POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Executive Summary of the Thesis

Physics-informed neural network for gravity field modeling around Didymos binary system

Laurea Magistrale in Space Engineering

Author: Marco Galeazzi

Advisor: Prof. Francesco Topputo

Co-advisor: Mattia Pugliatti, Prof. Fabio Ferrari

Academic year: 2022-2023

---

1. Introduction

The study of small solar system objects, such as asteroids and comets, are a key instrument to understand the formation of our solar system and the origin of life. Current estimates indicate that 16 % of the Near Earth Asteroid population is made of binary systems. Future missions that will investigate these binary systems could require planning a set of robust close proximity operations around small bodies. Such operations could be challenging and can be complicated by numerous factors such as the irregular shape and mass distribution of a body and its weak and uncertain gravitational field. Due to such factors, the orbital dynamics around small bodies could deviate from the ideal Keplerian motion. The knowledge of the dynamics driving the motion of a body in the vicinity of a binary system is then a key point for the success of the mission. In [3, 4] a new method to investigate the gravity field of an asteroid using a physics-informed neural network (PINN) is presented. The model seems to have a fast computation time once the PINN is trained, it can describe the potential of an asteroid with the same accuracy of other models while using a smaller number of parameters to do so and can be also modeled using only the acceleration measurements. The PINN model could be also theoretically used in-situ to model the gravitational acceleration.

This MSc Thesis main goal then will be to extend the work done by the author to a binary asteroid system. In particular, the Didymos-Dimorphos system gravitational field is studied. This work is divided in two parts. The first part is focused on modeling the gravity field of the binary asteroid system using PINN with data generated from some known model and compares the results with other traditional methods. Instead, in the second part, the PINN are trained from simulated acceleration measurements to understand to what extend PINN can be used to map the gravity field in-situ.

2. Methodology

2.1. Physics-Informed Neural Network

An artificial neural networks (ANN) is a computerized model inspired by the human brain's structure and functioning. It consists of interconnected nodes (neurons) organized in layers, with each connection having a weight, and aims to map input data to desired output by adjusting these weights during a training process. Essentially, it's a tool for solving complex problems and recognizing patterns in data. However, one of the disadvantages of using ANN to represent a function found in physics is that the learned representation may not satisfy the fundamental properties of said function. In [5] this problem is addressed and a method to train the ANN to ensure that the learned representations obey the differential equations that govern the system is proposed. To this end, PINN are introduced. PINNs add the differential equations into the cost function of a traditional neural network and use automatic differentiation to ensure that these equations are respected by the function learned by the network. These extra terms serve as a form of regularization in the training process which can lead to improved solutions that conveniently also satisfy important physics properties. The PINN model could also be useful to obtain a better approximation of the function in the presence of noisy training data. However, despite this additional robustness, these constraints could increase the amount of training time necessary for the PINN to converge given the computational complexity of calculating the derivatives of the function (especially when second order derivatives or beyond are considered). Also, in case multiple physics objectives are considered in the loss function, they could have competing gradient flow dynamics (i.e. the different objectives have different learning behaviors which may prevent some objectives from being leveraged during training). In order to avoid this, a weight could be associated at each physic objective and also a learning rate annealing algorithm can be implemented in order to adapt the weights during the training. The derivatives of the network output with respect to the network inputs are taken with automatic differentiation (AD). This is a method for calculating derivatives of a function's output with respect to its input variables. It provides precise and efficient gradients by decomposing the function into a sequence of elementary operations and applying the chain rule.

