AAS 19-491

NEURAL NETWORK TRAINED CONTROLLER FOR ATMOSPHERIC ENTRY IN MARS MISSIONS

Hao Wang[*], Dillon Martin[†], and Tarek Elgohary[‡]

We present a new method to design the controller of Mars capsule atmospheric entry using neural networks. Compared to Apollo controller as a baseline, the simulation of neural network controller reproduces the classical Apollo results over a variation of initial conditions, e.g. initial position. This leads to the potential of achieving landing accuracy requirements of future manned Mars missions. The data from Apollo re-entry simulation in Earth model is used for neural networks training. The neural network controller for Earth reentry is evaluated with Apollo real data. It is then adapted for the Mars environment and achieves the desired landing accuracy for a Mars capsule. The results show significant promise in using that approach for future Mars missions.

INTRODUCTION

Atmospheric entry requires a careful balance of landing accuracy, maximum acceleration, and heating requirements; however, landing accuracy is often sacrificed in order to meet other mission requirements.[1] While Earth re-entry missions have become routine, Mars missions often do not have the accuracy desired. Additionally, Mars landings present unique challenges. Not counting the most recent InSight mission, there have been 20 missions that intended to land on Mars, of which only 7 were successful. A summary of these missions is given in Table 1.

Most notably the landing accuracy for the majority of these missions is on the order of 100's of kilometers. To enable the next generation of Mars landers, future manned Mars missions will require a landing accuracy of 10-100 meters.[2] In order to achieve that, several technical challenges need to be overcome. One of which is the capability to land accurately while maintaining acceptable acceleration and heating requirements set by the vehicle and cargo.

So far there have been generally three generations of entry guidance algorithm. The first-generation entry guidance algorithm was designed for low-lifting capsule vehicles like Apollo. The Apollo entry guidance algorithm is broken up into different phases. In each phase, the Apollo vehicle flies the trim angle of attack without modulation. This algorithm heavily depends on the given reference trajectory and the command for the bank angle is generated to reduce the downrange error. To reduce the computational burden, this algorithm relies mostly on analytical, approximate, and empirical relationships, which affects the precision and applicability.[4–9] The second-generation algorithm is the space shuttle entry guidance algorithm. The shuttle has an obviously higher lift-to-drag ratio than

[*]PhD student, Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, FL 32826
[†]Undergraduate student, Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, FL 32826
[‡]Assistant Professor, Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, FL 32826

Table 1: Summary of Previous Successful Mars EDL Missions[2,3]

	Viking 1/2	MPF	MER-A/B	Phoenix	MSL
Year	1976	1997	2004/2008	2008	2010
Entry From	Orbit	Direct	Direct	Direct	Direct
Guidance	Unguided	Unguided	Unguided	Unguided	Apollo Guidance
Entry Velocity (m/s)	4700	7260	5400/5500	5500	5900
Entry Flight Path Angle ($deg.$)	-17	-14.06	-11.49/11.47	-13	-15.5
Ballistic Coefficient (kg/m^2)	64	63	94	70	115
Entry Mass (kg)	992	584	827/832	600	2920
Lift to Drag Ratio	0.18	0	0	0	0.24
Aeroshell Diameter (m)	3.5	2.65	2.65	2.65	4.6
3-σ Landing Ellipse Major Axis (km)	280	200	80	100	20
3-σ Landing Ellipse Minor Axis (km)	100	100	12	21	10
Total Integrated Heating (J/m^2)	1100	3865	3687	2428	6185
Peak Heating Rate (W/cm^2)	26	100	44	58	155

Apollo as well as longer flight time and downrange. Additionally, the shuttle lands horizontally. In this acceleration-based entry guidance algorithm, a reference longitudinal trajectory is defined by a drag acceleration vs Earth-relative velocity profile for high Mach numbers and a drag-vs-energy profile for lower Mach numbers.[4,10,11] The third-generation entry guidance algorithms in recent years originated from previous methods but rely much more on onboard computation for a real-time trajectory design and guidance solution. Predictor-corrector algorithms have shown great potential. These methods can adapt to large trajectory dispersions, no reliance on a given reference trajectory. However, their weakness is the lack of ways to enforce inequality trajectory constraints. Considering all their advantages and disadvantages, Ping Lu developed a Unified Method based on the same algorithmic principles but applicable to multiple vehicle configurations. The algorithm is relatively simple and highly robust using the bank angle to control the vertical component of the aerodynamic lift. A numerical predictor-corrector algorithm is used as the baseline algorithm.[4]

The Apollo reentry method is detailed in Re-entry guidance for Apollo by Morth.[12] It has been flight proven and is often the starting point of modern entry methods, including MSL, the most advanced Mars lander.[13,14] On the other hand, even though several papers about MSL are published, the access to real Mars mission data is still limited. The large uncertainties in Mars atmosphere model should be considered which have significant influence on results. Yu's review paper summarized the current navigation and guidance techniques for Mars pinpoint landing.[15] The paper

indicated that the landing process is a critical and dangerous phase of the entire Mars landing mission leading to higher accuracy and safety requirements. However, present GNC technologies are no longer suitable for future missions. The challenges for different phases including atmospheric entry, parachute descent, powered descent and landing involve limited navigation information, nonlinearity and uncertainty of the dynamic model, and weak control capability in deep space. It is noteworthy that the MSL mission, which has the most accurate landing so far, used Apollo re-entry guidance algorithm to guide the vehicle to the parachute deployment velocity.[15]

In recent years, research using machine learning in atmospheric entry, especially Mars entry was conducted.[16,17] However, most of them are combinations of machine learning and other control methods and set in idealized situations. For example, Jiang et al. combined reinforcement learning and Gauss Pseudospectral Method in Mars powered descent entry[16] and Li et al. discussed second-order sliding mode guidance with radial basic functions(RBF) neural network for Mars entry.[17] Recently, Gaudet et al. used deep reinforcement learning in Mars powered descent and pinpoint landing problem and it performed well with gaze heuristic method.[18] However, reinforcement learning has to care about both policy network and value network, which increased the complexity of the problem and computation requirement compared to deep neural networks.

Considering constraints and challenges discussed above, we are working on finding the simplest and most efficient way to control the Mars vehicle during entry, and showing the potential to be useful in future missions. We chose to use only deep neural network trained with Apollo simulation data, which are the most realistic data accessible, and replace the original control block of Apollo simulation by the trained neural network. After it has shown the capability to work in Apollo environment, it is transferred to Mars environment with more uncertainties. Even though the guidance algorithm of Apollo was designed 50 years ago, it is still used as a reference of recent Mars missions. This is also the reason that we used it as baseline and reference of this research discussing the descent phase before parachute deployment.

