AN ADAPTIVE ATMOSPHERIC PREDICTION ALGORITHM TO IMPROVE DENSITY FORECASTING FOR AEROCAPTURE GUIDANCE PROCESSES

A Dissertation
Presented to
The Academic Faculty

by

John J. Wagner

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2014

AN ADAPTIVE ATMOSPHERIC PREDICTION ALGORITHM TO IMPROVE DENSITY FORECASTING FOR AEROCAPTURE GUIDANCE PROCESSES

Approved by:

Dr. Alan Wilhite, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Robert Braun
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Richard Powell
Analytic Mechanics Associates

Dr. Eric Johnson
School of Aerospace Engineering
Georgia Institute of Technology

[October 07, 2014]

“Ad maiorem Dei gloriam”
(For the greater glory of God)

-- Ignatius of Loyola
Soldier, Priest, & Roman
Catholic Saint

To all those who work humbly, quietly, and peacefully for a better world…

ACKNOWLEDGEMENTS

There are many people whose thoughts, encouragement, and support contributed greatly to this work. First, I would like to express gratitude to my advisor and chairman, Dr. Alan Wilhite, for his guidance and insight throughout my years in his program. His advice and experience played a major role in the development of this thesis and in my work over the years. The opportunities he has afforded me have taught me valuable life-long lessons for which he has my thanks and respect. I would also like to thank Dr. Doug Stanley for his advice and mentorship throughout my time at NASA and Georgia Tech. His quick and penetrating intellect and his good humor not only enhanced the clarity and quality of my work but also made it exciting and enjoyable. I'd also like to thank my research mentor at NASA Langley, Dick Powell, who spent many hours patiently guiding me through the labyrinth of knowledge surrounding the POST trajectory simulator. His skill in flight dynamics and his concise, intuitive approach to problem solving were both invaluable and generously provided when I needed it most. I would also like to thank Dr. Daniel Schrage, Dr. Eric Johnson, and Dr. Robert Braun for their service on my committee and their insightful support over the past several years.

I would be deeply remiss if I did not include in my gratitude the leadership and members of the Atmospheric Flight and Entry Systems Branch at NASA. Their constant financial, intellectual, and personal support made my time at NASA a true privilege. My thanks especially to Jeff Herath, Walt Engelund, Jill Prince, Lou Glaab, and all the branch leadership, former and current. I'd also like to extend my warm thanks to my

fellow graduate students and colleagues at the National Institute of Aerospace for the friendship and good humor we've shared over the years. Similarly, thanks to my friends at Our Lady of Mount Carmel and the Hampton Roads Young Adults; your antics and kindness will remain with me as I make a new start in northern Virginia.

Finally, I'd like to extend my ongoing gratitude to my family, most especially to my parents, Joseph and Elizabeth Wagner. Their caring support, encouragement, sympathy, and faith have armed me for my daily battles and steadied me in moments of uncertainty. Thanks also to my young brother whose memory we cherish and whose presence is unmistakable. Thank you all for helping me shape a brighter and better future!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ADALINE	Adaptive linear neuron
ADS	Air data system
ALRM	Apoapsis lower/raise maneuver
BIBO	Bounded input, bounded output
CL	Closed loop
DIAL	Differential absorption lidar
DOF	Degrees of freedom
EDL	Entry, descent, and landing
EDLSA	Entry, descent, and landing systems analysis
GCM	Global circulation model
GNC	Guidance, navigation, and control
HYPAS	Hybrid predictor-corrector aerocapture scheme
IAD	Inflatable aerodynamic decelerator
IMU	Inertial measurement unit
IMU/ADS	Inertial measurement unit coupled with an air data system
LAT	Vehicle geocentric latitude, deg
LONG	Vehicle longitude, deg
LOS	Line-of-sight
LRRP	Laser risk reduction program
MADALINE	Multiple adaptive linear neurons
MCD	Mars climate database
MER	Mars exploration rover
MGS	Mars global surveyor
MM	Mesoscale model
MRO	Mars reconnaissance orbiter
MSL	Mars science laboratory
NN	Neural network
NPC	Numerical predictor-corrector
OCT	Office of the chief technologist
OL	Open loop

PCM	Plane change maneuver
PCR	Principal component regression
PENS	Plastic ensemble neural system
PLS	Partial least squares
POST2	Program to optimize simulated trajectories 2
PRM	Periapsis raise maneuver
S/C	Spacecraft
SEADS	Shuttle entry air data system
UTC	Universal time coordinated
WTA	Winner-take-all
a,b	Exponential model parameters
A_{body}	Spacecraft body acceleration vector, m/s^2
a_{drag}	Drag acceleration, m/s^2
$[b]$	ADALINE offset/bias vector
C	Centrifugal force, N
C_D	Drag coefficient
C_L	Lift coefficient
D	Drag force, N
di_{err}/dt	Time rate of change of the inclination error, deg/s
e	Instantaneous error of a buffered solution, % density error
e_i	Error signal of the i^{th} ensemble member, % density error
$[e_i]$	Vector (discreet) form of error signal of the i^{th} ensemble member, % density error
e_{min}	Error of the worst buffered solution, % density error
$F10.7$	10.7 cm wavelength full disc solar emission, 10^{-22} W m^{-2} hz^{-1}
F_L	Lower root boundary in the bisection algorithm, km
F_U	Upper root boundary in the bisection algorithm, km
G	Gravitational force , N
h	Altitude, km
h_o	Reference altitude, km

H	Basis matrix
$h(t)$	Guidance estimated altitude history, km
i	Ensemble member index (1,2…n)
i_{db}	Inclination deadband boundary, deg
i_{err}	Inclination error, deg
i_{ref}	Reference inclination for deadband boundary, deg
i_{tol}	Inclination tolerance for deadband boundary, deg
I	Induced drag force, N
j	Altitude vector index or along-track vector index (j = 1,2…m)
k	Index of buffered solutions (k = 1,2…b)
K	HYPAS density scale factor
L	Lift force, N
L/D	Lift-to-drag ratio
L_s	Solar longitude, deg
m	Vehicle mass, kg
N	Nadir lift force component, N
p_{dyn}	Dynamic pressure, Pa
R^2	Coefficient of determination
r	Orbital radius, km
RMSE	Root Mean Square Error
$RMSE_i$	Root Mean Square Error of the i^{th} ensemble member
S	Aerodynamic reference area, m^2
S_L	Aerodynamic reference area to compute lift force, m^2
S_D	Aerodynamic reference area to compute drag force, m^2
T	Cross-track lift component, N
V	Vehicle surface relative velocity, m/s
V_{FS}	Free stream velocity, m/s
V_{iner}	Spacecraft inertial velocity vector, m/s
V_{scale}	Deadband corridor scale velocity, m/s
V_{shift}	Deadband corridor shift velocity, m/s
V_W	Atmospheric wind velocity, m/s

w_i	i^{th} ensemble member unscaled weight
$w_{i,min}$	Minimum unscaled ensemble weight
$\hat{x}$	Generalized estimated parameters vector
X_{iner}	Spacecraft inertial position vector, m
$\tilde{y}$	Measured data observations vector
$\hat{y}$	Generalized estimated physical quantity vector
α	Angle of attack, deg
β	Vehicle bank angle, deg
β_{ss}	Angle of sideslip, deg
γ	Span of scaled residuals
Γ	Flight path angle, deg
ε	Orbital specific energy, km^2/s^2
$\theta(t)$	Guidance estimated latitude history, deg
λ	Aerocapture failure rate
μ	Gravitational parameter, km^3/s^2
ρ_A	Average model ensemble density signal, kg/m^3
ρ_{ADS}	Air data system measured density, kg/m^3
ρ_C	Cold model ensemble density signal, kg/m^3
ρ_{echo}	Ensemble echo density signal, kg/m^3
$\rho_{K\text{-}EXP}$	HYPAS-derived exponential density model, kg/m^3
$\rho_{K\text{-}Table}$	HYPAS-derived tabular density model, kg/m^3
ρ_{PENS}	PENS density model, kg/m^3
ρ_{KPENS}	KPENS density model, kg/m^3
ρ_i	i^{th} ensemble member density signal, kg/m^3
$[\rho_i]$	Ensemble density vector, kg/m^3
ρ_{LS}	Least squares density model, kg/m^3
ρ_m	Minimum model ensemble density signal, kg/m^3
ρ_M	Maximum model ensemble density signal, kg/m^3

ρ_{pred}	Predicted Density Field, kg/m^3
ρ_W	Warm model ensemble density signal, kg/m^3
ρ_∞	Free stream density, kg/m^3
$\rho(t-)$	Guidance estimated density history, kg/m^3
$\rho(t+)$	PENS/Guidance predicted density field, kg/m^3
σ	Standard deviation
τ_{dust}	Dust opacity (dust optical depth)
$\varphi(t)$	Guidance estimated longitude history, deg
$[\omega]$	Scaled weight vector
ω_A	Average model scaled weight
ω_C	Cold model scaled weight
ω_e	Top-level ensemble echo weight
$[\omega_{echo}]$	Ensemble echo scaled weight vector
ω_i	ith ensemble member scaled weight
ω_m	Minimum model scaled weight
ω_M	Maximum model scaled weight
ω_W	Warm model scaled weight

SUMMARY

As the cost and complexity of deep space missions continue to climb, the interaction of spacecraft with highly dynamic atmospheric environments will increase in frequency. Large and unpredictable departures of atmospheric properties from anticipated values, most especially density departures, can cause significant complications for entry guidance processes. These departures, or atmospheric dispersions, indicate the level of uncertainty in the collective knowledge of a planetary atmosphere. The Martian environment and its atmosphere in particular have been the subject of study for many years and yet a high degree of uncertainty still remains in contemporary density models. This uncertainty varies between 20-40% at aerocapture periapsis altitudes depending on location, season, time of day, solar activity, suspended dust distribution, and other variables and assumptions. In the face of high uncertainty, mission designers and engineers are compelled to apply large trajectory and design safety margins which typically drive the system design towards less efficient solutions with smaller delivered payloads.

Aerocapture is a method for inserting a spacecraft into an orbit about a planetary body with an atmosphere without the need for significant propulsive maneuvers. This can reduce the required propellant and propulsion hardware for a given mission which lowers mission costs and increases the available payload fraction. While the potential system level benefits of aerocapture are great, so too are the risks associated with this mission class in highly uncertain atmospheric environments such as Mars. Large magnitude density dispersions have a particularly acute effect on aerocapture trajectories due to the

interaction of the high required speeds and relatively low densities encountered at aerocapture altitudes. This allows the spacecraft to rapidly enter regions of significantly higher- or lower-than-predicted density which complicates the guidance and control problem. Unfortunately, many contemporary guidance algorithms rely on simple analytic or tabular atmosphere models for state prediction and correction. Some of these algorithms include an on-board density measurement capability to permit density model adaptation. This provides some measure of adaptability to the density prediction model but significant density prediction errors may still remain. The high level of uncertainty (i.e. error) associated with these elementary models impacts the guidance decision making process and ultimately increases the likelihood of one or more guidance failure modes.

This work develops a new class of adaptive density estimator called a Plastic Ensemble Neural System (PENS). The aim of this estimator class is to reduce the impact of the atmospheric uncertainty on the aerocapture guidance process by generating high fidelity, adaptable density prediction models. PENS makes use of a powerful forecasting tool known as ensemble modeling wherein several density models spanning the anticipated range of day-of-flight conditions are stored aboard the vehicle. These on-board models are derived from numerical weather simulations but are stored as neural networks in order to reduce data storage and processing requirements. The PENS algorithm constructs a linear combination of selected on-board neural networks to form the density prediction model. The weights applied to each of the neural models in the linear combination are the result of an associative learning process made possible by the introduction of a new construct known as an ensemble echo. By remembering accurate

solutions from earlier in the trajectory, the echo enables the prediction model to evolve with increasing atmospheric exposure by incorporating newly observed density information as it becomes available. In this manner, the ensemble echo seeks to identify and accentuate prediction models which balance good local and good long-range density prediction performance. This approach converts a computationally expensive numerical density forecasting problem into a much simpler problem which may be readily solved with limited on-board resources. By utilizing many atmosphere models but delaying judgment of model quality until *in situ* information is available, the PENS estimator produces density prediction models which limit the impact of atmospheric uncertainty on the guidance system.

The PENS estimator is applied to a numerical guidance system and the performance of the composite system is investigated with over 144,000 guided trajectory simulations. The results demonstrate that the PENS algorithm achieves significant reductions in both the required post-aerocapture performance, and the aerocapture failure rates relative to historical density estimators. Average reductions of 33% in both ΔV and failure rates relative to contemporary methods are commonly observed with much greater reductions in both quantities also observed in many scenarios.

CHAPTER 1

INTRODUCTION

On January 4th, 2004, the Mars Exploration Rover (MER) Spirit successfully touched down in Gusev Crater 13.4 km from its intended landing site. Twenty one days later, its sister ship, Opportunity, followed by making a successful landing at Meridiani Planum nearly 15 km from its targeted landing location.[1] These landings, while demonstrating a highly respectable degree of accuracy for unguided entries, were nevertheless many kilometers from their targeted impact point. Reconstruction of these entries has revealed density profiles lower than the best pre-entry predictions. This outcome is not unexpected given the high degree of dynamism in the Martian atmosphere. This dynamism can generate daily and even hourly changes in the atmospheric state which complicates efforts to predict the atmospheric density prior to arrival. Many entry vehicles, however, rely on knowledge of atmospheric properties, specifically atmospheric density, for trajectory prediction and control. Consequently, the uncertainty associated with many common atmospheric density estimators may allow guided entry systems to make improper decisions which increase the risk exposure of the vehicle in flight and may lead to one or more guidance failure modes. The ability to better predict density in highly dynamic atmospheric environments would enable more precise targeting of final states (such as landing sites or post-entry orbits) as well as a reduction in vehicle risk exposure.

The high level of atmospheric uncertainty coupled with the frequency of robotic missions to Mars make it an ideal testing ground for candidate methods to improve the performance of atmospheric density estimators. The current approach for mitigating the risks posed by atmospheric uncertainty is the application of large system and trajectory margins. For example, in order to permit a landing mission to more accurately target a

desired landing site, additional maneuvering propellant mass may be added to the vehicle. This permits the vehicle to retroactively correct for atmospheric dispersions after they have already impacted the trajectory and thereby permits the entry system to counteract greater levels of atmospheric density uncertainty. This historical margining approach has proven to be effective for unguided entry applications to date. However for both guided and unguided missions, if the extreme case atmospheric state predictions assumed to drive the system requirements are incorrect, the vehicle is still exposed to unanticipated risk despite the addition of these expensive design margins. If, however, knowledge of the atmospheric properties can be improved, guided entry systems would be able to not only achieve increased final state targeting accuracy but also successfully function with smaller system and trajectory uncertainty margins leading to more efficient vehicle designs.

1.1 Study Goals & Scope

The goal of this research, therefore, becomes the creation of an improved atmospheric density estimator which reduces the impact of atmospheric uncertainty on guided entry trajectories. This is accomplished through the introduction of an adaptive density prediction algorithm developed using artificial intelligence and numerical weather forecasting techniques. Guidance decisions based on a highly uncertain atmosphere model may lead to incorrect guidance decisions which increase the risk of failure of the entry. To remediate this issue, the adaptive prediction algorithm modifies the density model used by the guidance system to better agree with *in situ* density measurements.

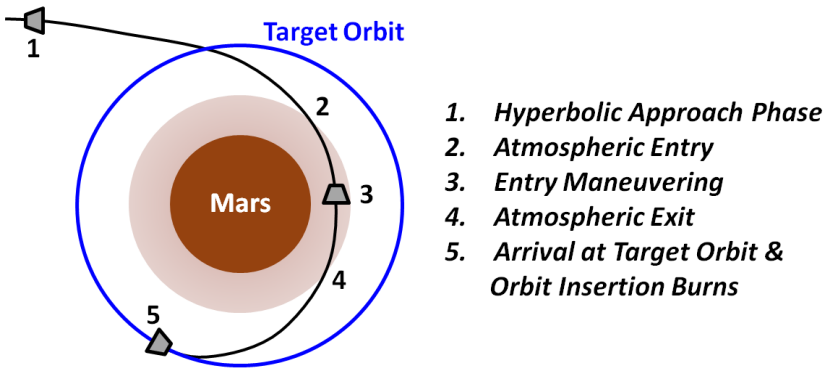


Figure 1.1: Fundamental Phases of Aerocapture Trajectories

While atmospheric dispersions can have a profound effect on any type of entry mission, they have an especially significant and unique impact on aerocapture missions. Aerocapture is a technique used to insert a vehicle into an orbit about a planetary body with an atmosphere without the need for large propulsive maneuvers (Figure 1.1). The vehicle enters the atmosphere from a hyperbolic approach trajectory using atmospheric drag to reduce its velocity. The atmospheric exit state is targeted to produce a Keplerian coast to the desired target orbit where a small insertion maneuver is performed. Note that an alternative approach for aerocapture is to seek any stable orbit rather than a specific post-aerocapture state. Propulsive orbit correction may then be used to reach the desired target orbit if necessary. It may even be possible to carefully design mission architectures such that a specific target orbit is not required. These approaches are more robust to atmospheric density uncertainty but also may require larger post-aerocapture ΔV budgets to account for atmospheric uncertainty. These alternative approaches are not examined herein in order to focus on improvement of the atmospheric guidance process.

In general, the expected density variability in an atmosphere decreases with altitude (see Figures. 1.2-1.3). Thus for landing missions, the vehicle begins at high altitude in a region of high atmospheric uncertainty and descends into regions of continually decreasing uncertainty. From the perspective of density forecasting, this is not as stressful a testing scenario as an aerocapture mission which must descend from a

region of higher uncertainty (at entry interface) to a region of lower uncertainty (at periapsis). The vehicle must then make density predictions concerning the outbound leg in order to safely guide itself back upward through regions of higher uncertainty. While the Martian atmosphere is in a state of approximate hydrostatic equilibrium, density structures such as gravity waves, density tidal waves, and planetary-scale flow structures[2-4] likely limit the utility of global analytic regression models. Because the outbound leg occurs in a completely different spatial region then the inbound leg, the use of inbound density trends to predict the outbound density profile will likely yield limited results. Throughout an aerocapture maneuver, density dispersions (i.e. density uncertainty) coupled with the high required speeds allow the vehicle to rapidly enter regions of density which are much higher or lower than predicted. These density pockets, shears, or long-term biases place the spacecraft at risk of violating hard constraints or controllability limits. This can lead to irrecoverable conditions such as skip-out or complete vehicle destruction. Even if the guidance rapidly compensates for the dispersion, soft failures such as large final state errors can occur which may jeopardize mission goals.

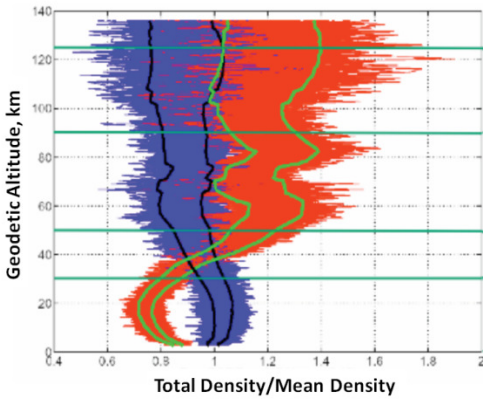


Figure 1.2: Density Variability in MarsGRAM 2005[5]
Blue = total density (τ_{dust} = 0.3) / mean density (τ_{dust} = 0.45)
Red = total density (τ_{dust} = 3.0) / mean density (τ_{dust} = 0.45)
Solid Lines are ±3σ values

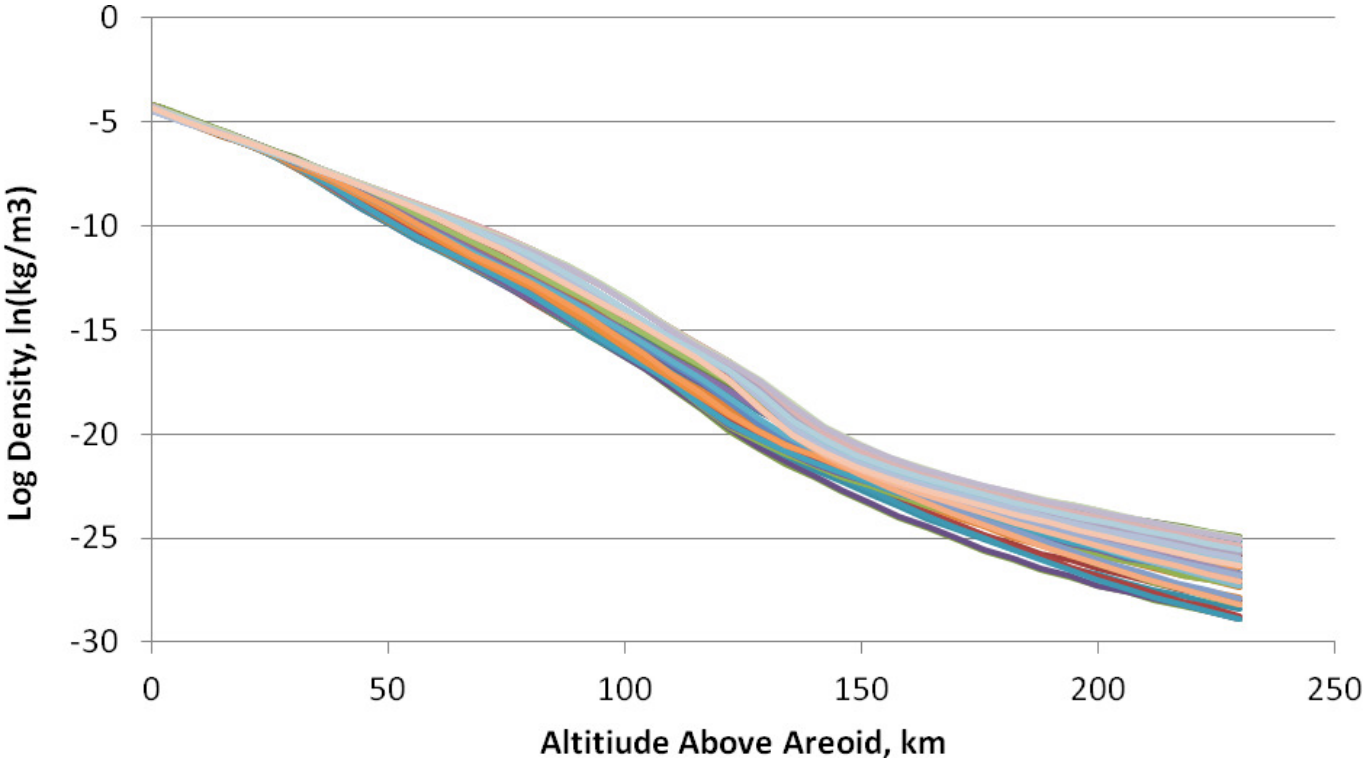


Figure 1.3: Temporal Density Variability with Dust & Solar Flux in Mars Climate Database

Vertical Altitude Profiles at 0°N, 0°E

5 dust & solar flux scenarios, 12 monthly profiles per scenario

It should be noted that the impact of atmospheric density on the entry trajectory will diminish with altitude. For aerocapture, the precise entry and exiting altitudes where the density no longer has a significant effect on the trajectory is largely a function of the desired target orbit. The target orbit dictates the entry and exiting flight path angles which largely determine the length of time the entry vehicle spends in the highly variable upper atmosphere. Rapid transits through this region, such as those produced by higher energy post-aerocapture target orbits, will tend to limit the effects of the density variability on the aerocapture trajectory. Slower, shallower transits through the upper atmosphere, typically produced by lower energy target orbits, will be more susceptible to density variations at in the upper atmosphere (see sections 3.3.7 and 5.2).

Therefore the nature of the target orbit greatly influences the maximum altitude at which the atmosphere no longer significantly affects the trajectory. However, in order to ensure that all interactions between the guidance system and the atmosphere are captured, all aerocapture trajectories examined herein are simulated from the entry interface altitude of 128 km through periapsis and back up to atmospheric exit at 128 km. These conditions were selected based on the work of Masciarelli, et al[6] because these conditions have a high likelihood of capturing the complete altitude region of interest for most aerocapture missions regardless of the target orbit.

1.2 Effects of Density Dispersions on Contemporary Guidance Schemes

Despite restricting the testing environment to aerocapture missions, it is instructive to examine a wider scope of literature for both aerocapture and landing in order to gain insight into the effects of density dispersions on entry guidance. Given the highly dynamic nature of the Martian atmosphere, it is reasonable to assume that the accuracy of atmospheric state predictions based on static models (analytic equations or tables) will always yield limited results. Unfortunately, unguided entries must rely on such static atmospheric state predictions. These predictions are typically derived through

probabilistic processes taking into account dispersions and multiple data sources such as wind field predictions from mesoscale models.[7-9] The entry state vector is controlled during the approach phase such that the center of the landing dispersion ellipse coincides with the nominal landing site. Because the onboard systems do not have any means of taking corrective action during the entry, this probabilistic process is effectively reduced to a static best estimate. If the predicted nominal atmosphere (i.e. the atmospheric state which occurs with the highest probability) is incorrect, the vehicle will be off-target.

For guided entries, such as the Mars Science Laboratory (MSL), the ability of the vehicle to correct its state vector during the entry enables the use of dynamic atmosphere models which can adapt to day-of-entry conditions. However, many proposed entry guidance algorithms for aerocapture[6,10-13] and landing,[14-16] including MSL,[17] utilize simple atmosphere models (i.e. an exponential/isothermal density model) to form state predictions and derive control signals and guidance gains. Many of these studies conclude that large atmospheric dispersion and density anomalies are a significant contributor to final state error and drive guidance failure modes.

For example, Fuhry notes that the efficacy of predictive guidance during the critical aerocapture exit phase is "greatly affected by the presence of unpredictable variations in atmospheric density."[11] This is caused by the dependency of the guidance on predictions of future vehicle control capability. "…The presence of density shears, which greatly change this control capability, can cause large target misses and fuel penalties."[11]

In studying Apollo-derived Mars landing guidance, Carman et al. found that dispersions in atmospheric density from the assumed exponential model can result in saturation of the bank command channel which inhibits the ability of the guidance system to reduce position errors.[14] Tigges and Ling developed a predictive equilibrium glide guidance scheme based on an exponential atmosphere assumption for a Mars sample return lander.[15] They found that constant density biases (unanticipated density shifts over an extended time period) generated large final state errors for entry from a 1 Sol orbit and

suggest the use of a circular orbit to reduce the errors associated with atmosphere dispersions. Note that a Sol is one Martian day lasting approximately 24 hours and 39 minutes.

In studying the effects of large atmospheric dispersions on Martian aerocapture guidance, Ess[18] notes that the use of exponential atmosphere models in the presence of density biases causes mispredictions in the desired drag deceleration profile. A finite time interval is required for the guidance to determine that significant error is accumulating and take action. This causes the corrective action of the guidance to lag behind the fluctuations in the mean density field. If the mean density shifts yet again while the guidance is still correcting for the first bias, the action of the guidance may be completely wrong for the environment the vehicle is currently experiencing. For example, if the vehicle flies into a higher-than-predicted density region, the guidance will compensate by commanding an increase in altitude to reach a more desirable density region. If the bias then reverses or disappears altogether, the vehicle is now at higher altitude than desired and therefore in a region of dangerously low density. Extremely low density degrades the control authority of lifting entries through attenuation of the lift vector magnitude. This situation can lead to a loss of control and possible hard failures such as skip out.

A plausible scenario illustrating this example is an unanticipated transit through a developing dust storm which could cause a large increase in mean density as the vehicle enters the storm and a large drop in mean density upon exit. A chance encounter with a developing dust storm during entry may seem unlikely. However, Knocke et al. notes that the atmosphere models for both MER entries were altered during the final approach phase due to a dust storm which reached its peak in mid-December 2003[1]. Therefore, of the seven successful landing missions at Mars, the entries of at least two were directly influenced by a dust storm.

For aerocapture at Earth, Skalecki et al. examined the effects of both uniform density biases over the entire entry as well as density “potholes” (i.e. persistent density

biases with a finite length which end at some point during the entry).[19] They concluded that uniform density biases presented little difficulty for their bank-modulated drag reference method. However, they note that pothole variations presented significant difficulty and, like Ess[18], acknowledge that density potholes create failure scenarios due to guidance lag. It was also noted that performance was sensitive to the point where the density pothole was applied and the most critical point was generally near periapsis. This matches intuition because peak deceleration should occur in the region of highest free stream density which occurs near periapsis.

Cianciolo, et al. conducted a study to parametrically assess the impact of pothole density dispersions on the nominal MSL entry trajectory.[20] The study notes that such pothole dispersions have been observed for shuttle orbiter entries at Earth where the density can shift by up to 60% over short durations.[21-23] The nominal MSL trajectory quickly dives through the highly variable upper Martian atmosphere then levels off around 13 km and enters near-level flight for more than one minute. It was determined that if the density potholes are strategically positioned, in this case below 20 km, the altitude of parachute deploy could decrease by 3-4 km and range errors at parachute deploy could approach 14 km.

From examination of the findings above, it seems reasonable to conclude that many entry guidance algorithms for both landing and aerocapture are vulnerable to at least one worst case density anomaly. Most seem susceptible to long lived density biases which begin during the trajectory or density potholes which both begin and end during the trajectory. Guidance algorithms appear most vulnerable near the region of peak deceleration which is an intuitively reasonable conclusion. If strategically positioned and of sufficient magnitude, density anomalies can disrupt vehicle guidance for both landing and aerocapture and result in large final state errors or complete guidance failure.

1.3 Guidance Failure Modes & Remediation Methods

While density anomalies are the root cause of the guidance disruption, the above literature also offers insight into how such worst case anomalies produce guidance failures. These effects include saturation of the controller[14], lags in guidance response caused by rapid density changes[18,19], and misprediction of remaining control authority[11,18].

Controller saturation occurs when the necessary command authority to execute a maneuver exceeds the available authority. Many Martian entry guidance schemes achieve control through modulation of the lift vector direction. Avoiding controller saturation, therefore, implies use of higher L/D to generate sufficient control authority in the presence of severe density dispersions. However, several studies[18,19,24] examined the effects of varying L/D and conclude that higher L/D may also introduce significant control difficulties and does not immediately ensure a successful entry. In addition, higher L/D shapes typically produce mass penalties due to packaging inefficiencies. Controller saturation could also be prevented by adding degrees of freedom to the vehicle to improve the formulation of the controls problem that the guidance must solve. This could include the addition of a speedbrake or inflatable decelerator to provide control over the ballistic coefficient or the addition of a body flap to modulate trim angle of attack. However, all of these solutions immediately generate a mass penalty.

As discussed earlier, guidance lag is caused by rapid large magnitude dispersions which prevent the guidance from keeping pace with the changes in the atmospheric density. Guidance lag could be reduced by the design of rapidly responding controllers but controller stability must still be considered. Even if a fast, stable controller is available, the vehicle must be capable of responding to the rapid control signals. Therefore, high roll rates and angular accelerations are required which increase the actuator (maneuvering jets or aerodynamic surfaces) requirements and ultimately add mass to the system.

Misprediction of remaining control authority, particularly important for aerocapture as noted by Fuhry[11], can be most directly addressed through more accurate prediction of the density field in front of the vehicle. Note that this is the only method which does not immediately lead to a mass penalty and even has the potential to reduce vehicle dry mass. An adaptive atmosphere prediction capability allows the vehicle to take *proactive* guidance action to counter upcoming atmospheric conditions. Proactive guidance could potentially yield a reduction in required angular rates and accelerations because the vehicle has more time to plan and execute guidance decisions. This would reduce the requirements on both the control system and control effectors thereby permitting smaller control surfaces or lower-thrust maneuvering jets. The additional planning time also provides a degree of protection against controller saturation. Proactive vehicle guidance would also allow a revision of the historical margining strategy which would reduce overdesign, enable the consideration of more landing sites, and increase the reliability of unproven techniques such as aerocapture.

1.4 Antecedents

Several methods for density prediction for entry vehicles have been previously developed. Perot and Rousseau developed a least squares fitting approach to tune a standard exponential atmosphere model using reconstructed density data taken in flight.[25] They use this model to approximate the density scale height during the critical exit phase of an aerocapture mission for two different guidance algorithms. Despite producing significant increases in final state accuracy their periapsis altitudes still vary by several hundred kilometers.

Cerimele and Gamble[10] also use reconstructed density data to determine a corrective multiplier for an exponential density model. This multiplier is taken to be the measured density divided by a reference density from the reference atmosphere used to derive the control equations. While their simulations and results assume Earth as the

central body and not Mars, they note that large final state errors can result from large magnitude atmospheric density dispersions. The same approach is also used by Masciarelli, et al.[6]

Fuhry employs a similar corrective multiplier to the Cerimele-Gamble scheme but incorporates the parameter into a Kalman filter to predict its future state based on current information[11]. Fuhry finds that large final state errors can occur in the presence of large magnitude dispersions (see Section 1.1). Another related approach is taken by Vijayaraghavan who foregoes the computation of a multiplier and estimates the parameters of an exponential model directly using a Kalman filter[12]. Dispersed trajectories are created by changing the reference density of the exponential model throughout the trajectory each time the guidance is called. It is determined that significant final state errors can occur in the presence of large magnitude dispersions.

Note that all these schemes have the potential to enter a guidance failure mode due to atmospheric dispersions and all are reliant, in some degree, on an exponential atmosphere model. It seems reasonable, therefore, to deduce that improved performance in the Martian environment may be possible if simple analytic atmosphere approximations such as exponential models are replaced with higher fidelity models.

It should also be stated that many guidance schemes[13-17,27-28] for atmospheric entry avoid direct treatment of the atmosphere in the guidance method, opting instead to correct for density dispersions after they affect the trajectory. For example, the Space Shuttle reentry guidance scheme determines control actions based on deviations of the actual drag acceleration from a reference profile. The vehicle guidance relies on multiple degrees of freedom to absorb density dispersions (recall the discussion in Section 1.2.). Bank angle modulation is used to control the lift vector direction for long-period corrections and maneuvers. Short-term corrections for dispersions and other deviations from the desired reference trajectory are accomplished through modulation of the angle of attack which changes the drag coefficient.[26] It should also be noted that Shuttle

guidance, while effective, relies on the relatively high terrestrial densities at lower altitudes coupled with its considerable L/D to ensure accurate final state targeting. In the Martian environment, the low atmospheric densities degrade control authority at slow speeds. This effect is exaggerated if the vehicle L/D is small as is typically the case for Martian entry vehicles. Lastly, it is desirable to avoid the addition of mass necessary to add guidance degrees of freedom such as angle of attack control.

The Apollo reentry guidance scheme uses an exponential atmosphere assumption to derive the equations used by the guidance system. The full Apollo reentry guidance routine requires an estimate of the atmospheric scale height as well as a reference density and a corresponding reference altitude[27,28]. During the final phase of the entry, deviations from the desired reference trajectory due to any perturbative effects including density dispersions are corrected by rolling the lift vector direction after the dispersion has already affected the trajectory. The MSL entry guidance algorithm, based on this final phase of the Apollo reentry guidance, makes no direct use of any atmospheric parameters.[14,17] (*n.b.* The guidance gains, however, are derived using an exponential atmosphere assumption.) Instead, the guidance accounts for dispersions by examining the departures of the sensed lift and drag accelerations from the reference profile. This approach, while simple and effective for Apollo due to the nature of the Earth's atmosphere, exposes MSL to risk throughout the entry. As is noted by Mendeck and Carman, "atmospheric density dispersion is a principal factor that degrades the guidance ability to converge to [the desired] drag versus velocity profile." They conclude that drag error caused by density dispersions is the principle contributor to landing site miss distance.[17]

From the above conclusions, it seems clear that inclusion of the atmosphere in a holistic systems approach is one possible approach to reduce mission risk. As noted earlier, this is particularly true for mission modes such as aerocapture which strongly depend on knowledge of the atmospheric density field.

1.5 Summary of Research Questions

As discussed in Section 1.1, the primary research question under study may be stated as follows: Can an adaptive atmospheric prediction algorithm be created to improve density forecasting for aerocapture guidance processes? Such a density estimator, by generating more accurate density forecasts, would reduce the impact of atmospheric dispersions on aerocapture guidance processes. The answer to this system-level problem requires knowledge and understanding of many cooperating disciplines such as vehicle guidance methods, atmospheric physics, numerical weather forecasting techniques, and machine learning. In order to properly approach the primary question, several supporting questions with narrower, more discipline-centric focuses must be addressed. These supporting questions decompose the primary question into smaller, more readily conquerable components. The supporting questions to be examined herein are:

(1.) How can atmospheric state uncertainty be reduced?

(2.) Can an ensemble neural system be utilized to form atmospheric density forecasts consistent with observed flight data? An ensemble neural system is a collection of computational models each trained to perform an identical task. It combines two powerful forecasting concepts; ensemble forecasting and artificial neural networks (NNs). Ensemble weather forecasting techniques utilize a series of weather prediction models. While each model has the same goal (e.g. predict density as a function of spatial position) the underlying assumptions, state equations, and data trends may differ between each of the models in the ensemble. By combining the outputs of these models using an averaging or other arbitration technique, ensemble forecasts have the potential to accurately predict the behavior of highly uncertain systems. A form of limited Monte Carlo analysis, ensemble forecasts are commonly used to

predict the trajectories of hurricanes or other highly dynamic weather systems. Artificial NNs are computational models based on human neurology which are capable of recognizing and learning from patterns embedded in complex, noisy datasets. The combination of these two tools creates a unique solution for the prediction of atmospheric density. For more information, see Sections 2.4 & 2.5.

(3.) How can atmospheric prediction experience gained in flight be leveraged to improve future state predictions?

(4.) How can a proactive guidance scheme be developed which leverages information from the improved atmospheric forecasts?

The answers to these questions will be sequentially explored in the sections and chapters to follow.

