	
positive	4		1780

L2 - Stable, Model 1

True Class \ Predicted Class	collision	negative	positive
collision	744	4	1
negative		112	
positive	13		2126

L1 - Unstable, Model 2

True Class \ Predicted Class	collision	negative	positive
collision	869	1	
negative	2	288	
positive	2		1838

L2 - Stable, Model 2

True Class \ Predicted Class	collision	negative	positive
collision	759	1	2
negative		107	
positive	4		2127

L1 - Unstable, Model 3

True Class \ Predicted Class	collision	negative	positive
collision	917	1	
negative	2	292	
positive			1788

L2 - Stable, Model 3

True Class \ Predicted Class	collision	negative	positive
collision	740		7
negative		124	
positive	2		2127

L1 - Unstable, Model 4

True Class \ Predicted Class	collision	negative	positive
collision	910		
negative		287	
positive	1		1802

L2 - Stable, Model 4

True Class \ Predicted Class	collision	negative	positive
collision	770		1
negative		125	
positive	4		2100

L1 - Unstable, Model 5

True Class \ Predicted Class	collision	negative	positive
collision	884		
negative	3	294	
positive	1		1818

L2 - Stable, Model 5

True Class \ Predicted Class	collision	negative	positive
collision	763	1	1
negative		112	
positive	8		2115

Figure 3.18: Comparison of the classification performance by means of the confusion matrix among five equivalent models

Now, to see where in the $C_J - \tau$ plane do the misclassifications take place, the test set for each model is used to predict the group for every case, leading once again to very similar results accross all the variations of the model: the misclassifications occur near the limits of the regions. As a means to quantify the variation of the results among the different models, the errors in the classification are summarized in table 3.3, along with their mean value and the standard deviation computed accross the set of 5 models. The coefficient between the standard deviation σ and the mean value μ, is called coefficient of variation and denoted like c_v, and it can be used to evaluate whether the fluctuations within a sample are big or small as compared to the mean value. As a rule of thumb, values of c_v greater than 1 indicate a big amount of dispersion of the error rates, whereas values smaller than 1 indicate little dispersion of the error. For our set of 5 models, the dispersion of both the *misclassification rate* and the *classification loss* is observed to be little since it takes values of the order of magnitude of 10^{-1}.

Table 3.3: Statistical error measurements of 5 classification models for the manifolds of L_1 (unstable) and L_2 (stable) trained with different partitions of the data

	Misclassification rate							
	Model 1	Model 2	Model 3	Model 4	Model 5	μ	σ	c_v
L1	3.00E03	1.66E-03	1.00E-3	3.33E-04	1.33E-03	**1.47E-03**	**9.89E-04**	**6.74E-01**
L2	6.00E-03	2.33E-03	3.00E-3	1.67E-03	3.33E-03	**3.27E-03**	**1.66E-03**	**5.07E-01**
	Classification loss							
	Model 1	Model 2	Model 3	Model 4	Model 5	μ	σ	c_v
L1	2.96E-03	1.67E-03	9.88E-04	3.35E-04	1.32E-03	**1.46E-03**	**9.78E-04**	**6.72E-01**
L2	6.01E-03	2.34E-03	3.09E-03	1.70E-03	3.33E-03	**3.29E-03**	**1.65E-03**	**5.02E-01**

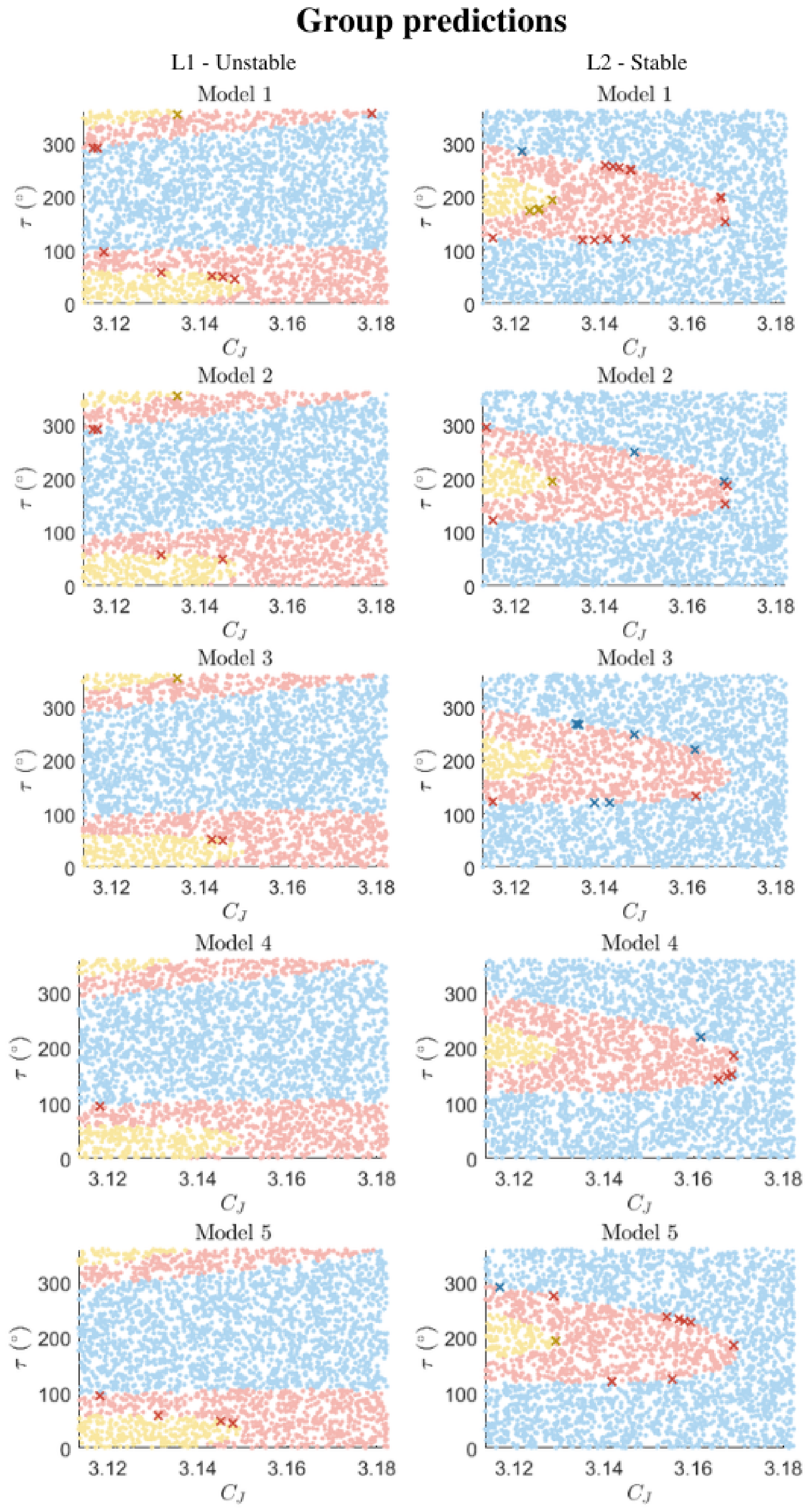


Figure 3.19: Comparison of the classification results among five equivalent models

After assessing the differences and similarities among different but equivalent classification models, a comparison among the performances of equivalent regression models is carried out. In order to do so, the Y and $\dot{Y}$ coordinates are predicted using the corresponding test set for each model. As a result, the predictions are shown in figures 3.20 and 3.21, and their respective relative error in figures 3.16 and 3.17. For all the models, a behaviour similar to what was described in the previous section is obtained; that is, for Y, an accumulation of higher error regions at the limits of the different regions can be perceived, while for the $\dot{Y}$, some higher errors seem to be concentrated around the heteroclinic connection area.

