Multi-Objective Autonomous Spacecraft Motion Planning around Near-Earth Asteroids using Machine Learning

CS 229 Final Project - Fall 2018
Category: Physical Sciences

Tommaso Guffanti
SUNetID: tommaso@stanford.edu

1 Problem Statement

A spacecraft (SC) capable to autonomously plan its motion while accounting for conflicting mission objectives in a Pareto optimal way would permit to accomplish complex mission tasks around highly uncharacterized celestial bodies such as near-Earth asteroids (NEAs). The two most common conflicting objectives of a space exploration mission are the maximization of the scientific output and the minimization of the control effort, i.e. the propellant required on board. If the targeted celestial body is sufficiently known and has a predictable orbital environment, both numerical and analytical tools can be leveraged on-ground to design spacecraft motion plans that account for the trade-off. On the contrary, if the celestial body and its orbit environment are largely uncertain, all plans elaborated on-ground may fail dramatically when implemented in space. A clear example are missions around NEAs having relevant dynamics parameters (i.e. mass, shape, rotation axis orientation and gravity coefficients) largely uncertain. In these missions, a spacecraft should be capable of autonomously plan its motion when an updated knowledge of the NEA environment is provided by the sensors and the navigation filter. In addition, the generated motion plan should account for the trade-off science-output vs. control-effort in an optimal sense.

2 Solution Approach using Machine Learning

To provide a spacecraft with such Pareto optimal autonomous planning capabilities is a huge challenge. In this project, the solution approach to the problem combines machine learning (both reinforcement and supervised) and numerical multi-objective optimization. In particular, assuming a value for the NEA dynamics parameters, an heuristic multi-objective optimization algorithm is used to generate a Pareto front describing the trade-off offered by various motion plans according to two conflicting cost functions. The two cost functions to be minimized provide metric of: 1) the control effort required to realize the motion plan ($J_{\Delta V}$), 2) the inverse of the quality/quantity of scientific output perceivable through realization of the motion plan ($J_{science}$). The Pareto front obtained relies on the assumption of the NEA dynamics parameters, when these parameters are changed different Pareto fronts are obtained. To identify a specific point on a Pareto front, which corresponds to a specific trade-off between the two conflicting costs, the multi-objective problem can be scalarized [1] as $J = \boldsymbol{\lambda} \cdot [J_{\Delta V}, J_{science}]^T = \lambda_1 J_{\Delta V} + \lambda_2 J_{science}$, where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2]$ and $\lambda_{1,2} \in (0, 1)$. λ_1 and λ_2 represent the relative weigths associated to the two objectives.

Casting this problem into a reinforcement learning (RL) setting, the state ($\mathbf{s}$) is defined as

$$\mathbf{s} = [\mathbf{p}_{NEA}, \boldsymbol{\lambda}] \in \mathbb{R}^{n_s} \tag{1}$$

where $\mathbf{p}_{NEA}$ are the uncertain NEA dynamics parameters. In this project

$$\mathbf{p}_{NEA} = [m_{NEA}, R_{NEA}, \boldsymbol{\mathcal{G}}_{NEA}, \boldsymbol{\alpha}_{NEA}] \tag{2}$$

where, m_{NEA} is the NEA mass, R_{NEA} is the NEA mean radius, $\boldsymbol{\mathcal{G}}_{NEA}$ are the NEA most relevant gravity coefficient, $\boldsymbol{\alpha}_{NEA}$ are the NEA orientation parameters with respect to the inertial frame. The policy ($\boldsymbol{\pi}$) is defined as

$$\boldsymbol{\pi} = \boldsymbol{\mathfrak{oe}}_{SC} \in \mathbb{R}^5 \tag{3}$$

where $\boldsymbol{\mathfrak{oe}}_{SC} = [a_{SC}, e_{SC}, i_{SC}, \Omega_{SC}, \omega_{SC}]$ are the spacecraft orbit elements around the NEA, i.e., a way to parameterize the spacecraft position a velocity around the NEA [2]. In this project, the goal will be limited to predict just 3 of the 5 components of $\boldsymbol{\mathfrak{oe}}_{SC}$, in particular, the orbit semi-major axis, a_{SC}, the orbit eccentricity, e_{SC}, and the orbit inclination, i_{SC}. The RL algorithm should match a given state $\mathbf{s}$ (i.e., NEA dynamics parameters and defined trade-off control effort-science output) with the corresponding optimal policy $\boldsymbol{\pi}^*(\mathbf{s})$, which represents the optimal spacecraft orbit configuration around the NEA given the specified trade-off. In order to do so it has to maximize a reward function – $R(\mathbf{s}, \boldsymbol{\pi})$ – defined as

$$R(\mathbf{s}, \boldsymbol{\pi}) = -J(\mathbf{s}, \boldsymbol{\pi}) = -\boldsymbol{\lambda} \cdot [J_{\Delta V}(\mathbf{p}_{NEA}, \boldsymbol{\pi}), J_{science}(\mathbf{p}_{NEA}, \boldsymbol{\pi})]^T \tag{4}$$