CHAPTER 2

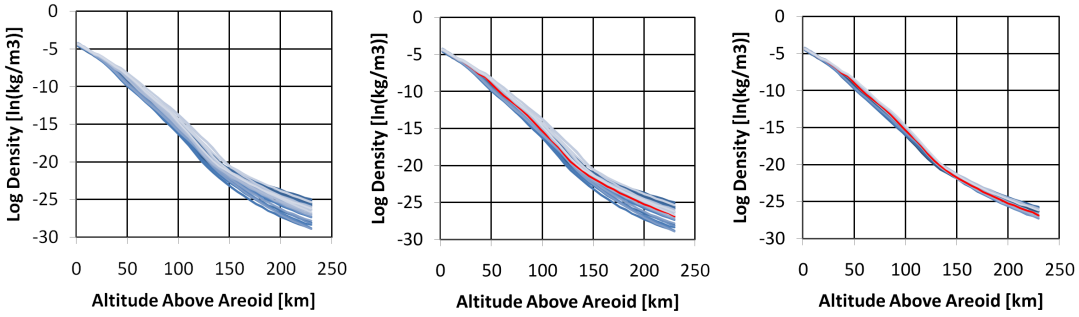
APPROACH & METHODOLOGY

2.1 Overview of Approach & Primary Research Question

The main challenge of the present study becomes the identification of a holistic approach to reduce the impact of atmospheric dispersions throughout the aerocapture entry process. Current methods of atmospheric density prediction for guided entries typically use one static atmosphere model as discussed earlier. Most contemporary static models have some measure of adaptive freedom; many are updated throughout the entry with an acceleration-based estimate of atmospheric density at the vehicle's current location. Therefore the term "static model" does not imply that the density estimator cannot adapt in flight. Rather, a static model implies that the underlying form of the model is fixed. For example, an exponential model can only capture vertical density trends and could never model horizontal density trends even if horizontal artifacts are directly observed during entry. Thus an exponential density model is always an exponential model regardless of the values of the input parameters or measured data used to adjust the model's behavior. Therefore even if a density measurement capability is available which provides density estimates at the current vehicle location, a static model may not be able to fully utilize this information due to the inflexibility of the underlying model dynamics. This inflexibility is a significant limitation when modeling a highly variable environment such as Mars. Therefore a method is sought which can change its underlying model dynamics to agree with observed conditions. Such a scheme is possible if the use of multiple models is considered. Employing multiple models allows information from several data sources to be combined to achieve the desired uncertainty reduction analogous to some hurricane path prediction methods. The approach used herein is to create a series of basis models which can be stored on-board the vehicle

(Figure 2.1.a). An Inertial Measurement Unit (IMU) and an Air Data System (ADS) are then used to sample the atmosphere from entry interface up to the current spatial location of the vehicle (the red line in Figure 2.1.b). Note that for the notional example shown in the figures, several outlying stored models do not accurately describe the observed density trends. These models can be eliminated from the prediction basis (Figure 2.1.c) and the remaining models can be combined in some manner to match the observed density history. This combinatorial model may then be used for forward-looking density forecasting.

If the observed atmospheric dynamics change, the problem may be reexamined and the basis model pruning process repeated to match the new conditions. By forming combinations of the most promising models and eliminating ill-fitting models, the underlying model dynamics may be continually adapted to suit currently observed trends.



(a) (b) (c)

Figure 2.1: Notional Example: Accomplishing Atmospheric Uncertainty Reduction

Vertical Density Profiles from Mars Climate Database at 0°N, 0°E

The problem now becomes finding a practical means of implementing the proposed scheme. Because the development of a series of accurate density models is central to the proposed approach, the identification of suitable Mars climate simulations from which to develop these models is critical. First, however, it is necessary to review several assumptions applied throughout this study.

2.2 Assumptions & Ground Rules

This study makes use of several general assumptions which apply throughout the remainder of the work. The prime assumptions stem from the availability of estimated or reconstructed density data. The entry vehicle is assumed to contain an IMU which is capable of tracking accelerations and angular rates as well as an ADS capable of inferring free stream dynamic pressure. This information is used to estimate the free stream density at the current spatial location of the entry vehicle. For the simulations herein, free stream density is estimated by combining the IMU information with measurements from an ADS. Entry vehicle ADSs are common and have been developed for the Space Shuttle,[29] for MSL[30] and for the X-15 research vehicle[31-33]. The approach used here is to compute free stream density ρ by combining dynamic pressure measurements ($p_{dyn} = \frac{1}{2}\rho V^2$) from the ADS with relative velocity measurements V from the INS. This approach makes no effort to determine or reduce the measurement uncertainty of either quantities because such procedures would require additional computational overhead.

Note that a viable procedure for computing free stream density with an associated uncertainty estimate is proposed by Dutta and Braun.[34] They also use two independent datasets from an IMU and ADS and produce estimates of atmospheric density which are independent of other uncertainty sources such as the vehicle aerodynamics. Their approach is notable because it allows separation of the various uncertainties and the production of more accurate free stream density estimates.[34,35] It should be stated that the quantification and reduction of other uncertainty sources (such as aerodynamic uncertainty) involves concepts from separate fields of study and is considered beyond the scope of the present work. However, likely velocity and dynamic pressure measurement errors are simulated in this study. These simulations reveal that the density estimator developed herein produces marked performance improvements in the presence of density measurement uncertainty.

Another common approach for estimating free stream density replaces data from an ADS with an on-board library of estimated aerodynamic coefficients as a function of the vehicle state and orientation. The density is then directly computed using

$$\rho_\infty = \frac{2a_{drag}m}{V^2 C_d S} \tag{2.1}$$

where a_{drag} is the measured drag acceleration, V is the relative velocity, C_d is the estimated drag coefficient, m is the vehicle mass, and S is the aerodynamic reference area. While this approach of using IMU data with an on-board aerodynamic database has been used extensively in real time for aerobraking operations[36-39], it is not adopted here. The atmospheric density estimator developed in this work requires a measurement of the local free stream density in order to make downrange forecasts. While the source of this estimate is assumed to be an ADS for convenience, the density estimator developed herein is fully capable of utilizing other methods as well. The tools and techniques for utilizing an aerodynamics model and accelerometer data to measure local free stream density are already well developed. Therefore, in order to place prime focus on the development of a novel density estimator, a suitable ADS is assumed to be available aboard the entry vehicle.

In addition to the above assumptions, the temporal variation of the atmosphere is assumed to be negligible over the several minutes of atmospheric flight. This permits the use of a single group of on-board density models rather than dynamically updating the models in flight or utilizing density models which are time dependent. Given the slow time scales over which atmospheric density typically changes, it seems reasonable to conclude that this assumption does not limit the applicability of the method.

2.3 Modeling of Martian Meteorology

In studying the accuracy of meteorological data necessary for successful aerocapture missions, Cerimele, Skalecki, and Gamble found that large density features (such as density pockets) which persist over several thousand vertical feet can generate

guidance failure modes.[19] Smaller features were not as significant because the vehicle was not in the feature long enough for it to significantly affect the trajectory. In addition, Ess noted that "significant density shifts are usually low frequency disturbances and [are] possibly predictable."[18]

Given these observations, it seems reasonable to utilize readily available data from Global Circulation Models (GCMs) to create the library of on-board basis models. These on-board models are each a static snapshot of spatially resolved density fields predicted by a suitable Mars GCM for one particular time, location, and set of physical inputs. Due to their global domains, GCMs produce results with coarse resolution to limit computational run time. Because large density structures are of prime interest and because aerocapture vehicles remain at high speed throughout the entry this coarse resolution does not limit the utility of GCM-derived basis models.

The use of higher resolution climate simulations such as regional mesoscale models (MMs) and large eddy simulations may be desirable in entry applications. However, Engelund et al. find that little validation data, such as density or temperature observations, is available for Mars MMs. The existing climate data sets are typically limited to a specific location, season, and time of day.[5] Conversely, GCMs are able to make use of these limited data sets for verification and validation due to their global domains. Despite the application of GCM data herein, the proposed approach is fully generalizable and could eventually utilize MMs for the development of the basis models when more accurate measured data sets are available for validation.

At present**,** the uncertainty of Martian atmospheric density ranges between 25-200% depending on altitude with an uncertainty of 20-40% near aerocapture periapsis altitudes.[5] This uncertainty is largely due to the highly variable dynamics of the Martian atmosphere which is driven by several principle factors[3,40], namely:

- Dramatic surface topography [the deepest point on Mars, Hellas Planitia is 7 km below the areoid (Mars equipotential datum) and the highest point on Mars, Mount Olympus, is 21 km above the areoid]
- Variations in surface solar heating & temperature-altitude profiles
- Bulk wind patterns, large flow structures, and interactions with topography

The last two items are largely governed by the Martian radiative balance. The atmospheric radiative balance is an energy conservation statement which describes how incoming solar energy is reflected from, dissipated by, or distributed in the atmosphere. The radiative balance, in turn, is affected by the amount of incoming solar energy (i.e. the solar flux incident on the Martian thermosphere) and the amount of suspended dust in the atmosphere. Higher levels of incident solar flux lead to more active solar forcing which affects global prevailing winds and temperature profiles. Suspended dust, lifted into the atmosphere by winds, can alter the radiative balance by absorbing and reradiating more energy to the surrounding fluid at a given altitude. Dust can also act as a shield or sunshade and decrease the amount of solar energy reaching lower altitudes. These mechanisms ultimately alter the temperature-altitude profile. Temperature variations directly impact density trends through the atmospheric equation of state. Therefore, the basis models developed for entry applications must account for both the dust distribution and the solar flux incident on the thermosphere. Because daily and seasonal property variations can be quite significant on Mars, season and entry time-of-day must also be considered. These four parameters, namely dust, solar flux, time, and topography are the most basic inputs to any candidate atmosphere model. Note that local topography and time are automatically accounted for when developing region-specific basis models for the known entry time.

In order to develop the proposed density forecasting scheme, two Martian climate simulations are required. One is needed to create the on-board library of basis models and the other is required to serve as a surrogate for the Martian atmosphere which the air data

system samples during the entry. The data in Figure 1.3 is taken from the Mars Climate Database (MCD), "a database of meteorological fields derived from GCM numerical simulations…validated using available observational data."[41] The GCM simulation used in MCD is derived from first principles and avoids inferences of global behavior based on sparse datasets, a limitation of many other available Mars climate models[42]. The MCD also includes several climate scenarios corresponding to various dust and solar flux conditions designed to bracket the global conditions on Mars[43]. As discussed, these two variables are the primary influences which drive the bulk behavior of the Martian atmosphere and therefore capturing their effects in any potential model is critical. For these reasons, the MCD is used to create the on-board model library. MarsGRAM 2005 is another GCM-derived Martian climate model based on a single year model integration.[44] It contains a flexible stochastic perturbation routine allowing the user significant control over the entry density profile. This makes MarsGRAM a useful choice for the surrogate atmosphere which functions as the true Martian atmosphere during the entry simulation.

2.4 Ensemble Weather Forecasting & Supporting Research Question 1

Creation of a methodology for adaptive atmosphere prediction using a series of basis models may be approached by examining contemporary weather forecasting techniques. Ensemble weather modeling is a form of Monte Carlo analysis where several nearly identical numerical models are constructed and integrated forward in time with small differences in initial conditions or model assumptions. Early use of weather ensembles is attributable to Leith who used ensemble averages to produce a Monte Carlo approximation to a stochastic-dynamic forecast (i.e. the forecast of a dynamical system subject to noise).[45] Commonly used to model tropical storm and hurricane systems on Earth[46,47], ensemble techniques are useful for weather prediction in the presence of uncertain parameters such as physical assumptions or initial state estimates[46,48]. Ensemble methods often combine several of the forecasts using one or more averaging techniques

to increase forecast accuracy. For example, Seidman applied ensemble averaging to a three-layer general circulation model and showed that more accurate forecasts may be made over longer periods as compared to the use of a single forecast.[49] In addition, several studies showed that predictions based on ensemble means typically produce better forecasts than individual deterministic forecasts at large spatial scales[50,51] such as those utilized herein. Vukicevic, et al. examined the forecasting skill of numerical weather ensembles based on both simple ensemble means and performance-weighted ensemble means.[48] They concluded that weighting ensemble members by a measure of performance reduces the mean absolute error of the ensemble-derived forecasts. This suggests that performance-weighted ensemble averages should be utilized to derive the density predictions.

Ensemble techniques have also been successfully applied to the Martian environment by Rogberg et al. who found that prediction efficacy over many Sols utilizing a perturbed GCM ensemble depends strongly on accurate knowledge of the atmospheric radiative balance.[52] This again highlights the necessity of capturing the driving influences which effect the Martian radiative balance such as the suspended dust distribution and the incident solar flux (see Section 2.1).

In the present study, the density ensemble (i.e. the on-board basis) is created utilizing subsets of eight climate scenarios from the MCD:

- Average (average solar flux & average dust)
- Maximum (maximum solar flux & average dust)
- Minimum (minimum solar flux & average dust)
- Cold (clear/minimal dust & solar minimum thermosphere)
- Warm (high dust but not a dust storm & solar maximum thermosphere)
- Dust Storm, Minimum Solar Flux
- Dust Storm, Average Solar Flux
- Dust Storm, Maximum Solar Flux

The first three scenarios consist of an average dust distribution derived from data observations from Martian Year 24, a year thought to be average. The Cold and Warm scenarios are designed to bracket the possible global conditions on Mars outside a global dust storm.[43] The last three scenarios are intended to model the range of behavior of the Martian atmosphere during a strong global dust storm at varying levels of incident solar flux. Taken together, these models provide a comprehensive prediction of the unperturbed atmospheric conditions on Mars. Various subsets of these eight climate models will be used to form experimental ensembles for testing purposes (see Section 5.3).

Figure 2.2 illustrates the differences between the various MCD atmosphere scenarios normalized against the average atmosphere. The figure is constructed using vertical MCD density measurements over the center of the Hellas Basin (Hellas Planitia will figure prominently in the remainder of this work, see Section 2.5). Note that the mean density trends of the eight scenarios become more dissimilar between 40–100 km. Most aerocapture trajectories reach periapsis in the 30 – 50 km range and, after periapsis passage, must transit this highly variable region between periapsis and 100 km. During this outbound leg of the trajectory, the vehicle is now traveling at significantly slower velocities then at entry interface. Therefore, due to its reduced momentum, the vehicle is now more vulnerable to density uncertainty in the outbound leg. By better predicting the mean density trends during the outbound leg, a proactive guidance scheme is able to reduce the impact of the high density uncertainty in this region on the trajectory.

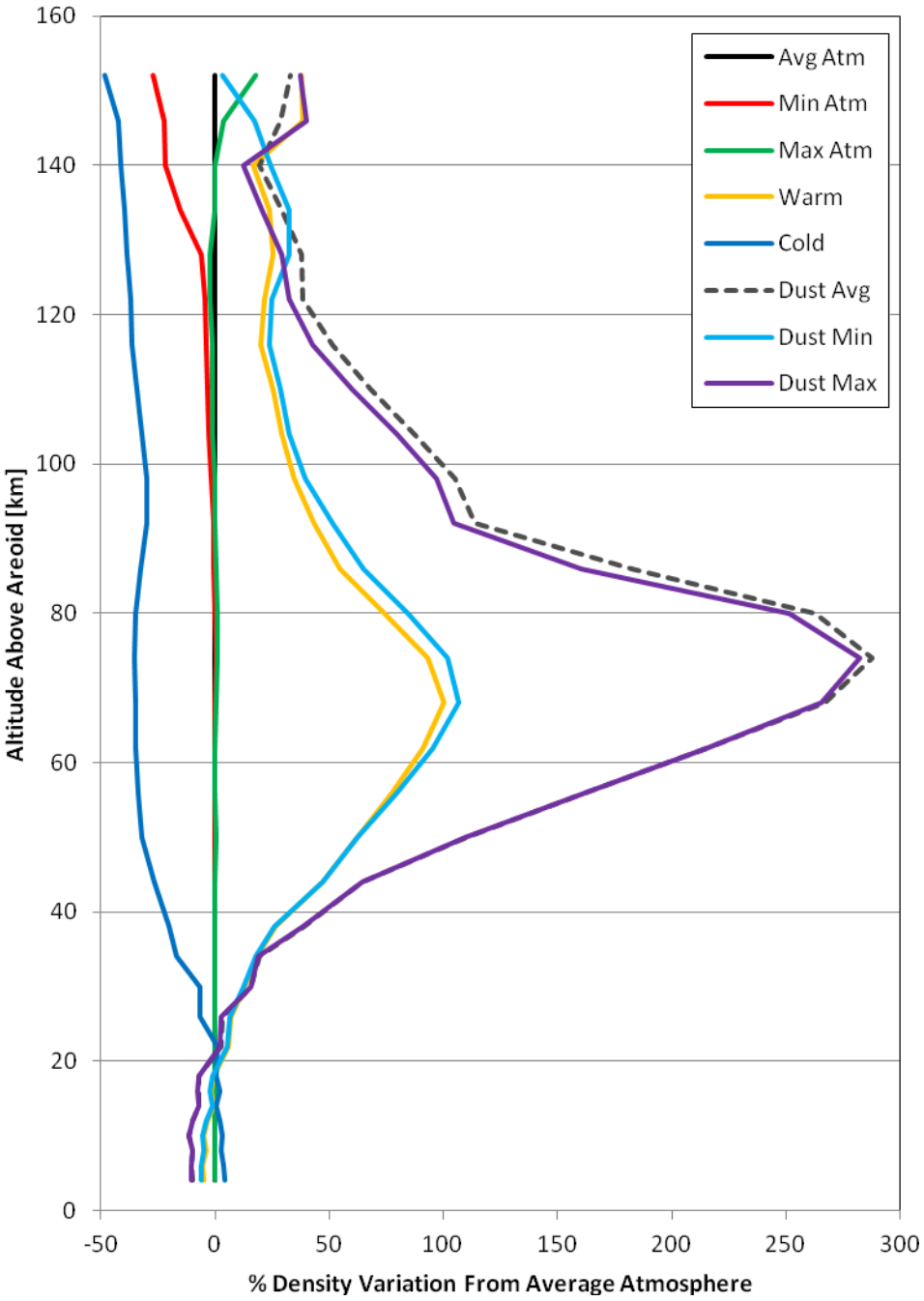


Figure 2.2: Ensemble Density Variations Relative to the Average Atmosphere

Note that each of the climate scenarios listed above accounts for two of the four basic inputs outlined earlier, namely dust and solar flux. Each of these scenarios in MCD provides atmospheric density as a function of latitude, longitude, and altitude over the globe for one Martian year. Specifying a specific region and time of entry limits the size of the data set to be modeled and also accounts for the remaining two basic atmosphere inputs. Even over constrained geographic and altitude regions, however, the amount of data available in the MCD is prohibitively large for direct on-board application. A storage method is required which permits a high level of data compression and allows natural interpolation between climate data points.

Several types of suitable models were considered for this role. It is desirable to utilize a model which offers an appropriate balance of data compression, accuracy, and pattern preservation/recognition capabilities. Tabular models offer a simple and readily implemented means of data storage. However, tabular models offer no natural means of data compression. Accurate pattern preservation is possible only if density data is gathered from closely spaced locations throughout the entry corridor. Therefore, thousands of density data points would need to be stored in order to form multiple spatially-resolved ensemble members. As the number of stored data points increases, so too does the time required to query the database to locate and interpolate between the stored density information.

Regression-based techniques such as response surfaces were also considered. These models can offer a high degree of data compression and are often represented by polynomials which may be rapidly computed. However, the assumed underlying equation is often a simple polynomial perhaps augmented with one or more transcendental functions. Early experimentation revealed that capturing the highly nonlinear patterns found in some regions of the atmosphere would likely require large numbers of overlapping models for each ensemble member. Therefore, linear regression techniques were not selected. More advanced regression techniques such as Gaussian process

regression (i.e. kriging) was also considered. Kriging models are often able to reproduce large datasets with high fidelity and offer accurate interpolation between data points. However, as Simpson et al note, prediction with a kriging model requires the inversion and multiplication of several matrices because a kriging model does not exist as a closed-form polynomial equation[53]. This would lead to larger computational demands during atmospheric flight which may be avoided by considering other methods.

Artificial NNs were eventually selected for use in the ensemble models because they are a "nearly universal approximator, and so are able to handle highly nonlinear or extremely large problems."[54] This makes NNs well suited to this application due to the size of and local nonlinearities present in the MCD datasets. Each NN training process results in a compact equation which closely mimics the training dataset, only with much fewer stored parameters. This high data compression ratio reduces density query time and on board storage requirements without significant loss of accuracy. Due to the continuous nature of the developed NN equation, interpolation between data points is automatically possible.

A series of neural networks are developed to summarize each of the eight climate scenarios in the MCD in order to form the on-board basis models. When working in concert with one another, the NNs form an ensemble neural system. For additional background on NNs, see Appendix A.

While NNs were chosen for the reasons highlighted above, it should be noted that the density estimation technique developed in the following sections can be readily generalized to use other modeling methods. NNs were chosen because they represent the best balance between data compression and modeling fidelity. However, future developments may present better alternatives which can be easily incorporated into the density estimator outlined below.

2.5 Ensemble Neural Systems

A neural ensemble or ensemble neural system is a NN learning paradigm where a collection of NNs is trained to perform the same task.[52,55-57] In this case, each NN is trained to represent one of the eight MCD climate scenarios and thus provides an atmospheric density estimate as a function of latitude, longitude, and altitude at the time of entry. The differences between the density estimates from the ensemble members correspond to the differences due to variations in the atmospheric physical assumptions (dust and solar flux). Commonly used to mimic an ensemble weather forecast, ensemble neural systems provide the same prediction quality but at a significantly reduced computational cost.[55,58-61] This reduced computational overhead not only reduces the on-board processing requirements of the entry vehicle but also permits faster computations which is critical for enabling time-sensitive decision making.

In a neural ensemble, the outputs of the individual NNs are typically combined in some manner. This combination could be as simple as the numerical mean of the ensemble members or an adaptive computation based on observed conditions. Ensemble combination often results in more accurate predictions with an improved generalization capability.[56,62,63] Therefore, combining the ensemble density estimates can lead to improved predictions not only in the vicinity of the entry vehicle, but also far ahead of the vehicle in entirely different regions of the atmosphere.

Figure 2.3 depicts the ensemble neural system architecture proposed here. It consists of the eight climate scenarios from the MCD connected by a top-level gating network. This network functions as a signal mixer and determines how to combine the signals from the individual NNs based on the information collected by the ADS. The ADS data consists of free stream density estimates along the flight path from entry interface up to the current position of the vehicle. The gating node must determine how to combine the ensemble signals to minimize the retroactive prediction error (i.e. the

prediction error from entry interface up to the current time). Algorithmic details concerning the function of the gating network are given in subsequent sections.

As discussed earlier, selection of the time and location of the entry is critical to both completely characterize the atmospheric state and to limit the amount of data to be modeled. In the present study, a notional aerocapture mission is used as the test bed for the proposed density forecasting method. This notional mission arrives at Mars on the second day of the eleventh Martian month which is near the end of the Martian dust storm season (see Table 2.1). Hellas Planitia (white box in Figure 2.4), the largest visible impact crater in the solar system, was chosen for the location of the test mission entry. Hellas Planitia is the deepest point on Mars, a fact which strongly contributes to the high levels of meteorological activity in the region. At least one global dust storm (one of the most extreme weather phenomena on Mars) was observed to nucleate from the basin in 2001.[64] In addition, topographically forced density perturbations such as gravity waves may propagate to high altitude.[7] Such perturbations when superimposed on larger scale features such as planetary-scale waves and tides create a dynamic testing environment.

As can be seen in Figure 2.3, more than one NN is associated with each climate scenario. This was necessary to maintain a high degree of fitting accuracy over the large spatial region selected for study. The nominal entry trajectory (black line in Figure 2.4) coincides with the 67.5° east longitude great circle which takes the vehicle over the middle of the basin. This notional polar orbit is a plausible aerocapture target as it offers global coverage for scientific or exploration/reconnaissance payloads. In order to allow sufficient maneuvering volume for future simulations employing an entry guidance algorithm (see Chapters 4 and 5), a large region was selected for modeling. This region extends from the South Pole to 45° north latitude and extends laterally for 11.3 degrees of longitude on either side of the nominal entry trajectory. The region is subdivided into four boxes (Figure 2.4) which overlap at their spatial boundaries in order to provide continuous coverage as the entry vehicle transits between boxes. One NN model is

created to approximate the predicted density field for each MCD climate scenario in each of the four boxes. Because there are eight climate scenarios and four boxes, a total of thirty two NN models are required to completely describe the density field in the target region.

The four boxes correspond to two different divisions of latitude and altitude. The lower altitude boxes extend from 4 km above the areoid up to 104 km or roughly up to the base of the Martian thermosphere. The minimum altitude of 4 km was selected in order to ensure local terrain clearance for the low-altitude boxes. The high-altitude boxes extend from 104 km up to 152 km. This ceiling was selected based on the walk-in phase of the Mars Global Surveyor aerobraking operations which determined that the atmosphere was "barely measurable" at 149 km.[36] To ensure that the majority of the sensible atmosphere is modeled, an additional 3 km margin was added to account for daily and seasonal scale height variability.

The two divisions in latitude were selected for modeling convenience. The original intent was to model the density field from the North Pole to the South Pole. However, the mission arrives during southern summer/northern winter. Due to the planet's axial tilt, the majority of the incoming solar energy is concentrated in the southern latitudes. This translates to large temperature gradients (and therefore large density gradients) between the equatorial regions and the northern polar region. These large gradients complicate modeling efforts because the atmospheric behavior in the northern polar region is markedly different from the atmospheric dynamics near the equator. In order to bypass this issue, the entire southern region is modeled (0° to 90° S latitude) but only half of the northern region is modeled (0° to 45° N latitude). This still provides ample overshoot and maneuvering volume for future guided simulations.

Thirty two single hidden layer perceptron NNs with logistic activation functions were created from the eight MCD climate scenarios. These NNs were trained using the JMP statistical software package with a third of the training data randomly withheld. This

withheld set was later used for cross-validation purposes to ensure a good NN model fit and prevent overtraining. See Appendix A for NN fitting statistics and equations. By partitioning the target spatial region into four smaller boxes, a high degree of fitting accuracy was maintained. All of the NNs have fitting error distributions which are approximately Gaussian and centered near zero. Of the 87,960 data points used for NN training, approximately 197 data points (or approximately 0.2% of the training data) exhibited an error greater than 10% when tested in the appropriate NN equation. These higher errors are typically confined near the edges of the four flight boxes. Note that the flight path of the baseline trajectory (see Section 2.7) has only minimal contact with the boundary regions of the four flight boxes. This offers additional justification for selecting a larger-than-necessary target region and for the divisions chosen for the four flight boxes. Modeling of a large target region moves the majority of the spatial boundaries far from the entry trajectory. Even guided simulations should not come near the edges of the target region provided that maneuvers resulting in extreme crossrange errors are prohibited. The nominal trajectory does come into contact with the internal boundaries of the four flight boxes but only for short downrange distances which limits the impact of the higher modeling errors. Also consider that the current uncertainty in the Martian atmosphere ranges between 25-200% depending on altitude.[5] This suggests that despite the modeling error, a reduction in uncertainty is still possible. Lastly, recall that the combination of NNs in an ensemble typically increases prediction accuracy and generalization ability relative to the direct use of a single deterministic forecast. Therefore the uncertainty associated with the composite ensemble prediction is often further reduced below the individual uncertainties associated with each ensemble member.

Table 2.1. Baseline Test Mission Temporal Characteristics

Earth (Mars) Year	2020 (35)
Martian Month	11
Day of Month	2
Solar Longitude, L_s (deg)	307.3
Sol Number	574
UTC of Entry	12:00:00
Julian Date	2459156

N.B. Martian month 11 occurs at the end of the annual dust storm season.

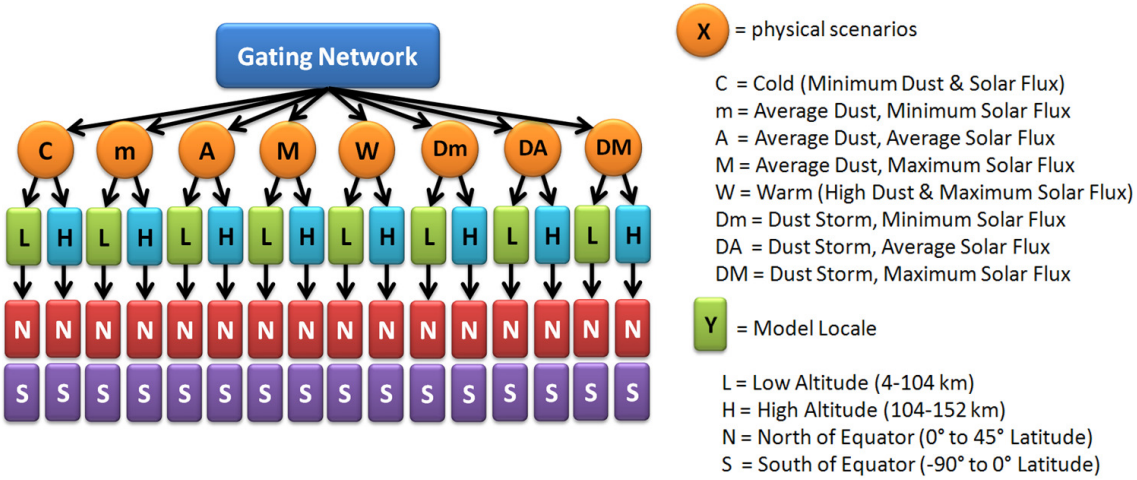


Figure 2.3: Ensemble Neural System Topology

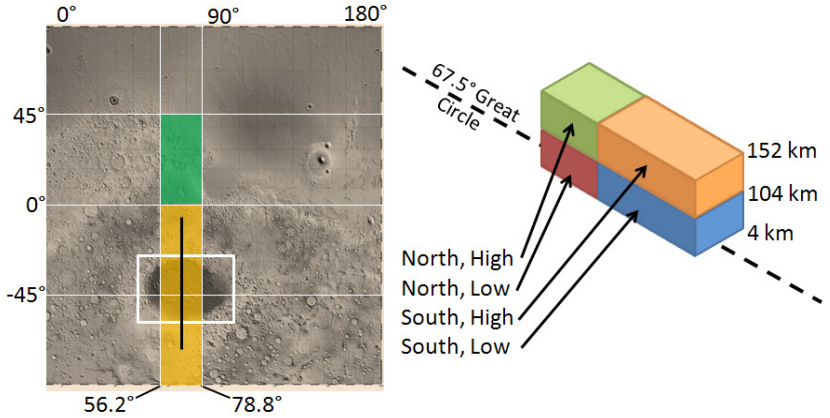


Figure 2.4: Martian Eastern Hemisphere with Entry Flight Box Overlay

2.6 Ensemble Linear Combination & Supporting Research Question 2

With the neural ensemble completely described, the current issue of interest is to determine how to best utilize the ensemble to form atmospheric forecasts consistent with the measured flight data from the ADS. The problem posed here is strikingly similar to a data assimilation problem in numerical weather modeling. Data assimilation is a process of combining measured weather data with data from a numerical weather prediction model while maintaining internal consistency in the combined model.[52,65-67] It is believed that the solution of a data assimilation problem using a Mars GCM and measured ADS data would generate a suitable prediction model. However, data assimilation techniques are computationally complex due to the need to iteratively correct and rerun numerical weather simulations. It is further hypothesized that the data assimilation problem may be readily solved if it is converted to a parameter estimation problem, as described later. This can be accomplished by constructing a linear combination of the ensemble members where the weights of the linear combination are adaptively tuned using a parameter estimation scheme. This allows the prediction model to be continually modified to better agree with the observed weather data from the ADS while obviating the need to rerun numerical weather models in the computationally constrained environment aboard the entry vehicle.

Parameter estimation problems, commonly used throughout engineering and the sciences, apply statistical and/or probabilistic measures to determine a quantity of interest.[68] Parameter estimation schemes have been used to identify aerodynamic coefficients or stability derivatives from measured flight data in real time.[68] They have also been used extensively by Johnson and Oh[69] and by Chowdhary and Johnson[70,71] to train a NN-based adaptive flight controller in real time using both background and online information.

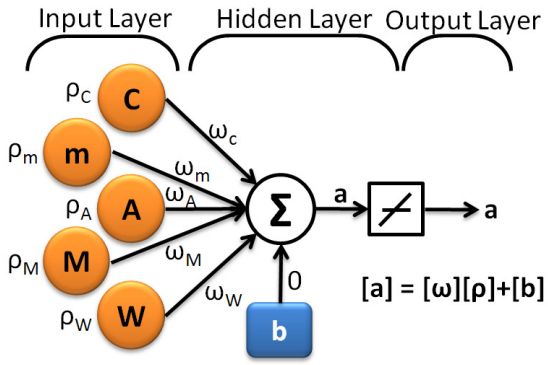


Figure 2.5: Example Adaptive Linear Neuron (ADALINE) Gating Network for Five Ensemble Member Inputs

As discussed previously, a gating network resides at the top of the neural ensemble and functions as a signal mixer. It determines how to best combine the members of the ensemble in order to maximize the retroactive prediction accuracy (i.e. minimize the error of the composite ensemble prediction from entry interface up to the current time). Because it is desired to form a linear combination of ensemble members, this gating network consists of a single adaptive linear neuron (or ADALINE). The ADALINE concept,[72,73] originally developed by Widrow and Hoff, accepts several input channels, weights each channel, and then determines the summation of the weighted inputs (Figure 2.5). Therefore, each ADALINE forms a linear combination of the inputs using some weighting strategy. The original work of Widrow typically involved the use of multiple ADALINE nodes in each network (a so called Multiple ADALINE or MADALINE network). In this application, however, if a parameter estimation scheme is available to adaptively compute the weights of the linear combination, only a single ADALINE is required for the gating network.

The ADALINE forms a linear combination from the eight ensemble members of the form

$$\begin{aligned} \rho_{pred} &= [\rho][\omega]+[b] \\ &= \rho_C\omega_C + \rho_m\omega_m + \rho_A\omega_A + \rho_M\omega_M + \rho_W\omega_W + \ldots \\ &\quad + \rho_{Dm}\omega_{Dm} + \rho_{DA}\omega_{DA} + \rho_{DM}\omega_{DM} \end{aligned} \tag{2.2}$$

where ρ_{pred} is the predicted density field, ρ_i is the density field associated with the i^{th} ensemble member and ω_i is the weight associated with the i^{th} ensemble member. Each density field in the equation is a function of altitude, latitude, and longitude within the limits of the target region discussed in Section 2.5. Though thirty two NN equations exist, only eight are active corresponding to whichever of the four flight boxes the vehicle is currently occupying. When the vehicle transits from one box to another, the NN equations for each of the ensemble density fields ρ_i change to reflect the eight newly active NNs. The weights ω_i are adjusted independently by the parameter estimation scheme and are not affected by flight box transitions.

Notice that the bias vector ***[b]*** in Figure 2.5 is currently not used in the present implementation and is set to the null vector. Biasing an ensemble member based on currently observed density data is problematic as the observed trends may change drastically over short time spans. This leads to an erratically shifting bias term which complicates the development of a consistent density prediction. The box in Figure 2.5 at the beginning of the output layer indicates that the output of the neuron (and therefore of the gating network) is a linear function of the input. In this case, the activation function is a 1:1 mapping such that the result of the linear combination becomes the output of the neuron. This is in contrast to many biologic and computational neurons which have nonlinear activation functions such as the logistic activation function employed in the thirty two NNs forming the ensemble members.

2.7 Baseline Aerocapture Reference Trajectory & Guidance

2.7.1 Entry Interface State

A single reference trajectory was defined using the Program to Optimize Simulated Trajectories (POST2)[74] for use in all early experimentation presented in Chapter 3. MarsGRAM 2005, the simulation chosen as the surrogate for the Martian

atmosphere sampled by the ADS, is called by the POST2 algorithm to supply atmospheric density estimates for a given position within the atmosphere. For all early experimentation in Chapter 3, a default atmospheric profile was utilized. This default profile is later replaced by a more appropriate MarsGRAM atmosphere with input values matched to the MCD inputs (see Section 3.3.2). However, this default MarsGRAM profile has little impact on testing because the nominal trajectory is only used with ideal density testing signals as will be explained in Section 3.3.1.

The entry interface conditions and vehicle characteristics are adapted from a study by Masciarelli, et al[6] on an analytic aerocapture guidance algorithm for a Mars sample return orbiter. Entry interface occurs at 128 km at a relative velocity of 5.9 km/s. The entry flight path angle is utilized by the POST2 targeting algorithm to control the apoapsis to 1000 km and was determined to be -9.67° from the local horizontal. It should be noted that Masciarelli uses a 1400 km apoapsis while a 1000 km apoapsis is used here for initial testing. To reach a lower apoapsis, the vehicle must expend more energy in the atmosphere. This generally requires that the vehicle dwell in the atmosphere for longer time periods over longer downrange distances. This, in turn, increases the amount of data returned from a single trajectory which is useful for algorithm development and data collection. Later, in Chapter 4, the apoapsis will revert back to 1400 km to match the original orbit proposed by Masciarelli. The vehicle begins at 70° S, 67.5° E at entry interface and moves directly north towards the equator (see Figure 2.4).

2.7.2 Reference Trajectory Atmospheric Guidance

The vehicle flies a lifting entry with a fixed L/D of 0.247 and a ballistic coefficient of 147.7 kg/m^2 corresponding to the values used by Masciarelli, et al. Because no longitudinal guidance is utilized in this phase of the study, the targeting algorithm selects a constant bank angle magnitude (in this case, 175.2° from the vertical or 4.8° from the nadir) which remains fixed throughout the entry. In subsequent chapters,

simulations employ longitudinal entry guidance where the lift vector is used to actively control the vehicle's trajectory (see Chapters 4 and 5). Bank angle modulated guidance solutions control the lift vector direction by changing the bank angle of the vehicle in flight. Note that some aerocapture trajectories fly with the lift vector predominantly pointing downward (i.e. nadir or towards the planet). This is useful to balance forces acting on the vehicle in the vertical direction (Figure 2.6). From the vehicle reference frame, the centrifugal acceleration vector is quite large and the nadir pointing lift vector is used along with the gravity vector to balance the centrifugal acceleration and prevent premature exit from the atmosphere. Remaining in the atmosphere is critical because atmospheric drag reduces the vehicle's velocity and allows it to attain a stable orbit. Bank-modulated guidance for longitudinal control selects the appropriate nadir lift component (labeled N in Figure 2.6) to control the vertical acceleration of the vehicle. This allows the vehicle to control its altitude and altitude rate to maintain some desired criteria such as equilibrium glide (i.e. all vertical forces sum to zero) or a desired drag acceleration profile.