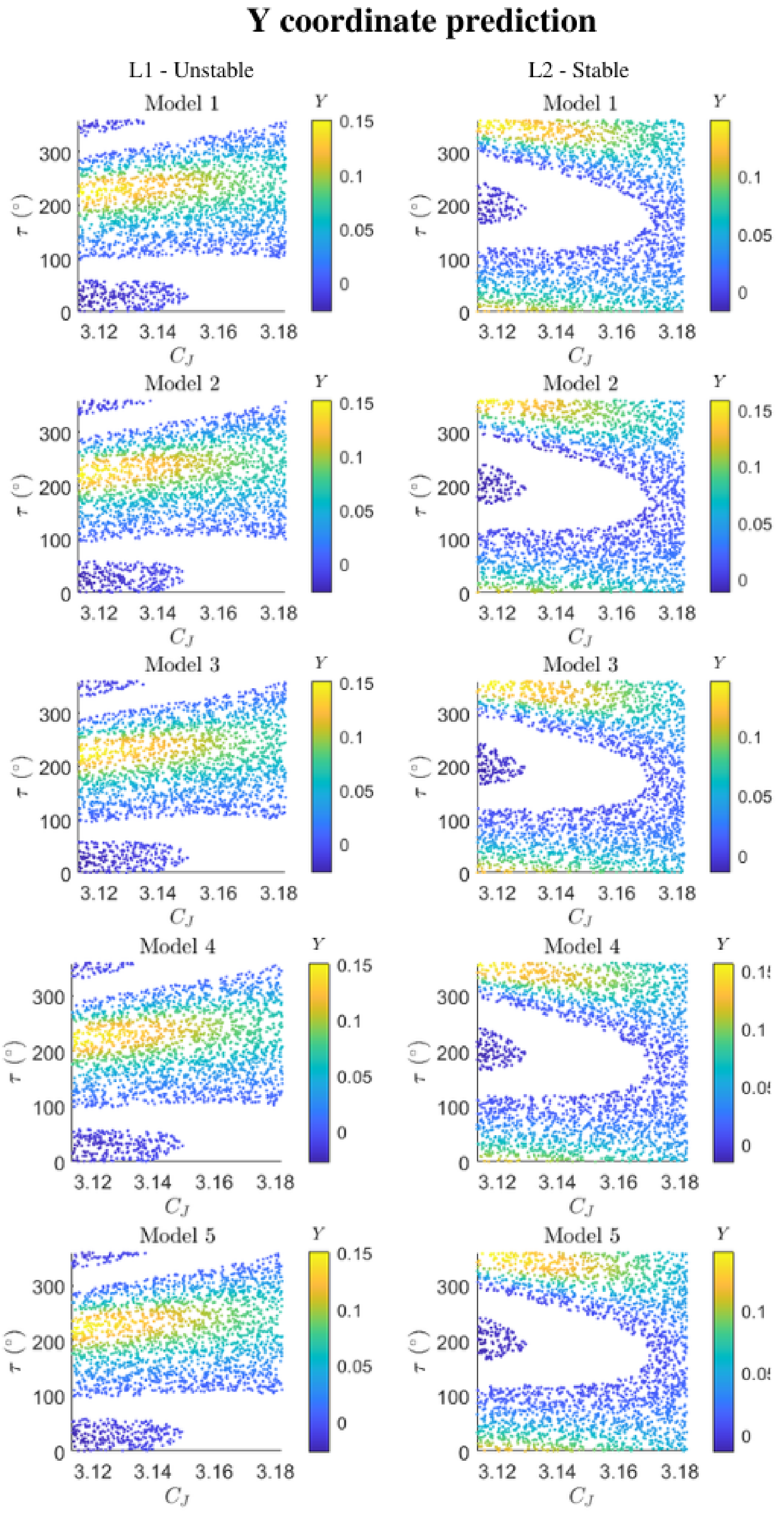


Figure 3.20: Comparison of the regression results in terms of the prediction of the Y coordinate among five equivalent models

