LONG SHORT-TERM MEMORY NETWORKS TO IMPROVE AERODYNAMIC COEFFICIENT ESTIMATION FOR AEROCAPTURE

by

DOMINIC A RUDAKEVYCH

B.S., West Point, 2023

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirement for the degree of

Master of Science

Department of Applied Mathematics

2025

Committee Members:

Stephen Becker, PhD

Manuel Lladser, PhD

Jay McMahon, PhD

ABSTRACT

Rudakevych, Dominic Anthony (M.S., Applied Mathematics)

Long Short-Term Memory Networks to Improve Aerodynamic Coefficient Estimation for Aerocapture

Thesis directed by Associate Professor Stephen Becker

Aerocapture is a method for orbital insertion from a hyperbolic trajectory being considered for NASA’s proposed 2030’s flagship mission to Uranus. By traveling through the planet's atmosphere to generate drag, aerocapture greatly reduces the fuel needed when firing retrograde thrusters, allowing for larger payloads or a less powerful launch vehicle. Despite these theoretical benefits, aerocapture has never flown on any planetary missions due to thin margins of error and in-situ corrections necessary to properly execute the maneuver. Critical to the guidance and control algorithms are the aerodynamic coefficients. We propose using a neural network to learn the nonlinear relationship between the raw sensor data and these aerodynamic coefficients. Specifically, we explore how network architectures designed for time dependent data, like Long-Short Term Memory (LSTM) neural networks, can produce aerodynamic coefficient estimates akin to that of a computational fluid dynamics (CFD) based lookup table, while providing more robust coefficient estimation when large environmental perturbations are experienced. Improving force and moment coefficient estimation would improve aerocapture by providing more accurate aerodynamic coefficients for use in guidance and control algorithms. This work considers multiple sensed data sources and aerodynamic coefficient data along with LSTM network architectures for model training to maximize an aerocapture maneuver’s success rate when tested in a Monte Carlo simulation. Along with this, sensitivity analyses were

conducted on model hyperparameters to account for relationship complexity. Results were compared against traditional aerodynamic coefficient lookup tables within the Fully Numeric Predictor-corrector Aerocapture Guidance (FNPAG) algorithm to draw conclusions for the model's performance.

CONTENTS

TABLES

Table

FIGURES

Figure

CHAPTER I

Nomenclature

ΔV	=	change in velocity
Δm	=	change in mass
L	=	lift force
D	=	drag force
C_L	=	lift coefficient
C_D	=	drag coefficient
ρ	=	atmospheric density
S	=	reference area
α	=	angle of attack
β	=	sideslip angle
CFA	=	longitudinal force coefficient
CFY	=	lateral force coefficient
CFN	=	normal force coefficient
CMl	=	rolling moment coefficient
CMm	=	pitching moment coefficient
CMn	=	yawing moment coefficient

Background

Aerocapture

With the release of a Decadal Strategy for Planetary Science and Astrobiology, the National Aeronautics and Space Administration (NASA) has highlighted the Uranus Orbiter and Probe (UOP) flagship mission as the highest priority mission for 2023-2032 [1]. Through UOP, NASA aims to put an in-situ atmospheric probe in orbit around Uranus for multiple years, where it will gather knowledge on this unexplored ice giant and the Uranian system. The three objectives of UOP are to explore Uranus' (1) origin, interior, and atmosphere; (2) magnetosphere; and (3) satellites and rings.

Currently, optimal launch opportunities in the early 2030's have UOP leveraging a variety of gravity assists to reach Uranus in a shorter cruise time. Once the Uranian system is reached, NASA will enter orbit from a hyperbolic trajectory. One theoretical method for achieving this is aerocapture. Through aerocapture, a satellite dips into a planet's atmosphere, creating drag and decreasing velocity. Through precise guidance, navigation and control (GNC), a satellite would be able to leverage this drag to slow down and reach orbital velocity without slowing so much as to reach sub-orbital velocity. Aerocapture offers three potential benefits over propulsive orbit insertion. First, the mass of hardware and propellant needed for an aerocapture maneuver would be less than the mass needed for the hardware and propellant to insert a satellite entirely propulsively [2]. This difference in mass could be used to transport more scientific payload. The second potential benefit would be a reduction to the overall trip time from Earth to Uranus. This is a product of how propellant and hardware mass (Δm) scale with the necessary change in velocity (ΔV) for different insertion techniques. For entirely propellant insertions, Δm increases approximately exponentially with ΔV, while for aerocapture, Δm increases linearly with ΔV [3]. This linear relationship allows for hyperbolic entry speeds infeasible with propulsive orbit insertion, and thus lower flight times. Finally, aerocapture could allow less costly launch vehicles because of the reduced propellant and hardware mass [2].