ENTRY, DESCENT, AND LANDING

A large portion of this paper focuses on bank angle. During entry there is often only one control parameter, the bank angle, thus the spacecraft flies in an "S" or a zig-zag pattern so that it never deviates too far from its target laterally. This algorithm puts emphasis on the location of the final reversal because the algorithm lends itself to higher lifting spacecraft that are likely to land on a runway and thus by having a finer control over bank reversals the vehicle is able to have a better controlled final heading constraint. For the proposed research high lifting vehicles will not be considered, because of the lack of a martian runway for landing; however, fine control over bank reversals can help improve accuracy significantly.

To achieve this increased accuracy requirement, robust guidance methods coupled with new spacecraft must be developed to handle large amounts of uncertainty including unknown flight characteristics, positional uncertainty, and most notably a significant amount of atmospheric uncertainty. This research will begin by developing a model to simulate atmospheric entry. The dynamic equations of the entry capsule are given below.

Equations

$$\dot{r} = Vsin(\gamma) \tag{1}$$

$$\dot{\theta} = \frac{Vcos(\gamma)cos(\psi)}{rcos(\phi)} \tag{2}$$

$$\dot{\phi} = \frac{Vcos(\gamma)sin(\psi)}{r} \tag{3}$$

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

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

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

Where, γ is the flight path angle, ψ is the heading angle, ϕ is the latitude, θ is the longitude, σ is the roll/bank angle, m is the mass of the capsule, D is the drag, L is the lift, G is the force due to gravity, r is the distance from the center of the planet, and V is the velocity of the capsule.

For Earth entry, the controller used for Apollo is implemented to validate the model and provide training data for the neural network. Next, the results of a trained neural network controller for Earth atmospheric entry are presented. All the simulations as well as neural network training and application are in MATLAB with Neural Network Toolbox.

APOLLO RE-ENTRY

First, the simulation of Apollo controller is implemented and run on Earth model. The following assumptions are made to simplify the problem: point Mass, constant mass, non-rotating planet, no thrust, drag acts in the direction opposite to velocity, lift is perpendicular to velocity and gravity is directed along a vector from the point mass to the center of the planet. These assumptions are consistent with the assumptions used for Apollo entry.[12]

Atmospheric modeling is crucial for an accurate simulation of a planned Mars mission, and many researchers are working on this topic; however, the uncertainty in atmospheric conditions can still be as high as 25-200%.[19–21] Due to this large amount of uncertainty little is lost by using a simplified atmospheric model along with Monte Carlo simulations that will propagate error throughout entry. Additionally, since Apollo's guidance will be implemented first on Earth, different models for Earth and Mars are needed.

The Earth's atmosphere is represented with a standard atmospheric model supplied by NASA.[22] This model has three separate curves to represent the troposphere, lower stratosphere, and upper stratosphere.

Starting from the surface of Earth and going up to 11,000 meters, the temperature and pressure are given by

$$T = 15.04 - 0.00649h \tag{7}$$

$$P = 101.29\left(\frac{T + 273.15}{288.08}\right)^{5.256} \tag{8}$$

where T is temperature in degrees Celsius,h is altitude in meters, and P is pressure in kilo-Pascals. Continuing up in altitude the temperature and pressure for the lower stratosphere, 11,000 to 25,000 meters, is given by

$$T = -56.46 \tag{9}$$

$$P = 22.65e^{1.73 - 0.000157h} \tag{10}$$

Finally, the pressure and temperature in the upper stratosphere, above 25,000 meters, is given by

$$T = -131.21 + 0.00299h \tag{11}$$

$$P = 2.488\left(\frac{T + 273.15}{216.6}\right)^{-11.388} \tag{12}$$

Using these pressures and temperatures the density ρ is computed as

$$\rho = \frac{P}{0.2869(T + 273.15)} \tag{13}$$

Finally, the speed of sound is calculated by following

$$a = \sqrt{\gamma RT} \tag{14}$$

where a is the speed of sound, γ is the ratio of specific heats, R is the gas constant, and T is again temperature. This can then be used to calculate the Mach number with

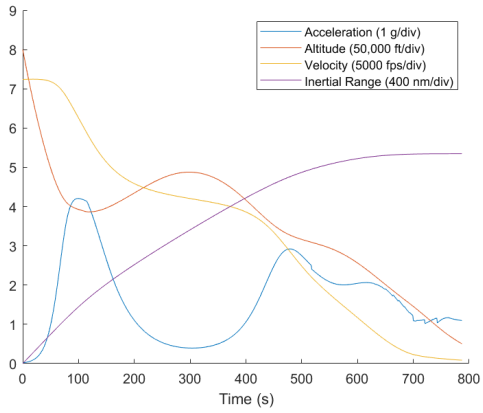
$$Ma = \frac{V}{a} \tag{15}$$

Initial conditions are obtained from Apollo report.[12]

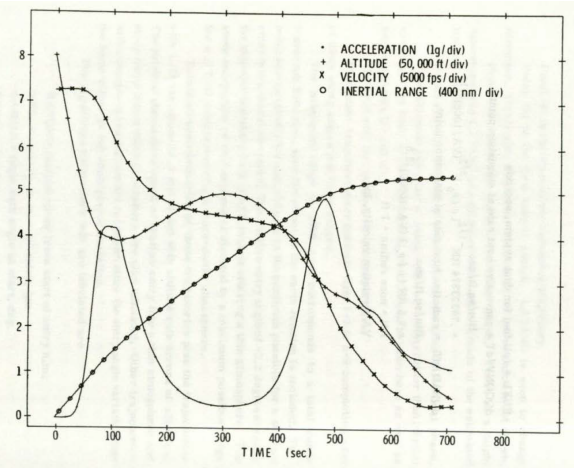
Table 2: Initial Conditions and Target on Earth[12]

Height h (m)	121920
Velocity V (m/s)	11033.76
Flight path angle γ (deg)	-6
Range (m)	3704000
Latitude ϕ (deg)	-12.7
Longitude θ (deg)	122.9
Heading angle ψ (deg)	61
Terminated height (m)	7620
Target (m) in Earth fixed frame	(-4754400.51, 3771311.48, 1964589.29)

The results are plotted in the following Figure 1a and Figure 2a. A similar case from Morth is shown as Figure 1b and Figure 2b for comparison. It is not hard to find that they are close showing that the Earth model and the controller are valid. However, since not all initial conditions are precisely known as well as accurate atmospheric and spacecraft flight characteristics, there are some discrepancies.

