Advanced Technologies for Hypersonic Propulsive, Energetic and Reusable Platforms
Technical Conference and Exposition
June 30th, 2021

NEURAL NETWORK TRAINED CONTROLLER FOR MANNED MARTIAN ATMOSPHERIC ENTRY

Jonathan Wells
Virginia Tech
Blacksburg, VA

Dr. Bennie Lewis
University of Central Florida
Orlando, FL

Dr. Tarek Elgohary
University of Central Florida
Orlando, FL

ABSTRACT

We submit a new method for atmospheric entry into the Martian atmosphere using a neural network trained controller. Traditionally, the controller for atmospheric entry is an algorithm that has internal logic with a specific purpose, such as calculating an escape orbit and staying below that limit[1]. The method we present here uses a neural network to control the bank angle of a spacecraft as it is entering an atmosphere. The results from a neural network from a previous group of researchers is examined. The neural network is trained using real data from the Apollo missions, and evaluated against the Apollo algorithm for entry. A new type of neural network is proposed to address some of the problems the previous group struggled with. Transfer learning will eventually be used to adapt the algorithm to the Mars environment. At this time, the neural network has not produced any results.

Keywords: Atmospheric Entry, Neural Networks

NOMENCLATURE

${}^{R}V$	Velocity of the Spacecraft
γ	Flight Path Angle
ψ	Heading Angle
ϕ	Latitude
θ	Longitude
σ	Bank Angle

1. INTRODUCTION

The purpose of this paper is to present a new method for controlling atmospheric entry. Atmospheric entry is an unavoidable design challenge that any manned mission must deal with. The primary design requirements of atmospheric entry are the forces on the spacecraft, atmospheric heating, and landing accuracy[2]. As manned missions to Mars become feasible, serious improvements to entry algorithms need to be made. It is possible for unmanned spacecraft to experience higher loads to achieve better accuracy but for manned missions, the force on the spacecraft will need to be limited to ensure the safety of crew. However, landing far from a desired landing area may have serious negative impacts on the goals of a manned mission. The purpose of the neural network trained controller is to develop a new entry algorithm that addresses both issues: limiting the forces on the spacecraft while still achieving acceptable accuracy.

The Apollo capsules featured a misaligned center of mass, allowing the vehicle to produce lift. The lift was perpendicular to velocity and could be made to rotate about the vehicle's central axis. This allowed the Apollo algorithm to choose an angle for this rotation, the bank angle, and control the spacecraft during reentry. To develop the Apollo algorithm, the engineers behind the Apollo algorithm first made several assumptions: point mass, constant mass, a non-rotating planet, no thrust produced by the spacecraft, drag is parallel to velocity, lift is perpendicular to velocity, and the force of gravity acts towards the center of the planet[1]. This paper will make the same assumptions. The general governing equations behind atmospheric entry can be cumbersome, but these assumptions make them decidedly less so. With these assumptions the following equations can be derived:

$$\dot{r} = {}^{R}V \sin(\gamma) \tag{1}$$

$$\dot{\theta} = \frac{{}^{R}V cos(\gamma)\cos(\psi)}{r cos(\phi)} \tag{2}$$

$$\dot{\phi} = \frac{{}^{R}V cos(\gamma)\sin(\psi)}{r} \tag{3}$$

$$\dot{V} = -\frac{D}{m} - \frac{G}{m}\sin(\gamma) \tag{4}$$

$$\dot{\gamma} = \left(\frac{L}{m}\cos(\sigma) - \frac{G}{m}\cos(\gamma) + \frac{{}^{R}V^{2}}{r}\cos(\gamma)\right)/V \tag{5}$$

$$\dot{\psi} = \left(\frac{L sin(\sigma)}{m cos(\gamma)} - \frac{{}^{R}V^{2}}{r}\cos(\gamma)\cos(\psi)\tan(\phi)\right)/V \tag{6}$$

While these equations may seem lengthy, it isn't difficult to understand them with the right perspective. The first three are kinematic equations describing the motion of the spacecraft relative to the planet in spherical coordinates. Equations 2 and 3 are divided by r to map them to the planet's latitude and

longitude. The next three equations describe the forces on the spacecraft. Lift, drag, and the force of gravity are all multiplied by trigonometric functions to shift them to the direction in which they act. The velocity squared terms are best understood as the change in latitude or longitude due to the velocity of the spacecraft. Figure 1 below help contextualize each of the variables.