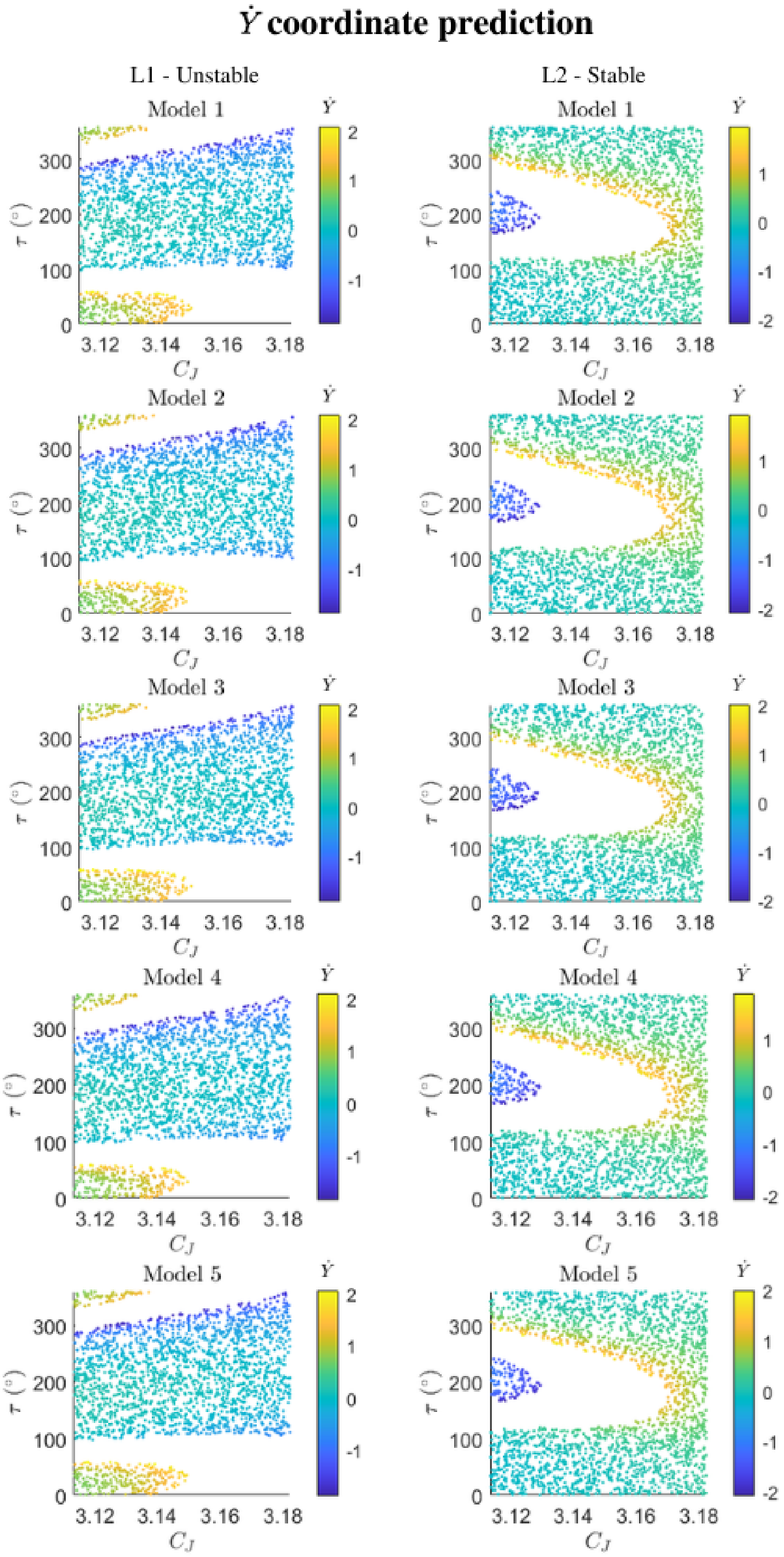


Figure 3.21: Comparison of the regression results in terms of the prediction of the Y coordinate among five equivalent models

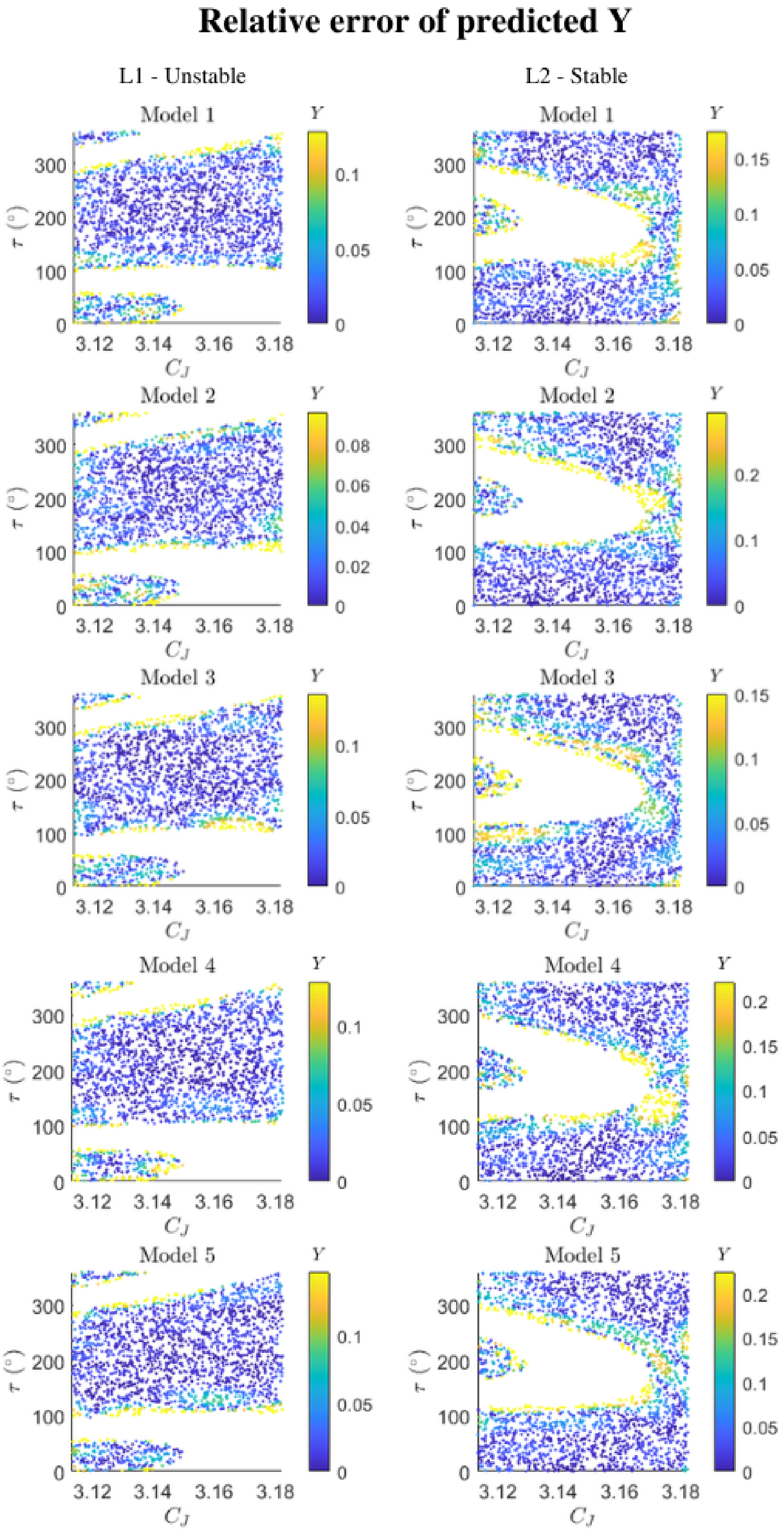


Figure 3.22: Relative error of the regression results in terms of the prediction of the Y coordinate among five equivalent models