Lateral guidance logic becomes necessary because the nominal lift vector, by design, is larger than necessary to counterbalance the centrifugal acceleration. This provides control margin to allow the vehicle to change its vertical acceleration and therefore its altitude and altitude rate. If the nominal lift vector were just sufficient to counterbalance the anticipated centrifugal acceleration, the vehicle would be forced to maintain full lift down (i.e. bank angle $\beta \equiv 0°$) and would be unable to control its altitude or altitude rate. The "oversized" lift vector requires a non-zero bank angle β in order to achieve the desired nadir lift component N (see right side of Figure 2.6). This nonzero bank angle also generates a cross-track component of the lift force T resulting in lateral movement.

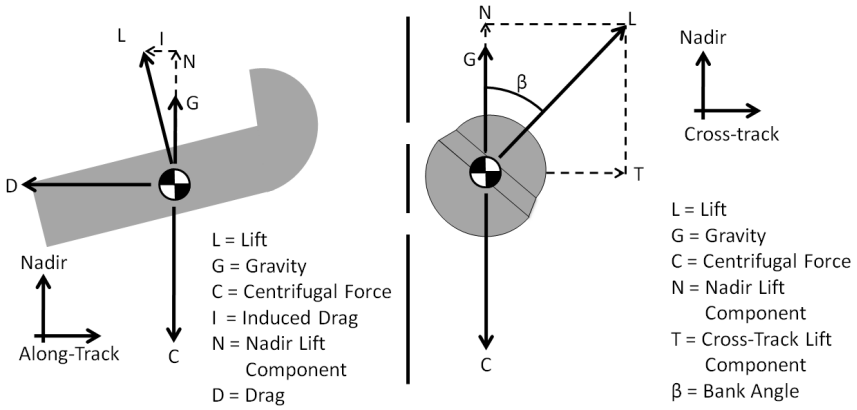


Figure 2.6: Aerocapture Forces in the Vertical (Left) and Cross-Track (Right) Planes

The lateral guidance logic prevents excessive crossrange position errors from developing due to this cross-track lift component by commanding bank angle reversals. These reversals typically maintain the same bank angle magnitude β but simply reverse the sign of the bank. This reverses the cross-track lift component direction while maintaining the same nadir lift component. Figure 2.7 is taken from the Cerimele-Gamble study[10] and demonstrates how bank angle reversal logic can be used to target a desired exit inclination (which in the figure is 28.5°). The straight line segments represent the lateral position of the vehicle relative to the entry axis which is the dashed line in the figure. When the vehicle reaches some pre-computed inclination or crossrange error, the sign of the bank angle is reversed. This lateral corridor or deadband typically contracts towards the end of the trajectory to limit crossrange error at atmospheric exit.

For the baseline trajectory, a simple lateral guidance scheme is utilized in order to approximate a fully guided trajectory. This is accomplished with a deadband bank angle reversal strategy based on the equation used in the Cerimele-Gamble aerocapture guidance.[10] This equation includes several constants which can be adjusted to suit the particular mission or trajectory requirements. The deadband provides bounds on the allowable inclination of the vehicle based on current vehicle velocity;

$$i_{db} = \pm\left[\frac{V - V_{shift}}{V_{scale}}\right]^4 + i_{ref} \pm i_{tol} \tag{2.3}$$

where i_{db} is the inclination defining the deadband boundary and V is the current relative velocity of the vehicle. The velocity shift V_{shift} controls the rate of contraction of the deadband corridor with higher values producing a tighter contraction and more accurate exit inclination targeting (likely at the expense of more required bank angle reversals). The scale velocity V_{scale} controls the scale of the deadband curves; higher values produce curves which are closer and closer to the target exit inclination. The reference inclination i_{ref} is the target exit inclination and i_{tol} is the inclination tolerance near the exit state. Because the vehicle flies through the entire trajectory with the lift vector oriented nearly in the vertical plane, the cross-track error accrues at a slow rate. Consequently, the baseline simulation only includes a single bank angle reversal and the accrued crossrange error is less than 200 km throughout the entire trajectory.

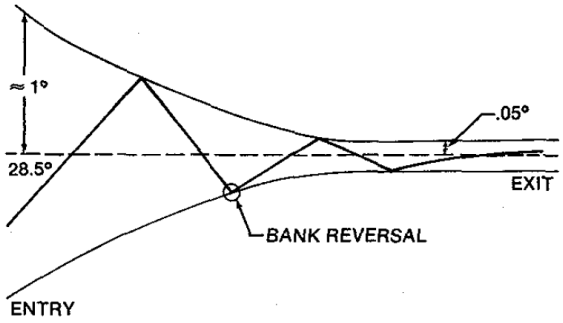


Figure 2.7: Bank Angle Reversal Strategy Used for Lateral Guidance[10]
(as viewed looking down on the vehicle from space)

2.7.3 Atmospheric Exit State & Reference Trajectory Summary

Atmospheric drag reduces the relative velocity of the vehicle to 3.66 km/s by the time the vehicle has climbed back to 128 km at 7.8° south latitude. The exiting flight path angle is 3.87° above the local horizontal in order to reach the desired apoapsis at 1000 km. The peak sensed acceleration of 1.16 Earth G occurs just prior to periapsis passage at

46 km about 200 seconds after entry interface. As can be seen in Figure 2.8, the trajectory ends after roughly 14.5 minutes elapsed time from entry interface.

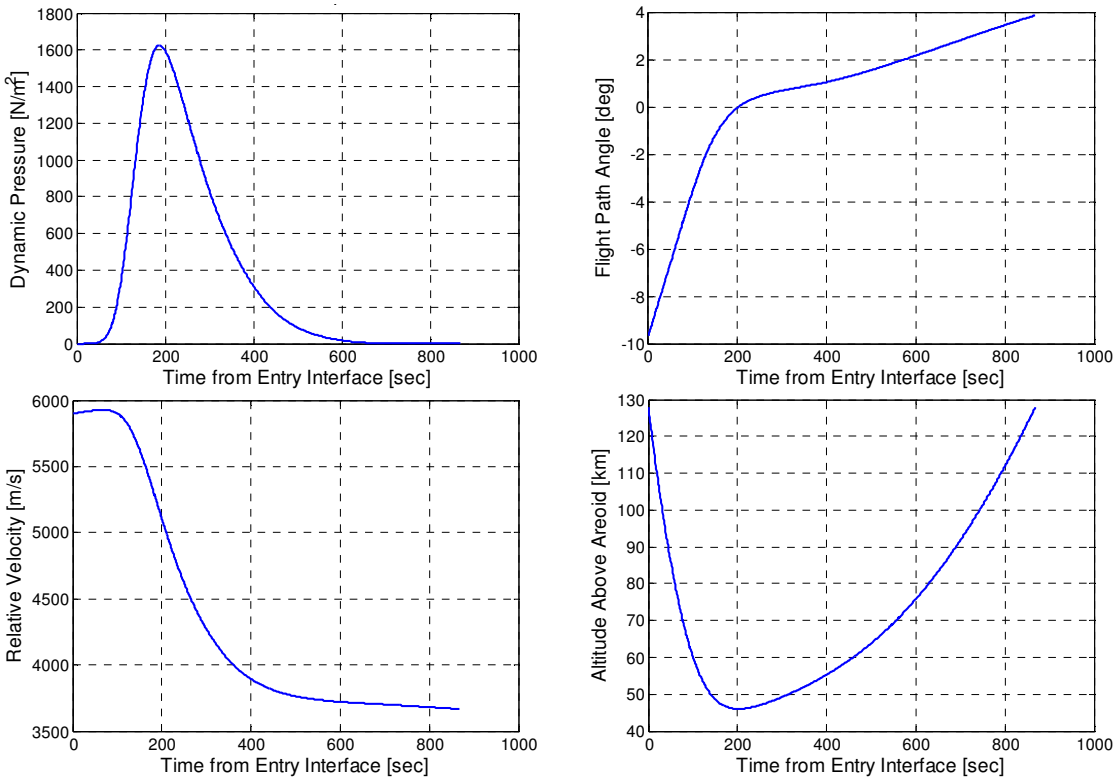


Figure 2.8: Baseline Aerocapture Reference Trajectory

CHAPTER 3

IMPROVING ATMOSPHERIC STATE PREDICTION EFFICACY

3.1 Testing Methodology

Testing of the prediction skill of the neural ensemble described in Chapter 2 is approached in several stages described by the author in reference 75 and summarized here. The first stage, presented in Section 3.2, examines the parameter estimation problem and several schemes used to adaptively solve this problem. Because the goal of this section is the identification of promising parameter estimation schemes, thirty six simple vertical atmosphere profiles were used to rapidly evaluate the prediction efficacy of each scheme. This approach also permitted evaluation of the parameter estimation schemes with a significantly larger ensemble to ensure the scheme is generally applicable. Following development and testing, the selected parameter estimation scheme was integrated into the neural ensemble and tested using an ideal testing signal (Section 3.3). This exposed several areas of improvement which are addressed with the application and testing of an ensemble echo, a new construct for ensemble weather prediction in computationally constrained environments (see Sections 3.3.2-6). The resulting construct is then tested in a more realistic testing environment using a MarsGRAM measured atmosphere signal in Section 3.3.7.

3.2 Adaptive Ensemble Weight Tuning via Parameter Estimation

The adopted linear ensemble combination (see Section 2.6) permits the adaptation of the prediction model to be accomplished with a parameter estimation scheme rather than a computationally complex data assimilation problem. Several parameter estimation strategies were examined to determine their suitability for application in aerocapture

guidance models. Several metrics are of importance for a viable parameter estimation scheme:

- *Algorithmic Simplicity*: The on-board guidance computational process must be computed multiple times by one or more processors to ensure fault tolerance. Because space-hardened avionics typically have slower processing speeds than their ground-based counterparts, computationally complex methods can increase time between guidance updates and increase the probability of an algorithmic fault. Methods which limit the frequency of complex operations are sought.
- *Prediction Accuracy*: Because the goal of the present study is to increase density prediction accuracy, the inclusion of this metric is clearly critical. The accuracy of the predictions must be sufficient to achieve an uncertainty reduction relative to the current knowledge state of the Martian atmosphere.
- *Prediction Robustness*: The algorithms considered here should consistently produce accurate density forecasts. Algorithms which only occasionally produce high accuracy forecasts or algorithms which sporadically produce forecasts with exceedingly poor accuracy are ill-suited to this application.
- *Stability*: The predicted density signal must clearly remain feasible throughout the entire prediction horizon, both near the vehicle and far ahead of the vehicle. Infeasible signals diverge to impossibly high or low densities for a given altitude. Oscillatory predicted density signals are not viewed as immediately problematic because real density features in the Martian atmosphere can be periodic in nature. Because this is the case, stability will be approached from a bounding argument. Algorithms are sought which produce bounded output signals given a bounded input signal (commonly referred to as Bounded Input, Bounded Output or BIBO stable).
- *Optimality*: Algorithms which include some notion of optimizing the ensemble combination will be given preference. The optimization must, of course, conform

to the other stated criteria, especially the goal of achieving algorithmic simplicity.

Four different parameter estimation strategies were developed for evaluation. These schemes are either adopted directly from the contemporary literature or are developed by combining one or more simpler techniques in order to fulfill the five stated algorithmic goals. Before outlining the estimation schemes, it is necessary to define the error signal for the ith ensemble member as

$$e_i(t) = \left(\frac{\rho_i(t) - \rho_{ADS}(t)}{\rho_{ADS}(t)} \right) 100\% \tag{3.1}$$

where ρ_{ADS} is the free stream density estimated by the air data system and ρ_i is the density from the ith ensemble member. Both are measured at the current location of the vehicle and both are functions of time as the vehicle proceeds through the atmosphere. The resulting error signal tracks how closely each ensemble member agrees with the observed density trend measured by the ADS. It is useful to develop an average error metric which is defined here as the root mean squared error (RMSE). The error signal is sampled at a fixed interval and stored in an error vector which tracks the error signal for each ensemble member over time. For the ith error vector $[e_i]$, the corresponding $RMSE_i$ may be expressed as

$$RMSE_i = \left[\frac{1}{m} \left([e_i]^T [e_i] \right) \right]^{1/2} \tag{3.2}$$

where m is the total number of elements in the error vector or, equivalently, the number of time steps from entry interface up to the current time.

In order to evaluate their performance-weighted strategy for producing an ensemble mean, Vukicevic et al. utilized a simple vertical atmosphere as a test case.[48] Following this approach, each of the parameter estimation schemes are tested using thirty six vertical atmosphere profiles (i.e. density is a function of altitude only) derived from MCD data taken at one location on Mars. These thirty six profiles, temporarily used in

lieu of the neural ensemble developed in Section 2.5, correspond to the vertical density column over the point of intersection of the equator with the prime meridian (0° latitude, 0° longitude). The density column is sampled once per Martian month over one full Martian year for three of the MCD climate scenarios, the Average Dust/Average Flux, Warm, and Cold scenarios. The twenty four vertical profiles from the Warm and Cold scenarios serve as a simplified on-board ensemble and the twelve remaining Average models each serve as a set of ADS measured data. To simulate the dynamics of an entry scenario, only the high-altitude region of these Average MCD profiles down to 86 km is used as measured data. Each of the parameter estimation schemes is used to construct twelve density predictions from 86 km to the surface corresponding to the twelve Average vertical profiles. These combinatorial predictions are weighted averages of the twenty four basis models formed using the method presented in Section 2.6. Utilizing this type of test for the parameter estimation schemes was intended to be stressful in two ways.

First, the twenty four member vertical ensemble, while larger than the eight member neural ensemble designed for integrated testing, is spread over the entire Martian year. Given the significant temporal variability of the Martian atmosphere, this requires each parameter estimation scheme to select the models whose combined density trends most closely agree with the measured data. Due to the high temporal variability and the large number of ensemble members, this forces the parameter estimation scheme to emphasize only a few models while de-emphasizing the others. This selectivity is necessary to achieve the algorithmic goals of prediction accuracy and robustness.

Second, only two basis scenarios, the Warm and Cold scenarios, are used which restricts the variability of the physical basis used for prediction. This requires accurate density prediction by combining models from only two sets of governing physical inputs (dust distribution and solar flux). This is beneficial because it allows testing of the

implicit assumption that linear combinations of Martian atmosphere models can produce valid prediction models using a limited physical basis.

In summary, the vertical ensemble used for testing was designed to have a high degree of temporal variability with only a limited set of physical inputs. This is in contrast to the more desirable neural ensemble which eliminates temporal variability (because the day and time of the entry will be known) and incorporates models based on a more varied set of physical inputs.

3.2.1 Winner-Take-All Scheme

The first and simplest algorithm is a winner-take-all strategy taken directly from the work of Maqsood et al.[57] This algorithm selects the single ensemble member whose average performance is closest to the measured density signal. This is accomplished by simply selecting the ensemble member with the smallest RMSE. The prediction results for the twelve Average MCD months, shown in Figure 3.1, are promising given the simplicity of the method.

The winner-take-all method is guaranteed to be BIBO stable as long as all the ensemble members are themselves stable (i.e. physically reasonable). Note that several of the predictions, notably months four and six, perform extremely well. In general, however, this performance is not consistent as can be seen by comparing months four and six to months with markedly poor performance such as months seven or ten. Despite its simplicity and occasional accuracy, the winner-take-all method does not exhibit good prediction robustness and does not consider the optimization of the ensemble combination because no combination exists.

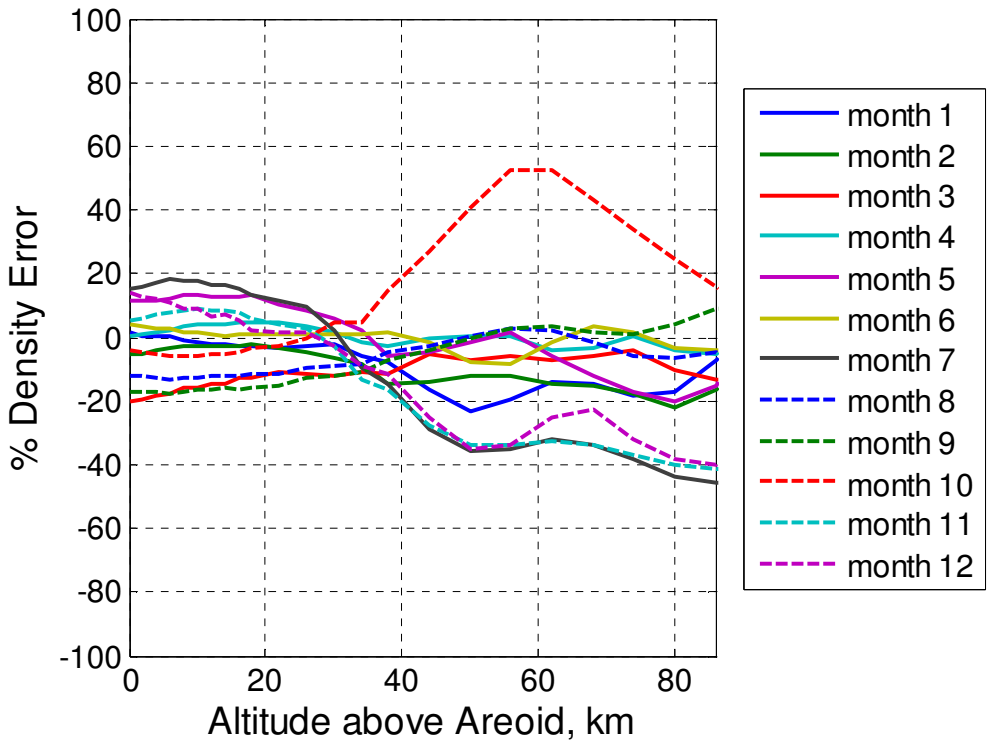


Figure 3.1: Low-altitude Prediction Using the Winner-Take-All Approach

3.2.2 Linear Least Squares Scheme

The second method is a direct implementation of the common linear least squares algorithm. This algorithm serves as the basis for many advanced estimation schemes such as linear sequential estimation, Gaussian least squares differential correction, and Kalman filters.[68] Therefore, inclusion of this method is crucial in order to determine if any of these methods will be directly applicable to the parameter estimation problem as it is currently formulated. The linear least squares algorithm computes the weights of the prediction model by minimizing the errors between the prediction model and the measured data. This accomplishes one of the desired algorithmic goals by incorporating some form of optimality into the prediction scheme. The derivation of and the theoretical justification behind linear least squares is well known and an excellent introduction can be found in Crassidis and Junkins.[68] The measured data is collected at regular time intervals and is stored in an observations vector $\tilde{y}$ where

$$\tilde{y} = [\tilde{y}_1 \quad \tilde{y}_2 \quad \cdots \quad \tilde{y}_m]^T \tag{3.3}$$

and $\tilde{y}_j$ are the measured density points collected over time by the ADS at the measuring frequency. A density estimate from the twenty four vertical ensemble members is requested at the same altitude as each of the measured points (Note that for a spatially resolved ensemble, the current latitude and longitude must also be known to compute a density estimate from the ensemble members). These density estimates form the prediction basis and are placed in the basis matrix H;

$$H = \begin{bmatrix} \rho_1(h_1) & \rho_2(h_1) & \cdots & \rho_n(h_1) \\ \rho_1(h_2) & \rho_2(h_2) & \cdots & \rho_n(h_2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_1(h_m) & \rho_2(h_1) & \cdots & \rho_n(h_m) \end{bmatrix} \tag{3.4}$$

where $\rho_i(h_j)$ is the density member from the i[th] ensemble member at the j[th] altitude level (i = 1,2...n and j = 1,2…m). The estimated parameters $\hat{x}$ of the ensemble linear combination are then directly determined with

$$\hat{x} = \left(H^T H\right)^{-1} H^T \tilde{y} \tag{3.5}$$

which is used to construct the linear approximation model $\hat{y} = H\hat{x}$. This estimation scheme was applied to the vertical ensemble with unanticipated results (Figure 3.2). The prediction scheme was able to construct very accurate linear combinations but only in the regions of the atmosphere where measured data is available.

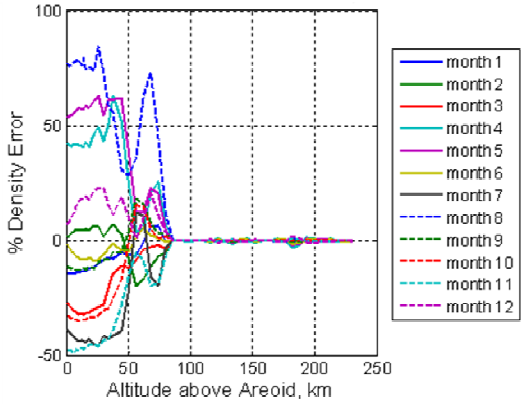


Figure 3.2: Linear Least Squares Prediction Using Vertical Ensemble

In the regions where no data is available and the prediction performance becomes important, the prediction accuracy degrades rapidly. As can be clearly seen, the results produced from this approach are also unstable and produce infeasible densities for many of the predicted months. This behavior is puzzling until more closely considering the nature of the ensemble. While each ensemble member is distinct, all of the members have very similar functional form to the other members. This interrelationship creates an approximate linear dependence among the ensemble members leading to numerical ill-conditioning when computing the inverse of computations based on the H matrix. As Farrar and Glauber[76] explain, as the correlation among the explanatory variables (the ensemble members in this case) grows, the matrix (H^TH) approaches singularity and the elements of $(H^TH)^{-1}$ begin to explode. This phenomena, known as multicollinearity, effectively means that the similarity between the ensemble members complicates efforts to allocate the explained variance. A nearly singular (H^TH) matrix indicates that the explained variance can be allocated in an almost arbitrary fashion among the explanatory variables.

Several computational methods exist for overcoming the limits imposed by multicollinearity. Techniques such as partial least squares[77] or principal component regression[78] first remove the collinearity among the explanatory variables by determining an appropriate matrix transformation. Then the classic linear least squares algorithm can be applied and the results interpreted in the original matrix space defined in the problem statement. Because these methods require a higher degree of computational complexity, the parameter estimation problem will be approached with simpler methods.

3.2.3 Weighted Average Scheme

An alternate technique was derived in an attempt to stabilize the parameter estimates $\hat{x}$ and increase prediction accuracy. Note that BIBO stability can be intrinsically assured by forming a weighted average of the ensemble members. Because each

ensemble member does not, by definition, diverge to unreasonable density estimates for a given spatial location, the weighted average must also be stable and contained by the most extreme ensemble members. The most extreme ensemble members, both in the vertical ensemble and in the neural ensemble, are derived from the Warm, Cold, and Dust Storm MCD scenarios. Because these scenarios are designed to bracket the possible conditions on Mars[43], weighted averages can still capture the possible variance in the atmospheric state within the limits of the simulation uncertainties.

To form a weighted average of the ensemble members, some weighting criteria must be determined. The approach taken herein was adapted from and is very similar to the Dynamically Averaging Networks concept developed by Maqsood et al[57] for classifier NN ensembles. A classifier NN uses a series of input information to make some judgmental decision about the input data. Useful for image and pattern recognition, classifier NNs could be used, for example, to separate a series of points in two-dimensional space into clusters.[79]

Maqsood defines a "certainty" parameter which quantifies how confident a NN member is in its classification judgment for a particular input. The weight of each member of the ensemble then becomes proportionate to its certainty such that more certain NNs are given higher weights. The ensemble weights are then scaled to sum to unity and consequently form an ensemble weighted average. For the present problem of density estimation, however, the NNs corresponding to each member of the ensemble are not classifier networks. Thus, the certainty parameter for the i^{th} ensemble member can be replaced with its root mean square error (RMSE). Recall that the RMSE quantifies the average deviation of the i^{th} ensemble member from the observed density trend measured by the ADS. Therefore, ensemble members with smaller RMSEs are closer to the measured trend and should be more highly trusted. Mathematically, the un-scaled weight w_i for each ensemble member is the inverse of the associated $RMSE_i$ or $w_i = 1/RMSE_i$. These un-scaled weights are then scaled such that the sum of the scaled weights is unity;

$$\omega_i = \frac{w_i}{\Sigma_i w_i} \text{ such that } \Sigma_i \omega_i = 1.0 \tag{3.6}$$

Despite its simplicity, this parameter estimation scheme has the potential to produce accurate prediction results. Figure 3.3 shows the prediction error-vs-altitude curves for the twelve predictions associated with the months of the Average climate scenario. The symmetric black lines denote the atmospheric state uncertainty of the vertical ensemble used for prediction. This uncertainty is computed by determining the percentage difference between the largest and smallest densities estimated by the vertical ensemble at each altitude level. Note that this uncertainty profile exhibits good agreement with the Martian atmospheric uncertainty given by Engelund et al. (25-200% depending on altitude)[5] as the uncertainty at ground level is 25.26%. Determination of the atmospheric uncertainty using the ensemble is appropriate because the ensemble should represent the communal state of knowledge of the atmosphere at entry. This is why the vertical ensemble was developed using the Warm and Cold scenarios as they are designed to bracket the likely atmospheric variance outside of a global dust storm. Improvement to this knowledge state during the entry represents an uncertainty reduction over the best pre-entry atmospheric state estimates.

It is useful to compare the results in Figure 3.3 to the winner-take-all (WTA) results in Figure 3.1. Note that the WTA predictions tend to uniformly under predict the correct density trend at middle altitudes (40-86 km) with the notable exception of month 10 which is a significant over prediction. This is in direct contrast to the RMSE weighted average results where the predictions both over- and under predict with equal frequency. However, note that several WTA predictions exhibit relatively high accuracy over wide altitude ranges despite the uniform under prediction tendency (for example, examine months 1-6, 8, & 9 at middle altitudes in Figure 3.1). Thus, the WTA scheme produces regions of higher accuracy at very low computational cost while the RMSE scheme produces more accurate predictions at low altitude and more balanced prediction errors at

middle altitudes. Because both the RMSE and WTA schemes have desirable properties, a method to combine the strengths of each is developed in the next section.

With respect to the five algorithmic goals stated earlier, the simplicity and stability goals have clearly been reached with the RMSE scheme. Robustness is respectable because no predictions are markedly worse than the others. Prediction accuracy, especially at high altitude, could certainly be improved as could the incorporation of some aspect of optimality. These are addressed with the addition of a basis pruning concept.

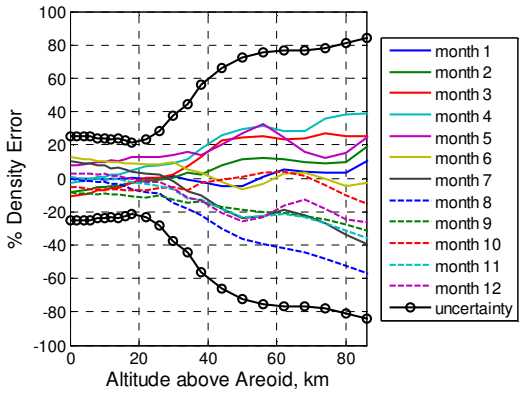


Figure 3.3: Low-altitude Prediction of Vertical Ensemble Using RMSE Weighted Average Scheme

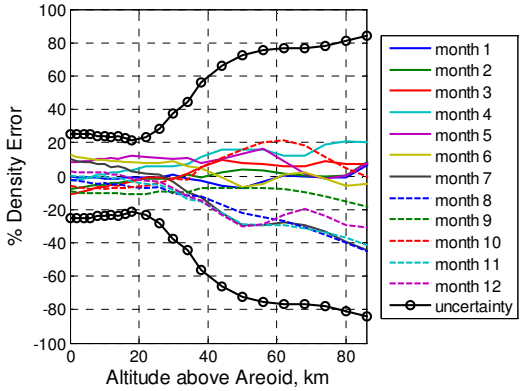


Figure 3.4: Pruned RMSE Weighted Average Scheme

3.2.4 Weighted Average Scheme with Ensemble Pruning

In studying the dynamics and general behavior of neural ensembles, Zhou et al[80] came to an interesting if somewhat counterintuitive conclusion. They found that predictions based on combining a subset of an ensemble rather than using all of the ensemble members often results in higher accuracy. This conclusion, shown to hold true in the present study, is likely due to the fact that inaccurate ensemble members assigned small weights may still generate large prediction errors. This suggests that one or more of the ensemble members should be pruned from the prediction basis during the ensemble combination process. Han and Shi[59] used this pruning concept to produce a "trimmed average" of a neural ensemble describing the water level along the Atlantic coast of Canada. This trimmed average combined all of the coastal NNs in their ensemble using a simple average except the NNs producing the most extreme water level values. They concluded that the trimmed average ensemble consistently outperformed the un-trimmed ensemble combination.

As mentioned earlier, combining the strengths of the WTA and the RMSE parameter estimation schemes would be quite useful. This goal may be approached by incorporating an ensemble pruning strategy such as that suggested by Zhou et al. However, an *a priori* trimming of the most extreme ensemble members is not useful in this case. Removal of extreme members seems most helpful when attempting to reduce the influence of anomalous empirical results on a forecast model. In this case, however, the conditions encountered during the entry may be similar to any member of the ensemble, including the extreme members. Therefore, ensemble pruning should proceed based on the proximity of the individual ensemble members to the measured density trend. This can be accomplished by ranking the members in order of decreasing RMSE and removing one member at a time from the prediction basis. After each member is removed, the average retroactive prediction performance (i.e. the average prediction error

from the current time back to entry interface) is computed for the currently active set of ensemble members.

Performance is first examined for the case where all members are active, then the member with the largest RMSE is removed and retroactive performance is retested, and so on. If one or more ensemble members have equal RMSE values, then they are eliminated from the prediction basis simultaneously. Pruning continues until only one member remains in the prediction basis (see Figure 3.5). At this opposite extreme, the single remaining member has the smallest RMSE and therefore corresponds directly to the winner-take-all scheme evaluated above. After the WTA boundary is reached, the algorithm reviews the retroactive prediction results and selects the subset of members which produces the smallest average error. This approach enables optimization of the ensemble combination without resorting to the use of computationally expensive zero-order or gradient-based optimization schemes. By selecting a subset of the ensemble as the prediction basis, this pruning concept produces accurate predictions over large altitude ranges like the WTA scheme but with more balanced prediction errors like the RMSE scheme.

Mathematically, pruning is simple to accomplish and is performed by setting the current minimum un-scaled weight $w_{i,min}$ to zero. This minimum weight corresponds to the ensemble member with the largest RMSE among the remaining active members. For example suppose a notional ensemble exists where the minimum un-scaled weight is currently w_2:

$$[w_i] = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_i \end{bmatrix} = \begin{bmatrix} w_1 \\ (w_i)_{\min} \\ \vdots \\ w_i \end{bmatrix} = \begin{bmatrix} 1/RMSE_1 \\ 1/RMSE_2 \\ \vdots \\ 1/RMSE_i \end{bmatrix} \Rightarrow [w_i] = \begin{bmatrix} 1/RMSE_1 \\ 0 \\ \vdots \\ 1/RMSE_i \end{bmatrix} \tag{3.7}$$

Upon scaling, the remaining weights will still sum to unity and still have magnitudes inversely proportional to their associated errors.

$$[\omega_i] = \begin{bmatrix} w_1 / \sum_i w_i \\ 0 / \sum_i w_i \\ \vdots \\ w_i / \sum_i w_i \end{bmatrix} = \begin{bmatrix} \omega_1 \\ 0 \\ \vdots \\ \omega_i \end{bmatrix} \text{ where } \sum_i \omega_i = 1.0 \tag{3.8}$$

Because the atmospheric dynamics may continually change throughout the entry, pruning of an ensemble member is reversible. Pruned models may reenter the basis at a later time if the accuracy of the member improves relative to the observed density trend. This is possible because the entire pruning process is repeated each time the parameter estimation scheme is called. For complete algorithm implementation details, see Section 3.3.

Testing of this pruning method supports the conclusions of Zhou et al. and reveals an interesting general trend. The first models to be eliminated are the most erroneous ensemble members and prediction accuracy typically increases following their removal (Figure 3.5). The prediction error typically reaches a minimum with some intermediate number of members retained. As pruning continues, members which have some degree of explanatory power for the measured trend are expelled and this generates an increase in prediction error toward the WTA boundary. Therefore, an optimum typically emerges as pruning progresses.

Empirical verification of this general trend is presented in Figure 3.6. Here, the maximum spread of the twelve prediction models is shown as a function of the number of pruned ensemble members. The maximum spread is defined as the difference between the largest and smallest prediction errors over the twelve models in the prediction regime (0-86 km). Note that as the number of pruned members increase from zero the spread decreases, indicating that the majority of the models are improving. If the majority of the models were not improving, then the spread would either increase or remain unchanged. The spread decreases until a minimum appears at eleven pruned members before rising again when useful members are unnecessarily pruned from several models. Thus, many

of the prediction models benefit from some type of pruning and an optimal pruning typically exists.

Figure 3.4 displays the twelve prediction models at the "optimal" pruning of eleven members (i.e. optimal in the sense of minimizing the total spread). Observe that the un-pruned results presented in Figure 3.3 have a spread of 96% while the optimally pruned results in Figure 3.4 have a significantly reduced spread of 66.4%. This conclusion is also directly observable by comparing the un-pruned and optimally pruned maximum spreads in Figure 3.6.

The local increase in maximum spread in Figure 3.6 as pruning proceeds from five to seven members is caused by the reweighting process. Members with intermediate accuracy are temporarily given higher weights when the poorest performers are eliminated. These intermediate accuracy members are eventually eliminated as pruning continues and the spread once again decreases.

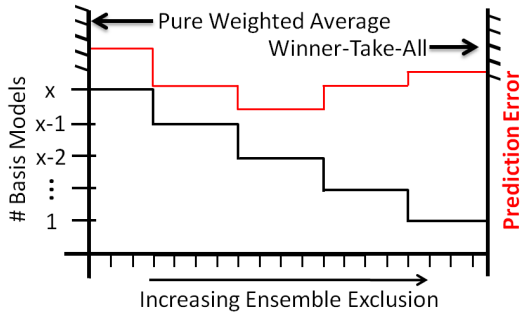


Figure 3.5: Typical Variation of Prediction Error in Ensemble Pruning Strategy

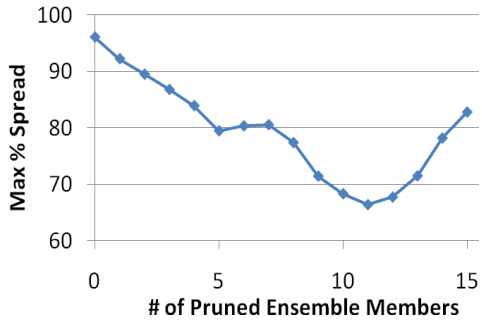


Figure 3.6: Effect of Pruning on Prediction Model Maximum Spread

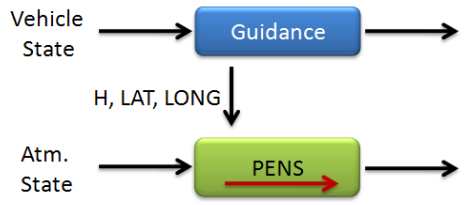
As demonstrated above, the RMSE parameter estimation scheme coupled with an ensemble pruning strategy produces robust prediction results with sufficient accuracy to reduce the atmospheric uncertainty, satisfying two of the five algorithmic goals. The algorithm produces a weighted average of an ensemble subset which by definition provides a BIBO-stable solution. The algorithm also provides a means of optimizing the prediction results via the ensemble pruning strategy which satisfies the optimality criterion of the algorithmic goals. Lastly, the proposed method maintains algorithmic simplicity by utilizing simple mathematical tools. Computationally expensive operations such as matrix inversions, numerical integration, or classical optimization routines are avoided in order to limit the necessary computational overhead.

3.3 Integrated Concept Testing

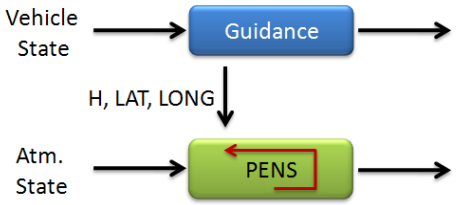
The flexibility provided by the pruning method enables the topology of the ensemble to continuously adapt to better match the current flight observations. This, in turn, allows the underlying dynamics of the density prediction algorithm to evolve in response to changes in observed conditions (recall from Section 2.1 that this was the primary research goal). This flexibility suggests that the proposed method is best described as a plastic ensemble neural system (PENS). In the preceding sections, the developmental testing of the parameter estimation scheme took place using the vertical ensemble with only one forward prediction at 86 km altitude. In an actual entry scenario, forward prediction should occur multiple times over the course of the entry in order to produce a continually improving estimate of the current atmospheric state. In the integrated testing phase, the parameter estimation scheme is married to the neural ensemble to simulate the real time application of the density prediction problem. Note that the dust storm models are not yet utilized in the neural ensemble because the test case atmospheres are bounded by the remaining five ensemble members. The performance of ensembles which include the dust storm models is evaluated later (see

Chapters 4 and 5). The integrated testing is approached in several stages of evolving complexity and fidelity:

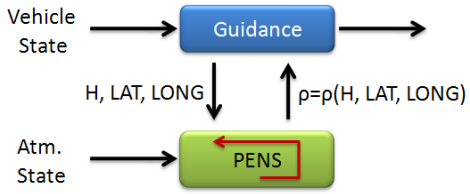
- *Phase I Testing*: In this first phase of integrated testing (Figure 3.7.a), a static trajectory is assumed and no guidance model is used. This essentially converts the PENS method into a fly-along payload which performs a series of forward-looking atmospheric predictions at different points in the trajectory but without the ability to influence guidance decisions. No internal feedback within the PENS algorithm is allowed to simplify the dynamical behavior for open loop proof-of-concept testing.
- *Phase II Testing*: The second phase of integrated testing (Figure 3.7.b) uses the same static trajectory and also lacks a guidance model. However, in this phase, internal feedback is created in the PENS method through the introduction of an ensemble echo, a new construct for computationally constrained ensemble forecasting. The ensemble echo improves the density prediction performance by enabling associative learning from in-flight experience. The echo provides the PENS algorithm with an evolutionary memory which stabilizes the prediction performance without the addition of significant computational complexity.
- *Phase III Testing*: The third testing phase (Figure 3.7.c) closes the guidance loop by examining the interaction of the echo-augmented PENS algorithm with a numerical predictor-corrector guidance method.