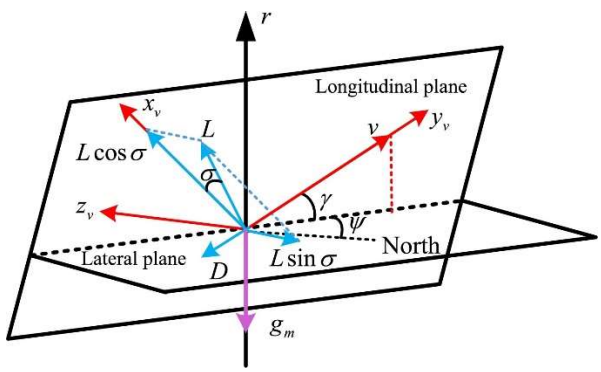

FIGURE 1: Forces acting on a spacecraft[6]

Notice that each of the variables, except the bank angle, cannot be controlled by the spacecraft and are only initial conditions at the point of entry. For any algorithm that is chosen for reentry, the flight profile should be chosen in congruence with the reentry algorithm. The flight profile will determine the initial conditions of reentry, and the algorithm will control the bank angle throughout the course of reentry.

It is important then, to consider different landing algorithms that have been investigated by researchers. The algorithm that was used during the Apollo missions is still used today and was recreated in this paper. It uses internal logic to act as a proportional controller, designed to follow a preprogrammed trajectory[1]. It was used during the Apollo-era due to its low computational power, but respectable results. Another method, the predictor-corrector method, numerically integrates an energy form of the above equations[7]. By doing this, the spacecraft can predict the trajectory it will follow as it descends through the atmosphere and adjust the bank angle accordingly to correct for any inaccuracies in the trajectory it wants to follow. There are also researchers working on methods to optimize the trajectory itself limit the maximum deceleration and heating that spacecraft experience as they descend through the atmosphere[8].

Both the Curiosity and Perseverance rovers used the architecture of the Apollo entry algorithm[3]. Until 2012, with the landing of the Curiosity rover, every Martian landing was unguided[4]. This limited the complexity of each of the missions, but greatly reduced the accuracy of each lander. By implementing a guidance system for atmospheric entry on the Curiosity rover, a greater accuracy could be achieved, and a more precise landing zone chosen. The Perseverance used a similar, although slightly modernized algorithm[3]. Both landers used the Apollo algorithm against other, more involved, algorithms because despite its age, the Apollo algorithm is still very good. In many different simulations, the Apollo algorithm was consistently the best or second best among a wide variety of categories, such as landing accuracy[3]. The neural network will use data from the apollo missions and will use the apollo algorithm as a baseline for training.

The Apollo algorithm uses several different phases to determine what the best bank angle through the atmosphere is. The phases are as follows: Initial Roll, Huntest, Down-Control, Up-Control, Kepler, Second entry[1]. The specifics of the Apollo algorithm are beyond the scope of this paper but can be explored in Reentry Guidance for Apollo by Raymond Morth.

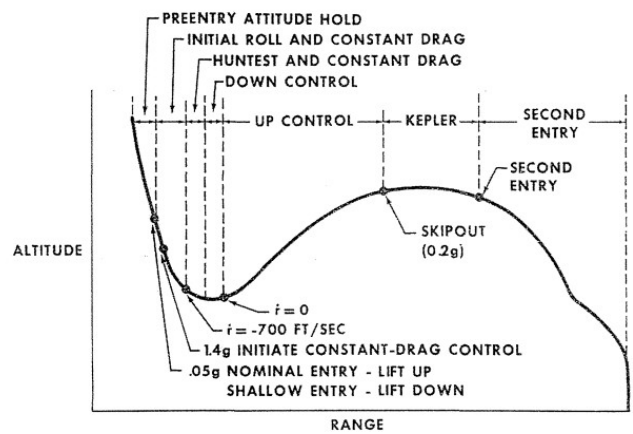

FIGURE 2: Apollo Entry Guidance Phases[5]

Each phase has its own internal logic for ensuring a safe and accurate reentry such as limiting the forces on the capsule, keeping the capsule from entering an escape orbit, or determining which phase should be entered next based on the current trajectory. A high-level view of the logic from the Apollo algorithm can be seen in Figure 3.