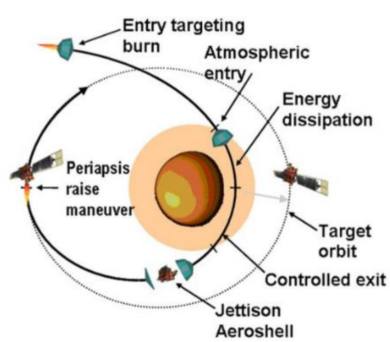


Fig. 1 Notional Aerocapture Maneuver [4]

Despite these potential benefits, aerocapture has not been used on a flight mission to date. While multiple flight tests and missions featuring aerocapture have been proposed, all have been unsuccessful on account of aerocapture's technological readiness level (TRLs). TRLs are a 1-9 scale system designed by NASA in the 1970s for estimating the maturity level of a particular technology [5]. At the extreme ends of the scale, TRL 1 indicates research has begun on a certain technology, but the work is still speculative, whereas TRL 9 suggests a fully functional technology which has been "flight proven" during a successful mission. The UOP flagship mission would be a multi-billion-dollar endeavor, meaning the risk tolerance is extremely low, and the TRL for the aerocapture maneuver would have to advance substantially on the TRL scale.

Estimating Lift and Drag Coefficients

The current state-of-the-art aerocapture guidance, the Fully Numeric Predictor-corrector Aerocapture Guidance (FNPAG), has been flight simulated in multiple environments, especially Earth and Mars [6, 7]. In these simulations, lift and drag coefficients are considered fixed on account of high velocity during aerocapture maneuvers [7]. Lift and drag coefficients are conventionally defined as

$$C_L = \frac{L}{0.5 * \rho * V^2 * S} \tag{1}$$

$$C_D = \frac{D}{0.5 * \rho * V^2 * S} \tag{2}$$

Equations (1) and (2) show the lift and drag coefficients have inverse square scaling with vehicle velocity. While (1) and (2) provide algebraic solutions for the coefficients, these are not usable online. This is because flights implementing FNPAG need accurate lift and drag coefficients to compute lift and drag forces, which are used in guidance and control for vital flight measurements including the L/D ratio and acceleration due to aerodynamics. Since the lift and drag forces are unavailable to the model, flights making online adjustments to lift and drag coefficients rely on the relationships between force coefficients and angle of attack defined below [8].

$$C_L = \mathrm{CFN}\cos\alpha - \mathrm{CFA}\sin\alpha \tag{3}$$

$$C_D = \mathrm{CFN}\sin\alpha + \mathrm{CFA}\cos\alpha \tag{4}$$

The longitudinal, lateral, and normal force coefficients used for estimating lift and drag coefficients are provided by lookup tables informed by wind tunnel and computational fluid dynamics (CFD) data which take angle of attack, sideslip angle, and Knudsen Number as table input. While these tables provide more robust lift and drag coefficient information for guidance, aerodynamics for destinations with demanding entry circumstances (such as Uranus) cannot be accurately modeled by extrapolation of laboratory data at lower Mach numbers [2].

Neural Networks

To account for uncertainties introduced by the Uranian atmosphere, we propose a data-driven technique for producing lift and drag coefficients during an aerocapture maneuver, using realistic numerical simulations to create a dataset. A neural network is a model designed after the biological learning process. The base learning unit, a neuron, takes in a collection of inputs to produce an output, which may then be fed to other neurons. Figure 2 shows a simple feed-forward neural network (FFNN) with three layers: an input layer of three neurons, a hidden layer of four neurons, and an output layer of two neurons. The connections (or weights) between neurons are updated during the learning process as the model identifies which relationships are relevant. For an error back-propagation learning scheme, weights are updated to travel the steepest gradient-descent to minimize the difference between the sum of squared errors between the model outputs and truth output values known during the training process [9].