Therefore, the optimal policy is defined as

$$\boldsymbol{\pi}^*(\mathbf{s}) = \boldsymbol{\mathfrak{oe}}^*_{SC}(\mathbf{s}) = \arg\max_{\boldsymbol{\pi}} R(\mathbf{s}, \boldsymbol{\pi}) = \arg\min_{\boldsymbol{\pi}} J(\mathbf{s}, \boldsymbol{\pi}) = \arg\min_{\boldsymbol{\pi}} \left\{ \boldsymbol{\lambda} \cdot [J_{\Delta V}(\mathbf{p}_{NEA}, \boldsymbol{\pi}), J_{science}(\mathbf{p}_{NEA}, \boldsymbol{\pi})]^T \right\} \tag{5}$$

Ideally, this optimization problem should be solved on-line, in order to obtain the optimal spacecraft motion plan ($\boldsymbol{\mathfrak{oe}}^*_{SC}$), given the updated $\mathbf{p}_{NEA}$ (passed by the navigation filter) and the selected trade-off ($\boldsymbol{\lambda}$) . Practically, this is not possible, since solving Eq.5 entails: 1) to run the numerical multi-objective optimization algorithm with input $\mathbf{p}_{NEA}$, which embeds the simulation of the spacecraft non-linear dynamics around the NEA and evaluates control effort and science cost functions, 2) to get the Pareto front and identify the Pareto point corresponding to the trade-off $\boldsymbol{\lambda}$. This is a non-convex problem which entails considerable computational effort, out of the possibility of any spacecraft CPU. Therefore, there is the need to leverage as much as possible off-line on-ground computation, and limit the on-line part implemented on the spacecraft computer as (ideally) a lookup table which given the current state provides the policy.

Here comes into play Supervised Learning (SL). By performing $m-$runs of the multi-objective optimizer for various NEAs dynamics parameters, a database of paired couples: dynamics parameters-Pareto fronts $\left(\mathbf{p}^{(i)}_{NEA}, PF^{(i)}\right)$, for $i = 1, \dots m$, is generated. Each $PF^{(i)}$ is discretized in n_{PF} Pareto points, each one corresponding to an optimal policy $\boldsymbol{\pi}^{*(i)}_k(\mathbf{p}^{(i)}_{NEA}) = \boldsymbol{\mathfrak{oe}}^{*(i)}_{k,SC}(\mathbf{p}^{(i)}_{NEA})$ associated to the trade-off $\boldsymbol{\lambda}^{(i)}_k$, for $k = 1, \dots n_{PF}$. Therefore, from the $i-th$ run of the multi-objective optimizer, n_{PF} state-policy pairs are generated as:

$$\left([\mathbf{p}^{(i)}_{NEA}, \boldsymbol{\lambda}^{(i)}_k], \boldsymbol{\pi}^{*(i)}_k\right) \quad \text{for } k = 1, \dots n_{PF} \tag{6}$$

Finally, after $m-$runs of the multi-objective optimizer, $n_{PF} \times m$ state-policy pairs are generated as:

$$\left([\mathbf{p}^{(i)}_{NEA}, \boldsymbol{\lambda}^{(i)}_k], \boldsymbol{\pi}^{*(i)}_k\right) \quad \text{for } k = 1, \dots n_{PF} \quad \text{for } i = 1, \dots m \tag{7}$$

The $n_{PF} \times m$ state-policy pairs can be partitioned in a training set, a development set, and a test set. A neural network (NN) is trained on the training set to then make predictions on the test set, the NN hyperparameters are optimized looking at the results achieved on the development set. The trained NN learns the functional relationship between states and policies. The trained NN can be then implemented on-board the spacecraft and used on-line between the navigation filter and the spacecraft controller as an autonomous motion planning unit (Fig.1). This unit takes as input the most recently estimated $\mathbf{p}_{NEA}$ and the selected trade-off $\boldsymbol{\lambda}$, and outputs the optimal spacecraft orbital configuration: $\boldsymbol{\pi}^* = \boldsymbol{\alpha}\mathbf{e}^*_{SC}$. This output is passed to the controller which takes action to reach this target configuration.