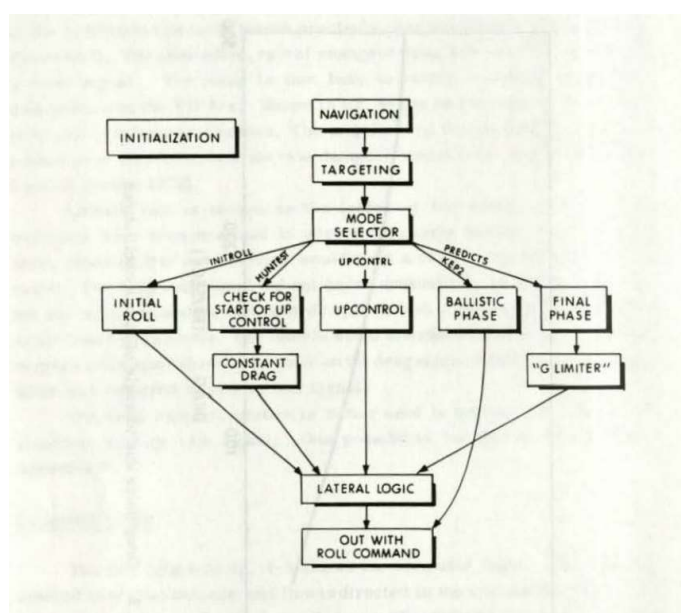

FIGURE 3: Apollo Algorithm Logic[1]

Within each block is the logic which controlled the Apollo capsules during reentry. For example, the constant drag block calculates a new L/D then runs a test depending on the drag to determine the sign of the bank angle. The result of this is constant drag on the spacecraft. The specifics of the logic were programmed into MATLAB. This will be discussed in the next section.

2. MATERIALS AND METHODS

To train the neural network, mission data from the Apollo missions were used. A simulation of an Apollo mission was created to ensure accuracy and provide a comparison to the neural network before a simulation of the Martian environment will be used with the neural network.

2.1 Apollo Mission Simulation

To first ensure that any simulations performed by the neural network were accurate, a simulation of the Apollo mission using the Apollo algorithm was first developed in MATLAB. Using data from one of the Apollo mission, the initial conditions for reentry could be replicated, and the results from the simulation can be compared. The results from the MATLAB simulation are shown in Figure 5, while the data collected from the Apollo mission are shown in Figure 5.

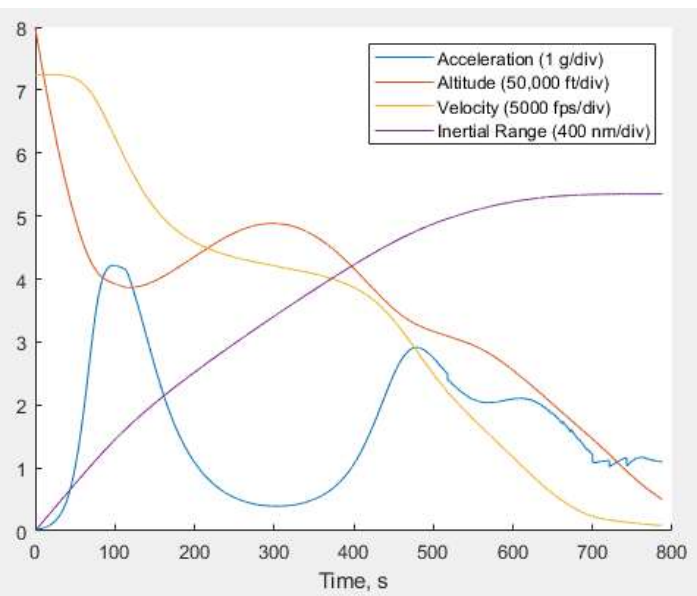

FIGURE 4: MATLAB simulation results for a -6 degree entry angle

It can clearly be seen that the two graphs are very similar. There are some small differences between the two. The second peak in acceleration in the MATLAB simulation is somewhat smaller than the Apollo simulation and the velocity and altitude lines do not intersect. This is likely due to uncertainty in the Apollo measurements. Due to the sensitivity of the system to initial measurements, it is possible that a slight change in initial position of velocity could have caused these changes. Additionally, weather effects would play a larger role in the lower parts of the atmosphere, which could explain part of the difference. In combination, these can explain the differences between the graphs. Although there are minor differences between the two, the MATLAB simulation replicated the results of the Apollo mission extremely well.