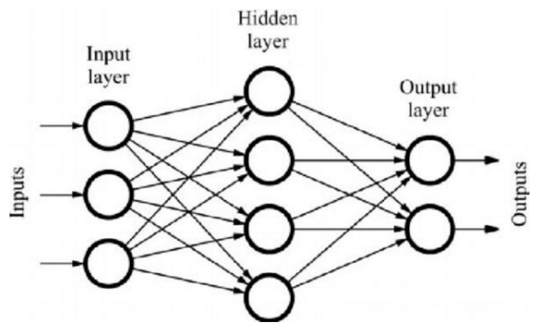


Fig. 2 FFNN Architecture [10]

The architecture of individual neurons allows neural networks to develop nonlinear mappings from inputs to outputs. Each neuron performs a mapping $F: R^n \rightarrow R$, wherein a vector of inputs, A_i, is multiplied by a respective vector of weights, w_i, and then adjusted by a bias vector, b_i, after which all the values are summed. The weighted input sum is then used as input to a nonlinear activation function F, which maps to the neuron output, Z. Equation 5 demonstrates this relationship

.

$$Z = F\left(\sum_{i=1}^{n} w_i A_i + b_i\right) \tag{5}$$

The ability for neural networks to learn nonlinear relationships comes from the use of nonlinear activation functions. Common activation functions include the Sigmoid function, the Rectified Linear Unit (ReLU), and the Tanh function [11]. In this work, the sigmoid and tanh activation functions are used.

While powerful, FFNNs are preferred for handling data in a single pass, while other network architectures learn trends in time-dependent data better. Recurrent neural networks (RNNs) are often utilized for their internal "memory" structure and ability to handle sequences of variable length. The key difference is allowing the output of a previous timestep, Z^{t-1}, to be used as input for the current timestep [12]. Long short-term memory neural networks (LSTMs) are a type of RNN designed to model temporal data and capture long term dependencies, and one of the most popular types of RNNs. LSTMs overcome certain weaknesses of general RNNs, such as the long-term time dependencies [13], which make them attractive for the task of estimating aerodynamic coefficients for an aerocapture maneuver. At each time step in an LSTM, four neural network layers, referred to as "gates", interact to maintain long-term time dependencies, as shown in Figure 3.

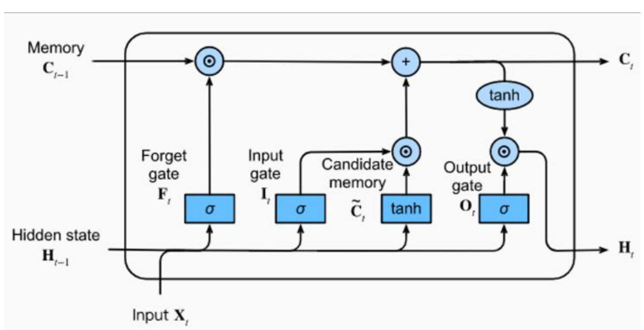


Fig. 3 LSTM Architecture [14]

At each timestep, the cell state (or "memory") and the hidden layer information from the previous timestep are passed in. The "forget gate" takes the previous output and the new input and decides what information from the cell state does not need to be maintained. The "input gate" and "candidate memory gate" then determine what new information will be stored in the memory. Using the new input, the "output gate" determines what information will be passed from the cell state to the next input. Equations 6-11, referring to Fig. 3, demonstrate how each neural network layer updates during one timestep. Note the equations for the gates (6, 7, 8, 10) are analogous to Equation 5, while the cell state and hidden layer equations (9, 11) are simply updating functions of the four gates.