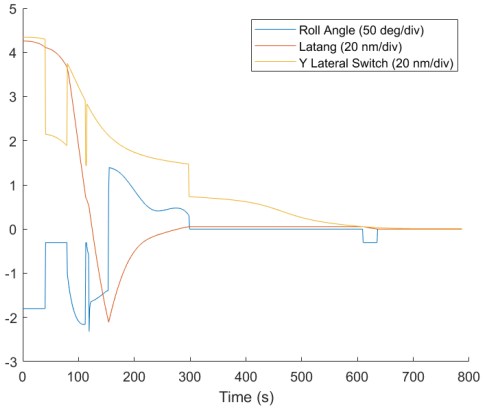


(a) Apollo Replication

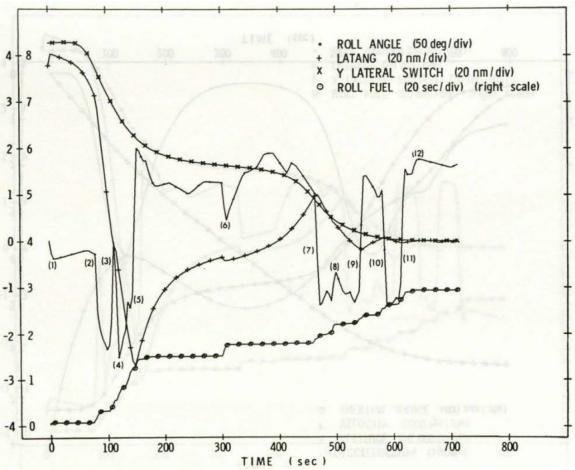


(b) Apollo[12]

Figure 1: Flight Characteristics Comparison



(a) Apollo Replication



(b) Apollo[12]

Figure 2: Control and Targeting Comparison

NEURAL NETWORK TRAINING

Artificial Neural Networks (ANN) are computing systems with interconnected group of nodes inspired by the biological neural networks in animal brains.[23] Deep learning is the application of multineuron, multi-layer neural networks which can perform regression, classification, clustering

and other tasks.[24] Since Hinton published the famous deep belief networks paper in 2006, deep learning technologies has shown their inherent capability of overcoming the drawback of traditional algorithms dependent on hand-designed features across disciplines.[25,26] With machine learning techniques, computers have the capability of constructing algorithms that can learn from data, and making data-driven decisions or predictions without being programmed explicitly.[26] It has shown the capability to be useful in solving nonlinear problems in complex system. Researchers in deep learning area are also working on topics like learning from fewer training data, reinforcement learning approaches to deep neural networks for complex systems, and Applications of deep neural networks in nonlinear networked control systems, which are related to aerospace control problems.[26]

Neural networks work by using a number of nodes and node connections, where each of these connections has a weight. The simplest form of neural network is a feed forward neural network, where for a given set of inputs there is an output without any loops or cycles. These neural networks are arranged in layers, where each layer contains a number of nodes.

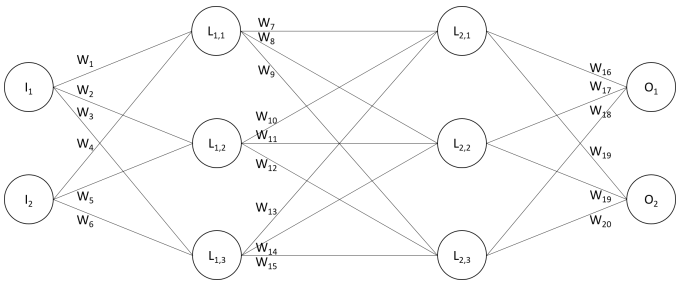


Figure 3: Neural Network Illustration

Figure 3 has an input layer with 2 nodes(left), 2 hidden layers with 3 nodes each(center), and an output layer with 2 nodes (right); however, this need not be symmetrical. Each line between the nodes represents a connection that has a numerical weight, W, associated with it. In neural network, a sigmoid neuron, the most basic neuron, can be considered as a single logistic node. Each one is connected to the input ahead of it, and a loss function is used to update the weights of the neuron as well as optimize the logistic fit to the incoming data.[24]

The use of neural networks produces a controller that can learn to control a vehicle with a wide variety of vehicle characteristics in a variety of atmospheric conditions. In our case, there are 3 inputs of the training neural network represent 3 components in distance from current position to the target. The output neuron represents the one control variable, bank angle. We will start from 3 hidden layers for Earth model. The neural network will be trained for the Apollo controller to test this new method and then adapted for Mars entry with more uncertainties.

For our simulations, it is assumed that the inputs and outputs in application are within the same range as inputs and outputs in training. As a result, the same normalization method is applied with the same maximum and minimum values.

To obtain the training data, we run the Apollo simulation for 216 runs with 500 meter variation in x,y,z components of initial position. The latitudes and longitudes of landing locations are shown in Figure 4 and the missed distances from the target are shown in Figure 5. The red filled point is the supposed target. From the plots, it is seen that all these points are close to the target and the missed distances are within the required range of Apollo mission guidance system, about 20 km.[12]

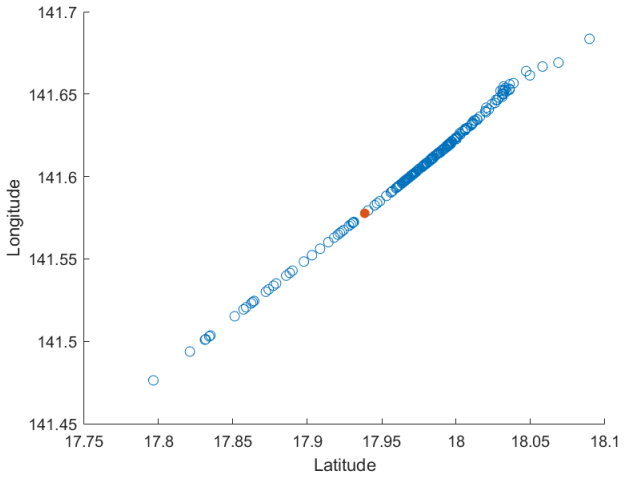


Figure 4: Latitude and Longitude Distribution for 500 meters variation in Initial Position

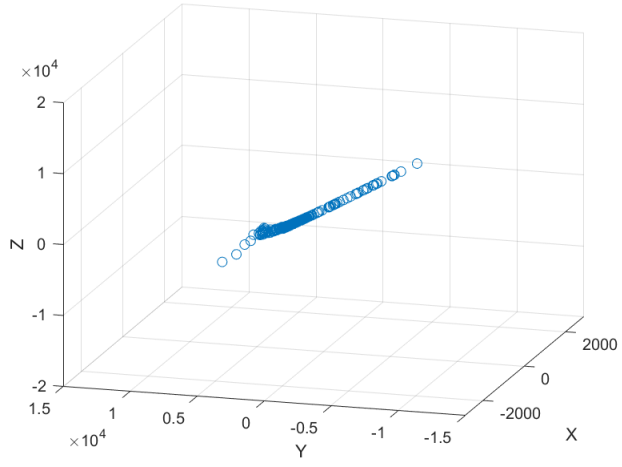


Figure 5: Missed Distance from Target for 500 meters variation in Initial Position

For training, the neural network is set with 5 layers; the first layer includes 3 inputs, 3 hidden layers contain 128, 64, 32 nodes, respectively, and the final output layer only has 1 output. The transfer functions connecting the first four layer are 'hyperbolic tangent sigmoid'. The final/output layer is connected to the previous layer via 'linear' transfer functions. Gradient descent is used as training method and the goal is 0.01. The learning rate is set as 0.01 and 1000 epochs/iterations are conducted. Table 3 shows the training results. Different results can be derived from different sets of training data with various number of runs and time steps. This is one of the best performance neural network in this experiment.