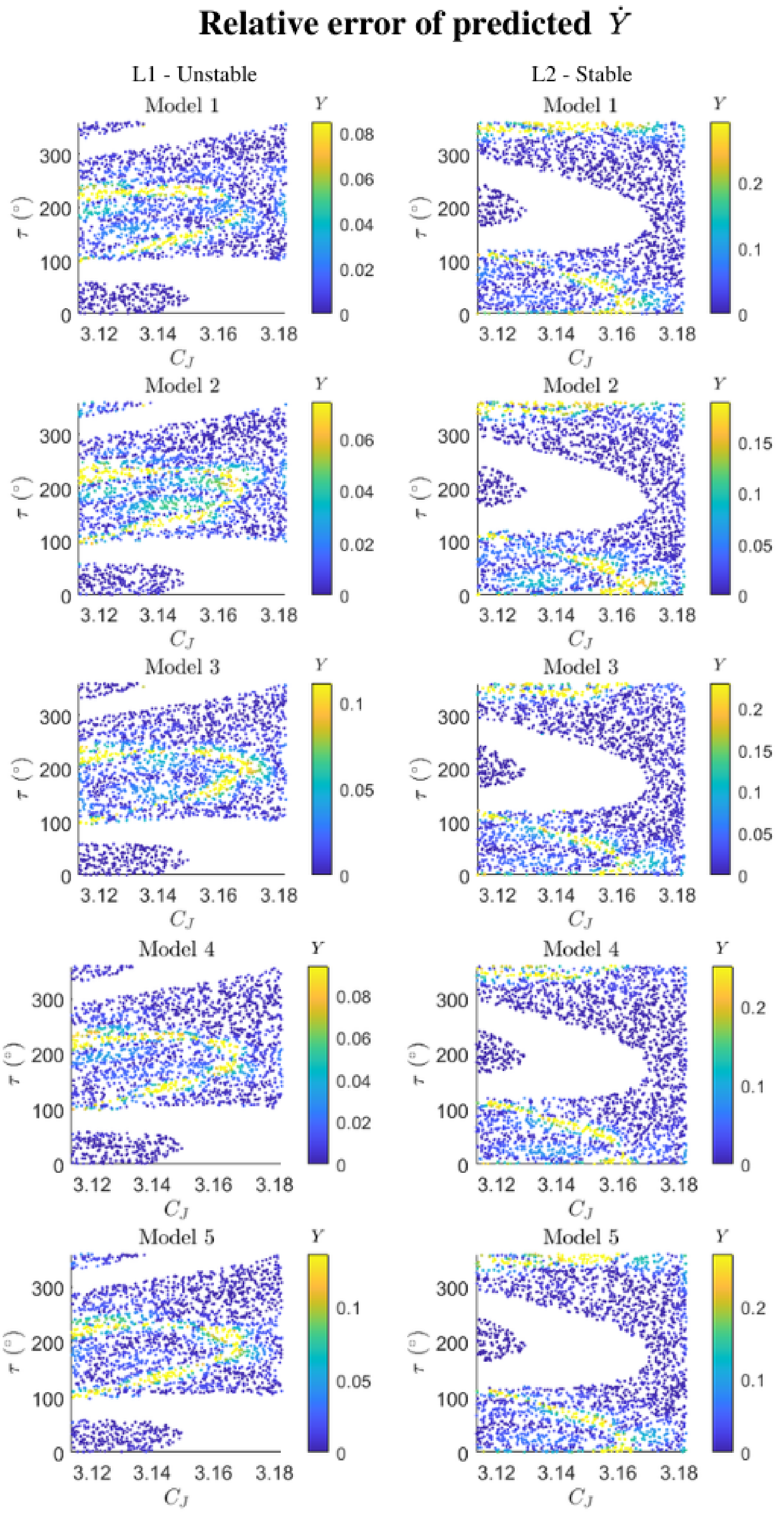


Figure 3.23: Relative error of the regression results in terms of the prediction of the $\dot{Y}$ coordinate among five equivalent models

As before, for a more detailed evaluation of the errors, the following error indicators are compared as well as , which are compiled in 3.4: MAE, MAPE, RMSE and RMSRE (see their formulas in section 3.4.1.). Again, with the aim of ensuring the robustness of the method proposed for the prediction of the manifold cuts with the Poincaré section, the coefficient of variation of the different errors is analysed. Since for all the cases that coefficient has an order of magnitude of 10^{-2} to 10^{-1} it is concluded that the regression ANNs created are robust.

Table 3.4: Statistical error measurements of 5 classification models for the manifolds of L_1 (unstable) and L_2 (stable) trained with different partitions of the data

	MAE							
	Model 1	Model 2	Model 3	Model 4	Model 5	μ	σ	c_v
$L_1(Y)$	7.19E-04	7.00E-04	7.43E-04	6.94E-04	7.16E-04	**7.15E-04**	**1.92E-05**	**2.69E-02**
$L_1(\dot{Y})$	2.18E-03	2.61E-03	2.53E-03	2.06E-03	3.20E-03	**2.51E-03**	**4.47E-04**	**1.78E-01**
$L_2(Y)$	1.30E-03	2.38E-03	1.55E-03	1.54E-03	1.49E-03	**1.65E-03**	**4.20E-04**	**2.54E-01**
$L_2(\dot{Y})$	5.54E-03	4.26E-03	4.21E-03	3.75E-03	5.10E-03	**4.57E-03**	**7.28E-04**	**1.59E-01**
	MAPE (%)							
	Model 1	Model 2	Model 3	Model 4	Model 5	μ	σ	c_v
$L_1(Y)$	3.26E+00	2.88E+00	3.63E+00	3.11E+00	3.46E+00	**3.27E+00**	**2.93E-01**	**8.97E-2**
$L_1(\dot{Y})$	5.87E+00	4.81E+00	5.62E+00	6.93E+00	10.35E+00	**6.72E+00**	**2.17E+00**	**3.22E-01**
$L_2(Y)$	5.01E+00	8.93E+00	5.75E+00	6.19E+00	6.26E+00	**6.43E+00**	**1.48E+00**	**2.31E-01**
$L_2(\dot{Y})$	9.97E+00	1.42E+01	1.42E+01	1.62E+01	1.02E+01	**1.29E+01**	**2.74E+00**	**2.12E-01**
	RMSE							
	Model 1	Model 2	Model 3	Model 4	Model 5	μ	σ	c_v
$L_1(Y)$	9.72E-04	1.02E-03	1.02E-03	1.02E-03	1.03E-03	**1.01E-03**	**2.27E-05**	**2.24E-02**
$L_1(\dot{Y})$	4.12E-03	5.03E-03	4.52E-03	3.11E-03	5.16E-03	**4.39E-03**	**8.27E-04**	**1.88E-01**
$L_2(Y)$	2.01E-03	4.12E-03	2.51E-03	2.40E-03	2.40E-03	**2.69E-03**	**8.25E-04**	**3.07E-01**
$L_2(\dot{Y})$	7.92E-03	6.19E-03	7.30E-03	5.35E-03	7.54E-03	**6.86E-03**	**1.06E-03**	**1.55E-01**
	RMSRE							
	Model 1	Model 2	Model 3	Model 4	Model 5	μ	σ	c_v
$L_1(Y)$	6.95E-02	6.62E-02	8.31E-02	6.67E-02	7.56E-02	**7.22E-02**	**7.12E-03**	**9.85E-02**
$L_1(\dot{Y})$	8.66E-01	6.08E-01	9.86E-01	9.75E-01	2.77E+00	**1.24E+00**	**8.67E-01**	**6.99E-01**
$L_2(Y)$	8.31E-01	2.18E-01	1.26E-01	1.13E-01	1.21E-01	**1.32E-01**	**5.07E-02**	**3.83E-01**
$L_2(\dot{Y})$	5.23E-01	3.39E+00	3.05E+00	3.27E+00	6.70E-01	**2.18E+00**	**1.45E+00**	**6.66E-01**

To ensure that the predictive engine formed by a combination of ANNs can be used in the range of application with confidence, an analysis of the sensitivity of the networks to the specific data used for training, such as the one presented here, is required. Only when its performance does not show great variance among the equivalent models can the model be considered valid.