2.2. Environment

In order to train the Neural Network, a reference model is created in order to generate the data required for the training. The gravity field of Didymos is modelled using a polyhedron while for Dimorphos a triaxial ellipsoid is used as a reference. For both models a constant density is assumed. Three reference frames are considered in this work, the two body reference frame of each asteroid and an inertial reference frame centered in the barycenter of the binary system. In case the total acceleration $\ddot{\boldsymbol{r}}$ provided by the binary system must be computed, it can be determined as the summation of the acceleration provided by each asteroid as [1]:

$$\ddot{\boldsymbol{r}} = \boldsymbol{a}_{didymos} + \boldsymbol{a}_{dimorphos}$$

Where $\boldsymbol{a}_{didymos}$ and $\boldsymbol{a}_{dimorphos}$ represent the acceleration provided by each asteroid in the inertial frame.
The acceleration provided by each asteroid is computed in the body frame of the same asteroid and both accelerations must be rotated in the inertial reference frame.
The contributions of the solar radiation pressure and of the third-body effects of the Sun were not considered because for the training of the PINNs only the acceleration provided by the asteroids are required. In case the PINNs are trained using measurements of the total acceleration, it is assumed that the instrument used is able to measure directly the gravity field as it is in line with real space instruments used.

2.3. Method implementation

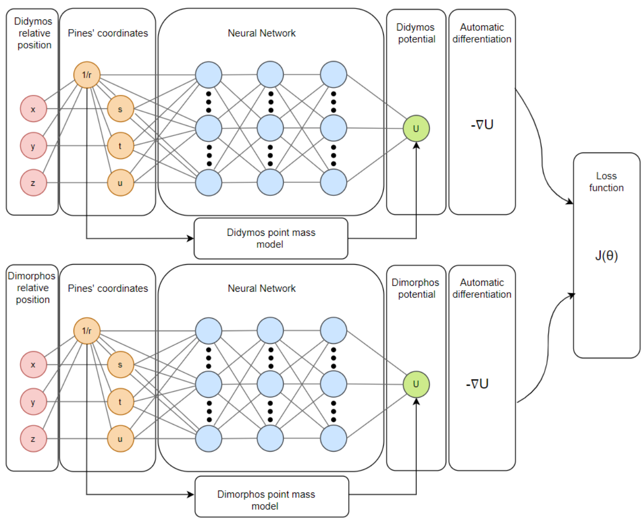

Figure 1: Architecture of the model used for mapping the gravity field when trained from total acceleration measurements

The architecture of the PINNs used to model the gravity field is build by taking inspiration from [3, 4]. The model was implemented in Python with the usage of TensorFlow 2.10. In case only total gravitational acceleration measurements are known, the architecture consists of two separate networks trained in parallel with a common loss function as shown in Figure 1.
In case the network is trained from some known model, two separate PINNs can be trained independently. Each PINN will be used to map the potential of a single asteroid. The input for each PINN is the relative position of the field point with respect to the corresponding asteroid with coordinates in the body frame of the same celestial body and returns as an output the potential of the asteroid in the field point. To train the PINNs, the following algorithm is used:

Algorithm 1 How to train the Network

```
Collect training data from a known model or from acceleration measurements
Non-dimensionalize the training data
for epoch in n_epochs do
    Convert the input position (r=norm([x, y, z]), x/r=s, y/r=t, z/r=u)
    Compute the output of both networks
    Add the point mass model to both networks output
    Re-scale the output of both networks
    Auto differentiate the potential with respect to the cartesian coordinates expressed in the body frame to compute the accelerations
    Compute the loss function
    Update the parameters of the network
end for
```

After the training, the total gravitational acceleration in a field point can be computed using automatic differentiation and by computing the contribution of the singular asteroid acceleration. Then, both accelerations are rotated from the body frame to the inertial reference frame and are added together. The loss function used for the training depends if the network is trained from total acceleration measurements or a known model. In case the networks are trained from some known models, two different loss functions can be used. Each loss function will be used to impose that the negative gradient of the potential of an asteroid is equal to the acceleration provided by the asteroid:

$$J(\boldsymbol{\Theta}) = \frac{1}{N}\sum_{i=0}^{N} \frac{\|\boldsymbol{a}_{true,i} - (-\nabla U_{NN,i})\|}{\|\boldsymbol{a}_{true,i}\|}$$