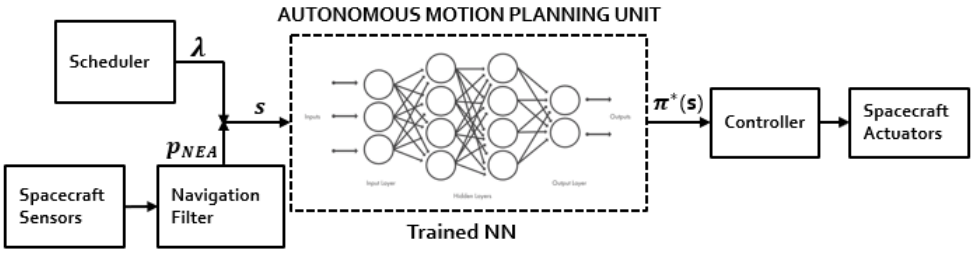

Figure 1: Autonomous spacecraft motion planning.

3 Dataset Generation

The multi-objective optimizer used is the Multi-Objective Particle Swarm Optimization (MOPSO), which is shown in literature to provide high level performances in terms of time of convergence and full reconstruction of the global Pareto front [3]. The Pareto front resulting from a single run of the multi-objective optimizer ($PF^{(i)}$) is presented in Fig.2. The control cost ($J_{\Delta V}$, x-axis) is expressed in control effort (velocity variation ΔV) required to maintain the orbital configuration $\boldsymbol{\alpha}\mathbf{e}^{*(i)}_{k,SC}$, which corresponds to a single point on $PF^{(i)}$, associated to the trade-off $\boldsymbol{\lambda}^{(i)}_k$. $J_{\Delta V}$ is therefore an index of orbit stability, and is evaluated by propagating the spacecraft trajectory starting at the nominal configuration $\boldsymbol{\alpha}\mathbf{e}^{*(i)}_{k,SC}$ for 10 orbital periods under the effect of the NEA non-linear orbital dynamics associated to $\mathbf{p}^{(i)}_{NEA}$ [4][5][6], and computing the control cost required to keep the nominal orbital configuration [7][8]. The science cost ($J_{science}$, y-axis) is expressed in camera resolution (meter/pixel) modified to account for the NEA coverage achievable from $\boldsymbol{\alpha}\mathbf{e}^{*(i)}_{k,SC}$. The camera model used takes inspiration from the XCAM C3D CubeSat Camera [9]. In general terms, to minimize the science cost (i.e., to maximize the science output quality) the spacecraft should go closer to the NEA (to increase the resolution) and at higher orbit inclinations (to increase the coverage). On the other hand, the closer to the NEA the more control action is required to maintain the orbit due to the more intense dynamics. Therefore there is a trade-off, and as a consequence a Pareto Front, $PF^{(i)}$, of non-dominated optimal orbital configurations each identified by a specific $\boldsymbol{\lambda}^{(i)}_k$ (Fig.2).

The multi-objective optimizer has been run $m = 96$ times (~ 48 hours), therefore capturing 96 possible NEA dynamics parameters configurations ($\mathbf{p}^{(i)}_{NEA}$ for $i = 1, \dots 96$). Taken uniformly through the whole spectrum of $\mathbf{p}_{NEA}$, 6 sample configurations have been extracted as the development set and 5 as the test set. This results in approximately 3400 state-policies pairs for the training set, 200 for the development set and 200 for the test set.

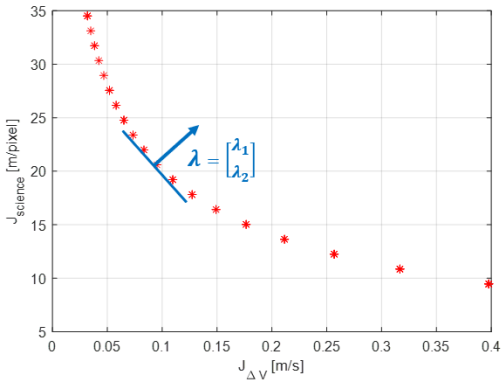

Figure 2: Pareto Front.