To summarize, it can be said from the observations made for the different equivalent models that the method developed for creating an ANN predictor that infers the manifold crossings with the chosen Poincaré section is valid, since no major fluctuations of the results are observed when comparing models that were created using the same methodology and shuffeled variations of the data.

3.5. Results from the ANN curve fitting method

The results from the *ANN curve fitting* method are compared here against the results from the cubic interpolation explained in the Methodology section. Figures 3.24 and 3.25 show a comparison between said methods.

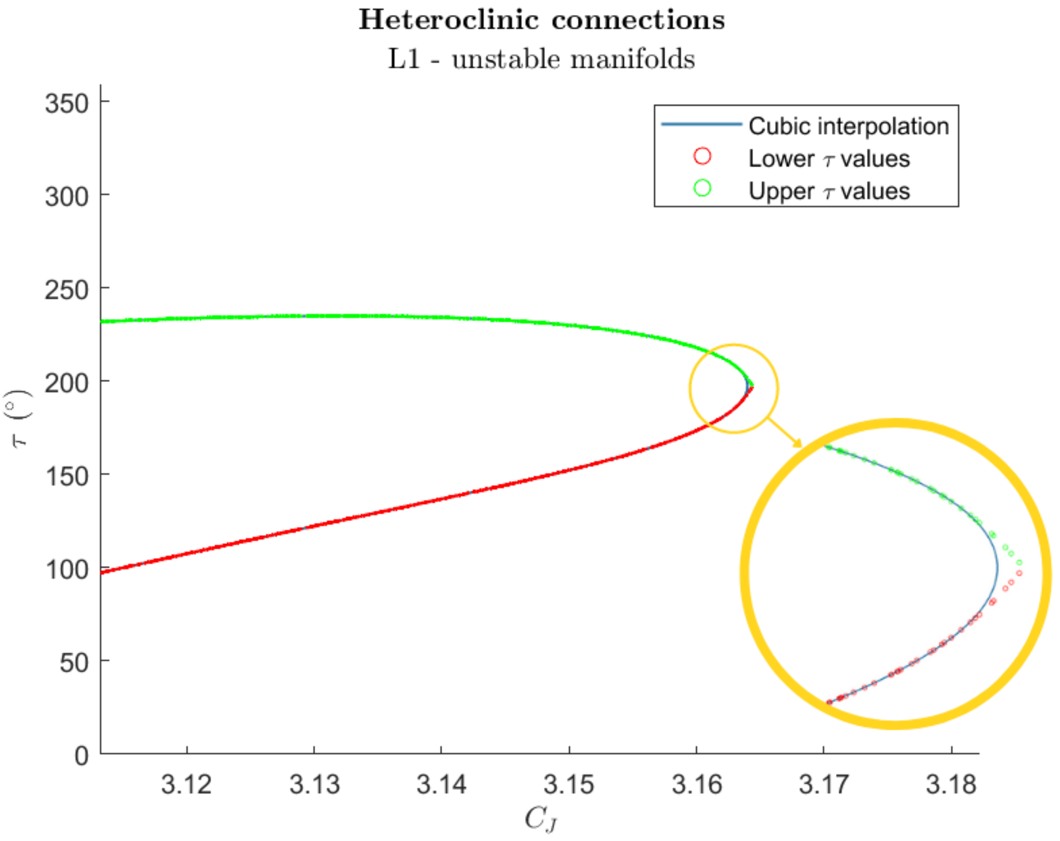


Figure 3.24: Predicted pairs of C_J and τ values of the unstable manifolds of L_1 that form heteroclinic connections by means of the ANN curve fitting method vs. the approximation of the curve through cubic interpolation

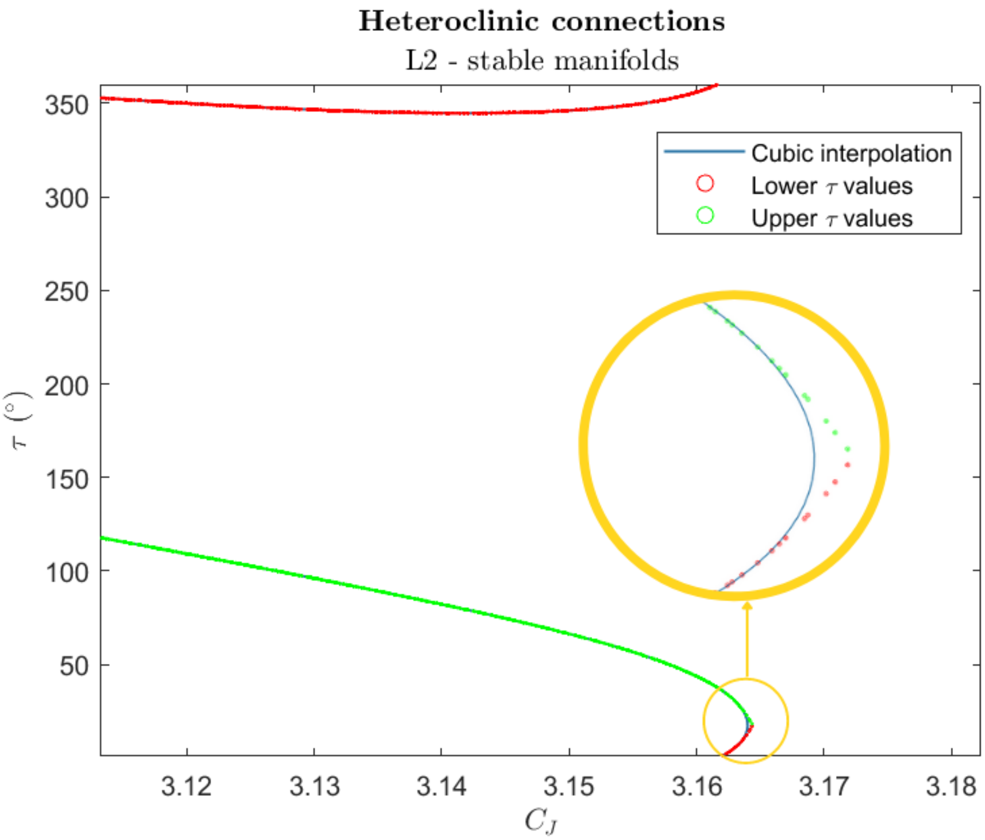


Figure 3.25: Predicted pairs of C_J and τ values of the stable manifolds of L_2 that form heteroclinic connections by means of the ANN curve fitting method vs. the approximation of the curve through cubic interpolation

On the one hand, the predictions of the trained ANNs are calculated for 1000 random values of C_J in the range shown in equation (1000 for the upper branch and 1000 for the lower branch, according to the criteria indicated in the legend), and on the other hand, a function in polar coordinates is fitted by means of a cubic interpolator. By plotting the predictions over the interpolated function, it is possible to observe that with the ANN method significant errors are found when approaching the limit where the upper and lower branches meet. This is due to the fact that, as it is not possible to train a neural network to map an input value to two different output values, the choice was made to train a pair of networks separately: one for the upper branch and another for the lower branch, for each libration orbit. That is why the neural network does not behave as well in the extreme, where the slope changes substantially but there are not many points of the training set, so the network is not capable of picking up that change.