NEURAL NETWORK USED IN EARTH MODEL

Using the same initial conditions of the Apollo case, Table 2, the neural network control is applied to the Earth entry problem. The trained neural network will be used as a replacement of the original

Table 3: Neural Network Training Results

Iterations:	1000
Time:	0:52:09
Performance (mse):	0.0825
Gradient:	0.0297

control block in this application, as shown in Figure 6.

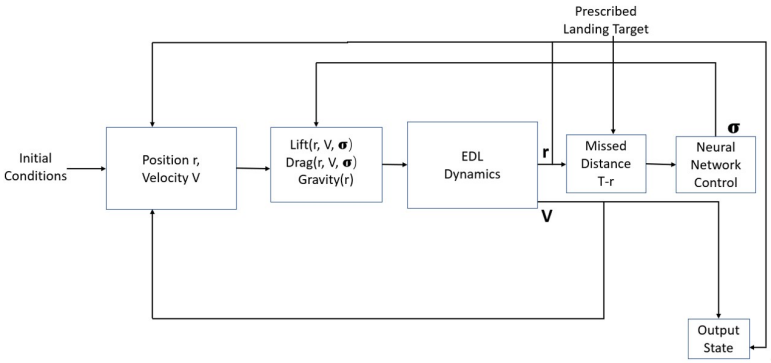


Figure 6: Applied Neural Network in Control System

All other environmental parameters are kept the same except the control block is removed.

The run with the same target gives a final state as shown in Table 4:

Table 4: Final State in Earth Application

step:	164	dt:	4.7
Position:	-4781326.45, 3752079.20, 1935429.73	norm:	6378484.87
Velocity:	115.87, -58.95, -32.47	norm:	134.00
Target:	-4754400.51, 3771311.48, 1964589.29	norm:	6378614.71
Missed:	26925.94, 19232.28, 29159.56	norm:	44104.04

The flight characteristics and bank angle behavior are shown in Figure 7 and Figure 8.

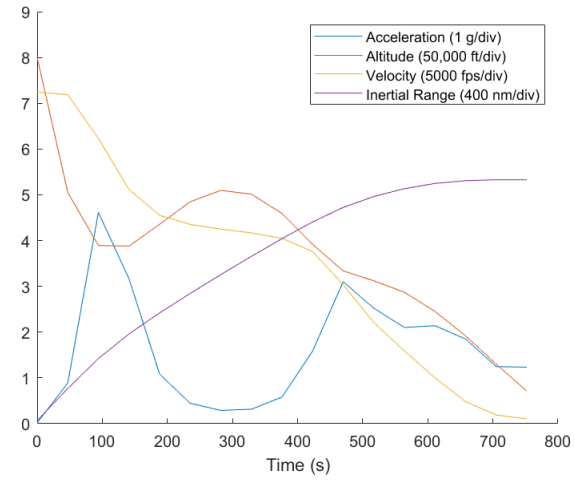


Figure 7: Neural Network Controller in Earth Model Flight Characteristics

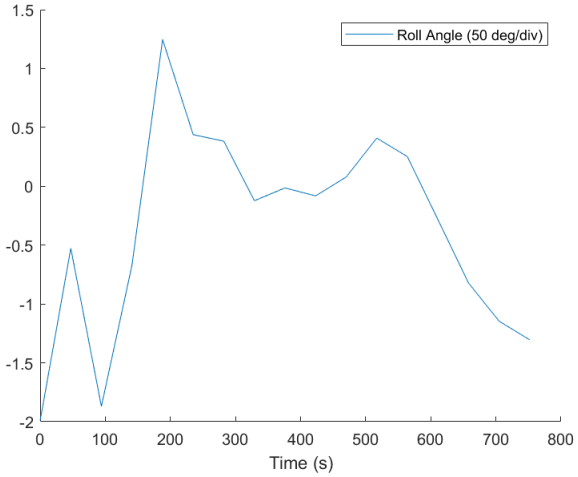


Figure 8: Neural Network Controller in Earth Model Bank Angle

Figure 7 closely matches the Apollo case with the original control block in Figure 1b but the curves are not so smooth because of the increase of time step. During the application, different time step sizes were tried to find the best performance. The current step size 4.7s led to better results meaning closer to the target than smaller steps like 1s or 0.1s as in training data.

Monte Carlo simulations with varying initial positions are conducted for 1000 runs. Each component of initial position varies in normal distribution with a nominal initial position of $r = [-3440503.766641493, 5318209.107111008, -1427443.817166626]^T$ and a standard deviation of 500/3 m in x, y, and z directions. The results are shown in Figure 9 and Figure 10.

The red dot shows the actual landing location of the nominal trajectory.

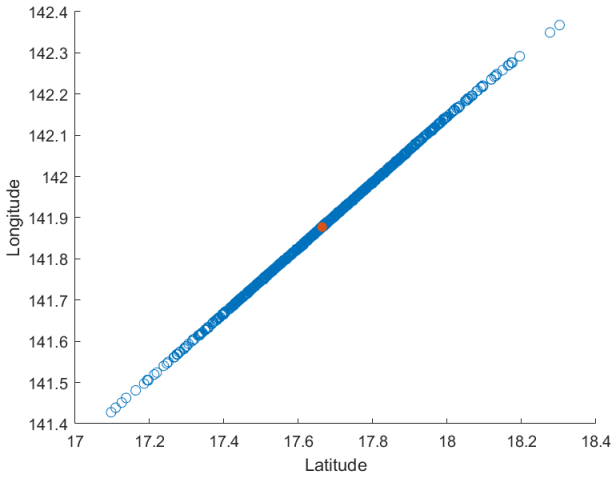


Figure 9: Latitude and Longitude Distribution in Monte Carlo Simulation of NN Controller in Earth Model