4 Results

The neural network (NN) weights have been trained on the training set using a mean square error (MSE) loss function. The NN hyperparameters have been tuned and optimized according to the performances provided on the development set [10]. Inputs and outputs of the NN have been normalized, the inputs between -1 and 1 (since they can be negative), the outputs between 0 and 1 (since they are all positive). The normalization factors are the maximum considered values of the various inputs and outputs. Initially, a randomized search varying number of layers and activation functions has been performed to identify a suitable NN configuration. This initial search made the NN model to converge to a configuration with four hidden layers: three layers with hyperbolic tangent activation function, plus one layer with sigmoid activation function. Justification regarding these activation functions is found in the fact that the normalized inputs have value comprised in $(-1, 1)$, whereas the normalized outputs have value comprised in $(0, 1)$. As mentioned previously, in this project the focus is on predicting a subset of the optimal spacecraft orbit elements, i.e., semi-major axis, a_{SC}, eccentricity, e_{SC}, and inclination, i_{SC}. To do so, a possible way to go is to train a single NN (with three output neurons) to minimize the global MSE on the prediction of a_{SC}, e_{SC} and i_{SC}. Another way to go is to design 3 separated NN, each one trained to minimize the prediction MSE of either a_{SC} or e_{SC} or i_{SC}. More accurate results on each of the three outputs are obtained pursuing the second path, therefore 3 NN have been trained. For each NN an hyperparameters optimization procedure has been carried out, focusing on the number of neurons per layer and on the batch size. For each NN, results of the hyperparameters optimization procedure are reported in Fig.3, where the contour plots of the root mean square error (RMSE) obtained on the development set are represented. The number of epochs considered for each training is always 10000 with and early stopping tolerance of 1000 epochs (i.e. the training is interrupted if the loss function does not improve after 1000 epochs). The optimizer used is Adam [11]. Looking at Fig.3 on the top left, the lowest RMSE(a_{SC}) is obtained with a 150 neurons per hidden layer and using at each epoch a batch size of one third of the training dataset. Looking at Fig.3 on the top right, the lowest RMSE(e_{SC}) is obtained with a 175 neurons per hidden layer and using at each epoch a batch size of half of the training dataset. Looking at Fig.3 on the bottom, the lowest RMSE(i_{SC}) is obtained with a 75 neurons per hidden layer and using at each epoch a batch size of on tenth of the training dataset. The dependency of the accuracy results from the batch size can be justified with the fact that batches of different size contain information about a different number of possible $\mathbf{p}_{NEA}$. Finally, results of prediction accuracy of the three optimized NN on the test set are reported in Tab.4 (last column), these are compared with the accuracy shown by the same NN on the training and development sets. The three NN show interesting

but still moderate accuracy results. To provide a way of evaluation, around NEA of different size, typical a_{SC} values are comprised in $(10, 70)$km, typical e_{SC} values are comprised in $(0.01, 0.4)$ and typical i_{SC} values are comprised in $(90, 180)$deg. An improvement of approximately one order of magnitude in both a_{SC}, e_{SC} and i_{SC} would be needed to make the NN predictions exploitable on-board a spacecraft. The first way to follow is surely to enlarge the dataset, which up to now is limited to only 96 possible NEA dynamics parameters configurations.