$$\mathrm{F_t} = \sigma(\mathrm{W_F} * [\mathrm{H_{t-1}}, \mathrm{X_t}] + \mathrm{b_F}) \tag{6}$$
$$\mathrm{I_t} = \sigma(\mathrm{W_I} * [\mathrm{H_{t-1}}, \mathrm{X_t}] + \mathrm{b_I}) \tag{7}$$
$$\tilde{\mathrm{c}}_\mathrm{t} = \tanh(\mathrm{W_C} * [\mathrm{H_{t-1}}, \mathrm{X_t}] + \mathrm{b_C}) \tag{8}$$
$$\mathrm{C_t} = \mathrm{F_t} * \mathrm{C_{t-1}} + \mathrm{I_t} * \tilde{\mathrm{c}}_\mathrm{t} \tag{9}$$
$$\mathrm{O_t} = \sigma(\mathrm{W_O} * [\mathrm{H_{t-1}}, \mathrm{X_t}] + \mathrm{b_O}) \tag{10}$$
$$\mathrm{H_t} = \mathrm{O_t} * \tanh(\mathrm{C_t}) \tag{11}$$

Depending on whether the predictive task of the network is regression-based or classification-based, the final outputs of the LSTM will pass through a final linear or SoftMax activation function.

CHAPTER II

Monte Carlo Simulations

To train the LSTM model, data was generated from a Monte Carlo (MC) simulation in Matlab. MC simulations are a process by which researchers may computationally examine outcomes of a complex task by providing parameters and known distributions of unknown input parameters. The MC simulation used in this research models a satellite performing an aerocapture maneuver to achieve planetary orbit around Uranus. The simulation uses preloaded data provided by NASA regarding planetary constants, satellite dimensions and aerodynamic properties, as well as an atmospheric profile for Uranus. By perturbing the entry flight path angle, aerodynamics, onboard instrument measurement readings, and navigation rotational initial states, the MC simulation can generate thousands of unique flight paths for examining the performance of new aerocapture technologies. The training data uses six-degrees-of-freedom motion modeling and assumes perfect navigation knowledge during guidance and control.

The explanatory variables used to train the LSTM model are necessarily variables available in-situ for the model to be of any utility for a satellite performing an aerocapture maneuver. Because of this, training variables include the satellite's quaternion representation of rotations (q), linear velocity (v), angular velocity (w), altitude relative to the planetary body (alt), and the atmospheric density (ρ). In training, the model learns the non-linear relationship between these variables and the force and moment coefficients (CFA, CFY, CFN, CMl, CMm, CMn) described in section II. 1000 perturbed trajectories were generated for training data with these variables available. With the computing power available, each MC simulation took roughly 40 secounds to generate.

Sensitivity Analysis of Hyperparameters

Hyperparameters play a key role in how the relationship between explanatory and response variables are established in a neural network during training. Minibatch size refers to how many simulated trajectories the model can view before performing gradient descent and back propagating to update weights. Hidden units refer to the number of neurons in the LSTM layer. The number of hidden units in a neural network scale with the complexity of the predictive task [9]. To find an optimal hyperparameter configuration, sensitivity analysis was performed across these dimensions. Because of long training time, each LSTM network was trained over 100 epochs. The minimization target was the mean absolute error (MAE) between the predicted and true force and moment coefficients across each trajectory. The results of this analysis are shown in Table 1. Hyperparameter values of 200 hidden units and a minibatch size of 64 are shown to be the optimal, among all tested hyperparameters, for minimizing the difference between target and prediction values.

Table 1: Sensitivity Analysis for Training Hyperparameters, showing the MAE

Minibatch Size / Hidden Units	32	64	128
10	0.035384	0.036822	0.041822
100	0.029583	0.027413	0.039612
200	0.028078	0.024402	0.033858

CHAPTER III

Offline Performance

Before in-situ guidance implementation of the LSTM network, a direct comparison of neural network to lookup table performance was conducted. For a testing set of 1000 trajectories, truth and lookup table values were logged for the target force and moment coefficients. The LSTM was provided input variables, and predictions were made offline. Each trajectory was filtered to only include data where atmospheric density (ρ) was above 0.00001 so the model would learn coefficient patterns for traveling through an atmosphere, where guidance and control would be active during aerocapture. By calculating absolute error at each timestep and averaging across an individual trajectory, a direct comparison of lookup table and LSTM network is possible. Table 2 shows the proportion of runs where the LSTM outperformed the lookup table for each response variable.