In case the model is trained from real acceleration measurements, a single loss function is used to impose the percentage error between the total acceleration measured and the sum of the two accelerations provided by each asteroid equal to zero. The accelerations are rotated from the body frame of each asteroid to the inertial frame before summing them. In case the potential is known, the percentage error between the true potential and the potential obtained from the networks can be imposed equal to zero. Finally, the curl of the acceleration and the laplacian of the potential can be also imposed equal to zero (or equal to the Poisson's equation when inside a body in the case of the laplacian). The curl and the laplacian can be computed using AD, however this will increase the training time and the memory needed as second order derivatives are computed. In this work, all the different terms of the loss function are tested to see which ones produce the better performances. In addition to these loss functions, an ANN is also considered. In this case, the output of the Neural Network is not the potential of the body but instead are the three components of the acceleration.
All inputs and outputs are non-dimensionalized in order to have values ranging from -1 to 1. 1/r, s, t, u are used as input of the network in order to improve the performances. The inverse of the radius is used in order to have values ranging from 1 to zero after the dimensionalization. The point mass model is added to the output in order for the PINN to not map this prominent and easily observable contribution. In this way, it can focus on mapping the potential's perturbations. The potential is rescaled by dividing the output by the cube of the radius in order to improve the performance far away from the body.

3. Results

3.1. Training with a known model

The distances of the field points considered for the training are between the surface of the asteroid up to a distance of 5 radii. The radius for each field point is generated using a random uni-

form distribution. All the field points are generated only in the exterior of the asteroids.
Between all the different loss functions used to train the model, the one that includes the potential and the acceleration seems to perform slightly better with respect to the other loss functions considered. In case the point mass model is included in the model, an adaptive weight of the terms of the loss function is necessary if the laplacian and the curl term are included. If the adaptive weight is not used, the model in this case converge to the point mass model independently on the number of data and epochs used for the training. When the adaptive weight is included, the performance are pretty similar to the case where only the potential and the acceleration where included in the loss function. However, the time of training and the memory needed to train the model are much greater due to second order derivatives that must be computed.
A comparison of the PINN methods implemented in this work with the traditional methods is made below:

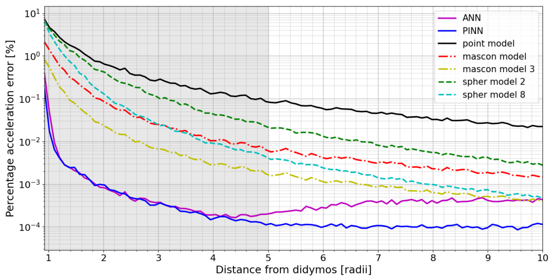

Figure 2: PINN Didymos model compared with traditional models

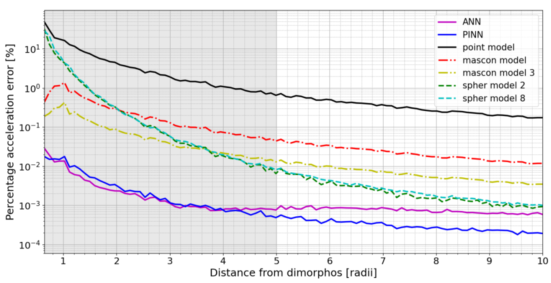

Figure 3: PINN Dimorphos model compared with traditional models

The polyhedral and the ellipsoid models are used as a reference, the mascon model is created by dividing the polyhedron in a collection of tetrahedra and assigning the point mass in the center of each tetrahedra. In particular, a single mass was considered per tetrahedra for mascon model, while 3 masses where considered for mascon model 3. For the spherical harmonics, the spherical harmonics coefficients were computed from the polyhedral shape. A spherical harmonics of order 2 and order 8 were considered. As we can see, the PINN methods performs better with respect to the other models in proximity of the surface. As the distance from the asteroid increases, it seems that the spherical model gets closer with the performances and at distances near 10 radii the spherical model has errors in the same order with respect to the PINN model.
The computational time can also be tested in order to understand which method compute faster. To do this, 1000 different field points are generated and the acceleration is computed for each point individually. The time to compute each single acceleration is measured and the mean time is retrieved. The computer used for the time comparison has an NVIDIA Quadro P1000 as a graphic card and an Intel core i7-8850h CPU.