(a) Phase I Testing



(b) Phase II Testing



(c) Phase III Testing

Figure 3.7: Progression of PENS Testing Phases

3.3.1 Phase I Testing

The first phase of testing was designed to establish the open-loop dynamics of the PENS method without feedback internal to PENS and in isolation from an active guidance system. The baseline aerocapture trajectory defined in Section 2.7 was utilized as the static trajectory for this analysis. The system block diagram for this phase is presented in Figure 3.8. The ensemble density vector $[\rho_i]$ consists of five estimated densities, one for each of the five MCD climate scenarios (recall that the dust storm models are not yet utilized), computed by the neural ensemble developed in Section 2.5 at the current vehicle location. The percent difference between each of the elements of $[\rho_i]$ and the estimated freestream density ρ_{ADS} is computed using Equation 3.1. This results in an error vector which has one element for each ensemble member. As time progresses, a series of error vectors accumulates and are stored in an error matrix; each vector becomes

one row of the matrix and each column is the error history of one ensemble member. The resulting error matrix is used to compute the retroactive root mean square error for each of the ensemble members from entry interface up to the current time. The RMSE for each of the ensemble members is then passed to the parameter estimation scheme which computes the optimal ensemble weight vector $[\omega]$. The weight vector is passed to the ADALINE gating network which constructs the ensemble linear combination representing the current predicted density field.

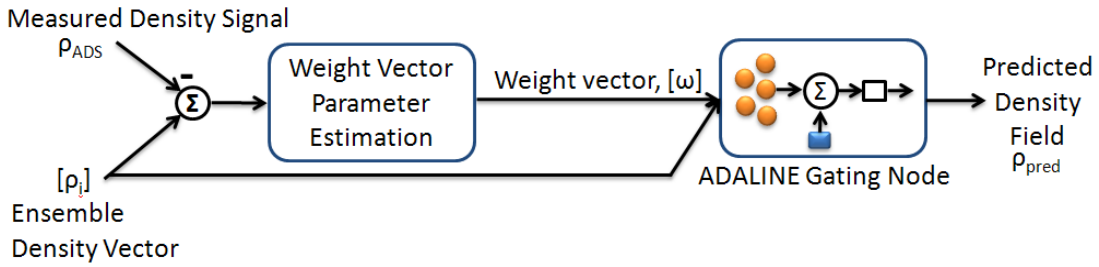


Figure 3.8: Feedforward PENS Block Diagram used for Phase I Testing

In order to characterize the prediction performance of the Phase I, feedforward-only system, an ideal testing scenario was created. The position history of the trajectory was used to create a replacement for the default MarsGRAM density profile based on a simple linear combination of two of the ensemble members. Recall that the ensemble NN models provide density as a function of altitude, latitude, and longitude. A replacement density profile can be constructed by evaluating one or more NNs using the position history provided by the nominal trajectory. Note that a change in the density profile will alter the position history of the nominal trajectory. However, for the sake of testing consistency, the nominal trajectory was assumed to be fixed for the following tests. Early testing proved that if any one of the NN models were used to replace the reference trajectory density, the PENS method was consistently able to identify the correct NN model. The goal of testing a simple linear combination was to determine if the PENS method could identify an ideally correct linear combination if one actually existed. The position history was supplied to the Average and Minimum NN models which resulted in

two density profiles. These profiles were then averaged with equal weight such that the ideal solution is ω_{avg} = 0.5 and ω_{min} = 0.5. The Average and Minimum NN models were selected because their density profiles are similar to one another for the nominal trajectory. This creates a challenging testing environment which seeks to determine if the PENS method has the requisite sensitivity to delineate the two correct models from among all the possible choices.

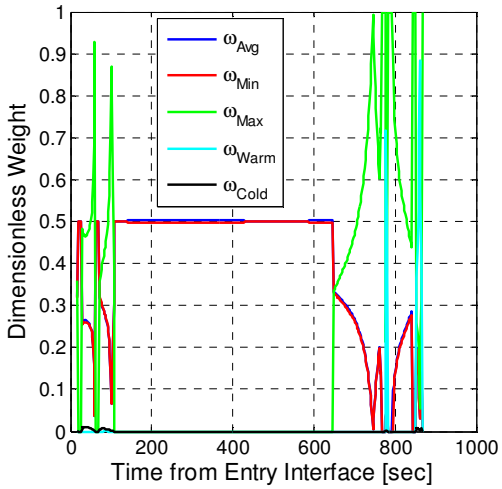


Figure 3.9: Ideal Case Ensemble Weight Vector History

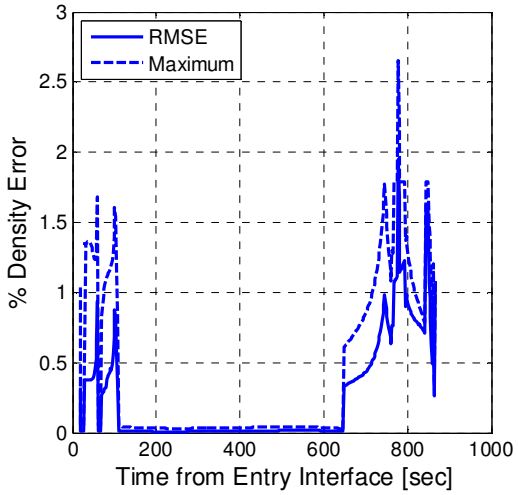


Figure 3.10: Forward-Looking Prediction Accuracy

Figure 3.9 presents the ensemble weight vector history for the ideal testing scenario. Note that the method correctly determines the ideal solution for a large region

of the trajectory between 109 and 647 seconds. Although difficult to identify from inspection of the figure, the method also correctly identifies the ideal solution for the initial 6 seconds and again between 62 and 68 seconds. As can be clearly seen, however, the PENS method loses sight of the ideal solution over several large intervals, mostly in higher altitude regions. This is due to the fact that the three core NN models with the average dust parameterization (the Average, Minimum, and Warm NNs) have very similar density trends (Figure 3.11) between roughly 60 to 120 km for this particular season, time of day, and location. The core models become more distinct near periapsis (Figure 3.12) at 46 km and also begin to diverge from one another in the thermosphere above 136 km. This similarity of the core models at middle altitudes explains why the density prediction error (Figure 3.10) remains small despite the fact that the ideal solution is not identified throughout the entire entry. Figure 3.10 is a RMSE/Maximum error plot. This type of plot is used repeatedly throughout this chapter and therefore a detailed explanation of this plot type and the data it presents is available in Appendix B.

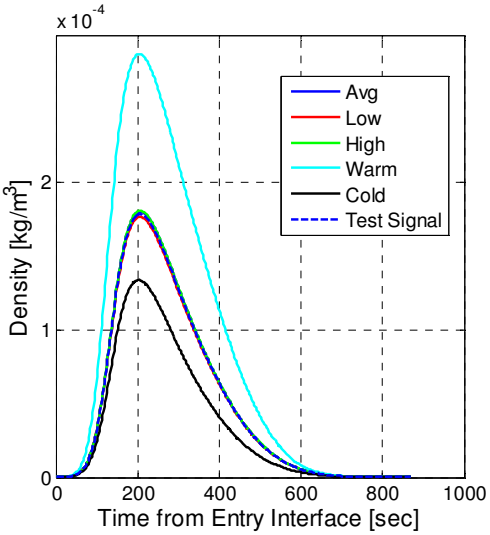


Figure 3.11: Ensemble Density History Along Nominal Reference Trajectory

In order to perform the most stringent test possible, the PENS testing algorithm constructs a series of density predictions from many points in the trajectory. In this manner, the testing algorithm proceeds to step through the trajectory at four second

increments, simulating the motion of the vehicle through the atmosphere. The ADS gathers density data from entry interface up to the current position of the spacecraft. At each time step, the PENS method is used to make a single density prediction spanning the remainder of the trajectory. The testing algorithm then compares each predicted density field with the true density (which is the ideal test signal in this case). The performance of each density prediction is individually reported and no time-averaging techniques are used.

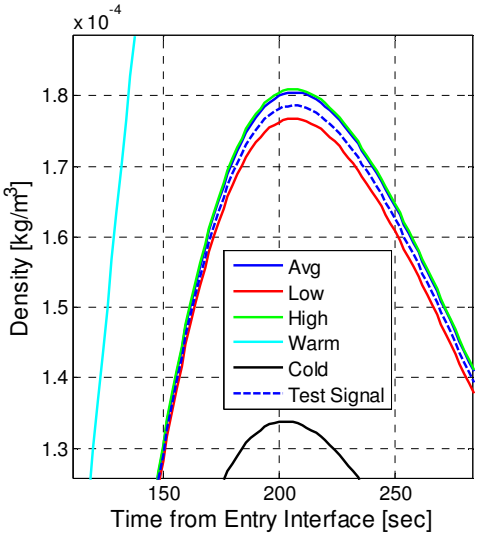


Figure 3.12: Ensemble Density History Near Periapsis

This approach is designed to estimate the performance of the PENS algorithm if it were only permitted to perform a single prediction throughout the entire trajectory. This approach not only provides a stringent testing scenario, it also permits many predictions to be easily presented and compared. Most importantly, it also enables understanding of the evolution of prediction accuracy as more measured density data from the ADS becomes available. For example, in Figure 3.10, it can be clearly seen that the availability of recent ADS information after 600 seconds does not positively impact prediction accuracy. (In fact, the ADS information is being used very inefficiently at this point, a shortcoming that will be addressed in the following sections.) The density error data

presented in Figure 3.10 represents the prediction error from over 400 separate density predictions made from locations throughout the reference trajectory. The solid line indicates the average prediction error at a given prediction time while the dashed line represents the maximum error. For example, if only a single PENS prediction is made at 647 seconds, the average prediction error over the remainder of the trajectory is 0.34% while the maximum error is 0.62%.

As mentioned, the PENS prediction loses sight of the ideal solution at several locations. Because the test signal is a linear combination of the Average and Minimum models, it remains confined between these two pure signals and never intersects them. When one of the other ensemble members (typically the Maximum model in the above test) intersects the test signal, the local error of the intersecting member becomes very small relative to the other ensemble members. This inhibits the growth rate of the average (RMSE) error of the intersecting member. If the intersecting member remains close to the test signal over a sufficiently long interval, the RMSEs of the pure members temporarily exceeds the intersecting member. This causes the intersecting member to become more highly weighted by the parameter estimation scheme. Note that the prediction errors still remain low because the intersecting member must be sufficiently close to the test signal over an extended period for this local aliasing to occur. Despite the low error magnitudes, this local aliasing may still cause difficulty if the PENS algorithm is used multiple times in a single trajectory. If an intersecting member is only locally similar to the test signal and is significantly different far in the future, local aliasing may cause the far-field predicted density trend to rapidly change as the parameter estimation scheme temporarily trusts the wrong model. Therefore, it is necessary to seek an augmentation to the current implementation which will combat local aliasing and increase prediction accuracy.

3.3.2 Phase II Testing & Supporting Research Question 3

The goal of Phase II testing is to determine how atmospheric prediction experience gained in flight can be leveraged to improve future state predictions. The current PENS implementation has no means of remembering good solutions or learning from in-flight experience. However, because the Phase I PENS implementation was able to determine the ideal solution over an extended interval, it is conjectured that a feedback capability that remembers good solutions for future consideration will improve performance. Because more than one good solution may exist and because these solutions may occur at any time during the entry, conventional simple feedback may not be appropriate. Simple feedback only permits the current state vector to be retained and will not readily store a series of high quality solutions. In addition, simple feedback alone will not enable associative learning. Multiple high quality prediction solutions may exist but only one can be utilized by the vehicle guidance at any given moment. Therefore, associative learning is highly desired in this application as a method for determining which of the stored solutions should be trusted based on the observed density trends.

Any augmentation to the Phase I PENS algorithm to permit solution retention and associative learning should process several desirable characteristics. Recall from Section 2.6 that both Johnson and Oh[69] and Chowdhary and Johnson[70,71] have trained a NN-based adaptive flight controller in real time using both background and online information. Prior NN-based adaptive controllers only used instantaneous states to tune the adaptive gains which limited the adaptability of the controller. Their new approach utilized stored state information (background information) as well as the current state (online information) to simultaneously train a NN. They demonstrated that this approach results in faster adaptation to unknown model dynamics and enhances the long term learning ability of the controller. This algorithm was successfully flight tested on an unmanned aerial system, the Georgia Tech GTMAX rotorcraft which is based on the Yamaha RMAX helicopter.[81] Due to the success of this approach, a similar method utilizing both

background and online data will be developed here to augment the adaptive training process for the ADALINE gating network.

Another desired characteristic is the ability to simulate Hebbian learning, a theory of learning and memory in human neurology. First proposed by Donald Hebb in 1949, it attempts to explain the physical mechanism underpinning associative learning.[82-85] The basic theory states that persistent use of a given chain of neurons leads to an increase in the efficiency of that chain whereas non-use of a chain results in an active decrease in the synaptic efficiency. Because it has been postulated that memories are stored in chains of biological neurons, Hebbian theory offers an explanation as to why often-used memories are easy to recall whereas seldom used memories fade over time. In artificial neural models, Hebbian learning is characterized as a reinforcement learning method. According to Rojas,[79] "each input produces a reinforcement of the network weights in such a way as to enhance the reproduction of the desired output." This highlights a necessary key feature of the desired feedback structure required to facilitate Hebbian learning. Each set of density observations from the ADS should be used to determine which density prediction model (or set of models) has the highest performance. A Hebbian algorithm then reinforces the trust or weight associated with these more accurate density models while suppressing less accurate models. Rojas also states that linear associators (i.e. neurons such as the ADALINE which form a weighted average of their inputs) are predominantly used in building networks capable of associative learning. Therefore, the Phase I architecture is a well-suited foundation for the feedback approach proposed here.

The third and final desirable characteristic for the feedback augmentation concerns the performance of individual solutions. If the PENS method is repeatedly called throughout the entry, a series of solutions in the form of ensemble weight vectors will be developed over time. The pruned RMSE parameter estimation scheme developed in Section 3.2 produces weight vectors which attempt to minimize the average retroactive prediction error. Therefore, up to this point, performance has always been based on an

average quality metric. However, a solution vector that has both a small average error and a small instantaneous error at the current vehicle location is highly desired. Considering local performance is yet another method of discriminating among the possible solution vectors in order to identify the best option for accurate density prediction.

Therefore, the desired feedback augmentation is one which:

- Simulates Hebbian learning
- Allows simultaneous training of a NN using both background and online data
- Balances good average (long-term) performance with good local (short-term) performance

All three of the stated goals may be accomplished through the introduction of an *ensemble echo*, a new construct for adaptive ensemble forecasting. The Phase II system (Figure 3.13) consists of the entire Phase I architecture (black lines) augmented with an adaptive feedback structure (blue dashed lines). The creation of an ensemble weighted average begins with the same steps as the original Phase I implementation. Note that each step described below is indicated on Figure 3.13 by a red number in parentheses which corresponds to the step number (e.g. step number 1 is marked with a red (1) in Figure 3.13).

1. Each element of the ensemble density vector is subtracted from the free stream density measured by the ADS. This creates an error vector whose length is equivalent to the number of ensemble members.
2. The error vectors are stored in an error matrix where each row represents an error vector from one particular time step and each column represents the error history of a single ensemble member.
3. The pruned RMSE parameter estimation scheme is used to determine a scaled weight vector $[\omega]$ which is passed to the ADALINE gating network in order to form the ensemble linear combination.

Up to this point, the implementation described here is identical to the steps utilized in the Phase I architecture. In Figure 3.13, these steps are equivalent to proceeding along the forward cascade (i.e. the black, solid lines) from the density estimate ρ_{ADS} to the production of an estimated density field ρ_{pred}.

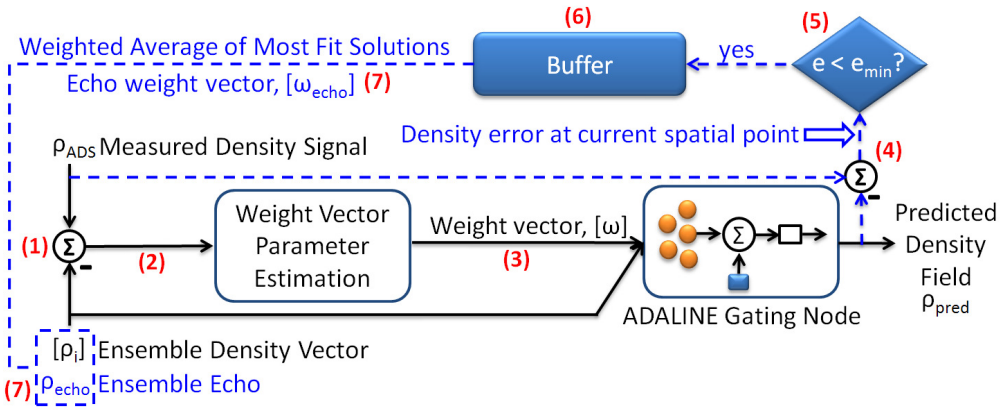


Figure 3.13: Phase II PENS Block Diagram: Implementation of the Ensemble Echo

The feedback structure implemented in Phase II testing begins with the predicted density field;

4. Once the predicted density field has been established, the measured density signal from the ADS is compared to the estimated density at the current vehicle location using the newly created ensemble linear combination. The result is a density error e at the current location of the vehicle which is computed in a manner analogous to Equation 3.1;

$$e = \left(\frac{\rho_{pred} - \rho_{ADS}}{\rho_{ADS}} \right) 100\% \tag{3.9}$$

5. The value of the current density error e is used to determine if the current solution defined by the weight vector $[\omega]$ is of sufficient quality to warrant retention for future consideration. If the current error e is less than some minimum error value e_{min}, then the weight vector $[\omega]$ is stored in a buffer along with its associated instantaneous error e.

6. The buffer itself consists of a user-defined number of saved solution vectors along with the associated instantaneous errors of each weight vector. The largest instantaneous error currently in the buffer (i.e. the worst stored solution) serves as the decision criteria e_{min} for storing new solutions. If a candidate solution has an absolute instantaneous error $|e|$ which is smaller than the largest absolute error in the buffer, then the candidate replaces this worst buffered solution. Thus, the worst buffered solution is always in jeopardy of being expelled from the buffer by a solution with a smaller instantaneous error. In this manner, the buffer is designed to continuously evolve during the entry by seeking to store solutions with ever improving local performance. Early testing revealed that larger buffers take significant time to fill to capacity, elongating the convergence period of the PENS model. Shorter buffers do not have the storage capacity to remember a large number of high-quality solutions and may be unable to retain important models. It was found that a buffer size of 30 solutions fills quickly while also providing sufficient memory capacity For example, if the ADS sampling frequency is 1 Hz, the buffer fills in quickly and begins to stabilize after roughly one minute. After this point, expulsion/retention of bad/good solutions in the buffer becomes less frequent and the PENS model converges to a quasi steady-state.
7. A weighted average of the buffer members is formed where the weights of each of the stored solutions are the inverse of the instantaneous stored errors;

$$[\omega'_{echo}] = \left(\frac{1}{b}\right)\sum_{k=1}^{b}\frac{1}{|e_k|}[\omega_k] = \left(\frac{1}{b}\right)\left[\frac{1}{|e_1|}\begin{bmatrix}\omega_1\\ \vdots \\ \omega_n\end{bmatrix}_1 + \frac{1}{|e_2|}\begin{bmatrix}\omega_1\\ \vdots \\ \omega_n\end{bmatrix}_2 + \cdots + \frac{1}{|e_b|}\begin{bmatrix}\omega_1\\ \vdots \\ \omega_n\end{bmatrix}_b\right] \quad (3.10a)$$

where b is the number of stored solution vectors and e_k and $[\omega_k]$ are the kth instantaneous error and the kth stored solution vector respectively. Note that the number of buffered solutions b changes during the startup period when the buffer

is not yet filled to capacity. Once the buffer capacity (30 solutions for all results herein) is reached, b remains constant for the remainder of the trajectory. The resulting vector $[\omega'_{echo}]$ is not a true weighted average because the $(1/|e_k|)$ 'weights' applied in Equation 3.10a are not constrained to sum to unity. In order to form a true weighted average the vector elements must sum to unity which requires a rescaling of the vector:

$$[\omega_{echo}] = \frac{[\omega'_{echo}]}{\sum_{i=1}^{n} \omega'_{echo,i}} \tag{3.10b}$$

The echo weighted average $[\omega_{echo}]$ is structurally identical to the stored weight vectors $[\omega_k]$; all are unit vectors whose lengths n are equivalent to the number of ensemble members. Just as each stored solution vector $[\omega_k]$ may be used to define an ensemble linear combination, so too may the echo weight vector $[\omega_{echo}]$. This linear combination defined by the echo weight vector is a predicted density field called the *ensemble echo*. While linearly dependent on the original ensemble, the echo is treated as if it were a new and independent ensemble member. Therefore, following the first iteration, the weight vector parameter estimation scheme operates on n+1 ensemble members rather than only n members as in the Phase I architecture.

The formation of the ensemble echo completes the feedback loop. Note that because the ensemble echo is a linear combination of the original ensemble, it contains no new or additional information. It does, however, highlight the most important components of the original ensemble and functions as an evolutionary estimate of the current atmospheric state. As such, the echo should become more highly trusted (i.e. more highly weighted) the longer the vehicle is in the atmosphere.

It should be noted that instantaneous errors which are exactly zero are not considered for entry into the buffer. This is because the inverse of one or more $e_k = 0$ in Equation 3.10 leads to infinite weights on one or more of the buffered solutions.

Applying an infinite weight allows a single solution (or a small handful of solutions) to drive the behavior of the ensemble echo. While it is algorithmically possible to apply either an infinite or extremely large weighting to one or more of the buffered solutions, it is mathematically unjustifiable. Recall that the prediction model for density is a linear combination of the ensemble members (see Equation 2.2). Because there are n unknown ensemble weights ω_i, a single density measurement provides an indeterminate system with n unknowns but only one equation;

$$\rho_{ADS} = [\rho_{pred}] + e = [\rho_C \omega_C + \rho_m \omega_m + \rho_A \omega_A + \rho_M \omega_M + \rho_W \omega_W] + e \qquad (3.11)$$

Therefore, there exists an infinite number of solutions which produce an identically zero instantaneous error, $e \equiv 0$. Many of these solutions are only locally valid because an instantaneous error of zero often signifies an intersection of the candidate model and the measured density signal. Because such solutions would dominate the behavior of the echo and may only provide locally valid predictions, zero error solutions are not permitted into the buffer. However, as will be shown, if an accurate zero error solution exists then there also exists a nearby neighboring solution that has the same predictive power but does not generate an algorithmic singularity.

3.3.3 Hebbian Learning

The ensemble echo accomplishes the three stated feedback goals. To satisfy the first goal, the algorithm simulates a basic Hebbian learning process through both reinforcement of good solutions and active suppression/depression of bad solutions. Synaptic reinforcement is accomplished by the interaction of the buffer, ensemble echo, and the parameter estimation scheme. Note that the synapses of interest here are the ensemble weights found at the junctions between the ensemble NNs and the ADALINE network. Good solution vectors are stored in the buffer and effectively re-presented to the parameter estimation scheme via the echo. If the ensemble echo is accurately describing the atmospheric dynamics, it is highly weighted by the parameter estimation scheme

which causes a reinforcement of the appropriate weights in the ADALINE gating network (see discussion in Section 3.3.4 for details).

Synaptic suppression is accomplished by several mechanisms working in concert. Because all weight vectors are unit vectors, a reinforcement of some ensemble members immediately leads to the active suppression of the weights of the remaining members. In the limiting case, if one member were assigned unit weight, then the remaining members must necessarily all have zero weight. Also, recall that the worst buffered solution is continually in jeopardy. If a better solution is identified, this worst solution is removed from the buffer and the synaptic activity (i.e. the weights) associated with it are deleted. In addition, solutions which do not display promising local performance are not stored in the buffer and are effectively "forgotten." Lastly, ensemble members which are completely inappropriate for the observed density trend are actively pruned from the prediction basis.

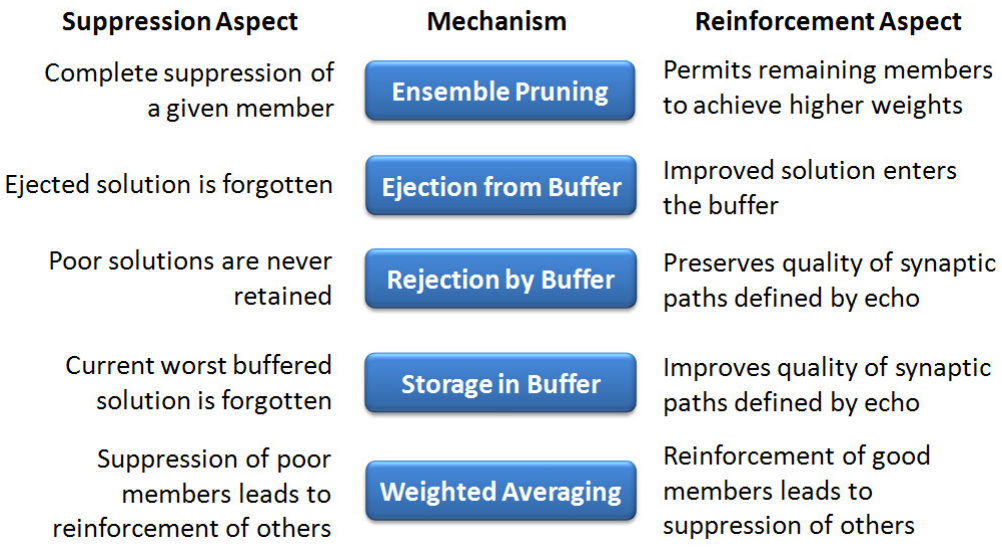


Figure 3.14: Duality of Synaptic Suppression and Reinforcement Mechanisms

Figure 3.14 illustrates the various mechanisms used to accomplish synaptic reinforcement and suppression. It is interesting to note that a duality exists in these reinforcement and suppression mechanisms. That is, the same mechanism accomplishes

both reinforcement and suppression such that the two functions may be viewed as opposite faces of the same coin. The first mechanism shown in the figure, ensemble pruning, removes one of the ensemble members from the prediction basis for a period of time. This completely suppresses the pruned member while permitting the remaining ensemble members to achieve higher (i.e. reinforced) weights reflective of their respective predictive abilities. Thus both reinforcement of useful members and suppression of members with high prediction error are accomplished by the same mechanism.

The next three reinforcement/suppression mechanisms in Figure 3.14 describe the behavior of the solution buffer which is used to define the ensemble echo. By ejecting poor solutions from the buffer, the synaptic activity (i.e. weights) associated with that deleted solution are suppressed. This ejection permits a better solution to enter the buffer which reinforces the synapses (weights) associated with better performing models.

By rejecting poor models (i.e. poor performing models are not considered as candidates for buffer storage), the weights associated with these poor solutions are suppressed by not preserving them for future consideration. Rejection of a candidate model by the buffer also preserves or reinforces the weights of the high-quality solutions currently stored in the buffer.

The buffer will only store models whose local error performance exceeds the performance of the worst model currently in the buffer. This worst model is then replaced by the new incoming candidate model which by definition exhibits better performance. In this manner, the buffer is evolutionary and will only accept new models if there is quantitative evidence that this new solution will improve the performance of the ensemble echo. The evolutionary storage of ever better models reinforces the synaptic activity associated with the ensemble members which best describe the observed density trend. Reflexively, deleting the worst stored model suppresses the weights associated

with that model ensuring that PENS is continually able to reevaluate the conclusions it draws from the measured atmospheric observations.

The last mechanism, weighted averaging, also leads to the simultaneous suppression of poor performing models and the reinforcement of accurate models. By enforcing that the summation of the model weights equal unity, the weighted averaging mechanism ensures that the final prediction models will be bounded inside the extreme limits of the ensemble. If new density observations reveal that a particular ensemble member is more accurately describing the local density trend, the parameter estimation scheme will begin increasing the weight associated with that model. In order to maintain the weighted average, the weight summation must remain unity. Therefore reinforcement of one synaptic path (i.e. increasing the weight applied to one ensemble member) immediately and reflexively leads to the suppression of other paths. Conversely, if the average error of one ensemble member continually increases as more observations are gathered, the parameter estimation scheme will suppress that member. This leads to an automatic reinforcement of the remaining models.

In this manner the same mechanisms which accomplish synaptic suppression also cause synaptic reinforcement. The duality of these mechanisms is useful as they create a Hebbian learning architecture in an efficient and complimentary manner.

3.3.4 ADALINE Network Training

The second goal of the feedback loop was to utilize both stored state information (background data) and current state information (online data) to simultaneously train the ADALINE gating network. This process may be understood by considering the flow of information around the system loop depicted in Figure 3.13 starting at the solution buffer. The buffered solutions function as background data consisting of saved state information. This data is used to define the ensemble echo via the feedback loop of Figure 3.13 and Equation 3.10. The echo is in turn used to train the ADALINE network through the

parameter estimation scheme (in the forward cascade of Figure 3.13 which is represented by solid arrows). The final product of the parameter estimation scheme is the current system state, defined as the weight vector of the *n* ensemble members. If this weight/state vector is found to produce accurate density predictions, it is saved in the buffer and the next iteration begins. The manner in which the parameter estimation scheme produces the weight/state vector (i.e. trains the ADALINE network) is crucial as it must somehow combine the information contained in the echo with the information in the other *n* ensemble members. This combination process proceeds by redistributing the echo components among the original *n* ensemble members. Recall that the echo is defined as a linear combination of the original ensemble which is treated as an independent atmosphere model:

$$\rho_{echo} = \sum_{i=1}^{n} \omega_{echo,i} \rho_i \tag{3.12}$$

where *n* is the number of original ensemble members and $\omega_{echo,i}$ are the elements of the echo weight vector $[\omega_{echo}]$ computed in Equation 3.10. Because the echo is treated as an independent ensemble member, the Phase II prediction model now has *n+1* ensemble members:

$$\rho_{pred} = \omega_e \rho_{echo} + \sum_{i=1}^{n} \omega_i \rho_i \tag{3.13}$$

where ω_e is the top-level weight applied to the echo model by the parameter estimation scheme. Note that because the echo is not actually linearly independent, it is possible to group like terms. Substituting Equation 3.12 into Equation 3.13 and grouping terms yields

$$\rho_{pred} = \omega_e \sum_{i=1}^{n} \omega_{echo,i} \rho_i + \sum_{i=1}^{n} \omega_i \rho_i = \sum_{i=1}^{n} (\omega_e \omega_{echo,i} + \omega_i) \rho_i \tag{3.14}$$

In this manner, the redistributed echo weights function as a history-based correction for the current state estimate ω_i. Thus the ADALINE network is trained using both online information (the current state ω_i) and background information (the historical

correction $\omega_e\omega_{echo,i}$). Note that because the weights of Equation 3.13 are elements of a unit vector and because Equation 3.14 is a direct re-statement of Equation 3.13, the summation of the n corrected weights ($\omega_e\omega_{echo,i} + \omega_i$) is still unity. If the ensemble echo accurately estimates the current atmospheric state, then the parameter estimation scheme assigns a higher weight ω_e to the echo. This, in turn, forces the vector of redistributed weights ($\omega_e\omega_{echo,i} + \omega_i$) in Equation 3.14 to more closely resemble the echo weight vector $[\omega_{echo}]$. This reinforcement learning process enables the PENS method to apply experience gained in-flight to the current forecast model in a computationally economical way. The echo concept, however, still provides a significant amount of flexibility. The ensemble echo can be marginally weighted or even pruned if it suddenly fails to model the atmospheric dynamics at the current vehicle location.

The redistribution of the weights explains the origin of the name *ensemble echo*. The redistribution process occurs between the parameter estimation scheme and the ADALINE network. The parameter estimation process takes place with $n+1$ ensemble members but only n synapses are required in the ADALINE network. Thus the echo represents a summary of past states which imitates an ensemble member. Like an echo, this imitation ensemble member disappears as state information progresses along the forward cascade of the system block diagram which is represented by the solid arrows in Figure 3.13.

3.3.5 Multiple Prediction Horizons

The final goal of the feedback structure is to create a capacity for balancing good long-term performance with good short-term performance. A high-quality candidate solution (i.e. a weight vector defining an ensemble linear combination) should produce small prediction errors both near the vehicle and far in the past near entry interface. Therefore, balancing of good local and average performance is important for producing high-quality solution candidates. This goal is automatically accomplished by the

dynamics of the ensemble echo. Note that the parameter estimation scheme produces solution candidates with good average performance based on the RMSE metric. The buffer which is used to define the echo examines the local performance of all of the solution candidates and only stores those with increasingly smaller local error. Thus, the solutions which define the echo have all been observed to perform well both locally and far from the vehicle.

3.3.6 Phase II Ideal Testing Results

With all three goals of the feedback structure accomplished, the prediction performance of the Phase II architecture was tested using the same ideal testing signal used for Phase I testing described in Section 3.3.1. Recall that this test signal was created from a simple linear combination of two of the ensemble members (ω_{Avg} = 0.5 and ω_{Min} = 0.5). Also recall that during Phase I testing, the ideal test signal was correctly identified over several intervals, including the initial 6 seconds near entry interface. The ideal prediction performance of the Phase II system is presented in Figures. 3.15-18. As may be seen by inspection of Figure 3.15, the ensemble echo is trusted completely from the beginning of the trajectory. Because the parameter estimation scheme is able to correctly identify the ideal solution at the outset of the simulation, it is immediately stored in the buffer. After the first prediction iteration, the buffer consists of the first stored weight vector only which in this case is the ideal solution. Therefore, this single vector defines the ensemble echo for the second prediction iteration (see Figure 3.16). The parameter estimation scheme compares the echo to the measured density signal, determines that the echo is almost identical to the measured signal, and places full trust in the echo for the second iteration. At the end of this iteration, weight reinforcement occurs and the second stored buffer entry is almost identical to the first within the numerical precision of the computation scheme. Subsequent iterations are practically identical to the second

iteration. Because the ideal solution is identified so early, the density prediction error (Figure 3.17) is practically zero as expected.

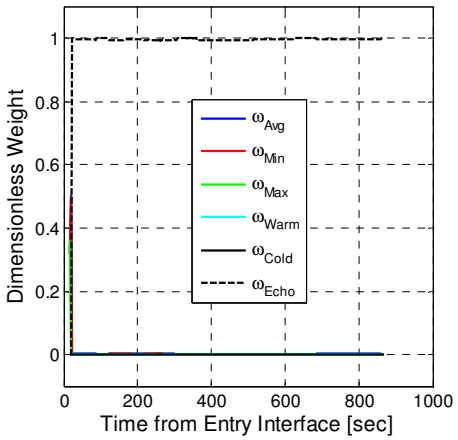


Figure 3.15: Ideal Case Ensemble Weight Vector History

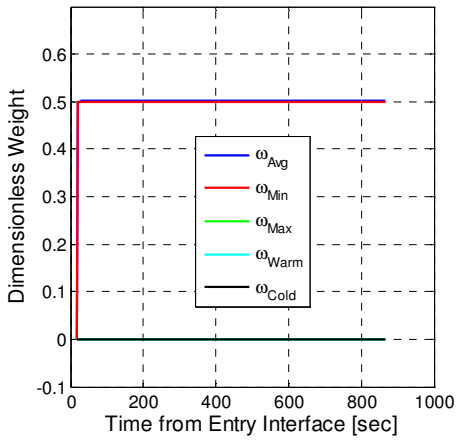


Figure 3.16: Ensemble Echo Weight Vector History

As noted before, solutions with exactly zero instantaneous error $e = 0$, are not permitted into the buffer. Therefore the solutions identified by the Phase II architecture are not precisely equivalent to the true ideal solution of $\omega_{Avg} = 0.5$ and $\omega_{Min} = 0.5$. As may be seen in Figure 3.18, however, this architecture is able to identify nearby neighboring solutions which closely approximate the ideal solution. The behavior of the Phase II system represents a significant performance increase over the Phase I system and

suggests that a basic associative learning process based on in-flight experience is occurring.

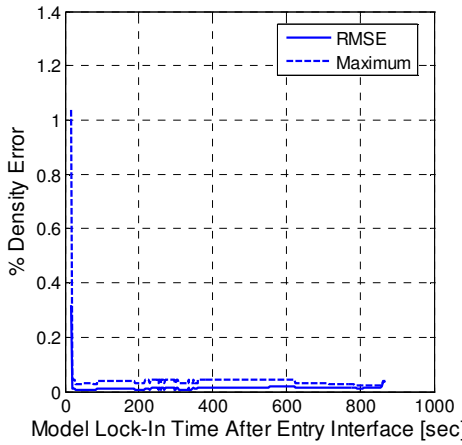


Figure 3.17: Forward-Looking Prediction Accuracy

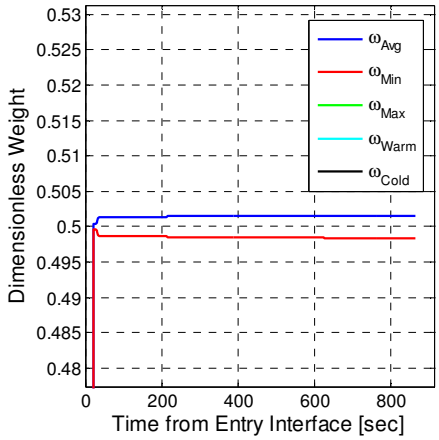


Figure 3.18: Detail View of Echo Weight Vector History

3.3.7 Phase II Dissimilar Testing Results

Up to this point, all prediction scenarios have utilized ideal test signals derived from the MCD atmosphere data, the same data source used to create the on-board neural ensemble. In order to test the Phase II architecture with a higher level of fidelity, a dissimilar atmosphere derived from a completely different Martian climate simulation is required. The intent is to create a MarsGRAM testing atmosphere with similar physical parameters as used in the MCD but utilizing a completely separate climate simulation

whose equations of motion followed an independent developmental history. Recall from Section 2.3 that four parameters, the suspended dust distribution, incident solar flux, time, and topography are the most basic inputs to any atmosphere model used herein. The topography over the entry corridor has already been accounted for in the neural ensemble modeling process. The default MarsGRAM atmosphere was modified to match the Julian entry date with the value used in the MCD simulations (see Table 2.1).