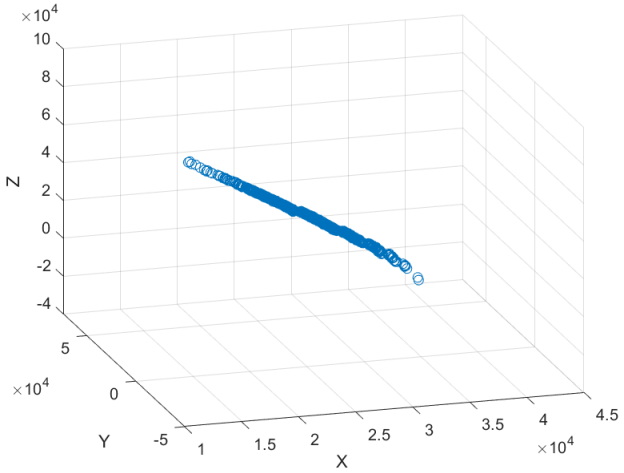


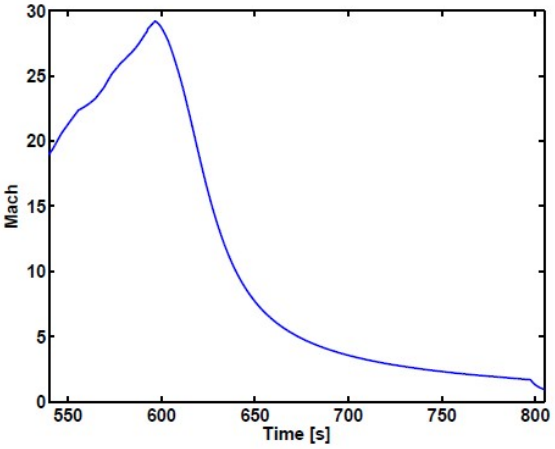
Figure 10: Missed Distance in Monte Carlo Simulation of NN Controller in Earth Model

From the plots, the final landing position is still close to the target considering the scale. The missed distance are within a reasonable range. This result built confidence that the neural network can work as a replacement of the control block since they produce similar landing performances and reach the landing goal.It has potential to work in Mars environment as well.

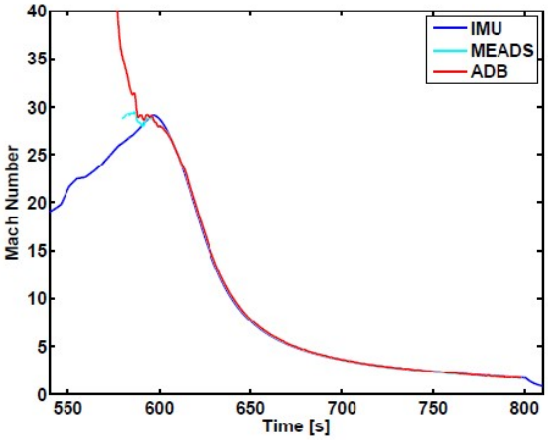
NEURAL NETWORK USED IN MARS MODEL

Because of the complexity and uncertainties of current Mars atmosphere model, in this application a constant temperature of 170K is assumed during the entry process. The real MSL data showed that the temperature varied between about 150K-200K which made this assumption reasonable. However, different methods lead to different temperature ranges as shown in Figure 11c and

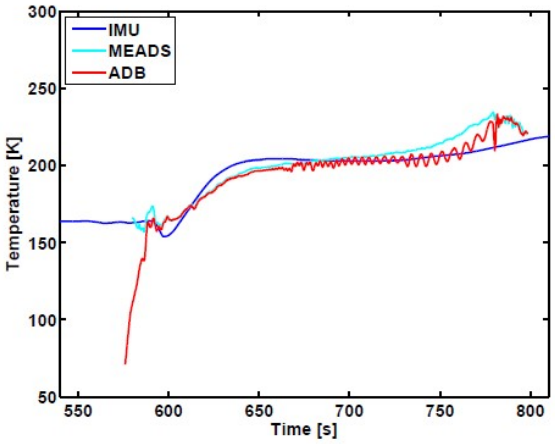
Figure 11d.[27] This number is decided also considering the maximum Mach number as in Figure 11a and Figure 11b.[27]



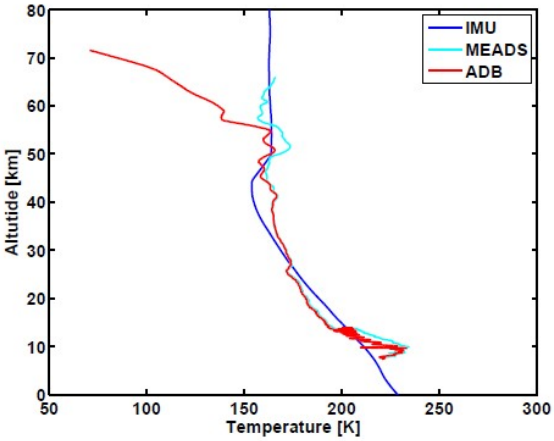
(a) MSL Nominal Trajectory Mach Number



(b) Aerodynamic State Reconstruction Mach Number



(c) Atmosphere Reconstruction Temperature vs. Time



(d) Atmosphere Reconstruction Temperature vs. Altitude

Figure 11: Mach Number and Temperature in MSL Flight Reconstruction Algorithm[27]

The uncertainties in Mars atmosphere model are much larger than in Earth model. It is hard to find one accurate enough although there are several Mars atmospheric models available. Considering reasonable entry conditions like speed of sound and realistic Mars environment, Equation (16) from Li's paper and real data from MSL mission are used in this application.[17,27] The density is calculated based on the density on the surface of Mars, the radius of the Mars, the constant scale height and current position.

$$\rho = \rho_0 e^{(-r-r_0)/h_s} \tag{16}$$

Where, ρ is the density, $\rho_0 = 0.0158 kg/m^3$ is the density on the surface of Mars, r is the current position, $r_0 = 3386.6 km$ is the radius of the Mars and $h_s = 9354.5 m$ is the constant scale height.

The speed of sound is calculated with the same formula as in Earth model. All parameters are from NASA's model.[28] The real MSL situation and the simulation based on MSL are used as reference when the trained neural network is applied to the Mars model.[17,27] Table 5 shows the initial conditions from the simulation in Li et al. and MSL report.[17,27]

Table 5: Initial Conditions[17] and Target on Mars

Height h (m)	135400
Velocity V (m/s)	6000
Flight path angle γ (deg)	-11.5
Latitude ϕ (deg)	0
Longitude θ (deg)	0
Heading angle ψ (deg)	90
Terminated height (m)	6000 by estimation
Target (m) in Mars fixed frame	(3237225.27, 51680.98, 1024523.02))

The results are shown in Table 6 for time step 1s and 5s. The flight characteristics and bank

Table 6: Final States in Mars Application

step:	252	dt:	1.0
Position:	3229743.45, 67929.61, 1047098.68	norm:	3395920.03
Velocity:	-306.12, 442.65, 450.94	norm:	702.14
Target:	3237225.27, 51680.98, 1024523.02	norm:	3395871.87
Missed:	7481.82, 16248.63, 22575.66	norm:	28803.75
step:	49	dt:	5.0
Position:	3237199.89, 68853.83, 1023530.03	norm:	3395853.02
Velocity:	-288.88, 446.22, 410.51	norm:	671.63
Target:	3237225.27, 51680.98, 1024523.02	norm:	3395871.87
Missed:	25.38, 17172.85, 992.99	norm:	17201.55

angle behavior with 1 second time step are shown in Figure 12 and Figure 13.