Model	Time needed [s]
Point	3.52 e-6
Polyhedral Didymos	2.49 e-1
Ellipsoid Dimorphos	5.63 e-3
Mascon	1.07 e-2
Mascon 3	3.19 e-2
Spherical 2	1.01 e-4
Spherical 8	6.09 e-4
ANN	3.21 e-3
PINN	5.50 e-3

Table 1: Computational time

Besides the point model, the spherical harmonics performs the fastest. It should also be noted that the computational time of the PINN and of the Ellipsoid model is pretty similar. It would then be unnecessary to train the PINN in the case of Dimorphos as the PINN model would take the same amount of time to compute the accelerations with respect to the reference model and would only perform worse with respect to it. A combination of PINN and spherical harmonics or point model could be used in order to

compute the acceleration of a field point. PINN could be used to compute the acceleration of field points in proximity of the surface while faster models could be used when they are located far away from the asteroid. In this way, the accuracy would still be high while the overall computational speed is increased.
In proximity of the asteroid the performance of the network becomes worse, especially in the case of Didymos. To solve this, the number of data in proximity of the surface can be increased. For example this can be done using an exponential distribution of the field points in proximity of the surface. These data are added to the ones generated normally. Using data in proximity of the surface does increase the performance in that region. However, the performance in the rest of the domain gets worse when using the same number of data. The generation of data in proximity of the surface would then depend on the requirements of the mission. If operations in close proximity of the surface are required, it is suggested to generate them, otherwise not. Similar results could be obtained by changing the loss function in order to give more weight to field points in proximity of the surface.

3.2. Training with acceleration measurements

In order to train the model with total acceleration measurements, data are generated in proximity of both Didymos and Dimorphos. The field points are generated using a uniform random distribution as before. However, in this case, the field points used for the training must be the same for both Didymos and Dimorphos. The training set will be made by the union of samples taken in proximity of both asteroids. The loss functions trained in this section do not consider the potential as it assumed unknown. The training and the validation data are generated in a time domain between 0 and 250 hours from the initial position and orientation of the asteroids. The time domain is chosen by assuming that a 30 seconds measurement time is required in order to measure each single acceleration. This value is in line with real instruments used in space applications. The test set is generated in a time domain that follows the time domain of the training set in order to understand if the PINN is able to map the gravity field even for positions and orientations of the asteroids never seen before. By changing the sampling time domain, the performance of the network do not change much. This is because the acceleration of the asteroids is modelled individually and, given the same position of the field point in the body frame, the acceleration would be the same independently on the position and on the orientation of the asteroids. For this reason, by also changing the orbit and asteroids orientation the results do not change.
The results obtained from this training are pretty similar to the ones obtained from the case where each network was trained with its own model when the same number of data and epochs are considered.
In order to understand better the possible performances of the model in real life applications, some errors are added to the measurements. A root mean square error (RMS) in the range of $[10^{-6}, 10^{-8}]$ is assumed.