The solar flux incident on the thermosphere is often approximated by the 10.7 cm wavelength full-disc solar emission as it is considered a general indicator of solar activity. Commonly abbreviated as F10.7 and expressed in units of 10^{-22} W m^{-2} hz^{-1}, this flux value for the average solar activity models in the MCD is approximately 130.[86] The F10.7 value in MarsGRAM was also set to 130 to match the MCD nominal value.

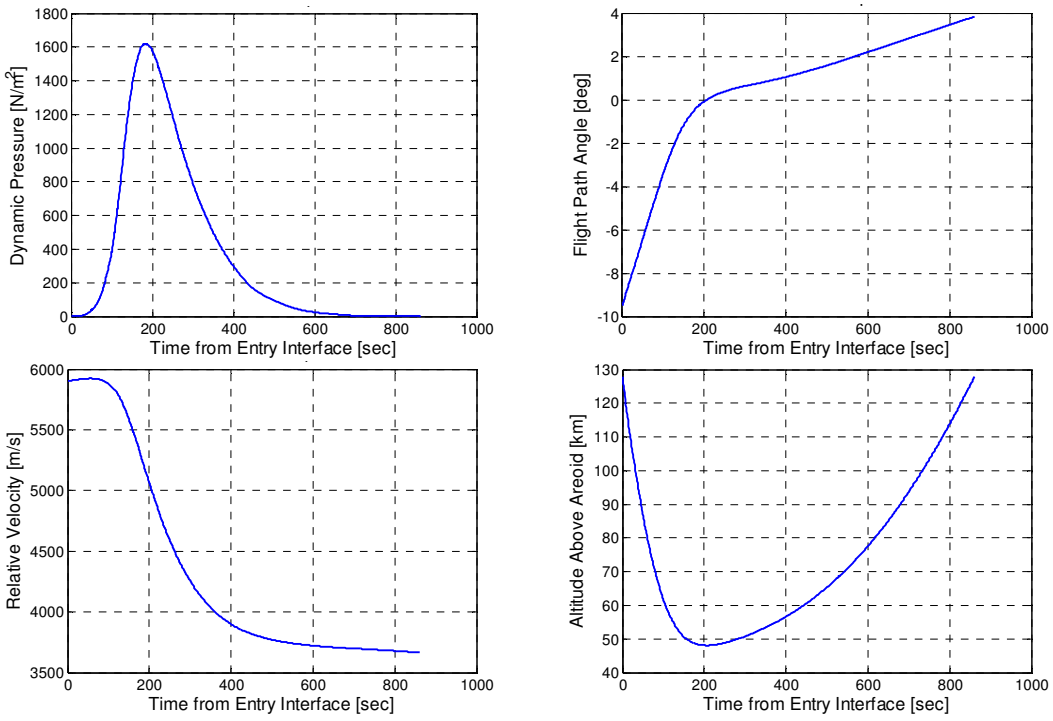


Figure 3.19: Aerocapture Reference Trajectory Using Dissimilar MarsGRAM Atmosphere

Lastly, the dust distribution in MarsGRAM was set to a programmatic default value which mimics a seasonal variation in dust optical depth as observed by the Viking

missions.[44] The dust properties are not exactly matched to the MCD in order to provide a stringent testing scenario. Because the Viking dust parameterization should reflect average or near-average conditions on Mars, the Phase II PENS system should be able to combine the members of the ensemble to mimic the Viking dust behavior. The baseline aerocapture simulation was rerun in POST2 in order to properly capture the impact of the newly defined MarsGRAM atmospheric model on the trajectory. The resulting trajectory (Figure 3.19) is very similar to the original baseline trajectory used for Phase I and early Phase II testing. The nominal MarsGRAM density profile used for the trajectory in Figure 3.19 is found in Figure 3.23 where it is compared to the ensemble members. The most notable changes are a slight increase in periapsis altitude from 46 km to 48 km and a small decrease in bank angle magnitude from 175.2° to 175° from the vertical.

In order to evaluate the prediction efficacy of the PENS method several contemporary density estimation methods were also computed for comparison. These schemes have either been proposed in the open literature or are close derivatives of previously proposed schemes. These schemes are described below and are also described in Table 3.1 for convenience. The first scheme is an exponential atmosphere model taken from the work of Perot and Rousseau[25] of the form

$$\rho_{LS} = e^{b} e^{a(h-h_o)} \tag{3.15}$$

where h is the current vehicle altitude, h_o is an arbitrary reference height, and the parameters a and b are computed using a least squares fitting approach based on ADS measured data. The parameters a and b are computed in two ways; the first method utilizes all of the measured density data from entry interface up to the current location of the vehicle. The second approach only uses the most recent 30 density measurements to permit some form of adaptation to developing atmospheric conditions and to prevent early atmospheric measurements from biasing the density predictions.

The second group of density prediction schemes is based on the work of Cerimele and Gamble[10] who defined an aerocapture guidance scheme commonly referred to as

HYPAS (or Hybrid Predictor-Corrector Aerocapture Scheme). HYPAS nominally uses a table look-up in order to estimate the density at the current location of the vehicle. In order to account for variations in density from the nominal atmosphere, HYPAS computes a table multiplier derived from density measurements. The model used by Cerimele and Gamble is implemented in three variations:

1. Instead of selecting an arbitrary nominal atmosphere, the exponential model defined in Equation 3.15 is used in place of the nominal table look-up. The HYPAS-derived density model becomes

$$\rho_{K-EXP} = K\rho_{LS} \quad \text{where} \quad K = \rho_{ADS} / \rho_{LS} \tag{3.16}$$

   In order to maximize the predictive ability of the method, the value of the density multiplier K is recomputed for each density prediction.

2. Like the exponential model of Equation 3.15, the HYPAS density model is also computed using only the most recent 30 density measurements. For this short HYPAS model, the multiplier K is computed using the short Least Squares model to maintain algorithmic symmetry.

3. Lastly, an assumed table model is used where the predicted density is computed with:

$$\rho_{K-Table} = K\rho_{Table} \quad \text{where} \quad K = \rho_{ADS} / \rho_{Table} \tag{3.17}$$

   This final model most closely matches the original Cerimele-Gamble approach and utilizes the average ensemble member as the tabular model.

For all of the above models utilizing the multiplier K, the multiplier is recomputed at each density prediction to allow rapid adaptation to evolving density trends. Historical estimators have typically been implemented with a low-pass filter but given the rapid adaptability of the PENS estimator, the low-pass filter for all multipliers was not used in order to provide a balanced comparison.

Table 3.1: Summary of Density Estimators

Short Name	Full Name	Model Form	Measured Data	References	Comments
PENS	Plastic Ensemble Neural System	$\rho_{PENS} = \omega_e \rho_{echo} + \sum_{i=1}^{n} \omega_i \rho_i$	From entry interface to current location	75	New
KPENS	State-Corrected PENS	$\rho_{KPENS} = K\rho_{PENS}$ $K = \rho_{ADS} / \rho_{PENS}$	From entry interface to current location	75	New
Exp	Exponential Model	$\rho_{LS} = e^b e^{a(h-h_o)}$	From entry interface to current location	25	Contemporary
Short Exp	Short Exponential Model	$\rho_{LS} = e^b e^{a(h-h_o)}$	Last 30 density measurements only	25	Contemporary
KExp	State-Corrected Exponential Model	$\rho_{K-EXP} = K\rho_{LS}$ $K = \rho_{ADS} / \rho_{LS}$	From entry interface to current location	10,25	Contemporary
Short KExp	Short State-Corrected Exponential Model	$\rho_{K-EXP} = K\rho_{LS}$ $K = \rho_{ADS} / \rho_{LS}$	Last 30 density measurements only	10,25	Contemporary
Table	State-Corrected Table Model	$\rho_{K-Table} = K\rho_{Table}$ $K = \rho_{ADS} / \rho_{Table}$	Current Location	10	Contemporary

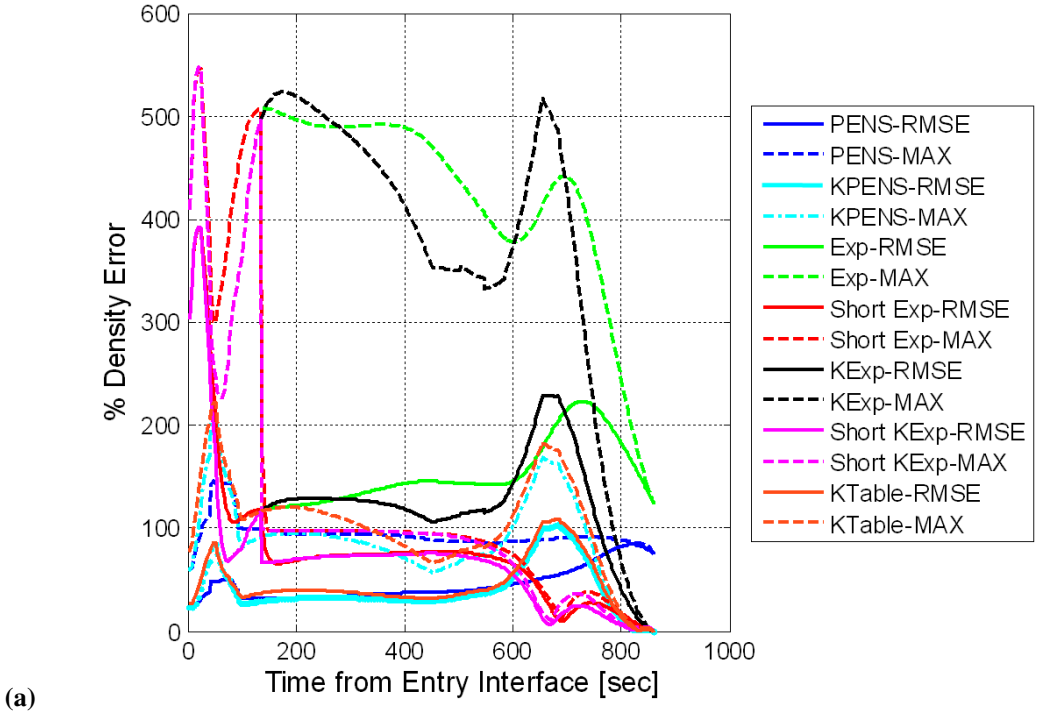
The original PENS density prediction model described in Equation 3.14 has no density multiplier to allow adaptation of the model to the measured density at the current location of the vehicle. This model relies fully on the physics of the underlying ensemble members to produce accurate long-term density predictions. In highly dispersed atmospheres, however, significant perturbations may exist. The underlying density field may have significant and long-lived noise components superimposed upon it which may generate significant prediction error even if the underlying model is correctly characterized. In order to obviate this possibility, a derivative of the PENS method referred to as KPENS, will also be examined which utilizes a density multiplier:

$$\rho_{KPENS} = K \sum_{i=1}^{n} (\omega_e \omega_{echo,i} + \omega_i) \rho_i \quad \text{where} \quad K = \rho_{ADS} / \rho_{PENS} \tag{3.18}$$

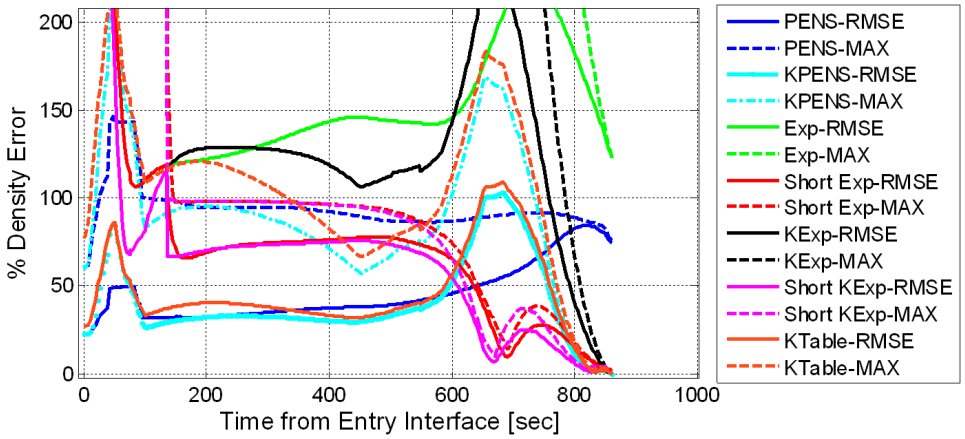
The density prediction performance of the Phase II PENS method using the dissimilar MarsGRAM atmosphere is presented in Figure 3.20. (A detailed explanation of this type of RMSE/Maximum error plot used throughout this chapter is available in Appendix B.) Like the analogous density error figures examined earlier (Figure 3.10 and Figure 3.17), the plot compactly summarizes the density prediction error statistics for over 400 individual prediction iterations. The contemporary density estimation methods just described are also included in the plot for comparison. The average density prediction error of the PENS method near periapsis (at 200 seconds) is approximately 32% while the KPENS method produces an error of 32.5%. Therefore, if no further prediction updates are made from periapsis to atmospheric exit, the PENS (KPENS) expected error at any point would be no greater than 32% (32.5%) with a maximum prediction error of approximately 95% (95%). This is in contrast to the contemporary methods which all have larger average and maximum prediction errors at periapsis. In fact, the PENS method produces predictions with smaller prediction errors than all of the contemporary methods for the majority of the trajectory. Only two of the contemporary methods out-perform the PENS/KPENS methods far beyond periapsis near 600 seconds. Inspection of Figure 3.19 reveals that this occurs at an altitude of roughly 80 km and at a very low dynamic pressure, far smaller than the peak pressure near periapsis. Because the dynamic pressure is so small at this point, the vehicle no longer has sufficient control authority to make gross trajectory adjustments using aerodynamic forces alone. Accurate density predictions must be available to the guidance system at a significantly earlier point, preferably near or before periapsis, in order to be useful for accurate path planning.

It should also be noted that both of the contemporary models which outperform the PENS method near 600 seconds use only the last 30 ADS density measurements. The analogous exponential $\rho_{K\text{-}EXP}$ and Least Squares ρ_{LS} models (the black and green curves respectively in Figure 3.20.a) which utilize all of the ADS density measurements exhibit

markedly worse prediction performance. Observe in Figure 3.20.a that, initially, the short and long models coincide. The short models are initially set equal to the long models at



(a)



(b)

Figure 3.20: PENS Prediction Accuracy Using Dissimilar MarsGRAM Atmosphere

start-up until 50 ADS density measurements have been gathered near 130 seconds. This is to prevent early high-altitude density measurements which typically have a higher degree of uncertainty (see Figures. 1.1-1.2) from biasing the prediction results. After 50 measurements have accrued, the 30 most recent measurements are used for prediction

and the short models become independent of the long models. At this point the predictive accuracy of the short models markedly improves, a trend which persists for the remainder of the trajectory.

The consistently superior performance of the short (30 data point) models relative to their long model counterparts suggests that the exponential approximation of the Martian atmosphere is only locally valid. This conclusion is significant because it implies that exponential models may not provide guidance systems with sufficiently accurate density predictions over great distances. If the vehicle guidance is not provided with enough notice concerning upcoming density trends, robust path prediction becomes quite challenging. It is this failure to provide accurate density approximations far ahead of the vehicle that leads to failure modes such as guidance lag, controller saturation, and misprediction of control authority highlighted earlier (see Section 1.3).

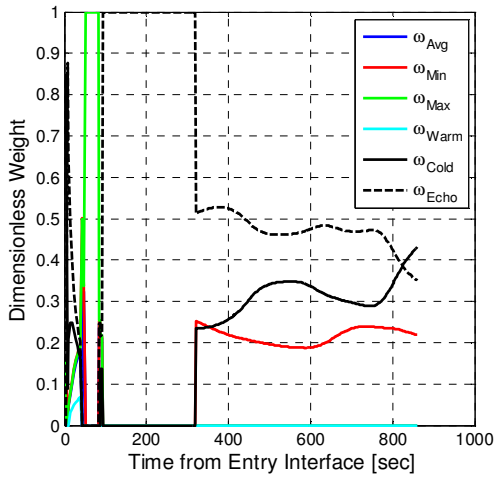


Figure 3.21: Full Model Ensemble Weight Vector Time History

Another feature of note is the consistency of the PENS density prediction error over the majority of the trajectory. Despite considerable variations in both the full model weight history (Figure 3.21) and the ensemble echo weight history (Figure 3.22) during the convergence period, the PENS prediction error remains relatively consistent for the remainder of the trajectory. This demonstrates that the adaptive capability of the PENS

method is successfully able to alter the underlying density model dynamics to successfully predict the upcoming density trends. This is in direct contrast to the four contemporary models based on the omnipresent exponential atmosphere assumption whose underlying dynamics are static. Because the contemporary models are less able to adapt to the observed density trends, the variations in the predictive error signal becomes more pronounced.

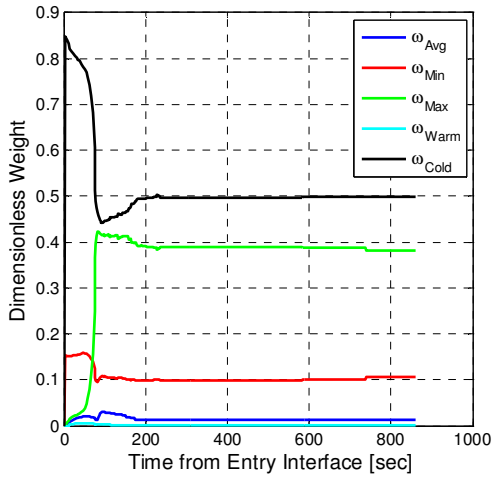


Figure 3.22: Ensemble Echo Weight Vector Time History

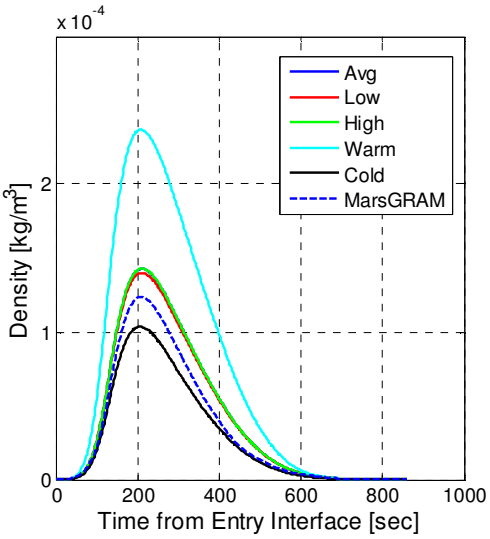


Figure 3.23: Ensemble & MarsGRAM Histories Along Dissimilar Reference Trajectory

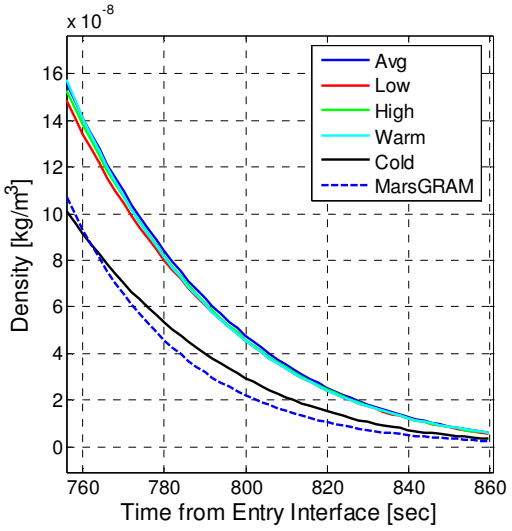


Figure 3.24: Ensemble & MarsGRAM Density Histories At High Altitude Near Atmospheric Exit

The predictive accuracy should continually improve as the PENS algorithm observes and learns from the atmospheric density trends. However, this appears not to be the case for the present simulation example using the dissimilar atmosphere model. Note that as time progresses in Figure 3.20, the average prediction error increases especially towards the end of the trajectory. This increase in average error is largely generated by the high altitude dissimilarity between the MarsGRAM and MCD density trends. Despite the fact that the MarsGRAM density model appears to be well within the bounds defined by the neural ensemble (see Figure 3.23), late in the trajectory near 760 seconds, the MarsGRAM density trend actually drifts outside the prediction basis (Figure 3.24). It is interesting to note, however, that the MarsGRAM thermosphere in this region near 760 seconds (corresponding to altitudes roughly 100 km and above) is well described by an exponential atmosphere as is visible in Figure 3.20. This clearly differs from the MCD thermospheric predictions which utilize a different model parameterization which is poorly described by a simple exponential model.

The difference between MarsGRAM and MCD makes perfect prediction impossible because the predicted density trend, as a weighted average of the ensemble,

must always be contained by the ensemble. This generates a non-reducible error at the end of the trajectory which appears in all of the PENS density predictions. However, because this error is small in magnitude and late in the trajectory, a net reduction in density prediction error is still possible. The presence of this non-reducible error explains why the average prediction error begins to increase towards the end of the trajectory while the maximum error remains relatively consistent. By the last prediction of the trajectory, the average and maximum error are one-and-the-same and the average and maximum error curves merge. Therefore, the increase in average prediction error is not due to a failure of the associative learning capability of the PENS algorithm but rather due to the differences in the atmosphere models used for testing.

Note the performance of the table model is respectable near periapsis (~200 sec), producing an average prediction error which is only 8.5% higher than the PENS/KPENS estimators. The table model becomes locally more accurate than the PENS estimator during the outbound leg of the trajectory between 338 and 564 seconds. However, inspection of Figure 3.20.b shows that the KPENS estimator is 2-4% more accurate in the same window. Note also the similarity in the prediction accuracy (the RMSE and the maximum) trends produced by the KPENS and table models. This similarity is completely due to the use of the multiplier K. In fact, both PENS models are identical save for the use of the multiplier in the KPENS version of the estimator. Therefore, the KPENS estimator would be identical to PENS if K = 1.0.

The multiplier K permits the prediction models to closely track the density trend near the entry vehicle but the increase in near-term prediction accuracy is not without cost. Inspection of Figure 3.20.b reveals that both the Table model and KPENS estimator exhibit higher error when predicting from high altitude. Early in the entry near 50 seconds, the average prediction error rises to approximately 75%, an increase of almost 50% over nominal. Recall that any given point in the figure summarizes the density prediction error from that given point to atmospheric exit. Therefore, if no further density

predictions are made, the expected prediction error will remain near 75% for the remainder of the trajectory. Likewise, the prediction error rises again at high altitude near the end of the trajectory to over 100% average error at 675 seconds, an increase of more than 60% in only a few seconds. This increase in prediction error is due to the large spread in observed atmospheric densities. At high altitude, very low density values permit the production of large K values from small absolute density differences. At lower altitudes, the reverse is true wherein higher densities may produce smaller K values despite the fact that large absolute differences may exist. For example:

At high altitude:

$K = \frac{\rho_{ADS}}{\rho_{PENS}} = \frac{2(10^{-7})}{10^{-7}} = 2$ despite the fact that the absolute difference is only 10^{-7} kg/m^3

At low altitude:

$K = \frac{\rho_{ADS}}{\rho_{PENS}} = \frac{1.5(10^{-5})}{10^{-5}} = 1.5$ despite the fact that the absolute difference is now much larger at $5(10^{-5})$ kg/m^3, 50 times the high altitude density difference.

However, this apparent limitation of the multiplier K may not, in fact, militate strongly against its use. Note that all the prediction models employing a multiplier approach zero error near atmospheric exit, a direct result of forcing the prediction model to closely follow the observed density trend. In addition, these prediction models must all be sequentially updated throughout the entry when implemented in an atmospheric guidance algorithm. This regular update process may completely counterbalance the long-term biasing effects of the multiplier. Given this possibility, as well as the prevalence of scalar multipliers in contemporary density prediction models, the KPENS model will be included in future experimentation.

As noted in Chapter 1, the impact of atmospheric density on the entry trajectory will diminish with altitude and will, at some point, become negligible. The precise entry and exiting altitudes where the density no longer has a significant effect on an

aerocapture trajectory is largely a function of the desired target orbit[87]. The target orbit dictates the entry and exiting flight path angles which largely determine the length of time the entry vehicle spends in the highly variable upper atmosphere. Rapid transits through this region, such as those produced by higher energy post-aerocapture target orbits, will tend to limit the effects of the density variability on the aerocapture trajectory. Slower, shallower transits through the upper atmosphere, typically produced by lower energy target orbits, will be more susceptible to density variations in the upper atmosphere.

It should be noted, however, that even during high altitude flight where the density has little or no impact on the trajectory, measurement of local free stream density during the early phases of entry can still provide a valuable learning opportunity for the PENS algorithm. Note in Figures 3.21 and 3.22 that, prior to 100 seconds (which corresponds to an altitude of about 60 km according to Figure 3.19), the ensemble weights change significantly. This indicates that PENS is learning from the trends observed at high altitude. This high altitude information is used to construct the early PENS density models which are then refined as the vehicle approaches periapsis. Because the control authority of the entry vehicle reaches its peak near periapsis, the availability of an accurate density model at periapsis significantly reduces the risk of an incorrect and irreversible guidance decision. Therefore, all aerocapture trajectories examined herein permit the PENS algorithm to observe the free stream density trends even at high altitude.

3.4 Summary of Initial Concept Testing

The PENS method has been shown to produce improved density forecasts with superior error performance over contemporary density prediction techniques. Existing density forecasting schemes for real time applications in vehicle guidance systems often rely on simple analytic or tabular atmosphere models.[6,10-17] Such models may not

adequately characterize the current atmospheric state due to an inflexibility in the underlying density model dynamics. This inflexibility often generates significant forecast error with increasing distance from the entry vehicle. This observation is supported by the work above which demonstrates that simple density models are often only regionally valid in the Martian environment. These models may not provide the guidance method with sufficiently accurate forecasted density trends with enough lead time for safe path planning and prediction. If not remediated in some manner, the high degree of atmospheric uncertainty present in the on-board density models increases the uncertainty in the guidance decision making process. This then places a significant burden on the guidance system which must remediate vehicle state errors due to atmospheric dispersions via active course correction. Not only is this strategy operationally expensive but it also places the vehicle at risk of entering one or more of the discussed guidance failure modes.

CHAPTER 4

APPLICATION TO AEROCAPTURE GUIDANCE

This chapter is dedicated to addressing the fourth supporting research question which is the development of a proactive entry guidance scheme which is capable of implementing the PENS density estimator. First, a survey of available entry guidance schemes is conducted which reveals that a Numerical Predictor-Corrector (NPC) is the best suited platform for testing the impact of the PENS estimator on aerocapture guidance. In the first sub-section of this chapter, an NPC guidance scheme which implements the PENS estimator is introduced and the guidance sub-models are enumerated and described. The second sub-section of this chapter is devoted to discussing possible sources of error and the approaches used for mitigating or simulating these errors. In the third sub-section, the NPC guidance scheme is subjected to verification and validation testing. Verification testing will reveal that the behavior of the guidance-implemented PENS estimator matches the behavior of the original standalone algorithm used for PENS development in Chapter 3. Validation testing will reveal that the PENS-augmented guidance scheme is capable of producing trajectories which exhibit a high degree of agreement with results available in relevant Mars aerocapture literature.

4.1 Bank Modulated Guidance Schemes

As discussed in Section 2.7, the bank angle of the entry vehicle is used to modulate the lift vector in order to control the vehicle in the atmosphere. Many such bank modulated entry guidance schemes exist and the pertinent question becomes which type of banking guidance scheme is best suited to implement the PENS density estimator. In general, banking entry guidance schemes can be divided into two broad categories; reference path tracking schemes and predictive path planning schemes.[88]

Reference path tracking schemes such as those implemented aboard the Apollo Command Module[27,28] and the Space Shuttle[26] attempt to track a preplanned reference trajectory which is created using nominal conditions and nominal models. The role of the guidance system is to correct for any errors which result from the departure of the actual entry conditions or models from the ideal nominal. Pertinent to the work herein, if the atmospheric model used to generate the reference trajectory has significant errors (i.e. a high degree of uncertainty) then the guidance must attempt to mitigate the effect of those errors by correcting the trajectory in real time.

Because there is no need to compute a complete trajectory on-board in real time, many (but certainly not all) reference path guidance systems require little numerical integration, iteration, or other arithmetically complex operations. This leads to a high degree of algorithmic simplicity which is often cited as the inherent advantage of this class of guidance scheme. This simplicity often does lead to a high degree of algorithmic robustness because computational redundancy and other fault tolerance measures are often computationally affordable. However, at Mars this algorithmic robustness does not translate to operational robustness (e.g. reliable performance in the face of large atmospheric dispersions).

Reference path schemes often achieve algorithmic simplicity through the application of simplifying assumptions applied to the dynamic equations of motion and the guidance models, especially the atmosphere model. A common, convenient, and seemingly inexpensive assumption made by many Mars reference path schemes[10,12-14,16,26] is to utilize the common exponential atmosphere model. However, several studies[14,17,19,20] of reference path methods at Mars report that significant guidance faults and failures occur due to atmospheric dispersions (i.e. differences between the modeled atmosphere and the true atmosphere).

As discussed in Section 1.4, some reference path schemes, such as the modified Apollo guidance used on MSL[14,17], forego the use of any atmospheric model. Instead,

these guidance systems track departures of the sensed lift and drag accelerations from a reference profile. The influence of the atmosphere, then, is indirectly treated by compensating for errors in the drag acceleration. While this approach seemingly bypasses the issue of atmospheric prediction altogether, it can quite easily lead to prevalent failure modes (see the failure modes discussion in Section 1.3). Pre-mission studies[17,20] of the MSL guidance scheme have concluded that the drag error caused by density dispersions is the principle statistical contributor to landing site miss distance at the parachute deployment point. Stated another way, the differences between the nominal predicted atmosphere used to construct the reference trajectory and the true atmosphere are the primary cause of final state targeting error.

Several reference path schemes, like Shuttle guidance[26], include the ability to compute an approximated reference trajectory in real time. However, the equations of motion are often greatly simplified using a series of assumptions (typically involving the nature of the atmosphere). If any of these assumptions do not hold, then the produced reference trajectory may be quite difficult (or impossible) for the vehicle's control system to follow. The vehicle is then forced to retroactively correct for dispersions after they have already impacted the trajectory. This, in turn, again forces the guidance towards prevalent failure modes.

At Earth, reference path methods such as those used by the Space Shuttle and Apollo programs are justifiably applicable. In both cases, reference path methods were suitable since alternate means of remediating the cumulative effects of density uncertainty were utilized. For Apollo, the recovery architecture permitted the vehicle to land in a relatively wide area whereupon search and rescue personnel would locate the vehicle. At Mars, future exploration architectures may require precision landing or the use of precise target orbits. Therefore, accepting wide 3σ final state dispersions like Apollo may not be possible. For the Space Shuttle, its high L/D provided greater control authority to correct for the cumulative effects of atmospheric density uncertainty prior to

landing. The use of higher L/D has already been explored for Mars and other environments (see Section 1.3). It has been found that higher L/D may actually introduce significant control difficulties and does not immediately ensure a successful entry. In addition, higher L/D shapes typically produce mass penalties due to packaging inefficiencies.

Thus, the strength of reference path methods is also the source of their collective weakness at Mars. Namely, the algorithmic simplicity achieved through the application of simplifying assumptions often leads to decreased accuracy and an increase in the demands on the guidance system. This generates increases in both failure rates and final state targeting errors in the highly dispersed Martian atmosphere. Thus, the effective compromise for most Mars reference path schemes is to trade accuracy and reliability for simplicity and speed.

The other major class of entry guidance schemes, predictive path planning methods, calculates a new trajectory from the current state to the desired final state using onboard prediction techniques. The key difference between reference path methods and path prediction is that the former tracks a reference profile which may or may not be recomputed in real time. The latter iteratively recomputes the trajectory by changing the control variables until the estimated final state is equal to the desired final state. Thus path prediction methods directly produce a control signal such as a bank angle-vs-time history as opposed to reference path schemes which produce a reference trajectory as the primary output.

Predictive guidance systems may be further divided into two mutually exclusive subdivisions; analytical and numerical prediction schemes. Common analytical path prediction methods[10,15,16,18] often utilize closed-form (or nearly closed-form) equations requiring little or no numerical integration to produce the estimated trajectory. This is often accomplished by introducing simplifying assumptions into the equations of motion and guidance models. Again, the atmosphere model is a prime target for discrimination

and often is reduced to an exponential atmosphere approximation. To reduce computational overhead, the equations of motion are often simplified further using analytic integration and algebraic manipulation following introduction of the analytic atmosphere model. The atmosphere, then, becomes hard-coded into the guidance equations and becomes quite challenging to replace during the entry. This is the case for the successful and prevalent Cerimele-Gamble[10] HYPAS aerocapture guidance scheme.

While the application of simplifying assumptions reduces the computational overhead just as in reference path schemes, the limitations are generally the same. In the present study, the ability to update/replace the guidance atmosphere model during the entry is a necessity. Most reference path schemes and analytical path prediction methods, however, do not offer this opportunity. Numerical path prediction schemes, on the other hand, do not have this limitation. In fact, for this class of guidance system, replacement of the on-board atmosphere model during the entry is completely possible. This makes numerical path prediction schemes an attractive option for PENS implementation.

Common numerical path prediction methods[11,88-92] integrate the equations of motion in real time while typically refraining from significant simplifying assumptions. Because on-board numerical integration is employed, replacement of the atmosphere with an updated model is possible because the atmosphere is not hard-coded into the guidance system. Each time the guidance is called, the equations of motion are numerically integrated from the current state to the final state using a guessed control signal such as a bank-vs-time profile. If the predicted final state is not equivalent to the desired final state, the control signal is updated and the process is iterated until the final state is equal to the desired state.

Because numerical guidance schemes require fewer assumptions, they have the potential to be highly accurate. Kluever[88] compared the performance of an analytic reference path scheme (MSL guidance[14]) with a numerical path prediction scheme for Mars precision landing. He concluded that while “predictive guidance delivers more

accurate terminal conditions compared with the reference-path guidance, it relies heavily on accurate onboard models of the vehicle and Martian atmosphere." This finding of Kluever suggests that a numerical path prediction scheme could produce more accurate final state targeting if improvements could be made in the accuracy of the on-board models such as the atmosphere. This notion further supports the use of a numerical prediction guidance scheme for PENS implementation. If the PENS algorithm does enhance the accuracy of the predicted density model as theorized, then the final state accuracy gains could become appreciable.

Further evidence of possible accuracy gains come from Putnam et al.[93] and Bairstow and Barton[94] who have evaluated the performance of a numerical predictor-corrector guidance system, PredGuid, for use on the Orion entry vehicle at Earth. PredGuid, while based on the Apollo entry guidance algorithm, has been augmented with a numerical predictor-corrector (NPC) to improve landing accuracy. The assumptions and heuristics used in the original Apollo entry guidance were found to produce inadequate landing accuracy for lunar-return entries. To improve the accuracy of this heritage reference path method, a NPC guidance scheme originally designed for aerocapture missions was used to replace portions of the original guidance. Both studies concluded that the addition of the NPC greatly improved the accuracy of the Orion entry guidance.

The fact that both PredGuid studies were able to realize performance gains at Earth while Kluever did not realize gains using an NPC at Mars is not surprising. The performance of an NPC is strongly dependent on knowledge of the atmosphere as Kluever noted. As already stated, the terrestrial atmosphere is much better characterized by the meteorological community than that of Mars. Also, the perturbations which may be expected at Earth are better understood and less problematic due to the greater control authority afforded by terrestrial atmospheric densities. Ultimately, the uncertainty inherent in the table look-up model used by PredGuid at Earth is smaller than the uncertainty associated with current Martian atmosphere models. Due to the highly

variable nature of the Martian atmosphere, this uncertainty is likely to persist even when climatologists reach a better understanding of the Martian atmospheric dynamics.

In summary, the use of an NPC guidance algorithm for atmospheric entry with an accurate on-board atmosphere model has been found to produce excellent final state targeting performance, exceeding that of classical reference path methods. Therefore, in addition to the ability to update the guidance atmosphere model, the possible performance gains suggested by the relevant literature further justifies the selection of a NPC scheme for PENS implementation.

4.2 Numerical Predictor-Corrector Guidance

The NPC guidance scheme selected for PENS implementation is similar to the PredGuid guidance scheme selected for use aboard the Orion entry vehicle. The NPC guidance scheme utilized herein was originally developed at the NASA Langley Research Center in support of the Aeroassist Flight Experiment[95] program. This algorithm has been modified, updated, and reconfigured to support several studies[89,91,92] of aerocapture at Mars. This atmospheric guidance algorithm is implemented in a three degree-of-freedom (3-DOF) version of the Program to Optimize Simulated Trajectories (POST2)[96]. The simulation environment (see Figure 4.1) is composed of two primary loops, both residing in the POST2 simulation framework. A simulated flight software bus uses on-board timers to activate pertinent subsystems of the spacecraft such as the guidance system, sensors, and actuators at specified intervals. This flight software, originally developed for the Entry, Descent, and Landing Systems Analysis Program[87,97-99] (EDLSA), is adapted for use here.