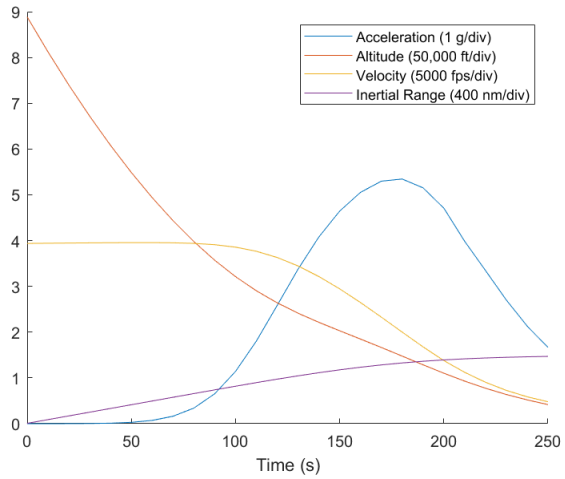


Figure 12: Neural Network Controller in Mars Model Flight Characteristics

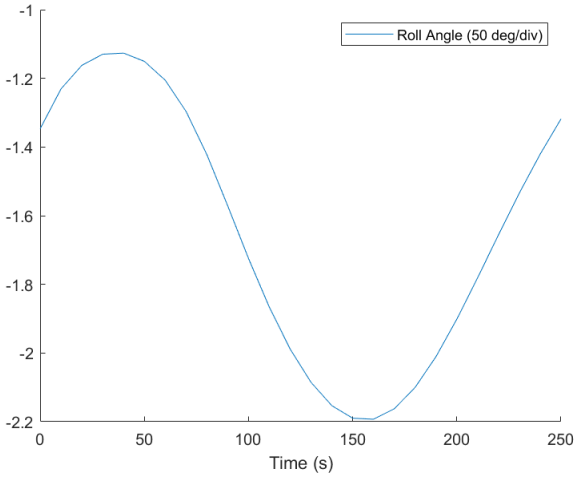


Figure 13: Neural Network Controller in Mars Model Bank Angle

However, when time step is increased to 5 seconds, a more accurate landing is shown with curves not smooth as before. The flight characteristics and bank angle behavior with 5 seconds time step are shown in Figure 14 and Figure 15.

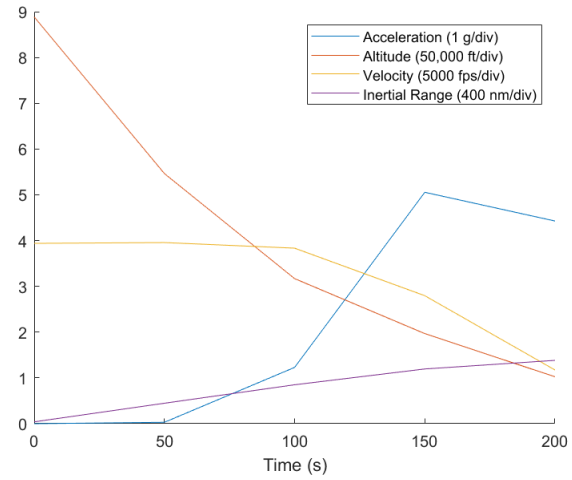


Figure 14: Neural Network Controller in Mars Model Flight Characteristics

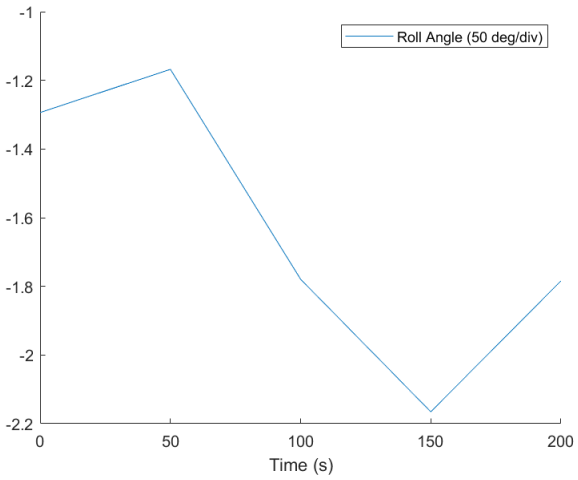


Figure 15: Neural Network Controller in Mars Model Bank Angle

It is seen that the capsule can land around 15 nautical miles designed as the requirement of track and range in Apollo guidance system with trained neural network controller.[12] With current low fidelity integration, the bank angle from neural network controller may change dramatically if the time step is too small. A high fidelity integration method has the potential to improve results.

Then the Monte Carlo simulations run 1000 times twice with varying initial positions and density. At first, each component of initial position varies in normal distribution with a nominal initial position of $r = [3525400, 0, 0]^T$ and a standard deviation of 500/3 m in x, y, and z directions. The results are shown in Figure 16 and Figure 17. Next the atmosphere density varies in normal distribution with a nominal density calculated by Equation (16) and a standard deviation of 10% of the nominal density. The results are shown in Figure 18 and Figure 19.

The red dot shows the actual landing location with nominal trajectory. The resulting final condi-

tions are still close to the target considering the scale. The missed distance are within a reasonable range.