3.6. Comparison of the three methods to compute heteroclinic connections

After separately analysing the results obtained from employing the three methods explained in sections 3.2.1., 3.2.2.1. and 3.2.2.2., they are compared in this section. It is particularly interesting to study how well the unstable and stable manifolds from the departure and arrival Lyapunov orbits that are computed to be the heteroclinic connections by the algorithms, match at the Poincaré section. In order to do so, three example cases are shown here, each with a randomly unique C_J value within the range where heteroclinic connections are known to exist, that has not been seen by the neural networks before. It is also worth mentioning here that when the random C_J value fell in the range where no heteroclinic connections exist, the three methods correctly predicted that no such transfers could be found.

As can be seen in figures 3.26, 3.27 and 3.28, the heteroclinic connections are successfully computed for all three cases. For the cases $C_J = 3.1176$ and $C_J = 3.1154$, in the solution of method (b) there is a gap between the unstable and stable manifold when they reach the Poincaré section that can be appreciated at first sight in the figures. For this reason, we can intuit that this method will not achieve an accuracy comparable to the reference method. However, it is inferred that both the classical method (with which it is possible to obtain an arbitrarily high accuracy), and the two new ANN methods developed in this work, can be considered equivalently good to be used as a seed in a realistic ephemeris model. This is because real trajectories of the bodies in the Solar System are not as close to the Restricted Three Body Problem model used here as a reference, and despite the fact that said model is an acceptable approximation in early stages of mission design where preliminary guesses are enough, it is necessary when defining a real mission to use more realistic models of ephemeris [3]. Some modern examples of ephemeris are Jet Propulsion Laboratory Development Ephemeris or JPL, Ephemeride Lunaire Parisienne and VSOP.

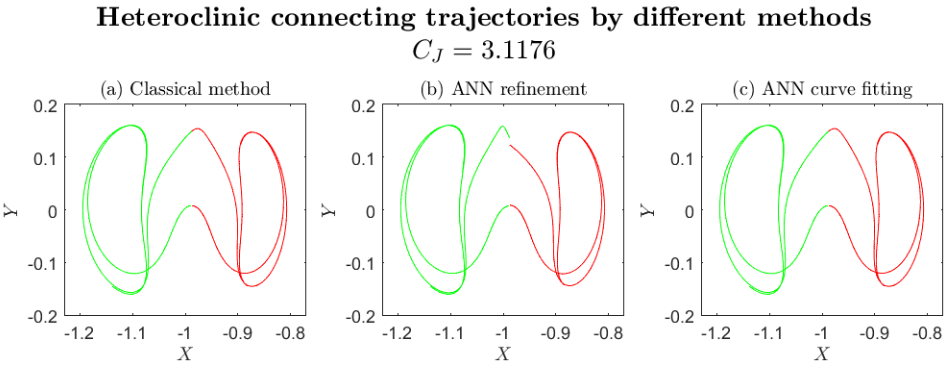


Figure 3.26: Case 1 - Representation of the trajectories of the planar heteroclinic connections from L_1 (in red) to L_2 (in green) computed though three different methods

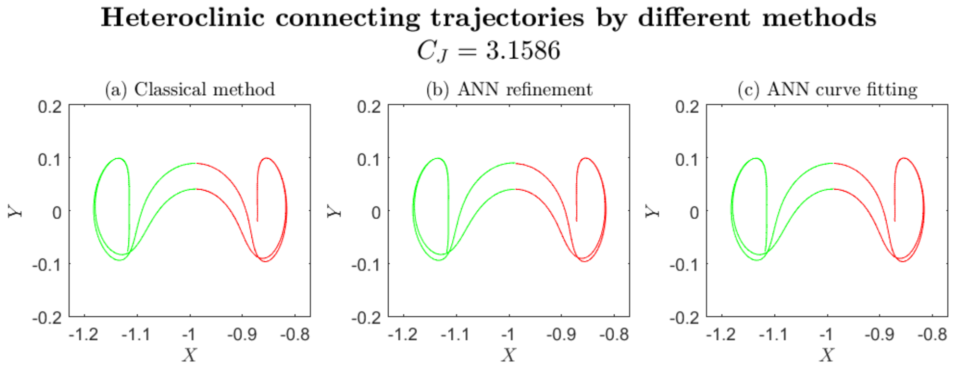


Figure 3.27: Case 2 - Representation of the trajectories of the planar heteroclinic connections from L_1 (in red) to L_2 (in green) computed though three different methods

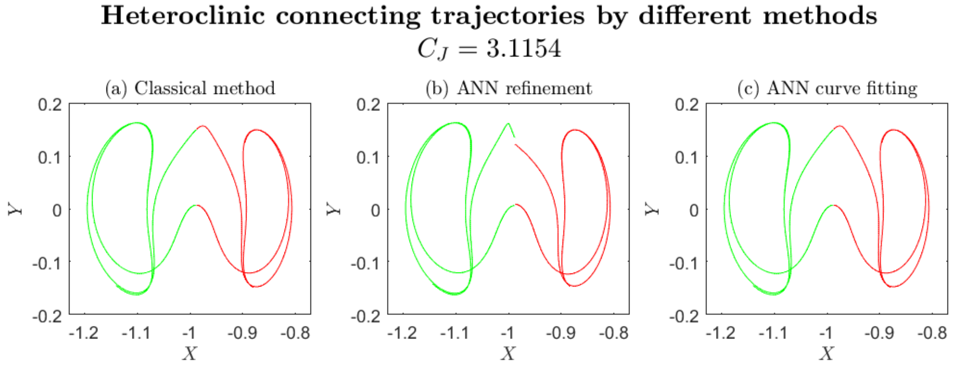


Figure 3.28: Case 3 - Representation of the trajectories of the planar heteroclinic connections from L_1 (in red) to L_2 (in green) computed though three different methods

In order to study the differences in the accuracy obtained for each method in detail, a compilation of the errors made by each one of them when computing the tau parameter where an heteroclinic connection is found (using the method classic as a reference) is shown in table 3.5. Moreover tables 3.6 and 3.7 show the matching gaps between the unstable and stable manifold, in terms of the Y and $\dot{Y}$ coordinates.