The outer brown loop in Figure 4.1 represents the real world Martian environment through which the entry vehicle must travel. The inner loop in blue represents the trajectory simulation environment on-board the entry vehicle which is used by the NPC guidance to predict the vehicle's flight path. The 3-DOF equations of motion are

implemented in the inner loop by the guidance system using a set of guidance sub-models (e.g. atmosphere model, gravity model, etc. For complete sub-model details, see Section 4.2.2). When called by the spacecraft bus, the inner loop conducts a series of on-board trajectory simulations from the vehicle's current state to the atmospheric exit conditions. If the predicted final orbit is not equal to the desired final orbit (red X in Figure 4.1), the control variables are updated and another trajectory prediction is made until the predicted final state matches the desired final state (green check mark in Figure 4.1). The guidance then uses this converged trajectory to issue commands to the control system to steer the vehicle.

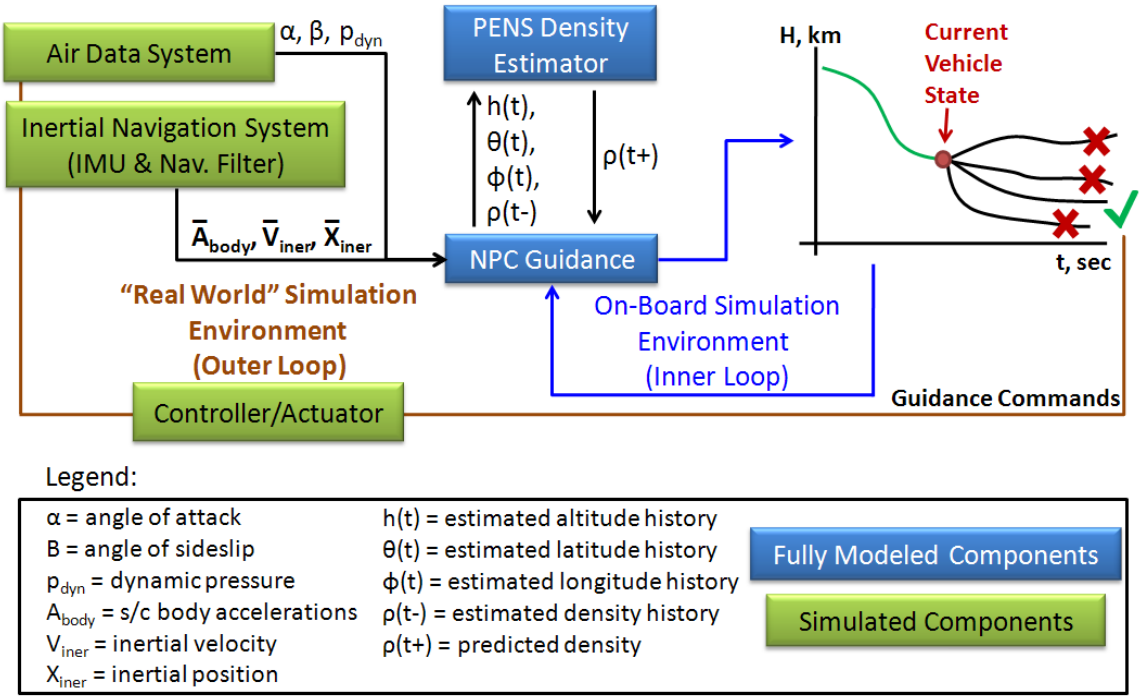


Figure 4.1: PENS-Augmented Numerical Predictor-Corrector Guidance Simulation Environment

The spacecraft bus operates at 10 Hz and, once per cycle, the bus commands the on-board sensor systems to collect and store information concerning the atmosphere and the current state of the vehicle. The vehicle sensor suite consists of two subsystems, the Air Data System (ADS) and the Inertial Navigation System (INS). The ADS estimates

atmospheric information based on pressure data, specifically the angle of attack α, the sideslip angle β_{ss}, and free stream dynamic pressure p_{dyn} through the use of flush-mounted pressure ports in the vehicle's external structure. The ADS may only collect atmospheric information from the vehicle's current flight environment. Unlike remote sensing systems, the ADS cannot collect atmospheric data from regions of the atmosphere that the vehicle has not yet transited.

The INS uses the on-board accelerometers and gyroscopes to calculate data products relating to the current dynamic state of the entry vehicle. The INS estimates and saves the vehicle sensed body acceleration vector, A_{body}, the inertial velocity vector, V_{iner}, and the inertial position vector, X_{iner}. Like the ADS, the INS saves the history of these quantities from entry interface up to the current location of the vehicle. Both the ADS and the INS are simulated components whose accuracy and performance have been tuned to replicate the typical performance of similar systems in the literature (see Section 4.2.2 for more information). Individual components such as pressure transducers or accelerometers are not modeled for simulated subsystems.

The only remaining simulated component, the controller/actuator model, also resides in the outer loop. This system receives the guidance commands which, in this case, are the desired bank angle magnitude and bank angle reversal time schedule. The controller/actuator module is simulated using a theoretical model called a pseudo-controller. The pseudo-controller, used extensively in previous entry studies[97,99-102] at Mars, simulates the vehicle attitude time response generated by the controller and actuator dynamics. In the absence of a pseudo-controller model, changes in the vehicle's attitude would become instantaneous rather than occurring over some finite time interval. The time response signal generated by the pseudo-controller has been designed to replicate the response of 6-DOF simulations of typically configured Mars entry vehicles. Thus, use of the pseudo-controller permits a 3-DOF simulation to approximate a full 6-DOF simulation by accounting for effects of the vehicle's rotational dynamics on the

simulation timeline. This, in turn, reduces the computational overhead associated with each trajectory simulation and also avoids the necessity of specifying a detailed vehicle inertia tensor and actuator subsystem. Instead, only the maximum vehicle angular acceleration and maximum angular velocity (5 deg/s^2 and 20 deg/s respectively, derived from the EDLSA[87,97-99] project) need be specified.

The remaining components of the simulation architecture (the NPC guidance system and PENS subsystem in blue in Figure 4.1) are fully modeled components whose behavior is of prime interest to the present study. At entry interface, the spacecraft bus places the guidance system in a standby mode and the vehicle bank is commanded to hold a nominal pre-entry bank magnitude which is 90° for the present study. This bank magnitude is in the middle of control margin between full lift up (at 0°) and full lift down at 180°, providing the maximum control authority to the vehicle at guidance initiation. Once the sensed acceleration reaches 0.05 Earth G indicating that the vehicle is now in the atmosphere, the guidance inner loop activates for the first time. The bus then calls the inner loop every 0.5 seconds (2 Hz) until the vehicle exits the atmosphere at 128 km. A detailed view of the computational process inside the inner loop may be found in Figure 4.2. The subsystems associated with black arrows (the density estimator and green sensor block) are not part of the inner loop but their data products directly support the inner loop computations and are included for clarity. The inner loop receives estimated state vector information (altitude, position, velocity, etc) from the INS and utilizes this information to conduct the on-board trajectory prediction from the current location to atmospheric exit. Prior to conducting each on-board trajectory simulation, the guidance selects a constant bank angle magnitude which it then maintains throughout the simulation. Bank angle reversals are still performed to prevent large final orbit inclination errors, but the magnitude of the bank angle, β, remains constant. The inner loop utilizes the sensor and density estimator information to create/update the guidance sub-models such as the atmosphere, aerodynamics, etc. (see Section 4.2.2 for a full description of the sub-

models). The results of the model evaluations are then used to perform the on-board trajectory propagation using the common 4th order Runge-Kutta method for numerical integration. The guidance propagates the trajectory up to the atmospheric exit altitude at which point the orbital elements are computed and compared with the desired orbital state. If the predicted apoapsis is not within the convergence tolerance, then a new bank angle is selected and the process repeats. The inner-loop iterates by performing another trajectory simulation until the predicted apoapsis is equivalent to the desired orbital apoapsis.

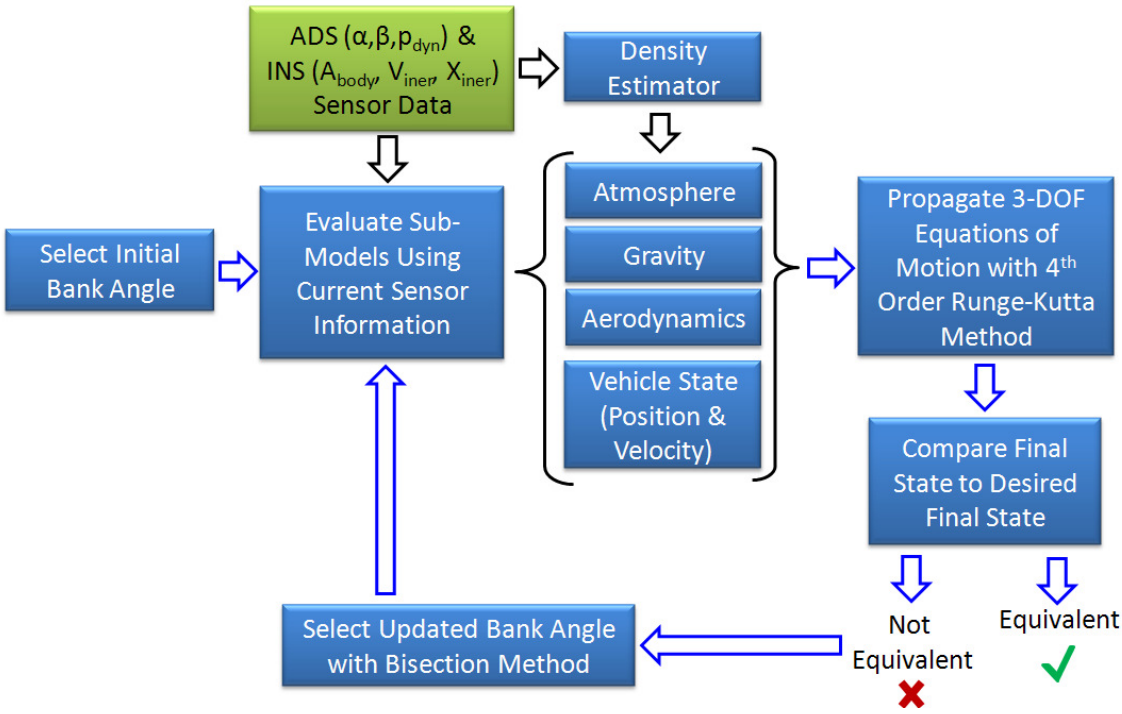


Figure 4.2: Detailed View of Inner Guidance Loop

The updated value of the guessed bank angle is determined using a bisection interval search routine. This algorithm, as implemented in the guidance, is a two phase root-finding method. In this case, the root the algorithm attempts to locate is the difference between the desired and actual orbital apoapses. At the root of this function, the desired and actual apoapses are equivalent and the inner loop may successfully terminate. In the first phase, the algorithm ensures that the root is "bracketed." As shown

in Figure 4.3, the algorithm requires that the objective function be continuous and that the initial two function evaluations have opposite sign. The intermediate value theorem[103] then ensures that at least one root of the function is contained (i.e. "bracketed") between the two bounding evaluations.

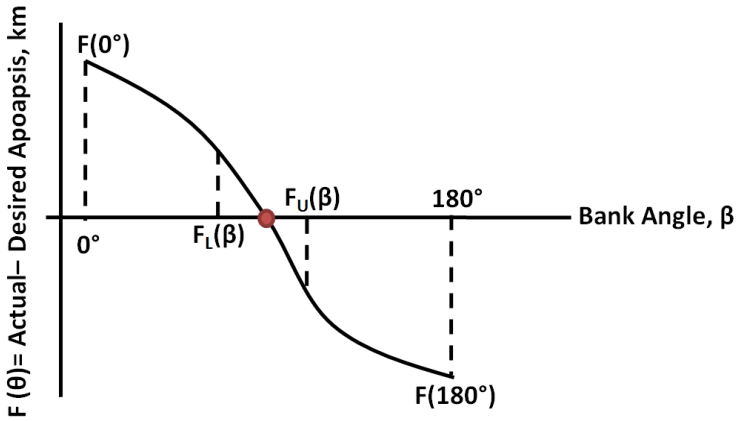


Figure 4.3: Inner Loop Bisection Algorithm

In order to bracket the root, the algorithm steps through the possible bank angles in 5° increments starting at the initial guessed bank angle. In each of the 5° increments, the first phase evaluates both the upper and lower bank angles (see Figure 4.3) in order to determine if the current upper and lower limits bracket the root. If the upper and lower limits have opposite signs, then the root is bracketed and the second phase may begin. If they do not, then the algorithm moves on to the next 5° increment and tries again. Because the bank angle magnitude β must be contained between 0-180°, the upper and lower bounds could easily be set to 0° and 180° respectively and the first phase of the algorithm could be skipped altogether. However, this approach is not used for two reasons. First, the bisection algorithm can take many function evaluations to begin homing in on the root, and the first phase is designed to prevent excessive evaluations by first searching the region around the initial guessed bank magnitude. Second, this

approach explicitly checks to ensure that a root is actually contained on the bank interval [0°, 180°]. If a root is not contained in the interval, this indicates that no feasible solution exists and the guidance commands either full lift up or full lift down (whichever minimizes the apoapsis miss distance) in an attempt to rescue the vehicle and save the mission.

Once the first phase is completed, the second phase begins in which the location of the bounded root is determined by finding the value of the function (by running an on-board trajectory simulation) halfway between the upper and lower bounds. This new function evaluation will be used to reevaluate the location of the root. The half interval containing no root is discarded while the other half interval, now with tighter upper and lower bounds, is retained. This halving process repeats until the location of the root is determined to within a convergence tolerance which is 5 km for all simulations herein. This value of 5 km was chosen as it was found in preliminary testing to offer satisfactory orbit targeting performance while preventing excessive of roll reversals.

As with the majority of bisection algorithms, this root finding approach is quite robust. It often identifies the feasible bank angle within 6-12 function evaluations, though occasionally more are required, particularly if the initial guess is very far from the root. The initial bank angle guess for the current inner loop call is determined by the converged value of the bank angle from the last inner loop call. As mentioned, the original bank guess at the first inner loop call just after entry interface is 90°. Between calls to the inner loop, the guidance commands the pseudo-controller to maintain the constant bank angle magnitude identified by the bisection algorithm and commands bank reversals on a timer using the reversal times saved from the converged trajectory prediction.

These bank angle reversal times are determined by the on-board trajectory simulation via the lateral guidance logic. The original lateral guidance logic described in Section 2.7 is replaced here in favor of a more general approach which is less dependent

on arbitrary tuning and better suited to this NPC scheme. The inclination error is a function of orbital specific energy, ε determined by the functions in Figure 4.4. The guidance linearly interpolates inside these functions to determine the allowable inclination error at the current vehicle orbital energy. If the current inclination error is outside the maximum allowed error, then the guidance commands a bank angle reversal by flipping the sign on the bank magnitude β. Positive bank angles denote right-hand banks whereas negative bank angles indicate left-hand banks.

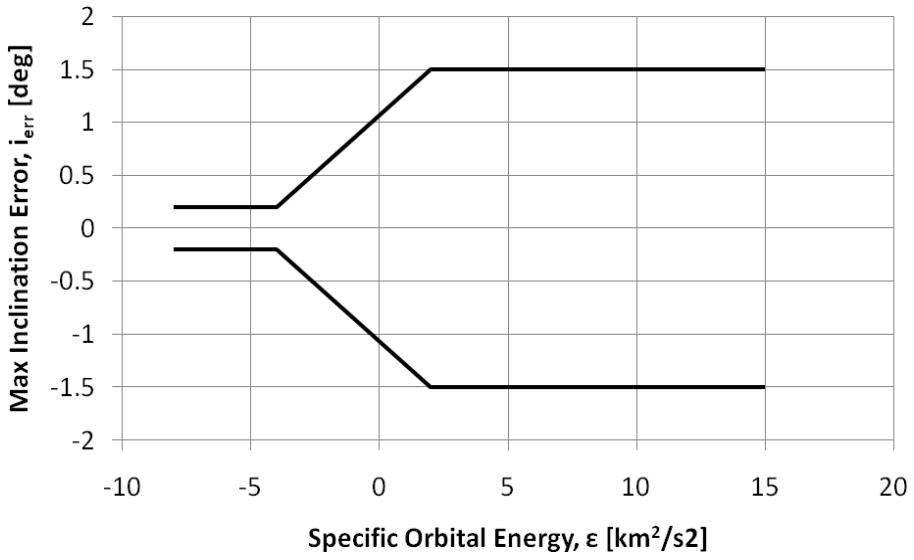


Figure 4.4: NPC Maximum Allowed Inclination Error Limits for Determining Bank Angle Reversals

If the current inclination error does not violate the maximum limits defined in Figure 4.4 outright, then the guidance examines the rate of inclination change. If the rate of change of the inclination error is high, indicating that the current orbital plane is rapidly diverging from the desired orbital plane, then the guidance artificially increases the current inclination error i_{err}. This approach ensures accurate orbital plane targeting by preventing excessive overshoot of the inclination error past the bounds in Figure 4.4 when the inclination rate of change is high. The rate of change of the inclination error di_{err}/dt is computed by determining the change in the inclination error, i_{err} over one

integration time step (one second in the on-board simulation). This rate is then added to the overall inclination error (i.e. $i_{err} = i_{err} + di_{err}/dt \cdot dt$) such that high rates inflate the inclination error. However, to limit the number of bank reversals, the rate term di_{err}/dt is limited to a maximum value of 0.2 deg/s. This limit prevents premature bank reversals (which place unnecessary demands the actuator and control systems) while still ensuring accurate orbital plane targeting.

4.2.1 PENS Implementation into NPC Guidance

The PENS density estimator receives position information collected by the INS; namely, the geocentric latitude, longitude, and geodetic altitude time histories from entry interface up to the current location of the vehicle. This position information is passed to the atmospheric neural networks (NNs) stored aboard the vehicle which are a compact representation of the ensemble models used for atmospheric forecasting (see Section 2.5). The NNs are used to construct the ensemble models along the vehicle's flight path from entry interface up to the current location. PENS also receives dynamic pressure information from the ADS and uses this data to estimate the atmospheric density by factoring in the velocity of the vehicle. (Recall the discussion of the free stream density computation in Section 2.2) This permits an estimated atmospheric density history along the vehicle's flight path to be constructed. At this point, the PENS algorithm has all the required information for a complete PENS iteration cycle to occur (for a complete list of PENS algorithmic steps, see Section 3.3.2). The PENS algorithm then constructs the density prediction equation (Equation 2.2) which defines the predicted density field ahead of the vehicle and passes the equation to the guidance system for use in the inner guidance loop. The PENS density model is updated at the guidance inner loop calling frequency such that the density model is always updated before the inner loop is activated. When the inner loop propagates the 3-DOF equations of motion, it utilizes the updated atmosphere model to produce the necessary density estimate required at each

integration step. As the accuracy of the density prediction model increases, so too does the accuracy of the predicted trajectory generated by the inner guidance loop.

All other historical density estimators interact with the guidance in a similar manner to the PENS estimator. The exponential atmosphere models are created using a least squares fitting process and require the altitude and estimated density histories but do not required latitude and longitude information. The table model, however, only requires the density estimate at the current vehicle location in order to generate the scalar table multiplier K. When these models are used in lieu of the PENS algorithm, they too are passed to the guidance inner loop and utilized in the same manner as described above.

The PENS algorithm may have minimal impact in the required computational overhead of the guidance process. In comparing the computational requirements of Mars aerocapture guidance algorithms, Rousseau et al.[104] used the number of lines of executable code as a proxy variable for guidance complexity and computational demand. To place the computational requirements of the PENS algorithm into perspective, the number of executable code lines in the NPC guidance algorithm is roughly a factor of five larger than the number of executable lines in PENS. It could be stated that any NPC algorithm is computationally demanding when compared to the reference path schemes prevalent in the Apollo and immediate post-Apollo eras. However, consider that an NPC guidance algorithm was recently selected for the Orion entry vehicle[93,94] due to the accuracy improvements realized by its use. As the speed of space-hardened avionics continues to improve, the use of computationally involved algorithms will certainly become more commonplace as is evidenced by Orion.

Even for robotic missions which typically have more limited computational resources, the PENS algorithm may still be considered computationally affordable. Many robotic navigation/guidance systems must numerically integrate six channels of acceleration information (three linear and three angular accelerations). These six channels must be numerically integrated twice to produce velocity and position information. In

total then, modern avionics must track and perform complex arithmetic operations, including numerical integration, on 18 channels of information (6 acceleration, 6 velocity, and 6 position channels). The PENS algorithm, in contrast, requires no complex arithmetic operations such as numerical integration, numerical differentiation, or large matrix operations. In addition, the PENS algorithm must only track *n+1* channels of information where *n* is the number of ensemble members and the additional channel represents the ensemble echo. Thus, the PENS algorithm, while certainly increasing in the computational overhead of the guidance system, should not apply an excessive strain on a modern avionics package. This must, of course, be balanced against the performance gains and risk reduction realizable through the application of the PENS algorithm which can be appreciable in many circumstances (see Chapter 5).

Table 4.1: Sub-Models Employed in Inner & Outer Guidance Loops

Sub-Model	Real World Simulation	On-Board Simulation	Knowledge Transfer	References
Aerodynamics	Constant L/D = 0.43 $C_L = 0.74$ & $C_D = 1.72$	Constant L/D ≈ 0.43 $C_L \approx 0.74$ & $C_D \approx 1.72$	none	29,31,32, 105-110
Sensors				
INS*	9-Channel ($\mathbf{A_{body}}$, $\mathbf{V_{iner}}$, $\mathbf{X_{iner}}$), noiseless, constant bias		via INS	110-112
ADS	3-Channel (α, β, p_{dyn}), noiseless, constant bias		via ADS	29,31,32, 105-109
Vehicle				
Mass (5736 kg)	constant, consistent with ballistic coefficient	constant, consistent with ballistic coefficient	perfect knowledge	
Reference Area ($8.65\ m^2$)	constant, consistent with ballistic coefficient	constant, consistent with ballistic coefficient	perfect knowledge	
Gravity	Harmonic (J2-J6)	Harmonic (J2)	none	
Actuator/Controller	pseudo-controller	pseudo-controller	N/A	97,99-102

4.2.2 Guidance System Sub-Models & Sources of Error

As stated in previous sections, both the guidance inner and outer loops utilize a series of sub-models in order to compute the 3-DOF equations of motion. These sub-models describe the characteristics and capabilities of the vehicle used for guided trajectory testing of the PENS algorithm. Table 4.1 presents a top-level summary of the

guidance sub-models used in both the outer loop (real world simulation environment) and the inner loop (on-board simulation environment).

The nominal vehicle characteristics utilized for the original reference trajectory in Section 2.7 were based on a sample return mission described by Masciarelli et al[6]. However, the growing emphasis in the future of Mars exploration will likely involve progressively larger payloads greater than one metric ton[113] to accommodate more extensive robotic and eventually human missions. Missions with larger robotic and human payloads often utilize an ellipsled[87,97-102] due to the greater packaging efficiencies and satisfactory aerodynamic characteristics achievable with this configuration (Figure 4.5).

Therefore a rigid aeroshell in the form of an ellipsled will be adopted as the nominal vehicle configuration for all guided trajectory simulations. The relevant impact of the vehicle configuration in the present study is to alter the atmospheric trajectory which then changes the information data stream and learning process used by PENS. While the vehicle configuration remains constant throughout the study, the effects of gross changes to the atmospheric trajectory will be examined (see Section 4.3.3). The adopted vehicle configuration, based on the ellipsled used in the EDLSA study[87,97-99], is considered a mid-L/D entry vehicle with an L/D of 0.43 and a ballistic coefficient of 385 kg/m^2.

Figure 4.5: General Ellipsled Geometric Configuration[98]

The mass of the vehicle (5736 kg) and the reference diameter (3.3 m) were selected to maintain the stated ballistic coefficient. This vehicle configuration corresponds to either a large robotic mission or a small support element of a human Mars architecture. However, Lafleur and Cerimele[114] observed that for a given set of initial conditions "an entry vehicle subject to a given bank-angle profile in a given planetary environment can, from a trajectory perspective, be completely described by [its] ballistic coefficient and lift-to-drag ratio. This description hinges on the assumption of a constant trim drag coefficient and [a] constant *L/D*, which is approximately true for hypersonic flight." Therefore, although it was necessary to decompose the ballistic coefficient into mass and reference diameter variables for entry into the POST2 simulation, the trajectory results generalize to any vehicle with similar L/D and ballistic coefficient. In fact a similar design is studied in the EDLSA project for delivering high mass (40 metric ton) human architecture elements.

Many of the inner-loop sub-models presented in Table 4.1 may have significant degrees of estimation error associated with them. In addition, other errors such as variations in the entry state or atmospheric density from expected conditions will also exist. In order to capture the impact of these unanticipated variations on the entry trajectory, all error sources identified from contemporary literature are included in the guided simulations as random variables. For example, the aerodynamic sub-model employs constant vehicle lift and drag coefficients (0.74 and 1.72 respectively) owing to the high-Mach hypersonic flight regime inherent in aerocapture trajectories. Due to pre-mission modeling uncertainty, however, the on-board estimates of the vehicle aerodynamics may contain a significant degree of error. To simulate the effects of this uncertainty, the lift and drag coefficients in the inner loop aerodynamic model are treated as random variables. The differences in the aerodynamic models will cause the on-board trajectory prediction to differ from the actual vehicle flight path. Because aerodynamic modeling inaccuracy has the potential to generate significant final state error[113], it is

treated as an error source and examined via simulation. The same approach, examination via direct simulation, is used for all similar error sources thought in contemporary literature to contribute to large errors in final state. Table 4.2 lists these error sources and the associated ranges and distribution shapes used to create the error dispersions. For all normal distributions, the limits stated in the table correspond to the effective ±3σ bounds. The normal distribution will, of course, exceed these bounds at large sample sizes. Therefore, the maximum and minimum observed dispersions are typically slightly more extreme than presented in the table. For example, in one guided trajectory Monte Carlo containing 36,000 cases (see Chapter 5), the observed extreme cases of the dispersed lift coefficient C_L were 0.64 and 0.84 whereas the ±3σ bounds of ±10% actually lie at 0.67 and 0.81. For uniform distributions, the stated limits represent the upper and lower distribution bounds.

As discussed in Section 4.2, the ADS uses flush-mounted pressure transducers on the external structure of the vehicle to determine the angle of attack, α sideslip angle, β and the dynamic pressure, p_{dyn} of the free stream. Systems such as these have been successfully employed on the North American X-15 high-altitude research vehicle[31,32,105], the Space Shuttle[29,106-108], and one is currently in flight to Mars aboard MSL[109]. The sensor sub-models (ADS and INS) are modeled as noiseless sensors with constant biases (see Table 4.1). This constant bias is applied on all channels of both the ADS and INS which cause persistent and non-reducible errors to accrue over the trajectory. While more detailed models exist with random noise and varying bias levels, the precise nature of the noise observed will depend on the type of sensor selected as well as the internal design of the sensor. In addition, many modern guidance systems often utilize multiple redundant sensors where the output of each sensor is fed to a common filter which produces smoothed estimates of the various sensed quantities. Therefore, a noiseless, constant bias sensor is utilized as a rudimentary proxy of a filtered guidance solution. These biases are randomly selected from the distributions in Table 4.2 in the Sensor Errors subsection. For

vector quantities, the bias is equally applied to all three channels under the assumption that all of the linear accelerometers are of the same manufacturing origin and suffer from similar error dynamics.

The error sources listed in Table 4.2 also include likely environmental errors such as variations in the initial entry state of the vehicle owing to uncertainty in the interplanetary navigation process as well as variations in the atmospheric state. Variations in the atmosphere, due to the differences between pre-mission atmospheric models and the true Martian atmosphere, are especially important given the nature of the present study. Recall from Section 2.3 that the density of the Martian atmosphere is greatly affected by the amount of dust suspended in the atmosphere and the incoming solar flux incident on the thermosphere. The dust load in the atmosphere is commonly measured with the dust optical depth, τ_{Dust} and the solar flux is measured using the 10.7 cm wavelength full disk solar emission (commonly abbreviated F10.7). Both of these parameters are dispersed to permit the mean atmospheric density trend provided by MarsGRAM to vary. Perturbations in density are then superimposed onto the smooth mean density trend with a newly developed perturbation generator (see Section 4.4). The perturbation magnitudes and the random number seed used to generate the perturbations are also dispersed.

Note that the aerodynamic errors are not rigidly correlated to errors in the center of gravity for all trajectories herein. An offset in the center of gravity would generate specific lift and drag coefficient errors which would impact vehicle performance. However, this approach is not taken here since the exact correlation often depends on the detailed aerodynamics of the entry vehicle.

Table 4.2: Dispersed Variables for Guided Trajectory Simulations

Variable	Units	Limits	Nominal	Distribution	References
Aerodynamic Errors					
Lift Coefficient, C_L	%	±10%	0	normal	87, 102,113
Drag Coefficient, C_D	%	±10%	0	normal	87, 102,113
Sensor Errors					
Body Accelerations	Earth μG	±10	0	normal	38, 111
Inertial Velocity	m/s	±2	0	normal	38, 87, 102
Inertial Position	km	±2	0	normal	38, 87, 102
Angle of Attack	deg	±0.1	0	normal	29,31,32, 105-109
Sideslip Angle	deg	±0.1	0	normal	29,31,32, 105-109
Dynamic Pressure	%	±2	0	normal	29,31,32, 105-109
Initial State Errors					
Velocity	m/s	±20	0	normal	87, 102
Flight Path Angle	deg	±0.25	0	normal	87, 102
Altitude	km	±3	0	normal	87, 102,115
Latitude	deg	±0.08	0	normal	87, 102,115
Longitude	deg	±0.04	0	normal	87, 102,115
Atmosphere					
Density Estimator	N/A	1:8	2(KPENS)	integer uniform	N/A
τ_{Dust}	N/A	0.1-3	0.45	uniform	44
Perturbation Seed	N/A	1:29999	none	integer uniform	44
Climate Set	N/A	1:3	3	integer uniform	N/A
Pert. Magnitude	%	±100%	0	uniform	44
F10.7	10^{-22} W/(Hz-m^2)	70:200	130	uniform	86

4.2.3 Effect of Atmospheric Winds on Density Measurements

The presence of significant sustained winds artificially increases or decreases the free stream dynamic pressure which the ADS measures in flight. Note that because the role of the ADS is to measure the free stream dynamic pressure p_{dyn}, wind components perpendicular to the plane of motion do not affect p_{dyn} and will not generate density measurement error. Also note that the angle of attack, α is unlikely to be significantly changed by Martian winds. An angle of attack change would occur if a significant vertical wind component exists and the vertical winds at altitude on Mars are typically

small. The vertical winds predicted by the MCD at the time, season, and spatial volume (described in Figure 2.4) of the test mission over Hellas Planitia have a maximum value of 3.3 m/s over all climate scenarios, including dust storm scenarios.

The free stream density is computed by combining information measured by both the ADS and the INS throughout the entry. The ADS measures the free stream dynamic pressure which may be expressed as

$$p_{dyn} = \frac{1}{2}\rho_{\infty}(V_{FS})^2 \tag{4.1}$$

where ρ_{∞} denotes the free stream density (the quantity of ultimate interest) and V_{FS} is the free stream velocity. Note that the free stream velocity V_{FS} is the sum of both the wind velocity V_W and the relative velocity of the vehicle V with respect to the surface of the planet ($V_{FS}=V+V_W$). In the nominal, no-wind case, V_W = 0 and density estimation is straightforward:

$$\rho_{\infty,estim} = \frac{2p_{dyn}}{V_{INS,estim}^2} = \frac{2\frac{1}{2}\rho_{\infty}(V)^2}{(V)^2} = \rho_{\infty} \tag{4.2}$$

In the case of significant wind, however, the above relation does not hold true. The INS estimates the inertial velocity of the vehicle by numerically integrating the sensed acceleration over time[116]. This produces an estimate of the change in velocity from an inertial reference state such as the inertial velocity at entry interface. The presence of significant head or tail winds will influence the rate of entry vehicle deceleration and will alter the inertial velocity-vs-time profile relative to the no-wind case. However, because the INS is able to sense the additional acceleration caused by wind, it is still able to estimate a correct inertial velocity within the system accuracy limits. Thus, the presence of wind does not cause inertial velocity biases or measurement errors. The INS then converts the unbiased inertial velocity into surface relative velocity V by factoring in the constant rotation rate of the planet. The dynamic pressure measured

by the ADS, however, is biased by wind as shown in Equation 4.3 because the wind field alters the free stream velocity.

$$p_{dyn} = \frac{1}{2}\rho_{\infty}\left(V_{FS}\right)^2 = \frac{1}{2}\rho_{\infty}\left(V + V_W\right)^2 \tag{4.3}$$

Thus, in the case of significant wind, Equation 4.1 becomes:

$$\rho_{\infty,estim} = \frac{2p_{dyn}}{V_{INS,estim}^2} = \frac{2\frac{1}{2}\rho_{\infty}\left(V + V_W\right)^2}{\left(V\right)^2} \neq \rho_{\infty} \tag{4.4}$$

The estimated density is not equivalent to the free stream density and the resulting error may be expressed as:

$$error = \left[\frac{\rho_{\infty,estim} - \rho_{\infty}}{\rho_{\infty}}\right]100\% = \left[\frac{\frac{\rho_{\infty}\left(V + V_W\right)^2}{\left(V\right)^2} - \rho_{\infty}}{\rho_{\infty}}\right]100\% = \left(\frac{\left(V + V_W\right)^2}{\left(V\right)^2} - 1\right)100\% \tag{4.5}$$

This relation has been used to explore the effect of the predicted wind field on density measurement errors. Figure 4.6 illustrates the impact of the meridional winds (i.e. the wind field component running parallel to lines of longitude) on the density measurement error. Due to the orientation of the entry trajectory which runs from the south toward the north along a line of constant longitude, the meridional winds are the head/tail wind component for this entry mission. As discussed, the remaining two wind field components, cross winds and vertical winds, do not impact the ADS free stream dynamic pressure measurements.

The Mars Climate Database (MCD) was used to examine the meridional wind components for all eight climate models (including high wind dust storm cases) over the entry corridor illustrated in Figure 2.4. In total, wind data was taken over 113,664 MCD grid points and summarized in the distribution on the right hand side of Figure 4.6. The 99th percentile case corresponds to winds of 100 m/s with a maximum observed wind speed of 185.6 m/s.

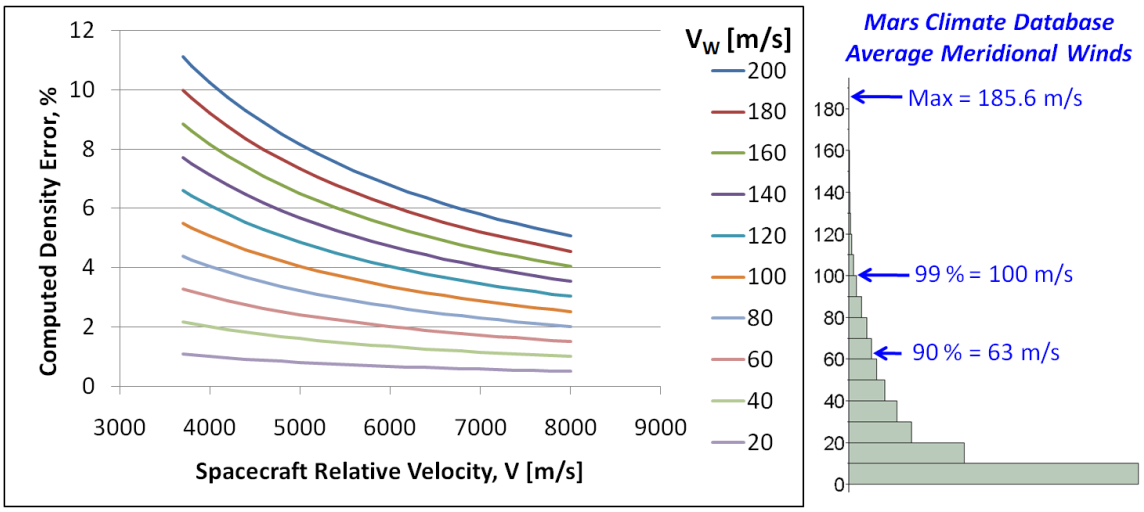


Figure 4.6: Effect of Sustained Winds on Density Measurement Error

Literature results are in general agreement with the MCD derived wind data. Moreno et al.[117] used the Plateau de Bure Interferometer, a radio telescope in the French Alps, to directly measure wind speeds in the Martian middle atmosphere (40-80 km). By examining Doppler line shifting of carbon monoxide in the Martian atmosphere, they found maximum average wind velocities near 100 m/s in the equatorial region. A later, more comprehensive study by Moreno using the same techniques found that typical middle atmosphere wind velocities vary between 70-170 m/s and exhibit strong temporal and spatial variability[118]. These results are intriguing because they are direct measurements of the conditions in the Martian middle atmosphere, which is rare unto itself. The altitude range over which these measurements were taken, therefore, make them uniquely applicable to aerocapture studies. In addition, these results seem to align well with the data from the Mars Climate Database (MCD) just discussed. High altitude wind information is also available. Thermospheric winds recovered from Odyssey and MGS aerobraking data by Crowley and Tolson[119] at altitudes of 100-160 km found evidence of sporadic winds in the 100-200 m/s range. The observed high speed wind patterns vary greatly in both time and space, with large changes in wind magnitude and direction (e.g. from +200 m/s to -200 m/s) observed in as little as 100 seconds.

Taken together, the above data suggests that 200 m/s is a conservative upper bound for the wind velocity in the middle and upper atmosphere. It should also be observed that the MCD-predicted speed of sound on Mars is approximately 200 m/s. Therefore, wind speeds greatly in excess of 200 m/s are supersonic and are likely to occur with lower probability. Over the range of likely aerocapture velocities, Figure 4.6 reveals that at the density measurement errors range from 6% to 11% for sustained 200 m/s winds.