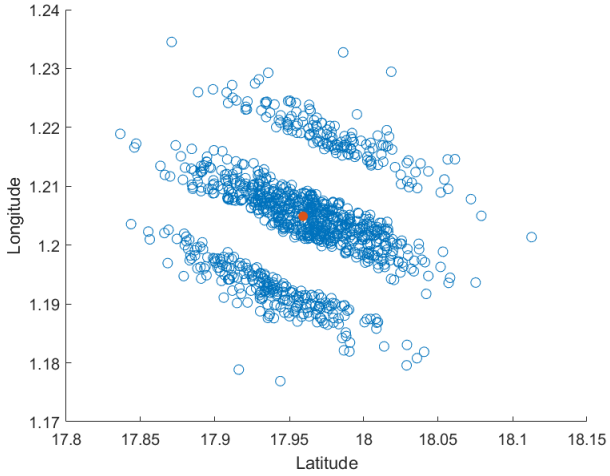


Figure 16: Latitude and Longitude Distribution in Monte Carlo Simulation of NN Controller in Mars Model Varying Initial Position

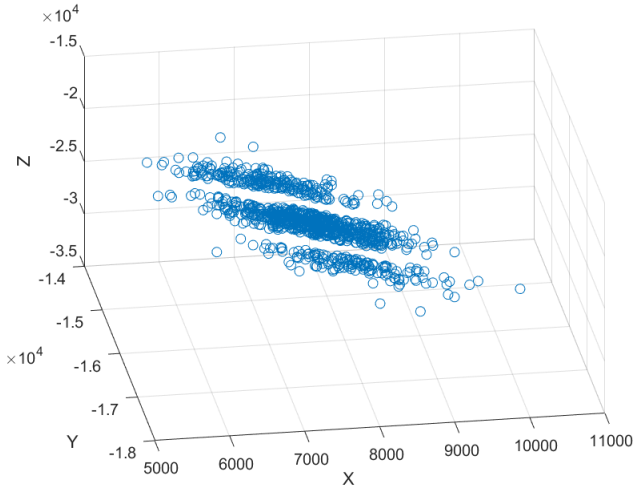


Figure 17: Missed Distance in Monte Carlo Simulation of NN Controller in Mars Model Varying Initial Position

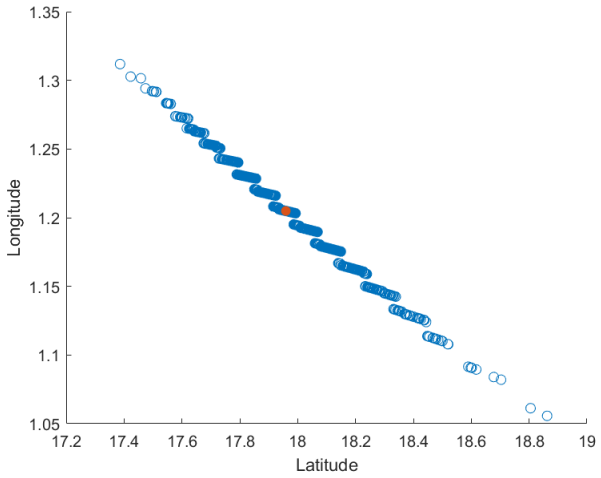


Figure 18: Latitude and Longitude Distribution in Monte Carlo Simulation of NN Controller in Mars Model Varying Density

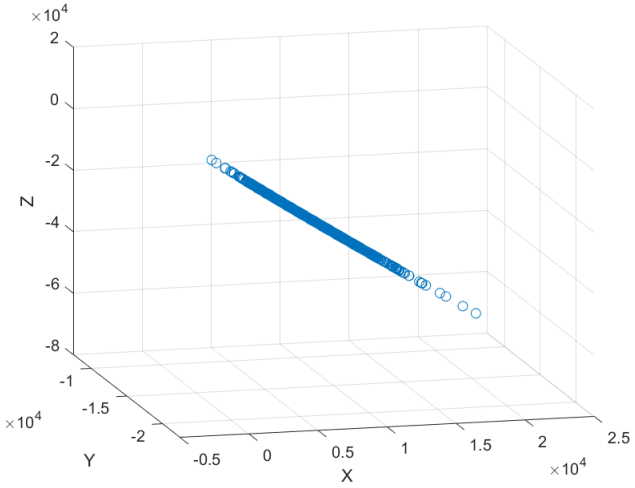


Figure 19: Missed Distance in Monte Carlo Simulation of NN Controller in Earth Model Varying Density

The results show that the neural network controller can still work in Mars and land the capsule on target within desired range even though they are only trained with data from Earth environment. This flexibility is notable as it provides a new approach for the control block design in new mission.

DISCUSSION

The results show that the neural network method works in both Earth and Mars cases. The Apollo simulation is successful since the plots of flight characteristics showed similar trends with real Apollo data. The data is then used to train the neural network. Several cases are tested to find the best performance of neural network. Finally, a case of 216 runs simulation including 1669934

points with time step 0.1 seconds is selected. It is applied as a replacement of the original control block in the same environment as Apollo case. It landed the capsule within a range of about 44 km. Although it is not within the desired range of 15 nautical miles about 27 km as stated in Apollo report, it provided a good starting point for Mars missions especially that most of the Mars mission landed about 100 km around the target.[12] The neural network is adopted to Mars model. All parameters are changed to Mars environment and the target is chosen within the designed range but the trained neural network is kept the same. The simulation showed that it can still land the capsule about 29 km around the target. The flexibility of the neural network application can be seen from this case. This result shows a more accurate landing than most of Mars missions. Additionally, this method is much more simple since the neural network can be trained on earth before the mission. Considering the neural network is trained with data from Apollo simulation with algorithm used in real Apollo mission, the results are more practical than in idealized situations.

Several sources of error affect the results presented. The most important one is lack of real data. All data used in this paper is from simulation and estimation. Although the Apollo results are based on the real Apollo re-entry algorithm which made this research more practical, not all required information is available. Mass, area and target position are all from estimation. For Mars model, even less information is accessible. If more real data of Mars entry is available in the future, the neural network can be trained with Mars data which will provide a more accurate and reliable way to design Mars entry controller.

In the Mars application, the maximum and minimum values of inputs and outputs used in normalization are from Earth model data because there is no real Mars data in this case. As a result, only when inputs and outputs are within the range of inputs and outputs in Earth model the neural network works well. As discussed above, if more Mars data is used to train the neural network, the limit of inputs and outputs will be changed.

Model accuracy is another important factor in application. Big uncertainties exist in Mars atmosphere model so far and a lot of assumptions are made to simplify the problem. The target is chosen only based on maximum range and lateral range, which is not the same as in real world since factors like shallow entry limits should also be considered. Furthermore, in our simulation the increments of velocity and position are calculated as simple products of time step and acceleration as well as velocity, where we may need a better integration method to reduce discrepancies with real data.

The simulation is not exactly the same as Apollo mission in several aspects. For the Apollo simulations, the present work did not use sensors data to update the velocity via the velocity increment detected as in the case of the real Apollo missions. In addition, the average-g method of updating velocity and position in real Apollo mission is not considered in our simulations. Moreover, interpolation was used in the final phase of Apollo mission to decide parameters (reference velocity for upcontrol, referece drag, etc.) where we looked up tables in Apollo report without interpolation in the simulation.[12] The current atmosphere model is slightly different from the one NASA used in the 1960's as well. Noise is not considered in all situations.

Different time step sizes were tried in both Earth model and Mars model applications to find the best performance of the controller. It is found that the larger time step within 5 seconds leads to better landing result (closer to target) than smaller step size. When the time step is small, 0.1 seconds, the bank angle as the output of neural network may experience several sharp changes and lead to some unexpected flight conditions due to the low fidelity integration method used in simulation.