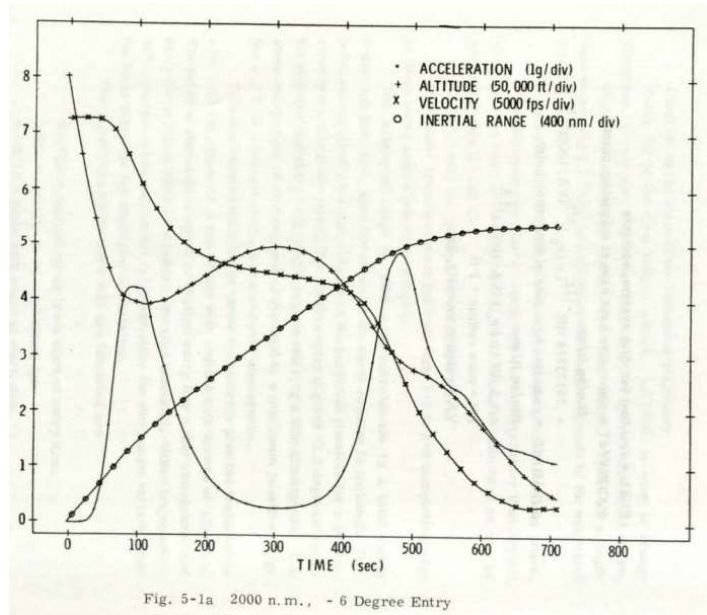

FIGURE 5: Apollo mission data for a -6 degree entry angle[1]

2.2 Neural Network Training

In recent years, neural networks have come much more popular. Neural Networks work by having a set of input neurons, which then pass through several layers of hidden neurons with their own weights and biases. As each neuron activates, it activates the next set of neurons according to their weights and biases. At the end of the network is a set of output neurons, which activate based on the activation of all the neurons in the network before it[9]. A traditional neural network is shown in Figure 6. These types of neural networks are called artificial neural networks (ANNs)[10].

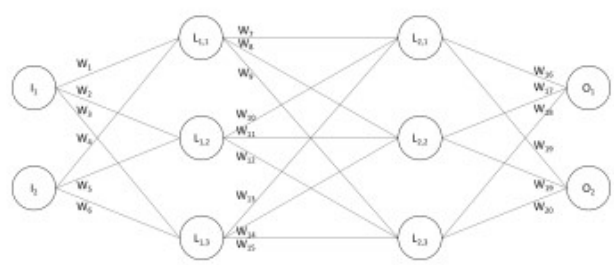

FIGURE 6: A Traditional Neural Network, also called an ANN[10]

To train a neural network, a large amount of data is needed. The input neurons activate according to the data, and information travels down the network until the output neurons activate. Initially the network responds poorly to the data and the output neurons do not give the desired outcome. Using a technique

known as backpropagation, the weights and biases of each of the neurons is changed to adjust the response of the neural network[9]. After training the network, the output neurons generally respond well to the data, and produce the desired outcome.

3. RESULTS AND DISCUSSION

The previous group of researchers planned to run a Monte Carlo simulation to test the robustness of the neural network trained controller that used an ANN. The ANN featured 8 hidden layers[11]. After training the network for several days, the simulation could be run. The results for an Earth and Mars environment are shown in Figures 7 and 8, respectively.

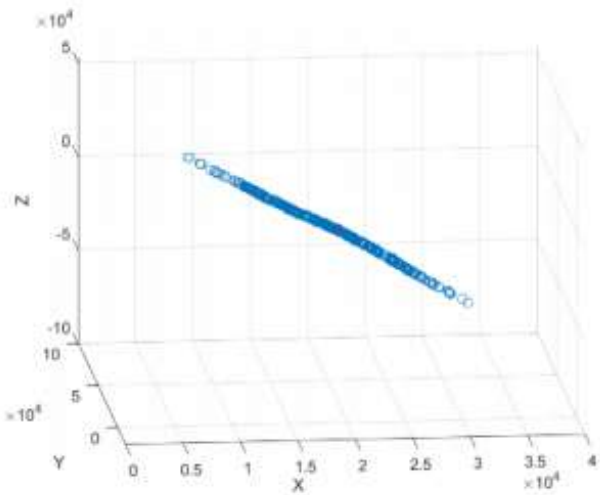

FIGURE 7: Missed distance in the Earth environment[11]