If the wind patterns on the day of entry are variable, as all of the observational data suggests, then the 6-11% errors in density measurement would occur sporadically and have little effect on density estimation. The PENS estimator functions as a linear adaptive filter, forming prediction models which are linear combinations of smooth (i.e. noiseless and unperturbed) atmospheric density models. Thus, the resulting density prediction models are noise free. The weights of each of the ensemble members in the linear combination would be affected only if the wind is sustained long enough to begin affecting the average error signals (i.e. the difference between each ensemble member and the measured density over time). Therefore, by design, rapid transient anomalies in either the true density or atmospheric winds will not cause significant PENS model changes.

For sustained winds, however, changes in the PENS density model will occur if the density bias is sufficiently large. Strong, long duration winds deceive the ADS (and PENS by extension) into believing that the atmospheric density is higher or lower than it actually is. However, adaptation to strong in-plane sustained winds is, in fact, beneficial to the aerocapture guidance. The goal of on-board density estimation is to provide the guidance with the ability to accurately predict the aerodynamic accelerations that the vehicle will encounter throughout the remainder of the trajectory. Consider the free body diagram given in Figure 4.7:

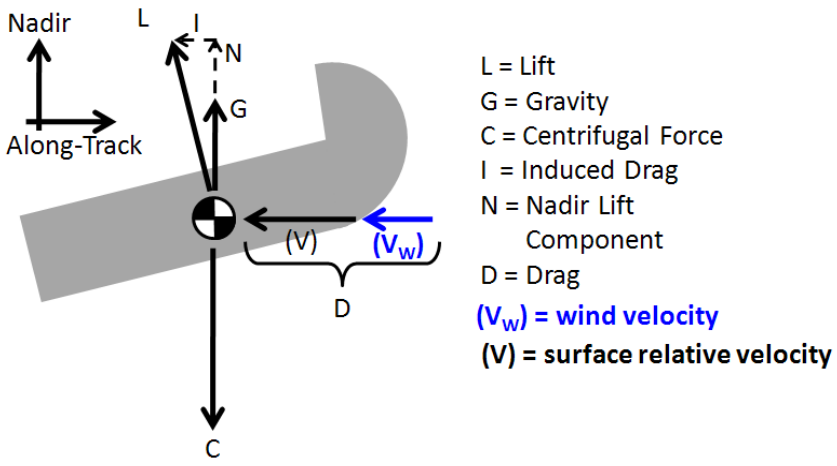


Figure 4.7: Effect of Wind Velocity in the Plane of Motion

The gravitational and centrifugal forces are not directly influenced by the presence of significant winds. The lift and drag forces, however, are effected due to their dependence on the free stream dynamic pressure. The following argument focuses on the drag acceleration but an analogous argument may be made for the lift acceleration. The true drag acceleration in the direction of motion may be expressed as

$$ma_{drag} = \frac{1}{2}\rho_\infty V_{FS}^2 C_D S = \frac{1}{2}\rho_\infty (V + V_W)^2 C_D S \tag{4.6}$$

Because no wind estimates are available on-board, the guidance utilizes the surface relative velocity V in place of the free stream velocity V_{FS}. This introduces an error into the estimated drag profile:

$$ma_{drag,estim} = \frac{1}{2}\rho_\infty V^2 C_D S \neq \frac{1}{2}\rho_\infty (V + V_W)^2 C_D S \tag{4.7}$$

However, recall from Equation 4.4 that the estimated density is biased by the wind field:

$$\rho_{\infty,estim} = \frac{(V + V_W)^2}{(V)^2}\rho_\infty \tag{4.8}$$

When the above relation is substituted into Equation 4.7, the estimated drag becomes:

$$ma_{drag,estim} = \frac{1}{2}\rho_{\infty,estim}V^2 C_D S = \frac{1}{2}\frac{(V+V_W)^2}{(V)^2}\rho_\infty V^2 C_D S = \frac{1}{2}(V+V_W)^2 \rho_\infty C_D S \quad (4.9)$$

which is equivalent to the true drag acceleration in Equation 4.6. Thus, if PENS is able to adapt to the wind-biased measured density, the resulting density model will counteract the effect of the wind error on the predicted aerodynamic accelerations.

The density measurement error continually increases throughout the entry as the velocity of the vehicle decreases (see Figure 4.6). However, PENS is able to continually adapt and learn from the observed environment. The prediction models generated by PENS, therefore, account for the current level of density measurement error caused by winds. Adapting to the current level of error permits more accurate drag prediction and limits the impact of wind-induced errors as the vehicle velocity decreases.

For example, consider the effect of wind-induced density error for the nominal 1400 km trajectory (discussed further in Section 4.3.3). The vehicle enters the atmosphere at 5.9 km/s and has slowed to just below 5 km/s at periapsis crossing. Inspection of Figure 4.6 reveals that the density measurement error due to 200 m/s sustained winds increased from 6.8% to 8% over this interval. If PENS can adapt to this 8% bias, then the impact of future increases in density measurement error on the predicted drag acceleration will be minimized. For the example given here, if PENS adapts to the 8% error near periapsis where control authority is high, the increase in drag prediction error over the remainder of the trajectory is approximately 11%-8% = 3%. This level of error is small and well inside the noise level of the Martian atmosphere. Therefore, because accurate trajectory predictions are available near periapsis, the guidance can successfully guide the vehicle through the outbound leg.

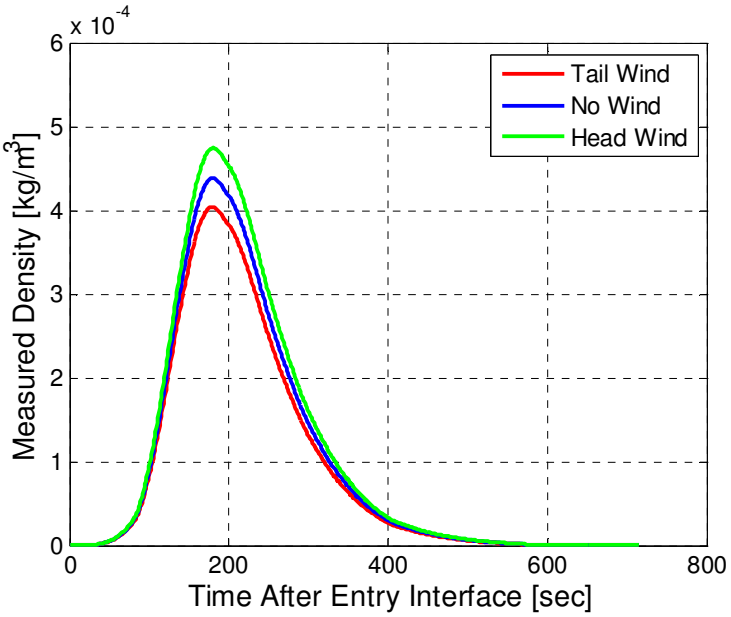


Figure 4.8: Measured Density Signal Generated by a 200 m/s Sustained Wind

The above conclusion depends, however, on the ability of PENS to adapt to the current level of density measurement error. Consider again the example above where PENS must adapt to the density error caused by 200 m/s sustained winds while in the region near periapsis. The effect of the wind on the measured density signal is shown in Figure 4.8 for the nominal 1400 km trajectory discussed in Section 4.3.3. This data was obtained by computing the wind-biased dynamic pressure given by Equation 4.3 and then computing the estimated density using Equation 4.4.

The PENS algorithm is permitted to observe the atmosphere from entry interface up to periapsis at 181 seconds. PENS then constructs a density prediction model from periapsis to the end of the trajectory. The error between the predicted density and the measured density is given in the top plot of Figure 4.9. Note the high degree of similarity between the no wind case and the 200 m/s head and tail wind cases despite the large differences in the measured density signals in Figure 4.8. Also observe that significant variations exist between the PENS density models used to construct these predictions as

evidenced by the large changes in ensemble weights across these cases (see Figure 4.9, bottom plot).

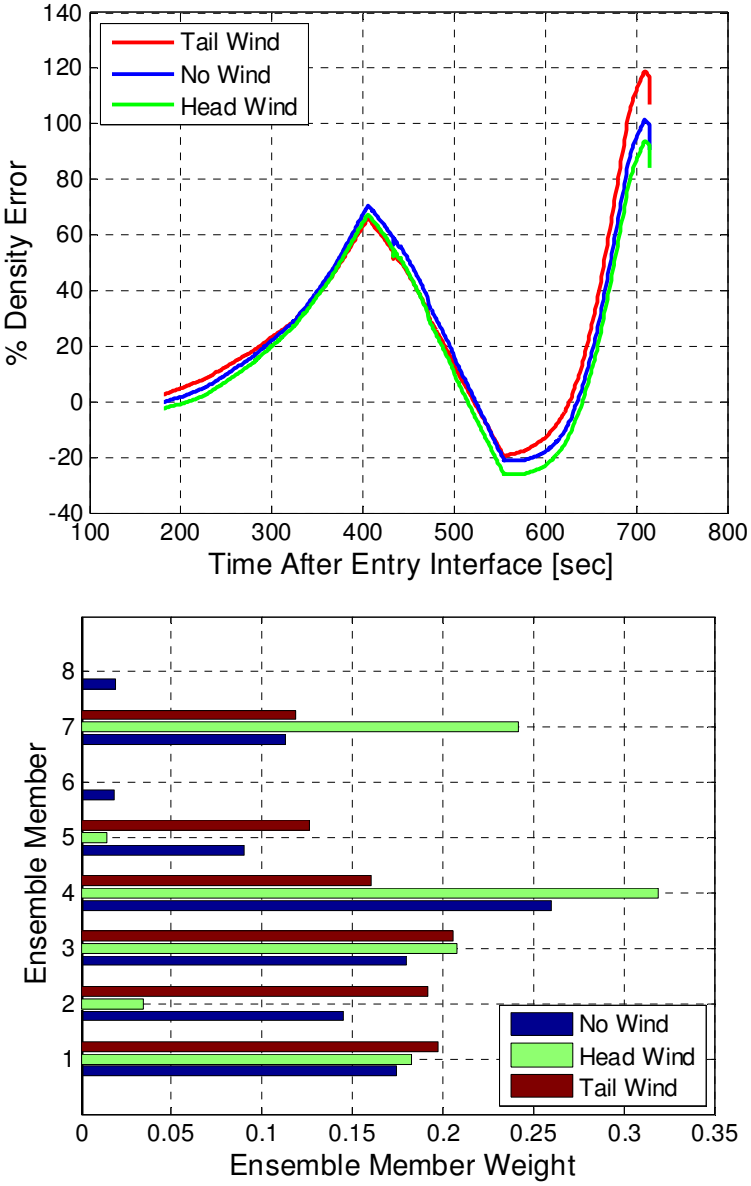


Figure 4.9: Prediction Performance from Periapsis with Respect to Measured Density (Top) and Associated Ensemble Weights (Bottom)

Therefore, in the unlikely event of strong and sustained winds over the majority of the entry, PENS is able to adapt to the wind-biased density measurements as predicted. Thus, the impact of sustained winds on the predicted drag acceleration will be minimal as already discussed. In the more likely case of highly variable wind fields, PENS will

assume that the sporadic density measurement error is due to atmospheric noise and ignore it. For these reasons no further treatment of atmospheric winds will be examined herein.

Other adaptive density estimators, however, may be more sensitive to wind then PENS. For reference, it is useful to note that that alternate methods for measuring the free stream density near the vehicle exist. McKenzie[120] describes a method for using an on-board laser-based instrumentation system for measuring the free stream density outside the window of the middeck crew hatch on the Space Shuttle. A pulsed ultraviolet laser is emitted normal to the longitudinal axis of the Shuttle. Light scattered backward by the ambient atmosphere is collected by an optical system attached to the inside of the hatch window and the collection optics are skillfully arranged such that only light scattered from the laser beam beyond the shock layer is detected. The intensity of the scattered light is time integrated and the resulting signal is "linearly proportional to the atmospheric density…and has no significant dependence on any other properties of the atmosphere or on the flight conditions." Such a technology may be useful on future Mars entry vehicles using similar ultraviolet Rayleigh scattering techniques or more modern techniques based on lidar. Note that this approach, like an ADS and unlike on-board lidar systems examined in Chapter 5, is designed for short ranges and only measures atmospheric density close to the vehicle.

Thus, the signals from multiple measurement systems could be combined to produce highly accurate free stream density estimates if they are required. However, literature concerning modern and historical ADSs indicates that high density estimation accuracies are possible. In fact, the Space Shuttle Columbia was equipped with the Shuttle Entry Air Data System (SEADS) which demonstrated 3σ measurement errors for p_{dyn} in the 2% range[29,106-108]. A similar system with even higher predicted accuracy (3σ p_{dyn} error of 0.45%) is currently aboard MSL bound for Mars[109]. As will be shown,

however, the SEADS flight-demonstrated level of accuracy is more than sufficient to guarantee significant aerocapture performance gains.

4.3 Verification & Validation of PENS-Augmented Guidance

The goal of this section is to demonstrate that the performance of the PENS-augmented NPC guidance matches both expectation and similar results from relevant aerocapture literature. The first phase of demonstration testing involves verifying that the PENS algorithm, originally developed in MATLAB, behaves as expected once translated into FORTRAN and integrated into the NPC guidance scheme. In the second phase of the demonstration, guided trajectories are generated and compared to similar simulations conducted in the literature. Following verification & validation, the nominal trajectories required for the guided trajectory Monte Carlo simulations in Chapter 5 are introduced and discussed.

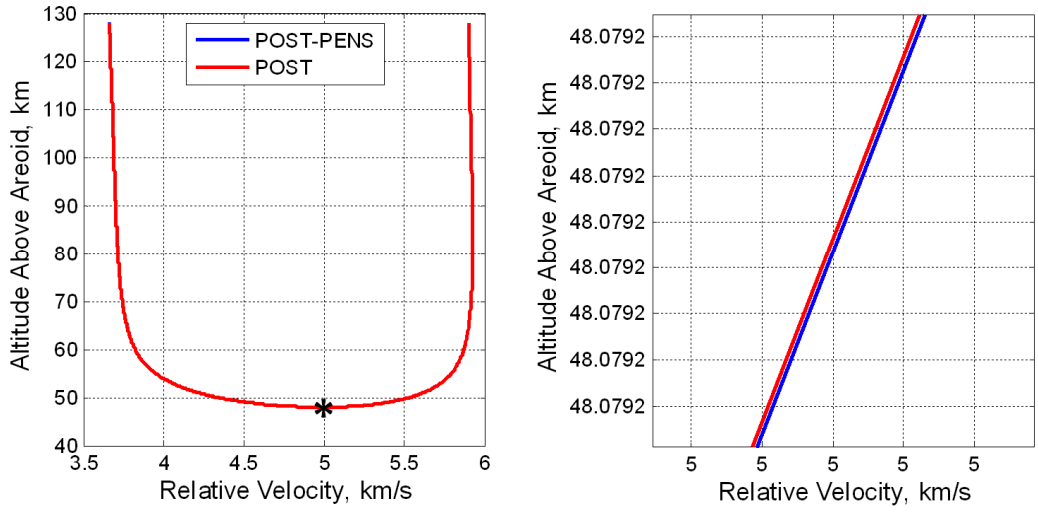
4.3.1 PENS-Augmented Guidance Verification

As mentioned, the original PENS algorithm has been designed, developed, and tested in MATLAB (see Chapter 3). The POST2 simulation framework, including the NPC guidance system, however, is written in multiple languages including C, C++, FORTRAN, and Perl. From among the possible choices, the PENS algorithm is converted into FORTRAN for two reasons. First, because the majority of the NPC guidance and the original POST2 trajectory code are written in FORTRAN, integration of the converted PENS algorithm into the guidance becomes more streamlined. Second, FORTRAN shares many similarities with MATLAB scripting code which reduces the required conversion and implementation time.

The successful implementation of the PENS algorithm into the NPC guidance and the POST2 framework required a myriad of modifications and changes within the original source code. To ensure the original functionality of POST2 is unaltered, a

trajectory simulation is conducted wherein both the guidance and PENS are present but inactive. This is to ensure that the PENS integration measures have not interfered with the basic trajectory engine (i.e. the equations of motion, sub-models, and the Runge-Kutta integration scheme) which comprises the outer loop of the guided simulation. Because the guidance must be inactive for this test, the trajectory used for verification purposes is a modified version of the reference trajectory given in Section 2.7. The original lateral logic is removed and replaced with a constant bank angle manually tuned to produce the desired target apoapsis. In addition, as mentioned in Section 2.7, the target apoapsis altitude is now returned to 1400 km to match the orbit used by Masciarelli et al.[6] The modified reference trajectory is then simulated using both an unaltered version of POST2 and then again using the POST2 code containing the NPC guidance and PENS. Because the standard unaltered POST2 contains no guidance system, deactivation of the guidance is necessary to accomplish this comparison.

Figure 4.10 displays the results of this first verification test which examines the functionality of the basic POST2 trajectory engine. As stated, the modified reference trajectory is computed twice; first using the original POST2 source code and then using the new POST2 source code containing the PENS algorithm. Both plots in the figure display an overlay of the altitude-relative velocity trend for the modified reference trajectory. The red curve is created using the unaltered version of POST2 and the blue trajectory is computed using the PENS-augmented POST2 code. Note that the differences between the trajectories are indiscernible on large scales (Figure 4.10.a). Figure 4.10.b on the right depicts an enlargement of the region near periapsis (denoted by a black asterisk in the opposing left-hand plot). As may be seen by inspection, the differences in the two trajectories are due to limitations in numerical precision, and are, for all practical purposes, identical.



(a.) **(b.)**

Figure 4.10: Verification of POST2 Outer Loop Functionality for (a.) the Complete Trajectory and (b.) Near Periapsis

A similar basic functionality test was conducted for the inner simulation loop to ensure that no PENS integration measures have adversely affected the on-board inner-loop trajectory engine. The NPC guidance was temporarily activated and the on-board sub-models were set to exactly match the outer-loop (i.e. "real world") sub-models. The results of this test revealed that both loops yield identical results to within numerical precision when their inputs are equalized. These first two tests indicate that the PENS implementation has not corrupted the basic functionality of the POST2 outer loop or the NPC guidance inner loop.

With the guidance inner and outer loops performing as expected, the PENS algorithm is now activated and tested. The NPC guidance is again deactivated to allow the performance of the PENS algorithm to be verified in isolation without being confounded with any guidance errors or anomalies. Recall from Chapter 2 that the PENS algorithm relies on an ensemble of models stored aboard the entry vehicle in condensed form as neural networks (NNs). The NNs in the original MATLAB algorithm were extensively verified and tested throughout development to ensure that the densities

replicated by the NNs were reasonable and consistent with their original MCD climate models (see Section 2.5 and Appendix A).

In the FORTRAN-based PENS algorithm, the NN equations and computation routines were extensively modified relative to the original MATLAB versions. If these NNs are incorrectly computed, then no degree of accuracy or consistency can be expected from the FORTRAN-based PENS algorithm. In order to insure that the NN densities are correctly computed, the PENS algorithm and NN subroutines from both the FORTRAN and MATLAB versions are applied to the same trajectory. Figure 4.11 depicts the percentage difference between the MATLAB and FORTRAN ensemble densities over the length of the trajectory. Inspection of the figure reveals that the percentage differences in the computed ensemble densities are very small and within the numerical precision of the algorithms and the computational hardware.

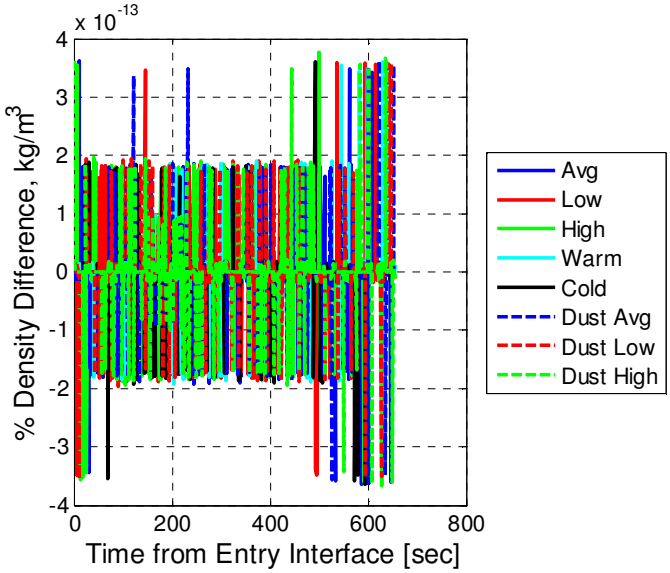


Figure 4.11: Comparison of NN Density Computations Between the FORTRAN-based and MATLAB-based PENS algorithms

Following computation of the ensemble densities, the PENS algorithm utilizes the differences between the measured atmospheric density and the on-board ensemble members to determine ensemble weights. Therefore, the next layer of PENS verification requires a comparison of the ensemble weights computed by both MATLAB and

FORTRAN. A comparison of the MATLAB-FORTRAN weight differences from the PENS estimator is presented in Figure 4.12 while the differences in the parameters of the historical estimators is given in Figure 4.13. The sharp spikes in the percent difference curves such as near 200 seconds in Figure 4.12 and near 550 seconds in Figure 4.13 are generated when the parameters in question approach zero. A small number close to zero appears in the denominator, generating the data spike in the percentage difference trend. The square-wave changes in Figure 4.12 are generated by changes in the ensemble weights.

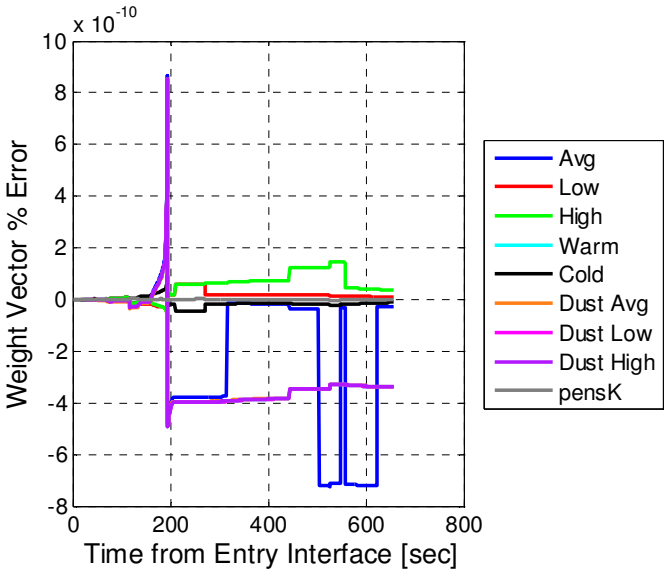


Figure 4.12: Comparison of MATLAB & FORTAN PENS Estimator Parameters

As an ensemble member becomes more trusted, its weight becomes a larger number and so too does the numerical imprecision associated with that larger weight. Notice, however, that all the percent difference curves in both plots remain quite small in magnitude. The maximum absolute error for all of the PENS parameters is 9×10^{-10} % and 8.4×10^{-8} % for the historical estimator parameters. This suggests that the data from both sources agrees to within numerical precision. Thus, the FORTRAN and MATLAB PENS algorithms are, for all practical purposes, arriving at the same parameter estimates.

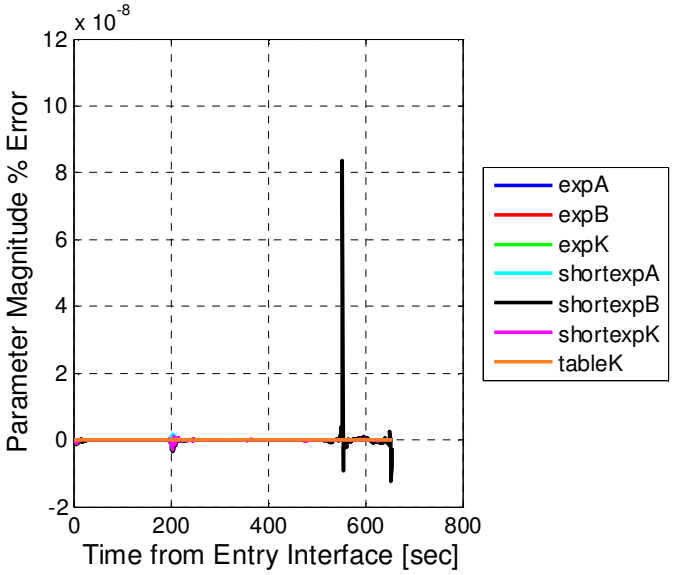


Figure 4.13: Comparison of MATLAB & FORTAN Historical Estimator Parameters

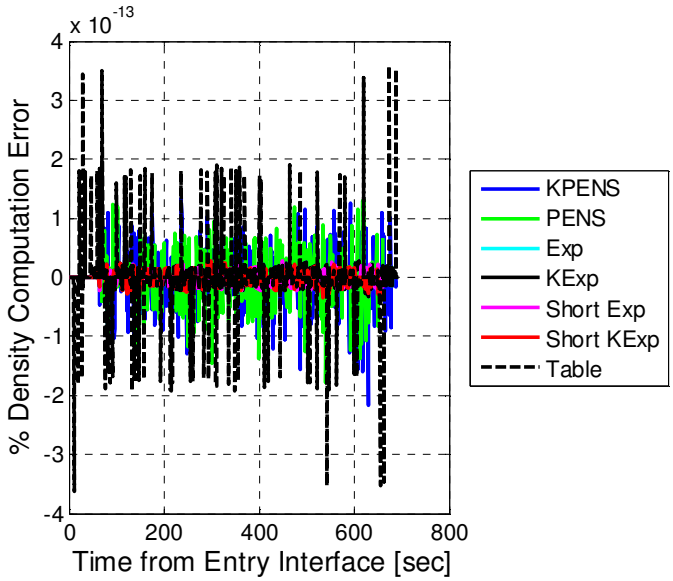


Figure 4.14: Comparison of MATLAB & FORTAN Computed Densities

The last layer of PENS algorithm verification corresponds to the last major algorithmic phase of the PENS iteration cycle. Following computation of the estimator parameters, the algorithm assembles the parameters into density predictions which are then used by the inner loop of the guidance for path prediction and planning. The density predictions generated by the original MATLAB-based PENS algorithm were verified

both manually and by comparing the predicted density trends to the true density (e.g. Figure 3.20). In order to ensure that the new FORTRAN density computation process is correct and accurate, the predicted densities were compared to the analogous MATLAB-computed densities at the current location of the vehicle. Figure 4.14 illustrates the density comparison between the MATLAB-based and FORTRAN-based PENS algorithm. Note that the percent density differences remain very small over the entire trajectory indicating that both algorithms produce the same results to within numerical precision.

4.3.2 PENS-Augmented Guidance Validation

With the POST2 version of the PENS algorithm verified to be functioning as expected, validation of the NPC guidance system is now examined. The goal of this phase of testing is to prove that the performance of the NPC guidance scheme matches similar systems developed in the relevant literature on Mars aerocapture. Because, however, the PENS algorithm is designed to alter trajectory performance, the PENS algorithm is active, but the model parameters are never used by the inner guidance loop. This permits the full integrated system to be used for testing without defeating the purpose of validation. This approach also proves that the PENS operations do not adversely impact guidance in any indirect way (no interference between subroutines, data storage, etc). The only permissible channel for the PENS algorithm to impact the guidance is through transfer of the density model to the inner loop. Because this transfer does not take place for the validation tests, then the PENS-augmented guidance should behave in a manner which closely matches similar systems in the literature.

The validation demonstration attempts to replicate a Mars aerocapture trajectory computed by Powell and Braun[92] using a similar NPC guidance system and earlier versions of both POST and MarsGRAM. The initial conditions, desired target orbit, vehicle characteristics, and atmosphere models for both the inner and outer loops are

matched to the Powell/Braun baseline. Because their atmosphere model is given in graphical form, minor differences between the trajectories exist owing to difficulties in interpreting the precise values of their nominal atmosphere model. However, despite this limitation, significant similarities between the trajectories produced by both systems are readily visible. More importantly, a high degree of similarity exists in the bank angle commands produced by both systems. This indicates that, while the atmosphere models may contain small differences, both systems produce similar commands in similar circumstances. Powell and Braun utilize three different atmosphere models (low, nominal, and high densities) and a lengthy attempt was made to match what was thought to represent the nominal atmosphere model in green in high-hand plot of Figure 4.15. The high and low density trends are created by Powell and Braun by finding the most extreme densities which bound all of the other density models produced by their older version of MarsGRAM. Attempts to utilize the low and high density models (blue and red curves in Figure 4.15 respectively) which are much simpler to reconstruct proved difficult. The extreme densities combined with the steep entry flight path angle to produce a sensitive system with very limited controllability margins in POST2.

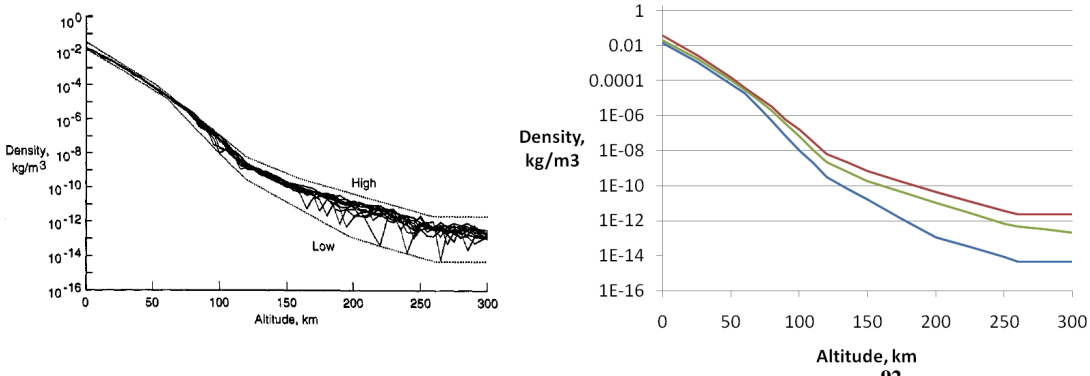


Figure 4.15: Atmosphere Models from Powell and Braun[92](right) and Reconstructed Validation Atmospheres (left)

It should be noted that their atmosphere is only a function of altitude, a common practice for the time. Utilizing a best-guess reconstruction of their nominal atmosphere, the altitude-vs-time histories (Figure 4.16) show excellent initial agreement. However, at

200 seconds just prior to periapsis, the differences in the atmosphere models begin to have cumulative effect and small differences between both trajectories develop. Note that shortly after periapsis, the NPC solution again approaches the Powell-Braun solution suggesting that low-altitude density differences are generating the disparity between the trajectories. A similar trend is discernable in the energy-vs-time histories from both guidance systems (Figure 4.17). A high degree of initial agreement gives way to larger differences at low altitudes. However, both guidance systems arrive at nearly identical exiting orbital energies indicating that the desired final state is reached by both systems. In fact, both guidance schemes reach the intended apoapsis altitude of 33,786 km during their respective atmospheric entries with extremely small residual errors (less than one kilometer each).

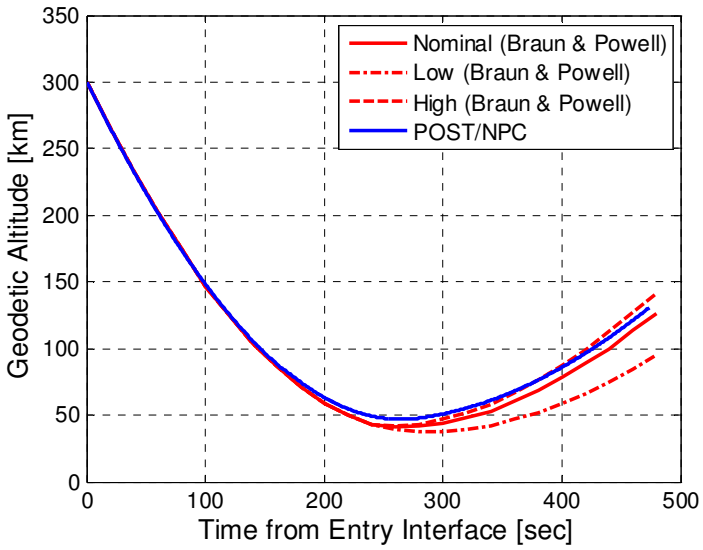


Figure 4.16: Comparison of NPC & Powell-Braun Altitude-vs-Time Histories

The last and most significant point of validation is the comparison of the bank angle command histories from both systems in Figure 4.18. It is important to mention that the Powell-Braun bank angle convention is reversed with respect to the normal convention used herein (their convention uses 0° bank as full lift down). While the results

in Figure 4.18 are reported in this convention to match the results from Powell and Braun, this is the only location in this work where this convention is used. All other bank angles use the convention of 180° as full lift down and 0° as full lift up.

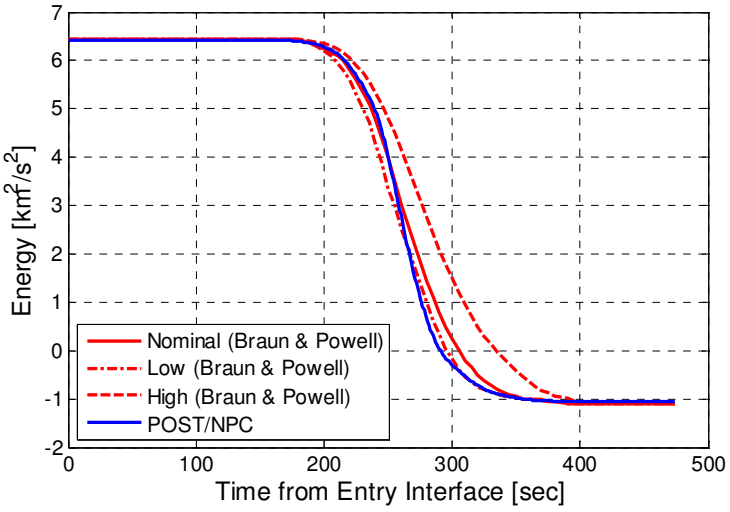


Figure 4.17: Comparison of NPC & Powell-Braun Energy-vs-Time Histories

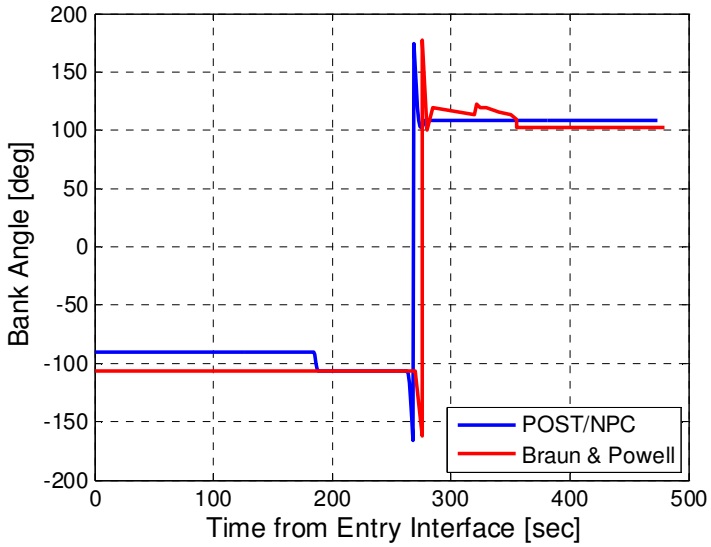


Figure 4.18: Comparison of NPC & Powell-Braun Bank Angle Histories

The commanded bank angle history is revealing because the goal of validation is to determine if both guidance systems make similar decisions in similar circumstances. Note that unlike the variables previously discussed, the Powell-Braun bank commands resulting from flight through the high and low density atmospheres are not available. Despite the differences in the atmosphere models, both systems make very similar guidance decisions. In fact, the nominal bank angle at entry interface for the NPC is -90°. Just prior to 200 seconds, the NPC guidance changes its bank to -106°, which exactly matches the bank commanded by the Powell-Braun guidance. Both systems execute very similar bank reversals near 270 seconds which occur only 7 seconds apart. Following the bank reversal, both systems hold similar constant bank angles as they exit the atmosphere. Thus, the NPC guidance is able to replicate the trajectory produced by a separate but similar guidance scheme for Mars aerocapture.

4.3.3 Nominal Guided Trajectories

With the PENS-augmented NPC guidance validated, the nominal guided trajectories are now produced and discussed. The EDLSA project, in examining the differences between low- and high-altitude orbits, found that low-altitude orbits generally require higher post-aerocapture ΔV but at a significantly reduced risk of aerocapture failure[87]. Conversely, it was also found that a high-altitude 1 Sol orbit requires, on average smaller post-aerocapture ΔV but the risk of failure, however, was notably elevated. (Recall that one Sol, or Martian day, is approximately 24 hours, 39 minutes.)

These results match intuition because aerocapture trajectories targeting high-altitude orbits must remove relatively small yet very precise quantities of energy from the orbit of the entry vehicle. In order to reach a 1 Sol apoapsis (which translates to roughly 34,000 km), the velocity of the vehicle must remain high throughout the aerocapture entry. Any errors in the atmospheric density estimation can then easily result in a hyperbolic final state and aerocapture failure. The ability of the vehicle to successfully

meet the exacting final state requirements depends on the atmospheric perturbations and the manner in which the guidance compensates for these perturbations. However, because the orbital energy remains high, small changes in ΔV create large changes in orbital energy, making high-altitude orbits affordable. For aerocapture entries targeting low-altitude orbits, the reverse trends are true. The aerocapture maneuver significantly reduces the orbital velocity of the vehicle which makes the risk of failure much less dependent on knowledge of the atmosphere. However, because the orbital velocities are lower, larger ΔV is required to create a given change in orbital energy.

The PENS algorithm has the potential to positively affect each type of orbit. For high-altitude target orbits, more precise atmospheric knowledge and a high degree of adaptability in the density estimator may significantly reduce the associated failure risk. This would then make high-altitude target orbits both affordable and less risky. For low-altitude target orbits, improved estimator accuracy and adaptability may permit more precise final state targeting which may lower the cost (i.e. ΔV) of these relatively safe yet expensive orbits.

Both types of orbits are also architecturally useful. High altitude, 1 Sol orbits offer a degree of predictability because the period is equivalent to the Martian day. This orbit then not only simplifies timeline planning and phasing but also requires a relatively small ΔV to escape Mars orbit. Low-altitude orbits, on the other hand, are required for certain architectural elements such as landing vehicles or orbital scientific payloads. Such orbits can be tuned to provide good ground coverage for communication, observation, mission support or a myriad of other reasons.