During the neural network training, the number of inputs were also reduced from the initial 11. At the beginning, time, acceleration, velocity, position and target are all considered as inputs. However, during the training, we found only the distance from current position to target is needed. Since the controller is a feedback system, acceleration, velocity and position are related and updated with time. Too many inputs may cause over-expressed issue. The current chosen inputs showed better training results than the original ones.

If more Mars real data is available in the future, the neural network trained with Earth data can be transferred to Mars environment with more layers and continued to be trained with new data. As a result, more accurate control performance can be obtained.

CONCLUSION

This method has application in design, serving as a benchmark for other controllers where comparison data is not readily available. Two test cases are mainly presented, a baseline case of an Apollo capsule re-entering Earth atmosphere, which is verified with the actual Apollo re-entry flight data available in the literature, and a theoretical capsule entering Mars atmosphere, which can land on the target area with the controller derived from the new method. The results show that the neural network trained with Apollo data can replace the traditional control block to land the capsule on target. The errors are within acceptable ranges considering the scale and all sources of errors. The neural network controller shows its flexibility since it has been only trained with Earth environment simulation data but works in both Earth and Mars models. The neural network training is simple and efficient compared to the traditional way as well as other machine learning methods. This new method will be beneficial for future mission development and can serve as a good baseline for future entry methods. It has potential and can be useful when big uncertainties exist and not much data is available.

More work is needed to determine the feasibility of neural networks for future missions; however, there are many areas that have not yet been explored. Aside from the addition of more computing power, the next steps may include implementing sliding window method and reinforcement learning, as these may allow for better prediction and handling of large uncertainties. Sliding window method may smooth the curve when bank angle changes and reinforcement learning is widely considered in controls which may use this research as basis, although both are more complex than simple neural network. More real data and more accurate atmospheric model simulation will be helpful to improve the work, and more reasonable range and target can be determined based on the accurate Mars model as well. In this research, simple numerical integration is applied as in Apollo. However, in the future with high order numerical integration is used, the results may be improved.

REFERENCES

[1] K. D. Hicks, "Introduction to Astrodynamic Reentry," tech. rep., AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH GRADUATE SCHOOL OF ENGINEERING AND MANAGEMENT, 2009.

[2] 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.

[3] S. Li and X. Jiang, "Review and prospect of guidance and control for Mars atmospheric entry," *Progress in Aerospace Sciences*, Vol. 69, 2014, pp. 40–57.

[4] P. Lu, "Entry guidance: a unified method," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 713–728.

[5] Z. Putnam, S. Bairstow, R. Braun, and G. Barton, "Improving lunar return entry range capability using enhanced skip trajectory guidance," *Journal of Spacecraft and Rockets*, Vol. 45, No. 2, 2008, pp. 309–315.

[6] C. W. Brunner and P. Lu, "Skip entry trajectory planning and guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1210–1219.

[7] C. W. Brunner and P. Lu, "Comparison of fully numerical predictor-corrector and Apollo skip entry guidance algorithms," *The Journal of the Astronautical Sciences*, Vol. 59, No. 3, 2012, pp. 517–540.

[8] M. Tigges, T. Crull, J. Rea, and W. Johnson, "Numerical skip-entry guidance," 2006.

[9] S. H. Bairstow, *Reentry guidance with extended range capability for low L/D spacecraft*. PhD thesis, Massachusetts Institute of Technology, 2006.

[10] J. C. Harpold and C. A. Graves Jr, "Shuttle entry guidance," *American Astronautical Society, Anniversary Conference, 25th, Houston, Tex., Oct. 30-Nov. 2, 1978, 35 p.*, 1978.

[11] J. C. Harpold and D. E. Gavert, "Space shuttle entry guidance performance results," *Journal of Guidance, Control, and Dynamics*, Vol. 6, No. 6, 1983, pp. 442–447.

[12] R. H. Morth, *Reentry guidance for Apollo*. Massachusetts Institute of Technology, Instrumentation Laboratory, 1966.

[13] G. Mendeck and L. Craig, "Entry guidance for the 2011 mars science laboratory mission," *AIAA Atmospheric Flight Mechanics Conference*, 2011, p. 6639.

[14] A. M. San Martin, S. W. Lee, and E. C. Wong, "The development of the MSL guidance, navigation, and control system for entry, descent, and landing," 2013.

[15] Z. Yu, P. Cui, and J. L. Crassidis, "Design and optimization of navigation and guidance techniques for Mars pinpoint landing: Review and prospect," *Progress in Aerospace Sciences*, Vol. 94, 2017, pp. 82–94.

[16] X. Jiang, R. Furfaro, and S. Li, "INTEGRATED GUIDANCE FOR MARS ENTRY AND POWERED DESCENT USING REINFORCEMENT LEARNING AND GAUSS PSEUDOSPECTRAL METHOD,"

[17] S. Li and X. Jiang, "RBF neural network based second-order sliding mode guidance for Mars entry under uncertainties," *Aerospace Science and Technology*, Vol. 43, 2015, pp. 226–235.

[18] B. Gaudet, R. Linares, and R. Furfaro, "Deep Reinforcement Learning for Six Degree-of-Freedom Planetary Powered Descent and Landing," *arXiv preprint arXiv:1810.08719*, 2018.

[19] W. Engelund, R. Powell, and R. Tolson, "Atmospheric Modeling Challenges and Measurement Requirements for Mars Entry, Descent and Landing," *LPI Contributions*, Vol. 1447, 2008, p. 9025.

[20] H. L. Justh, C. G. Justus, and H. S. Ramey, "The next generation of Mars-GRAM and its role in the autonomous aerobraking development plan," 2011.

[21] A. Chen, A. Vasavada, A. Cianciolo, J. Barnes, D. Tyler, S. Rafkin, D. Hinson, and S. Lewis, "Atmospheric risk assessment for the Mars Science Laboratory entry, descent, and landing system," *Aerospace Conference, 2010 IEEE*, IEEE, 2010, pp. 1–12.

[22] G. R. Center, "Earth Atmosphere Model," May 2015.

[23] M. v. Gerven and S. Bohte, *Artificial neural networks as models of neural information processing*. Frontiers Media SA, 2018.

[24] W. G. Hatcher and W. Yu, "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends," *IEEE Access*, Vol. 6, 2018, pp. 24411–24432.

[25] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, Vol. 18, No. 7, 2006, pp. 1527–1554.

[26] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, Vol. 234, 2017, pp. 11–26.

[27] C. Karlgaard, P. Kutty, J. Shidner, M. Schoenenberger, and M. Munk, "Mars entry atmospheric data system trajectory reconstruction algorithms and flight results," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 28.

[28] G. R. Center, "Speed of Sound," March 2018.