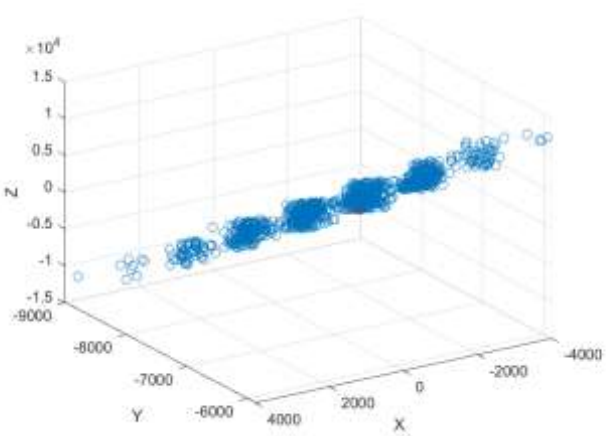

FIGURE 8: Missed distance in the Mars environment[11]

The red dot in both figures is the desired landing position, with no perturbation applied to the spacecraft before starting the simulation. While the results from the ANN are promising, the data that was collected for both environments used a linear model. When moving to a nonlinear model, the researcher encountered problems with the neural network. The controller could not achieve satisfactory error in the Earth environment even after training[11]. It was then decided to use a different type of neural network, a convolutional neural network, that might better address the problem.

4. CONCLUSION

Atmospheric entry on Mars poses a large problem to any manned missions that might take place in the future. Although current algorithms are sufficient to land probes on the surface, stricter requirement will be necessary for humans to take the journey. Several approaches exist that each utilize different methods to solve the problem. A previous research group used neural networks to produce promising initial results that had some weaknesses. Our goal is to use a convolutional neural network (CNN) to train a controller for atmospheric entry. Convolutional networks were initially designed for image and video processing. However, CNN's take in spatial inputs to process that data, which may make them an ideal candidate to address the problem of atmospheric entry[12].

ACKNOWLEDGEMENTS

I would like to thank Dr. Elgohary and Dr. Lewis for guiding my research this summer. Without their help I would not have nearly gotten as far as I did. I would also like to thank Dr. Gordon and Dr. Kauffman for their support of the program and Robert Burke for his excellent guidance of the students in the program. The HYPER program is funded by the NSF and DOD.

REFERENCES

[1] Morth, Raymond Harry., et al. *Reentry Guidance for Apollo*. Massachusetts Institute of Technology, Instrumentation Laboratory, 1966.

[2] D. Hicks, Kerry. *Introduction to Astrodynamic Reentry*. 2009

[3] Hall, Tim. "Apollo Algorithm Perseveres Yet Again to Guide Landing on Mars", *NASA JSC Features*, February 23, 2021. Available:https://roundupreads.jsc.nasa.gov/pages.ashx/1633/Apollo%20Algorithm%20Perseveres%20Yet%20Again%20to%20Guide%20Landing%20on%20Mars. [Accessed June 30th 2021]

[4] R. D. Braun and R. M. Manning, "Mars exploration entry, descent, and landing challenges," *Journal of spacecraft and rockets,* Vol. 44, No. 2, 2007, pp. 310–323.

[5] Ives, Dallas, et al. "Apollo-Derived Mars Precision Lander Guidance." *23rd Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, 2013, doi:10.2514/6.1998-4570.

[6] Gong, Youmin, et al. "Mars entry guidance for mid-lift-to-drag ratio vehicle with control constraints." *Aerospace Science and Technology Volume 107*. December 2020

[7] Ping, Lu. *Predictor-Corrector Entry Guidance for Low-Lifting Vehicles*. Iowa State University, 2008.

[8] Yu, Zhengshi, et al. *Design and optimization of navigation and guidance for Mars pinpoint landing: Review and prospect.* 2018.

[9] Hatcher, W. G., and W. Yu. "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends." IEEE Access, vol. 6, 2018, pp. 24411–32, doi:10.1109/ACCESS.2018.2830661.

[10] Wang, Hao, et al. NEURAL NETWORK TRAINED CONTROLLER FOR ATMOSPHERIC ENTRY IN MARS MISSIONS. 2019.
[11] Keosayian, Matthew, et al. *Neural Network Trained Controller For Atmospheric Entry in Mars Environment.* 2019
[12] Brownlee, Jason. "CNN Long Short-Term Memory Networks". Available: https://machinelearningmastery.com/cnn-long-short-term-memory-networks/ [Accessed July 26th 2021]