Table 3.5: Comparison of the τ parameter of the heteroclinic connections obtained by the three different methods. The reference value, τ_{ref}, is shown as computed by the classic method (a), as well as the difference between the reference and the τ obtained with the ANN refinement method (b) and with the ANN curve-fitting method (c). That difference for method *k* and manifold of L_i is computed as $|\Delta\tau^{L_i}_{(k)}| = |\tau^{L_i}_{(k)} - \tau^{L_i}_{ref.}|$

		L_1-unstable			**L_2-stable**		
		$\tau_{ref.}$	$\lvert\Delta\tau_{(b)}\rvert$	$\lvert\Delta\tau_{(c)}\rvert$	$\tau_{ref.}$	$\lvert\Delta\tau_{(b)}\rvert$	$\lvert\Delta\tau_{(c)}\rvert$
Case 1	1st H.C.	103.99	2.56E+00	1.12E-03	112.21	4.83E-01	1.70E-04
$C_J = 3.117648$	2nd H.C.	233.18	2.79E+01	6.44E-04	350.93	1.26E+01	2.48E-03
Case 2	1st H.C.	169.72	1.61E-01	6.06E-03	47.93	6.26E-01	8.77E-03
$C_J = 3.158603$	2nd H.C.	220.72	4.87E-01	3.20E-03	353.42	1.24E+00	9.67E-03
Case 3	1st H.C.	100.65	4.88E+00	4.78E-03	115.03	1.98E+00	7.32E-04
$C_J = 3.115433$	2nd H.C.	232.59	2.96E+01	4.24E-04	351.89	1.27E+01	2.11E-03

Analyzing the data for the τ parameter (see table 3.5), a clear difference is observed between the accuracy of the methods (b) and (c): while the values obtained from $|\Delta\tau_{(b)}|$ are in orders of magnitude between 10^{-1} and 10^{1}, the values of $|\Delta\tau_{(c)}|$ are in orders of magnitude between 10^{-4} and 10^{-3}. In other words, we obtain greater accuracy in calculating the τ using the ANN curve-fitting (c) method than using the ANN refinement method (b).

Table 3.6: Matching gap between coordinate Y of the unstable and stable manifold cuts with the Poincaré section, for the three different methods: (a) classic method, (b) ANN refinement method and (c) ANN curve-fitting method. The matching error for method k, is computed as $\epsilon^{Y}_{(k)} = |Y^{L_1}_{(k)} - Y^{L_2}_{(k)}|$

		$\epsilon^{Y}_{(a)}$	$\epsilon^{Y}_{(b)}$	$\epsilon^{Y}_{(c)}$
Case 1	1st H.C.	3.11E-14	1.19E-03	4.91E-07
$C_J = 3.117648$	2nd H.C.	1.01E-11	1.27E-02	5.91E-06
Case 2	1st H.C.	2.15E-13	3.30E-04	1.23E-05
$C_J = 3.158603$	2nd H.C.	7.99E-12	1.28E-03	9.65E-06
Case 3	1st H.C.	2.67E-13	2.58E-03	1.47E-06
$C_J = 3.115433$	2nd H.C.	1.02E-11	1.13E-02	3.78E-06

As for the data in tables 3.6 and 3.7, it shows how well the unstable manifold that departs from L_1 matches the stable manifold that approaches L_2 to form an heteroclinic connection. Once again, a better performance can be observed with method (c) than with (b) in terms of the closeness between the states of the ends of the manifolds. While for the classic method, said maximum distance is set to a fixed value (in our case the error both in Y and $\dot{Y}$ is limited to 10^{-10} at most), for the ANN methods greater errors are obtained: while the order of magnitude of $\epsilon^{Y}_{(b)}$ and $\epsilon^{\dot{Y}}_{(b)}$ is between 10^{-4} and 10^{-2}, for $\epsilon^{Y}_{(c)}$ and $\epsilon^{\dot{Y}}_{(c)}$ their order of magnitude is between 10^{-7} and 10^{-5}.

Table 3.7: Matching gap between coordinate $\dot{Y}$ of the unstable and stable manifold cuts with the Poincaré section, for the three different methods: (a) classic method, (b) ANN refinement method and (c) ANN curve-fitting method. The matching error for method k, is computed as $\epsilon^{\dot{Y}}_{(k)} = |\dot{Y}^{L_1}_{(k)} - \dot{Y}^{L_2}_{(k)}|$.

		$\epsilon^{\dot{Y}}_{(a)}$	$\epsilon^{\dot{Y}}_{(b)}$	$\epsilon^{\dot{Y}}_{(c)}$
Case 1	1st H.C.	1.56E-11	1.41E-02	7.09E-06
$C_J = 3.117648$	2nd H.C.	8.62E-11	1.99E-02	2.87E-05
Case 2	1st H.C.	3.28E-12	1.83E-03	7.73E-06
$C_J = 3.158603$	2nd H.C.	1.59E-11	1.62E-03	1.33E-05
Case 3	1st H.C.	1.87E-11	1.43E-02	4.71E-05
$C_J = 3.115433$	2nd H.C.	4.06E-11	2.68E-02	1.36E-05

Besides the above-mentioned indicators of accuracy, a major consideration that must also be taken into account when evaluating and comparing the usefulness of the two ANN methods in the context in which they are to be used, is the computational cost that is required to build the prediction engine related to each one, together with the corresponding time that that supposes. The prediction engine that uses the ANN refinement method does not require to be fed with the "exact" solution of the heteroclinic connections for the training process; the only training data needed to obtain a successful solution is a sufficiently densely populated collection of the Y and $\dot{Y}$ coordinates of the manifold cuts with the Poincaré section. On the other hand, in order to be trained, the predictor that uses the ANN curve-fitting method does require a collection of exact solutions (C_J and τ values) that provide the initial conditions for the heteroclinic connections. The significant advantage of not needing a collection of highly accurate solutions in the case of the ANN refinement method, is a great reduction of the computational cost when generating the training data, since the refining of the solution by means of successive propagations that make use of numerical integration is the most time-consuming part of the numerical implementation.

CONCLUSION

In this work, some new ANN-based methods for numerically computing heteroclinic connections between planar Lyapunov orbits in the Earth-Moon system for a given Jacobi constant is proposed.

To ensure that the predictive engine formed by a combination of ANNs can be used in the range of application with confidence, an analysis of the sensitivity of the networks to the specific data used for training is required, and only when its performance does not show great variance among the equivalent models can the model be considered valid. As for the predictive model developed in this work by means of the ANN refinement method, the results showed that it was able to produce robust results.

On the other hand, with regards to the hyper-parameter fitting of the networks for the ANN refinement method, the use of different types of activation functions resulted in the hyperbolic tangent type function being the best choice to minimise the error of the predictions, according to the error measurements provided by the k-fold cross validation data.

The two new ANN methods developed in this work, can be considered as good as the classical method to be used as a seed in a realistic ephemeris model. This is because real trajectories of the bodies in the Solar System are not as close to the Restricted Three Body Problem model used here as a reference, and despite the fact that said model is an acceptable approximation in early stages of mission design where preliminary guesses are enough, it is necessary when defining a real mission to use more realistic models of ephemeris [3].

Future work

A list of suggestions on how this research can be extended in future work is presented here:

- In order to obtain better results from the ANN refinement method, a broader hyper-parameter fitting can be performed such as the optimisation of the number of fully connected layers and neurons per layer.

- It would be interesting to see how the size of the datasets used in the training of ANNs influences their predicting performance and try to identify some criteria for determining the minimum amount of data that would provide a sufficiently accurate model of the problem, as well as the maximum amount of data that would be allowed before signs of overfitting start to appear.