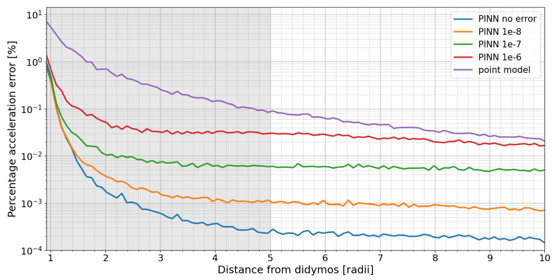

Figure 4: Error comparison in proximity of Didymos

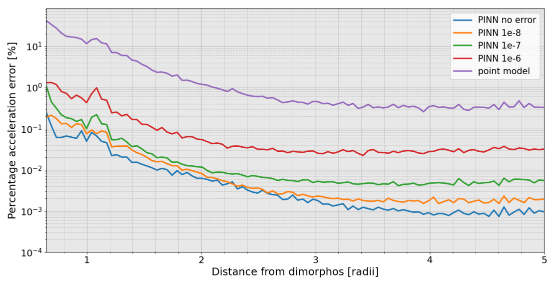

Figure 5: Error comparison in proximity of Dimorphos

The model is still capable of mapping the gravity field even in the presence of errors although with worse performances as the error gets bigger.
As the error in the measurements is increased,

the performance when the loss function includes the laplacian and curl term gets better with respect to the case with only the acceleration term when trained with the same data.
By training the PINNs using the point model as a reference, one problem arises. It could be that the mass that we have used is not the same of the real asteroid. If an error of the mass is assumed, the performance of the PINNs gets worse, especially when outside of the training data domain. Before training the model, an estimation of the mass of each asteroid shall be made in order to reduce this error.
In order to understand if the sampling of data in proximity of Didymos influences negatively with the performances of the model in proximity of Dimorphos and vice-versa, the number of data used for the training set in proximity of both asteroids is varied. In particular, the number of samples in proximity of Didymos and Dimorphos is changed. Only when a low number of samples are considered in proximity of an asteroid the mapping in proximity of the same asteroid gets worse. In both cases, it seems that when considering 3000 or more samples the model starts to converge in both regions for a total sampling time of 50 hours as the sampling for each measurement is assumed of 30 seconds.
It could be that the measurements near the surface of an asteroid are not permitted. To simulate this case, a domain from 2 to 5 radii of distance from both asteroids is considered for the training and validation set. In this case, only the performance of the model in proximity of the surface seems to be affected. Measurements in proximity of the surface would then be advised in order to map better the total acceleration in that region. In the region with a corresponding domain for both cases the performance seems pretty similar.

4. Conclusions

This work demonstrates that the PINN is able to produce high-accuracy models for the gravity field of the Didymos(65803) binary system. The PINN model can reach performances close to the best known gravity field models of both asteroids (polyhedral for Didymos and ellipsoid model for Dimorphos) while increasing the computational speed in the case of Didymos. In the case of Dimorphos, the computational speed is close to the reference model used, making its usage for modeling the gravity field redundant. Compared to spherical and mascon models, PINN performs better, especially in proximity of the surface. However, spherical model can compute much faster with respect to the PINN model. In order to map the whole gravity field, a combination of PINN and spherical harmonics could be implemented. PINN will be used to map the acceleration in proximity of the surface while spherical harmonics will be operated when considering field points far away from the asteroids.
The PINNs can also be used to map the gravity field in-situ from real total acceleration measurements using a fairly low number of data. When considering errors in the measurements, the PINN is still able to map the gravity field fairly well. In this case, it is recommended to impose the Laplacian of the potential and the curl of the acceleration equal to zero with the loss function while using an adapting loss weight. This will decrease the error while mapping the gravity field. However, this will increase the training time and the memory needed for the training.

References

[1] F. Ferrari, V. Franzese, M. Pugliatti, C. Giordano, and F. Topputo. Trajectory options for hera's milani cubesat around (65803) didymos. *J Astronaut Sci*, 68:973–994, 2021.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[3] J. Martin and H Schaub. Physics-informed neural networks for gravity field modeling of small bodies. *Celest Mech Dyn Astron*, 134(46), 2022.

[4] J. Martin and H Schaub. Physics-informed neural networks for gravity field modeling of the earth and moon. *Celest Mech Dyn Astr*, 134(13), 2022.

[5] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.