Table 2: Proportion of Trajectories where LSTM Outperformed Lookup Table

Response Variable	Percent
CFA	12%
CFY	**67%**
CFN	**100%**
CMl	25%
CMm	**100%**
CMn	0%

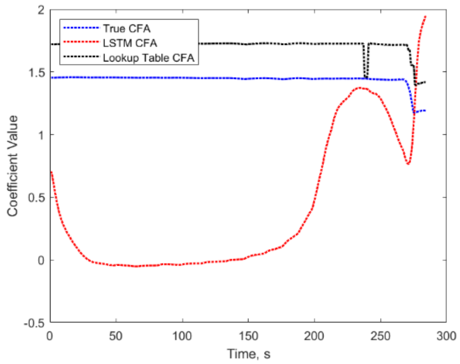
Table 2 shows that the LSTM outperformed the lookup table when estimating the true lateral and normal force coefficients, as well as the pitching moment coefficient. This is an expected result, as a satellite performing an aerocapture maneuver would experience high perturbations in lateral and normal forces, allowing the LSTM to develop a robust understanding of how the inputs relate to these coefficients.

Examining data for an individual trajectory quantifies how much the LSTM overperforms or underperforms the lookup table. Table 3 shows the MAE for the lookup table and the LSTM network for a single trajectory in the test data. Bolded values indicate whether the lookup table or the LSTM network performed better for each coefficient.

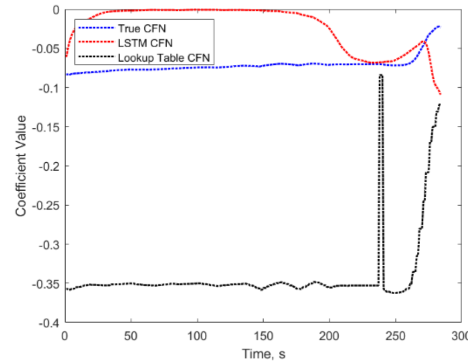
Table 3: Single Trajectory Comparison

Response Variable	Lookup Table MAE	LSTM Network MAE
CFA	**0.2717**	1.0830
CFY	0.0280	**0.0126**
CFN	0.2686	**0.0539**
CMl	**0.0015**	0.0040
CMm	0.0877	**0.0037**
CMn	**0.0039**	0.0091

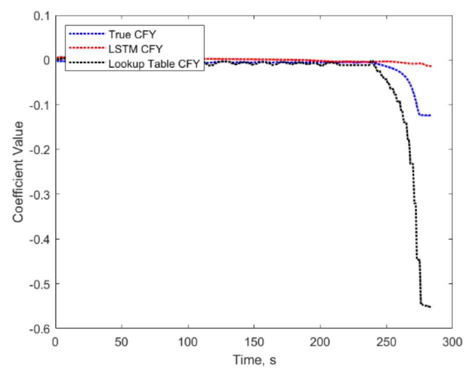
Once again, analysis shows the LSTM performing exceptionally well estimating the lateral force coefficient, normal force coefficient, and the pitching moment coefficient. Viewing an individual trajectory provides scale of improvement for each variable. While the lookup table has a mean absolute error 4x smaller than the LSTM when estimating the longitudinal force coefficient, estimates for the rolling and yawing moment coefficients are nearly identical. Figure 4 makes this more evident, showing the lookup table and LSTM results compared to the true force coefficient values across the trajectory. Appendix A provides the same plots for the moment coefficients.



a.) Longitudinal Force Coefficient



b.) Normal Force Coefficient



c.) Lateral Force Coefficient

Fig. 4 True Force Coefficients with LSTM and Lookup Table Estimates

Online Performance

Examining offline estimation offers insight about predictive capacity, but demonstrating in-situ improvements over the lookup table would be true progression toward improving aerocapture TRL. To access the LSTM online, the network was integrated into the MC simulation guidance block where the lookup table was previously being used. 1000 trajectories were simulated with the lookup table, and 1000 trajectories were separately simulated with the LSTM network. Each run had identical physics and perturbations, with the only variation being the implementation of the perturbed lookup table or the LSTM network in guidance. For each trajectory, success was determined by whether the maneuver successfully entered a scientific orbit (passed) or reached early termination (failed). The results are provided in Table 4.