- With the aim of seeing how well the proposed ANN-based methods generalize to other problems with different primary bodies, the same approaches could be extended and applied to the Sun-Earth system.

- Other similar ANNs can be designed and employed to predict if given a state in the neighbourhood of the small primary, its trajectory would propagate to be transit or non-transit orbits. The transit orbits are the trajectories that pass from one realm of the Restricted Three Body Problem to another [20]. Because the invariant manifolds of the libration orbits act as the separatrices of the transit and non-transit orbits, the ones that can be [15] are those inside the tubes. This could be a useful tool for mission design since given any state of a spacecraft going through a space of interest, its direction could be easily predicted at a very low computational cost.

- Similarly to the previous idea, other ANNs can be designed to classify the neighbouring states of a libration orbit depending on whether the would move away from the orbit to the right of the left.

BIBLIOGRAPHY

[1] Bishop, C.M. *Neural Networks for Pattern Recognition*. New York: Oxford University Press, ISBN 0198538642, (1995). 23

[2] Canalias E and Masdemont, J. J. “Homoclinic and heteroclinic transfer trajectories between Lyapunov orbits in the Sun-Earth and Earth-Moon systems”. *Discret and continuous dynamical systems*. **14**(2006), 261–279. 30

[3] Canalias Vila, E. Contributions to Libration Orbit Mission Design using Hyperbolic Invariant Manifolds [online]. Docoral thesis, UPC, Escola d’Enginyeria de Telecomunicació i Aeroespacial de Castelldefels, 2007. ISBN 9788469089736. [Retrieved: 24 May 2021]. Available at: < http://hdl.handle.net/2117/93242> 1, 7, 9, 31, 63, 67

[4] De Smet, S. and Scheeres, D.J. “Identifying heteroclinic connections using artificial neural networks”. *Acta Astronautica*. **161**(5), 192–199. (2019) 1, 7, 12, 15, 31, 42

[5] Everitt, B. *The Cambridge dictionary of statistics* [on line]. Repr. with corrections. Cambridge: Cambridge University Press, 1998. ISBN 0521593468 [Retrieved: 11 November 2021]. Available at: <https://archive.org/details/cambridgediction00ever_0/page/n7/mode/2up>. 27

[6] Garnham, Alan. “Introduction”. *Artificial intelligence : an introduction*. (Routledge Kegan Paul. 1988): 01–24. 15

[7] Garnham, Alan. “Knowledge representation”. *Artificial intelligence : an introduction*. (Routledge Kegan Paul. 1988): 24–44. 15

[8] Garnham, Alan. “Vision, Thinking and reasoning, Language, Learning, ”. *Artificial intelligence : an introduction*. (Routledge Kegan Paul. 1988): 57–270. 15

[9] Gómez, G.; Jorba, À.; Simó, C.; Masdemont, J. *Dynamics and Mission Design Near Libration Points. Volume III: Advanced Methods for Collinear Points*. Barcelona: World Scientific Monograph Series in Mathematics, 2001. 9

[10] Gómez, G.; Jorba, À.; Simó, C.; Masdemont, J. *Dynamics and Mission Design Near Libration Points. Volume I: Fundamentals: The Case of Collinear Libration Points*. Barcelona: World Scientific Monograph Series in Mathematics, 2001. 9

[11] Guzzetti, D. “Reinforcement Learning and Topology of Orbit Manifolds for Stationkeeping of Unstable Symmetric Periodic Orbits.”. *AAS/AIAA Astrodynamics Specialist Conference.***Portland (2019)**. 15

[12] Greer, Dale. “Constants and Equations for Calculations”. *USEnet SCI.SPACE*. Available at: <https://science.ksc.nasa.gov/facts/faq04.html>,(1994). 29

[13] Haykin, Simon. “Introduction”. *Neural networks and learning machines*. (Prentice Hall/Pearson): 11–17.(2009) 16, 18, 19, 21

[14] Jorba, A. and Masdemont, J. J. “Dynamics in the Center Manifold of the Collinear Points of the Restricted Three Body Problem”. *Physica D*. **132**(1999), 189–213. 9, 10

[15] Koon, W. S.; Lo, M. W.; Marsden, J. E.; Ross, S. D. “Heteroclinic Connections between Periodic Orbits and Resonance Transitions in Celestial Mechanics”. *Chaos*. **10**, 427–469. (2000) 68

[16] LaFarge, N., Howell, K. and Linares, R. “A Hybrid Closed-Loop Guidance Strategy for Low-Thrust Spacecraft Enabled by Neural Networks”. *31st AAS/AIAA Space Flight Mechanics Meeting, Charlotte, North Carolina (Virtual)*. Orlando, Florida (2021). 15

[17] MATLAB. “loss”. . [Retrieved: 25 May 2021]. Available at: <https://www.mathworks.com/help/stats/classificationlinear.loss.html>. 26

[18] MATLAB. “Cross-Validation”. . [Retrieved: 5 June 2021]. Available at: <https://www.mathworks.com/discovery/cross-validation.html>. 24

[19] Poincaré, H. “Sur le problème des trois corps et les équations de dynamique”. *Acta Mathematica*. **13**, 1–270. (1890) 3

[20] Qi, Rui Xu, Shijie Zhang, Yao Wang, Yue. (2012). Earth-to-Moon Low Energy Transfer Using Time-dependent Invariant Manifolds. AIAA/AAS Astrodynamics Specialist Conference 2012. 10.2514/6.2012-4431. 67

[21] Rosenberg, R. M. “Títol del capítol”. *Chapter 19: About Celestial Problems*. (Plenum Press. New York. 1977): 364–371. 3

[22] Ross, S.; Koon, W.; Lo, M.; Marsden, J. (2011). *Dynamical Systems, the Three-Body Problem, and Space Mission Design [on line].* 2011. [Retrieved: 13 June 2021]. Available at: <http://www.cds.caltech.edu/~koon/book/>. 12

[23] Sullivan, C.J. and Bosanac, N. “Using Reinforcement Learning to Design a Low-Thrust Approach into a Periodic Orbit in a Multi-Body System”. *20th AIAA Scitech Forum*. Orlando, Florida(2020). 15

[24] Szebehely V. *Theory of Orbit*. Academic Press, 1967. ISBN 9780123957320. 8

[25] The MathWorks, Inc. Machine Learning with MATLAB [Self-Paced Online Course]. [Retrieved: 24 May 2021]. Available at: <https://matlabacademy.mathworks.com/details/machine-learning-with-matlab/mlml>. 26

[26] Weisberg, M. “Robustness Analysis”. *Philosophy of Science*. **73**(5), 730–742. (2006) 51

[27] Williams David R. Breen, P. G.; Foley, C. N.; Boekholt, T.; Portegies Zwart, S. “Newton versus the machine: solving the chaotic three-body problem using deep neural networks”. *Monthly Notices of the Royal Astronomical Society*. **494**(2), 2465–2470. (2020) 1, 31, 34, 42

[28] Williams, David R. “Moon Fact Sheet”. *NASA Goddard Space Flight Center*. Available at: <https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html>,(2020). 29