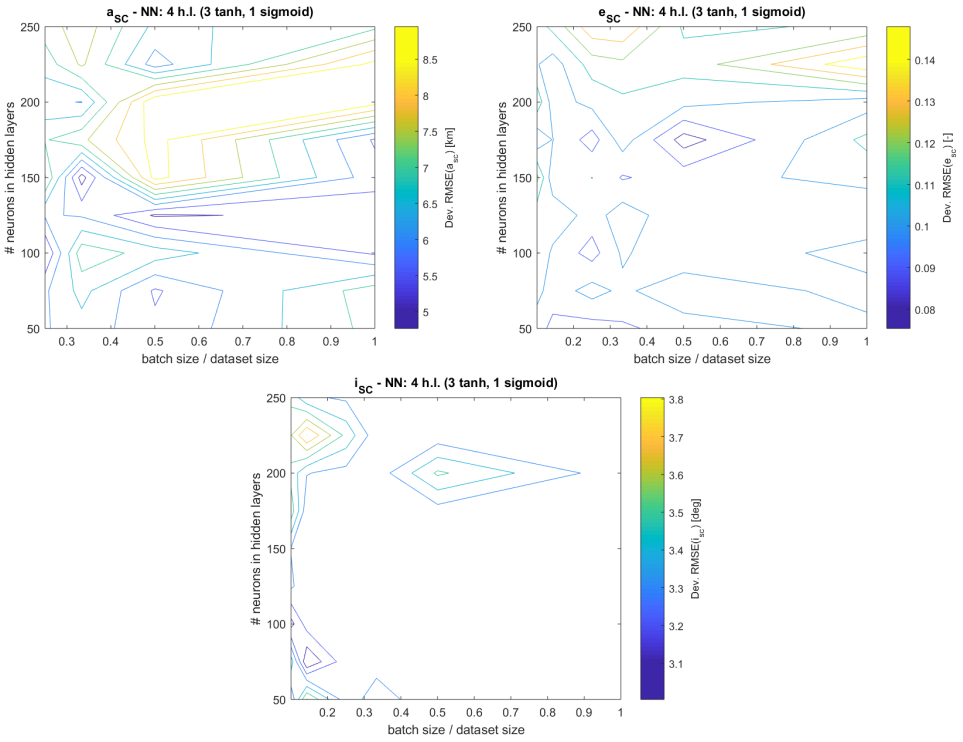

Figure 3: Hyperparameters tuning - a_{SC} NN (top left), e_{SC} NN (top right), i_{SC} NN (bottom).

	RMSE_{train}	RMSE_{dev}	RMSE_{test}
a_{SC} [km]	2.36	4.77	3.13
e_{SC} [-]	0.044	0.075	0.074
i_{SC} [deg]	0.39	3.00	2.43

Table 1: Accuracy results.

5 Conclusions and Way Forward

This project explores the use of neural networks (NN) to provide a spacecraft with autonomous multi-objective motion planning capabilities around a near-Earth asteroid (NEA). The trained NN has been shown to provide interesting but still moderate accuracy results. To improve the performances, the first way to follow is to enlarge the dataset. In addition, future work will explore ways to leverage information about the covariance of the estimated state, that the navigation filter outputs. In this sense, a possible way to go is to reformulate the problem as a stochastic Markov Decision Process.

Acknowledgements

I have developed this project advised by Prof. Simone D'Amico. The topic of this study has been motivated by the on-going research project: "Autonomous Nanosatellite Swarming using Radio Frequency and Optical Navigation", developed at Stanford's Space Rendezvous Laboratory in collaboration with NASA Ames Research Center and sponsored by NASA SSTP (Small Spacecraft Technology Program).

References

[1] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[2] Vallado, D. A., and McClain, W. D., *Fundamentals of Astrodynamics and Applications*, Space Technology Library, 2013.

[3] C.A. Coello Coello and G. Toscano Pulido and M. Salazar Lechuga, "Handling Multiple Objectives with Particle Swarm Optimization", *IEEE Transactions on Evolutionary Computation*, 2004.

[4] Scheeres, D.J., "Orbit Mechanics About Asteroids and Comets", *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 3, May-June 2012

[5] Koenig, A.W., Guffanti, T., D'Amico, S., "New State Transition Matrices for Spacecraft Relative Motion in Perturbed Orbits", *Journal of Guidance, Control, and Dynamics*, Vol.40, No.7, pp.1749-1768, September 2017.

[6] Guffanti, T., D'Amico, S., Lavagna, M., "Long-Term Analytical Propagation of Satellite Relative Motion in Perturbed Orbits", *27th AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, Texas, February 5-9, 2017.

[7] Kahle, R., D'Amico, S., "The TerraSAR-X Precise Orbit Control - Concept and Flight Results", *24th International Symposium on Space Flight Dynamics*, 5-9 May 2014, Laurel, USA (2014).

[8] Gaias, G., D'Amico, S., "Impulsive Maneuvers for Formation Reconfiguration Using Relative Orbital Elements," *Journal of Guidance, Control, and Dynamics*, Vol.38, No.6, pp.1036-1049 (2015).

[9] "XCAM C3D CubeSat Camera Datasheet," Northampton, United Kingdom, 2018.

[10] Ng, A.Y., *Machine Learning Yearning*, deeplearning.ai, 2018.

[11] Kingma, D., Ba, J., "Adam: A Method for Stochastic Optimization", Proceedings of the 3rd International Conference for Learning Representations, San Diego, 2015.

Link to project code: https://drive.google.com/open?id=1lIWeYJxLKEJ8ZN8-4VqRyWZo5s9M1rd5