Table 4: Pass Percentage for Online Implementation

LSTM Pass Percentage	Lookup Table Pass Percentage
47.4%	47.4%

An identical pass percentage when using the lookup table and the LSTM network in guidance for aerodynamic coefficients provides evidence of the full flight being robust to aerodynamic coefficient variation. This result reinforces both the sufficiency of lookup table implementation for aerodynamic coefficient estimation, as well as the potential for LSTM network capability in an online predictive space. Table 2 displays LSTM improvements over the lookup table, but table 4 suggests online improvements to aerodynamic coefficient accuracy are dwarfed by other, more dominant, flight variations such as atmosphere and gravity profile estimation.

Discussion of Results and Impact

With new interplanetary flagship missions being announced by NASA in the early 2030s, the allure of aerocapture for efficient planetary orbit insertion increases. One avenue for improving the TRL of aerocapture is improving online aerodynamic coefficient estimation. More accurate force and moment coefficient estimates during guidance and control would allow FNPAG to execute more efficient commands during an aerocapture maneuver. To this end, a LSTM neural network was introduced as an efficient tool for learning nonlinear relationships between sensed data and the target force and moment coefficients. After a sensitivity analysis to optimize training hyperparameters, the trained model was accessed offline and online, demonstrating an improvement in 3 of 6 force or moment coefficients in over 50% of simulated trajectories. Despite this, online tests of the model showed no difference in the flight performance when compared to the traditional lookup table used in guidance. This provides evidence to the robustness of an aerocapture maneuver to perturbations in aerodynamic coefficients, as well as the ability of neural networks to create predictions on coefficients akin to lookup table values without seeing any wind tunnel or CFD data.

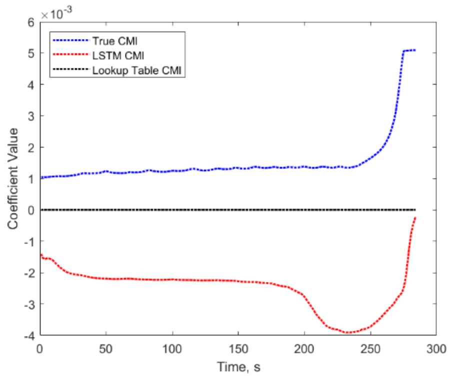
Prospects for Further Study

One key continuation of this research would be examining alternative loss and objective functions for model training. We used mean squared error as an optimization target during hyperparameter sensitivity analysis and training, but this is a surrogate function for our true goals. Ideally, any improvements to aerocapture would target outcomes like pass percentage of a MC simulation or fuel consumption during an aerocapture maneuver. Whether the model accurately predicts aerodynamic coefficients is without utility if it has no effect on true variables of interest, as shown when reviewing online performance.

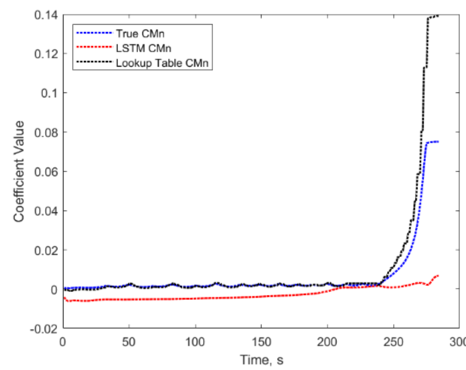
The utility of neural networks and machine learning open many doors of research in improving aerocapture. Multiple papers have explored how machine learning techniques may improve different nonlinear predictive tasks, including atmospheric composition estimation and local density approximation online [15, 16]. A potential continuation of this study would be using an LSTM network to do ensemble prediction, where all these models are combined into one. Because neural networks perform well at learning nonlinear relationships, there is high potential for modeling improvements when providing the network more inputs for analysis. A future paper coupling an ensemble LSTM approach with a more generalized loss function like pass percentage or fuel consumption would provide autonomy for the neural network to learn unknown relationships, leading to potentially drastic improvements in aerocapture viability.

Appendix

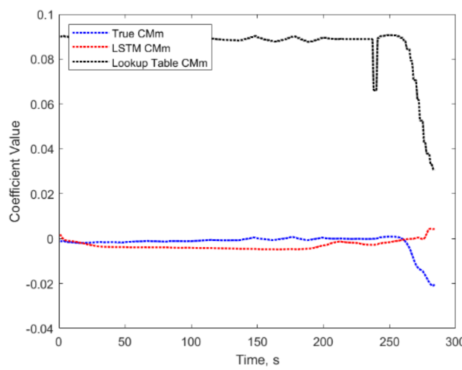
Appendix A: True Moment Coefficients with LSTM and Lookup Table Estimates



a.) Rolling Moment Coefficient



b.) Yawing Moment Coefficient



b.) Pitching Moment Coefficient

References

[1] National Academies of Sciences, Engineering, and Medicine. “Origins, Worlds, and Life: A Decadal Strategy for Planetary Science and Astrobiology 2023–2032”. Washington, DC: The National Academies Press, 2023, https://doi.org/10.17226/26522.

[2] Thomas R. Spilker, Mark Adler, Nitin Arora, Patricia M. Beauchamp, James A. Cutts, Michelle M. Munk, Richard W. Powell, Robert D. Braun, and Paul F. Wercinski. “Qualitative Assessment of Aerocapture and Applications to Future Missions”. Journal of Spacecraft and Rockets, 2019, 56:2, 536-545

[3] Jeffery L. Hall, Muriel A. Noca, and Robert W. Bailey. “Cost-Benefit Analysis of the Aerocapture Mission Set”. Journal of Spacecraft and Rockets, 2005, 42:2, 309-320

[4] Anderson, D.J., Pencil, Eric, Peterson, Todd, Dankanich, John and Munk, M.M. “In-Space Propulsion Technology products for NASA's future science and exploration missions.” 2011. 1 - 14. 10.1109/AERO.2011.5747507.

[5] Munk, Michelle, & Moon, Steven A., “Aerocapture Technology Development Overview”. NASA Technical Report Server, 2008.

[6] Matz, Daniel, Lu, Ping, Mendeck, Gavin and Sostaric, Ronald. “Application of a Fully Numeric Guidance to Mars Aerocapture”. NASA Technical Report Server, 2017.

[7[Chen, Jilin, Hongwei Han, Rui Teng, and Tong Qin. 2022. "Robust Near-Optimal Aerocapture Guidance Method Based on Saturation Function". Applied Sciences 12, no. 22: 11513. https://doi.org/10.3390/app122211513

[8] Lozano, Paulo. “Lecture Notes on Fluid Dynamics”. Massachusetts Institute of Technology, Fluid Mechanics and Aerodynamics”. https://web.mit.edu/16.unified/www/FALL/fluids/

[9] Berke, Laszlo, Patnaik, Surya, and Murthy, Pappu. “Application of Artificial Neural Networks to the Design Optimization of Aerospace Structural Components”. NASA Technical Memorandum 4389, 1993.

[10] Brotzer, J, Mosqueda, E and Gorro, Ken, “Predicting emotion in music through audio pattern analysis”. IOP Conference Series: Materials Science and Engineering. 2019, 482. 012021. 10.1088/1757-899X/482/1/012021.

[11] Pedamonti, Dabal, “Comparison of non-linear activation functions for deep neural networks on MNIST classification task”. Department of Computer Science, University of Edinburgh, 2018. https://doi.org/10.48550/arXiv.1804.02763.

[12] Eliasy, Ashkan and Przychodzen, Justyna, “The role of AI in capital structure to enhance corporate funding strategies”. 2020. Array. 6. 100017. 10.1016/j.array.2020.100017.

[13] Sak, Hasim, Senior, Andrew and Beaufays, Francoise, “Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling”. ISCA arXiv, 2014. https://doi.org/10.48550/arXiv.1402.1128.

[14] Zhang, A, Lipton, Z, Li, M and Smola, A, “Dive into Deep Learning” Cambridge University Press, 2023, Chap. 10.

[15] Roelke, Evan, McMahon, Jay and Braun, Robert, “Atmospheric Density Estimation Techniques for Aerocapture”, Journal of Spacecraft and Rockets. 2023, DOI: 10.2514/1.A35197.

[16] Justus, C, Duvali, A and Keller, V, “Atmospheric Models for Aerocapture Systems Studies”, AIAA Atmospheric Flight Mechanics Conference and Exhibit, 2004.