Because PENS may have a unique impact on both orbits and because both may have a high degree of architectural usefulness, both target orbits will be studied in the guided simulations. One nominal trajectory is created for each target orbit (Figure 4.19) and the orbit parameters and initial conditions for each are given in Table 4.3. Four of the six orbital parameters (apoapsis altitude, periapsis altitude, longitude of the ascending

node, and inclination) are controlled during the entry and targeted by post-aerocapture propulsive correction maneuvers. The remaining two orbital parameters, argument of periapsis and true anomaly are not targeted. The true anomaly describing the location of the vehicle in the target orbit largely depends on arrival time at entry interface which may be changed by altering the interplanetary transfer. The argument of periapsis describes the angular location of the periapsis vector in the orbital plane. Because the periapsis location is largely determined by the pre-planned entry interface location, this element is also not directly targeted. For more information about the post-aerocapture propulsive maneuvers used to reach the target orbit, see the introduction to Chapter 5.

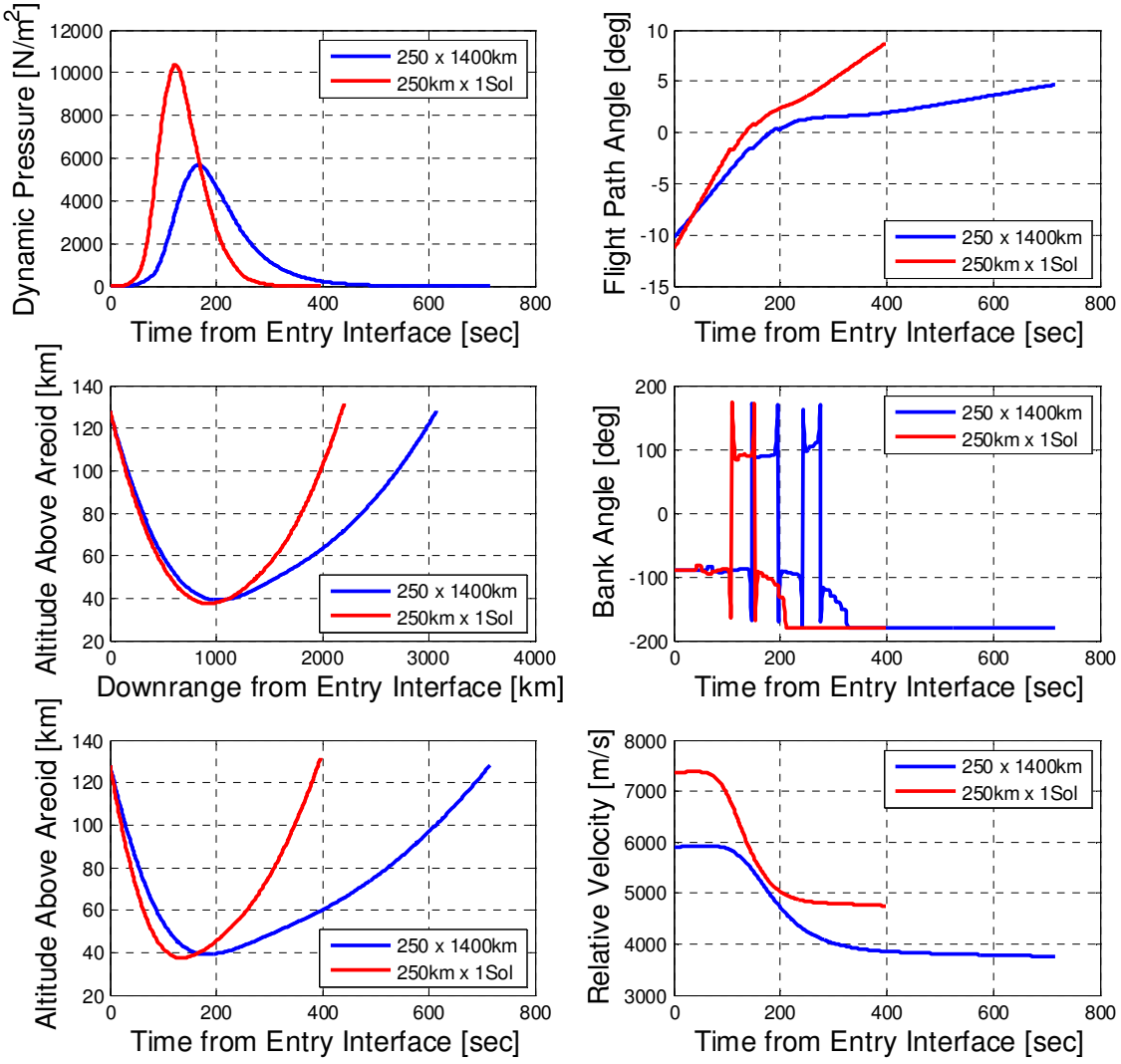


Figure 4.19: Nominal Guided Aerocapture Trajectories

The nominal trajectory resulting in a low-altitude orbit is similar to the original reference trajectory presented in Section 2.7. The major changes to this nominal trajectory relative to the original are due to the differences in the vehicle and the lateral bank reversal corridor. Recall that the vehicle was changed from a small capsule-shaped sample return vehicle to a larger ellipsled design useful for large robotic and human missions to enhance the applicability and extensibility of the study. In addition the bank reversal corridor was refined in order to better function with the NPC guidance. These changes resulted in altering the entry flight path angle from -9.67° to -10.27° below the local horizon. This permits the vehicle to attain the desired apoapsis with the longitudinal guidance deactivated.

Table 4.3: Target Orbital Conditions & Initial State for Nominal Entries

Target Orbital Conditions		
	Low Orbit	High Orbit
Apoapsis [km]	1400	33793
Periapsis [km]	250	250
Longitude of the Ascending Node [deg]	67.5	67.5
Inclination [deg]	90	90
Argument of Periapsis [deg]	N/A	N/A
True Anomaly [deg]	varies	varies
Entry Interface State		
	Low Orbit	High Orbit
Initial Altitude [km]	128	128
Inertial Entry Velocity [km/s]	5.9	7.36
Inertial Entry Flight Path Angle [deg]	-10.27	-11.33
Nominal Bank Magnitude [deg]	90	90
Initial Geocentric Latitude [deg]	70.0 S	70.0 S
Initial Longitude [deg]	67.5 E	67.5 E

Inspection of Figure 4.19 reveals an important difference between both trajectories, namely the longevity of the trajectories in the atmosphere. The trajectory targeting the high-altitude orbit enters the atmosphere over a kilometer per second faster than the trajectory producing the low-altitude orbit. The faster entry requires a steeper entry angle which penetrates to a lower altitude and then rapidly exits the atmosphere

once capture is assured. This steep entry is required to prevent the vehicle from quickly transiting through the atmosphere before sufficient drag losses accrue (i.e. skip-out). Once at low altitude deceleration occurs more rapidly for a vehicle at higher speeds because drag losses are generally proportional to the square of velocity. The vehicle must then rapidly exit the atmosphere before the high densities at low altitude remove too much energy and prevent achievement of the target orbit. The trajectory associated with the low-altitude orbit, however, is traveling at much slower speeds and is therefore required to linger at lower altitudes for a much longer time (nearly twice as long) and over a much longer downrange distance.

The differences in the distances and durations of these entries may become significant because it may impact the amount of learning that the PENS algorithm can accomplish in the entry. Shorter atmospheric exposure times and distances may translate to a lower level of associative learning because PENS is given less information from which general trends can be extracted. However, note that the trajectory resulting in the high-altitude target orbit has a larger exiting flight path angle. This permits the vehicle to transit the highly variable upper atmosphere quickly which will reduce the impact of the larger density perturbations typically found at high Martian altitudes. Thus, even though the high-altitude trajectory provides the PENS algorithm with less atmospheric exposure, the fast ascent may compensate for the decrease in available knowledge.

Both nominal trajectories are devoid of any atmospheric perturbations or variable dispersions such as those presented in Table 4.2 and both are produced using the KPENS density estimator in the inner guidance loop. The PENS/KPENS estimator parameters are presented in Figure 4.20 and illustrate that learning limitations due to limited atmospheric exposure may exist. Note that the trajectory resulting in the high-altitude target orbit (bottom row in the figure) follows very similar ensemble weight trends as the low-altitude trajectory, particularly in the first 100 seconds. However, after this time, differences in the ensemble weights begin to develop.

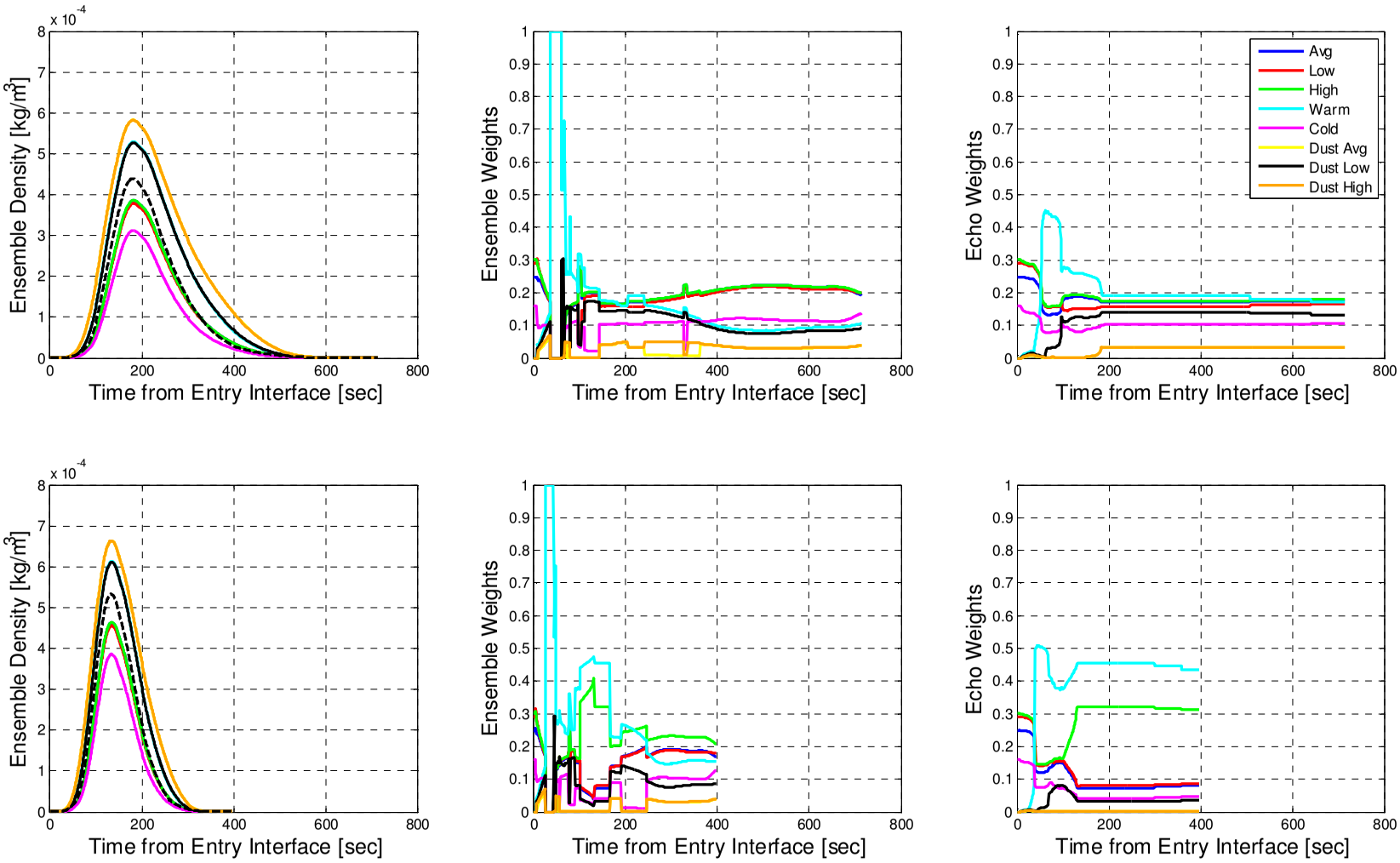


Figure 4.20: KPENS Estimator Quantities for Nominal Trajectories Targeting the Low-altitude Orbit (Top Row) and the High Altitude Orbit (Bottom Row).

Note how the ensemble echo in the high-altitude trajectory maintains higher weight on the Warm ensemble member relative to the low-altitude trajectory (i.e. in Figure 4.20, compare the light blue curves in the right-hand plots of both the top row and bottom rows). The same is true of several other ensemble members, especially after 100 seconds. Because all associative learning is due to the ensemble echo, these differences in the echo weights between both trajectories indicate that different associate learning processes are occurring. This, in turn, supports the notion that a faster trajectory may reduce the total amount of knowledge available to the PENS algorithm for learning. However, by 100 seconds, both trajectories are traveling in markedly different spatial regions of the atmosphere. Therefore, both vehicles are encountering different density trends which could also easily account for the changes observed in the associative learning processes. Further work will be presented in Chapter 5 when more data is available for statistical analyses.

Overall, both nominal guided trajectories attain adequate terminal performance. The low-altitude trajectory achieves an apoapsis error of 199 km over the target of 1400 km requiring a total ΔV of 72.4 m/s to correct to the desired target orbit (for complete description of post-aerocapture maneuvers, see the introduction to Chapter 5). The high-altitude trajectory achieves a much larger error of 7008 km above the target of 33,793 km but requires only 46.7 m/s to correct the orbit.

4.4 Atmospheric Dispersions & Perturbation Scheme

To increase the level of difficulty of the density prediction problem, perturbations are applied to the underlying density models provided by the MarsGRAM atmospheric simulation. MarsGRAM is prepackaged with its own set of density perturbations but, as will be shown, these do not always exhibit strong agreement with aerobraking observations. The proposed scheme combines several classes of perturbations which have been drawn from the relevant literature concerning the Martian atmosphere. This scheme

produces perturbed density trends which exhibit good formal agreement with density signals observed during aerobraking missions. In order to more fully agree with both observational data and theoretical predictions, these perturbations are then scaled to permit the perturbation magnitudes to grow exponentially with altitude.

4.4.1 Perturbation Magnitudes & Amplification with Altitude

Before describing the nature of the perturbations in detail, it is necessary to consider appropriate perturbation magnitudes. This question is critical because unjustifiably small perturbations will not stress the guidance system and will unfairly represent the variability of the true Martian atmosphere. Unjustifiably large perturbations will generate unrealistically pessimistic failure rates and lead to vehicle over-design. Fortunately, useful conclusions may be derived from density observations which have been recorded during aerobraking missions. Aerobraking missions consist of a series of high-altitude entries which gradually change the vehicle's orbit over time. Aerobraking periapsis altitudes are, therefore, generally much higher than aerocapture periapsis altitudes occurring near 90-100 km rather than 40-50 km for aerocapture. Despite this fact, aerocapture missions must transit higher altitude regions on the inbound and outbound legs of the trajectory. Therefore, knowledge of the density perturbation magnitudes in this region is quite useful. Three aerobraking missions have been successfully executed at Mars to date: Mars Global Surveyor (MGS) in 1997, Mars Odyssey in 2001, and Mars Reconnaissance Orbiter (MRO) in 2006[38]. Tolson et al. have examined every aerobraking orbit from all three missions and have extracted information concerning the nature and strength of the density perturbations[38]. Over a given orbit, the perturbed density signal is compared with a filtered/smoothed version of itself. The 39-second running mean is used to determine an average density to which the original noisy signal is compared (Figure 4.21, top of figure). The difference between the two signals

(bottom of Figure 4.21) permits a standard deviation of the density perturbations to be obtained which in this case is 0.12 or 12%.

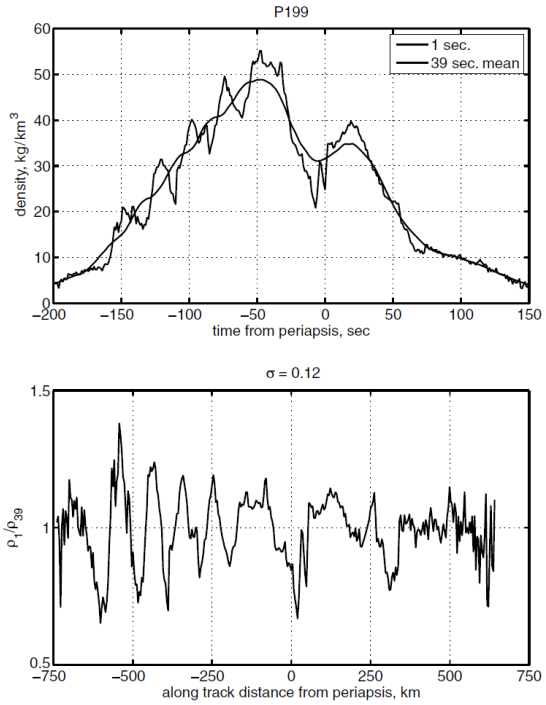


Figure 4.21: Extracting Perturbation Magnitude Information From Odyssey Periapsis 199[38]

Repeating this procedure over all available orbits produces a series of standard deviations which offers valuable insight into the density perturbation magnitudes which occur in high-altitude Martian orbits. Figure 4.22 displays these standard deviations as a function of periapsis latitude.

The strongest perturbations are seen early in the Odyssey and MGS aerobraking missions, labeled O1 and G1 respectively in the figure. Note that both of these points are near 60° North latitude which is over the winter mid-latitude jet, a high speed current analogous to the terrestrial jet stream. Fritts et al. have attributed the increase in variability in this region to the mid-latitude jet, theorizing that the jet permits small scale fluctuations to propagate to high altitude[2,13]. In order to develop realistic yet stressful atmospheric variations, the worst case density perturbations will be used. The worst case observed density perturbations occur during the Odyssey mission near 70° North latitude

and produce a standard deviation σ = 0.17 or 17%. Assuming that the density perturbations are normally distributed about the mean density, conventional design practice suggests that the 6σ perturbation magnitude, or approximately 100%, should be used for design. While stressful, this perturbation level is well within the variability captured by the neural ensemble, indicating that the PENS algorithm should be capable of functioning in this highly perturbed environment.

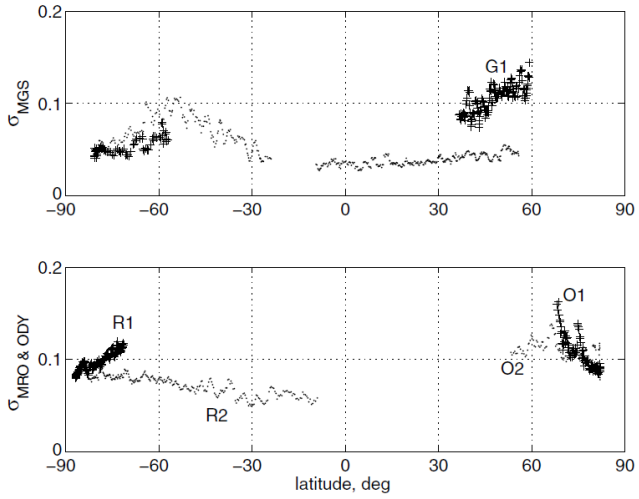


Figure 4.22: Standard Deviations of Observed Density Perturbations Over All Available Aerobraking Missions[38]

Utilizing the "6σ" worst case may seem overly conservative in some design contexts. However, this level of perturbations is likely to not represent the true 6σ perturbations in the Martian atmosphere. It should be noted that this observed worst case perturbation magnitude is derived from the spatially and temporally limited aerobraking data set. Given the highly variable nature of the Martian atmosphere, stronger perturbation magnitudes may very well exist, dependent upon the time, season, and location of the entry. Utilizing the known worst case provides stressful realism without exceeding justifiable perturbation limits.

In addition, it should be noted that the ±100% perturbation magnitudes derived from aerobraking standard deviations are based on the deviation of the 1-second density

trend from the mean density signal computed with a 39-second running average. As Figure 4.23 demonstrates, the 39-second mean atmosphere (bold dashed line) may also exhibit significant variations from theoretical models. The mean density trends in MarsGRAM, however, are generally symmetric about periapsis and do not exhibit such extreme variations. Because the proposed perturbation scheme is based on mean MarsGRAM models, utilizing large perturbation magnitudes will permit the generation of non-symmetric density signals with a high degree of formal similarity to observed aerobraking density trends.

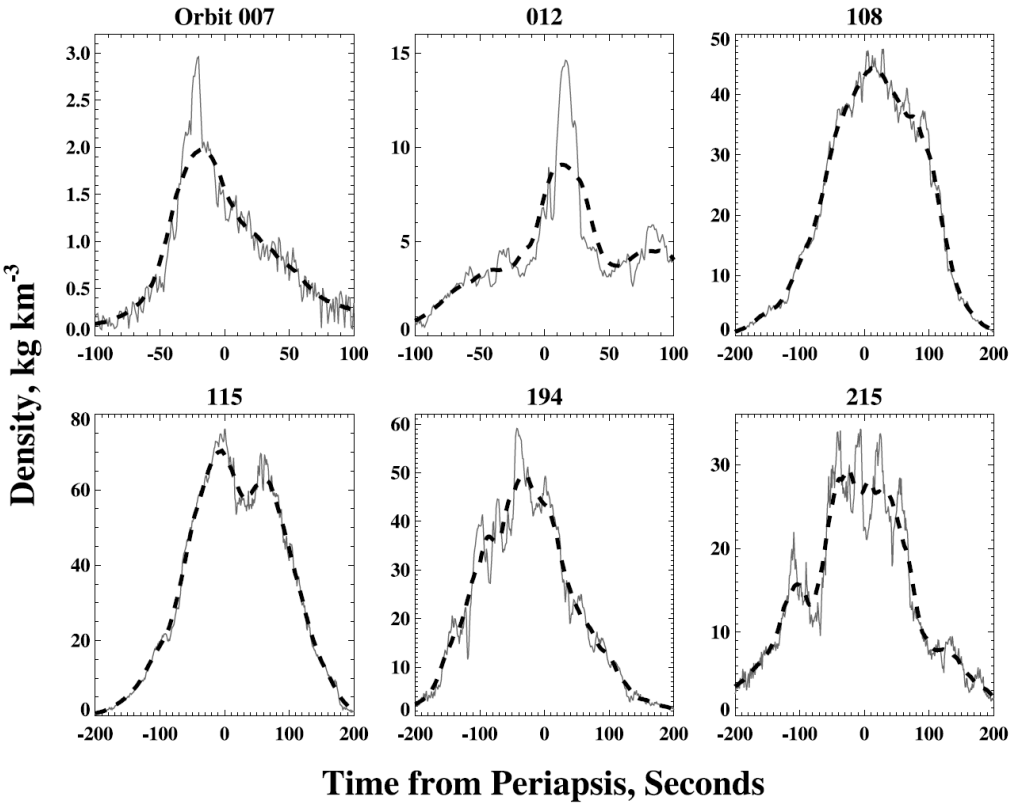


Figure 4.23: Odyssey Density Estimates at 1-Second Intervals (Thin Line) Compared with the 39-Second Running Mean (Bold Dashed Line)[121]

Tolson et al. state that small-scale density variations, such as those observed during a typical aerobraking entry, have generally been attributed to gravity wave activity propagating up to high altitudes. Tolson et al. conclude that "Almost every AB [Aerobraking] pass confirms those suggestions." An atmospheric gravity wave occurs

when a fluid parcel is forced out of stratified equilibrium by some perturbing force such as wind forcing an air mass over terrain features (typically referred to as orographic lift). The fluid parcel then moves vertically back toward its natural equilibrium altitude. However, due to its vertical momentum the fluid parcel overshoots the equilibrium location and then proceeds to oscillate towards equilibrium in similar fashion to a classical spring-mass-damper system. Attributing small-scale density perturbations to gravity waves is important as it impacts how the magnitude of the perturbations should vary with altitude. As Creasy et al. note, gravity waves grow exponentially with height; this permits small-amplitude disturbances near the surface to have profound effects at altitude.[122] In examining aerobraking data, they suggest that gravity wave sources other than terrain forcing (e.g. near-surface thermal contrasts, wind shear, convection, etc.) play a significant role. Similarly, recent versions of the MarsGRAM density perturbation scheme are also based on terrain-independent gravity wave models[123]. Prior versions of MarsGRAM utilized terrain-influenced gravity wave models but Justus and James[123] concluded that these models produced perturbation magnitudes which appear to change too rapidly in response to changes in local topography and temperature gradients. They now favor a simplified approach for varying the magnitude of the density perturbations with altitude; the standard deviation σ_D of the perturbation magnitudes may be expressed as

$$\sigma_D = 0.01(25 + z_s)\exp[(z-100)/40] \tag{4.10}$$

where z_S is the local topographic surface height and z is the altitude. Note that while this model is simpler than most terrain-influenced gravity wave models, it is still dependent on local terrain height. An even simpler approach which is fully independent of terrain relief is adopted here. The variation of the exponential term of Equation 4.10 with altitude is depicted as the blue line in Figure 4.24. Note that the magnitude of this term at 100 km, a typical aerobraking periapsis altitude, is unity. Because the proposed density perturbation scheme is based on aerobraking observations, the exponential term of

Equation 4.10 may be used to scale the density perturbation magnitudes with altitude while maintaining agreement with aerobraking observations. Thus, the growth of the perturbation magnitudes with altitude is consistent with the growth rate utilized in MarsGRAM while remaining independent of terrain relief as suggested by the relevant literature.

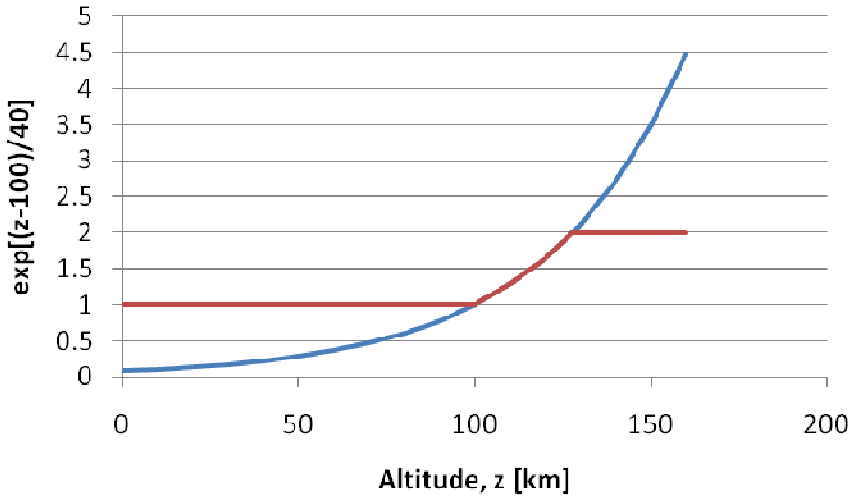


Figure 4.24: Growth of Density Perturbation Magnitudes with Altitude

Note that in raw form this equation would rescale the density perturbations near 50 km (a typical aerocapture periapsis altitude) to roughly one quarter of their value at 100 km. In the absence of significant density observations in the middle regions of the Martian atmosphere, reducing the perturbation magnitudes to small fractions of their observed values seems unjustified. Atmospheric phenomena other than gravity waves such as migrating and non-migrating tides, baroclinic instabilities, and atmospheric jets may play larger roles in the generation of density perturbations at these intermediate altitudes. For this reason, the perturbation magnitude multiplier will be limited to the interval [1,2] (i.e. the red line in Figure 4.24). This perturbation scheme is advantageous as it captures the exponential growth of the density perturbations with altitude but does not permit unjustifiably large or small perturbations to be created. Because the perturbations are based on a percentage deviation from the mean atmosphere, this scheme

also prevents the perturbations from becoming unjustifiably small as the mean density exponentially decays with altitude.

4.4.2 Perturbation Classes

In addition to perturbation magnitude, it is also necessary to prescribe the various classes of Martian density perturbations. Contemporary literature sources point to three major types of atmospheric density perturbations, namely; potholes[18-23], shears/biases[11,18], and random noise[2,36-39,44]. Recall from Section 1.2 that potholes are large magnitude density perturbations which have a finite length/duration (i.e. they have a definite beginning and end). The density inside the pothole is typically much higher or lower than the outside density and the boundaries of the pothole are typically sharp. Figure 4.25.a demonstrates an ideal simulated pothole while Figure 4.25.b-c demonstrates observed pothole phenomena (centered at 20 seconds and -10 seconds respectively) observed during Odyssey aerobraking operations.[38] By inspection, it is clear that the simulated ideal pothole is very different from the observed phenomena. Shortly, all three perturbation types will be combined in the same density signal and the result will be filtered to produce composite signals with high qualitative agreement to observations. However, first, the three perturbation types are introduced in isolation for demonstration purposes.

The second perturbation type, density shears, are similar to potholes in that they typically produce well-defined and large density changes. Often in the literature, shears are highly random phenomena which may occur multiple times with varying magnitudes[11,18]. However, because purely random perturbations are handled separately, shears will be defined here as single one-sided positive or negative changes in density. A simulated ideal density shear is plotted in Figure 4.26.a while a strong positive density shear observed during Odyssey aerobraking is visible in Figure 4.26.b near -15 seconds[38].

A second strong positive density shear observed during MGS aerobraking is visible in Figure 4.26.c near zero seconds (i.e. at periapsis)[36].

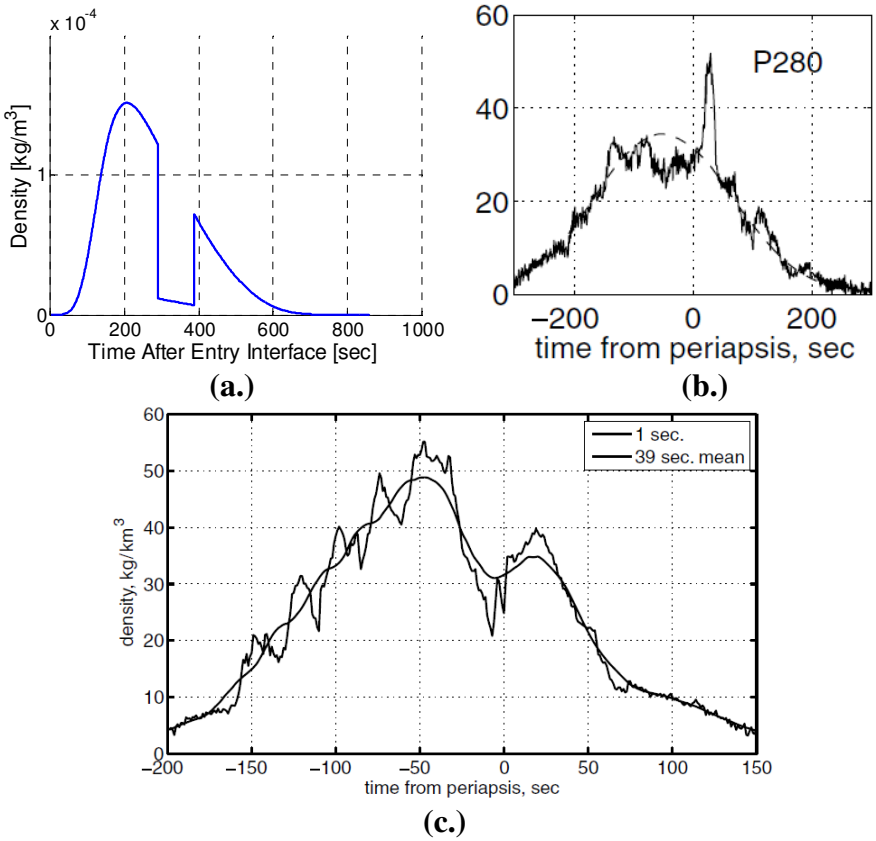


Figure 4.25: Examples of Pothole Density Perturbations: (a.) Simulated Negative Pothole, (b.) Odyssey Periapsis 280[38], (c.) Odyssey Periapsis 199[38]

The final type of density perturbations utilized herein is classic random noise. This noise signal is generated using a random number generator (i.e. uniformly distributed white noise) and is applied at every point in the density model. Figure 4.27.a is an example of high-magnitude pure random noise which will eventually become a perturbation component in the final density models. High magnitude noise such as this, taken in isolation, is physically unjustifiable based on observations in the Martian atmosphere (see Figure 4.27.b). However, when time-averaged filtering is applied, significant noise attenuation occurs.

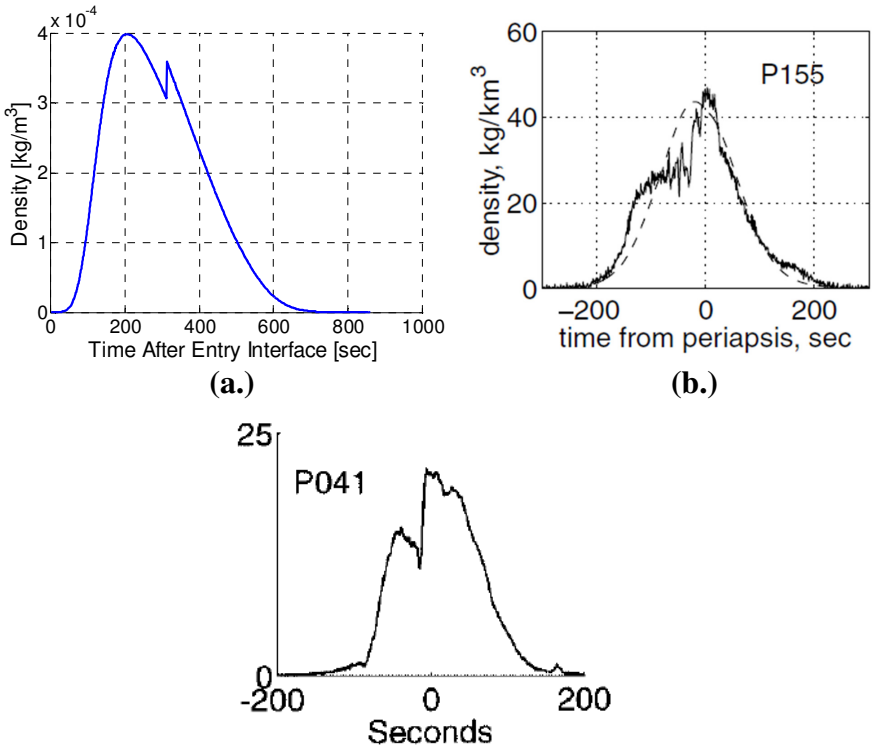


Figure 4.26: Examples of Shear Density Perturbations:
(a.) Simulated Positive Shear, (b.) Odyssey Periapsis 155[38], (c.) MGS Periapsis 41[36]

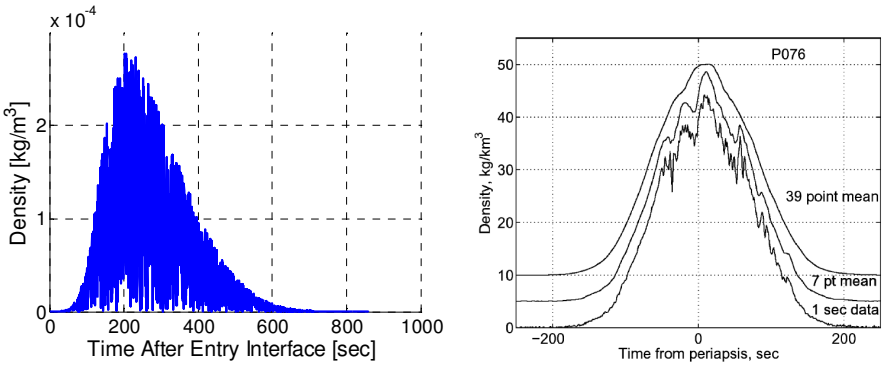


Figure 4.27: Examples of Random Density Perturbations
(a.) Simulated High-Magnitude Noise, (b.) Odyssey Periapsis 76[39]

Note that the mean density trend produced by MarsGRAM is a spatially resolved density field. In order to avoid the computational penalty required to generate a spatially resolved perturbation field, a perturbation vector of length 10,000 is created with each

entry initially set to unity. This vector will be used to store density perturbation multipliers. The perturbed density along the vehicle's trajectory is produced by multiplying the mean density at the vehicle's current location (from MarsGRAM) with a corresponding entry in the perturbation vector. At entry interface, the first element of the perturbation vector is used. Once the vehicle is three kilometers downrange of the entry point, the vector index is incremented by one and the second element of the perturbation vector is now used. This pattern repeats such that each element of the perturbation vector is utilized for three kilometers of downrange flight. Because the perturbation vector has 10,000 elements, it remains valid for roughly 30,000 km of downrange travel. This exceeds the circumference of the planet, ensuring that valid density perturbations will persist over the entire endo-atmospheric trajectory.

This approach ensures that the perturbations are fixed with respect to the Martian surface which is appropriate over the short duration of an average aerocapture entry. The foregoing method also assumes that the perturbations are only a function of the downrange distance from the entry point. This assumption is not limiting, however, because the typical cross-ranges traversed in the aerocapture trajectories herein are much less than the total downrange distances traveled (on the order of 150-200 km as opposed to 5000 km or more respectively).

To create the perturbation vector, each of the three perturbation classes is sequentially applied to the vector starting with the pothole component. Random pothole starting and ending locations from 1 to 10,000 are selected and a random perturbation magnitude on the interval [-100%, 100%] is chosen. For example, suppose the random starting and ending locations are 3000 and 4000 respectively and that a magnitude of -60% is chosen. In this case, the pothole will persist for 1000 entries in the perturbation vector and each element from index 3000 to 4000 in the vector is multiplied by 0.4 (because a 60% reduction in density is the same as multiplying the mean density by 0.4).

The shear and random noise components are added in a similar manner to the pothole component and the resulting density perturbation vector is filtered using the 39-point running average described above. As discussed, applying a running average filter is a common technique in aerobraking literature used to improve the signal/noise ratio of the density trend (see Figure 4.23, Figure 4.25.c, and Figure 4.27.b). It has been found that applying such a filter to the density model produces better qualitative agreement with observed aerobraking density trends.

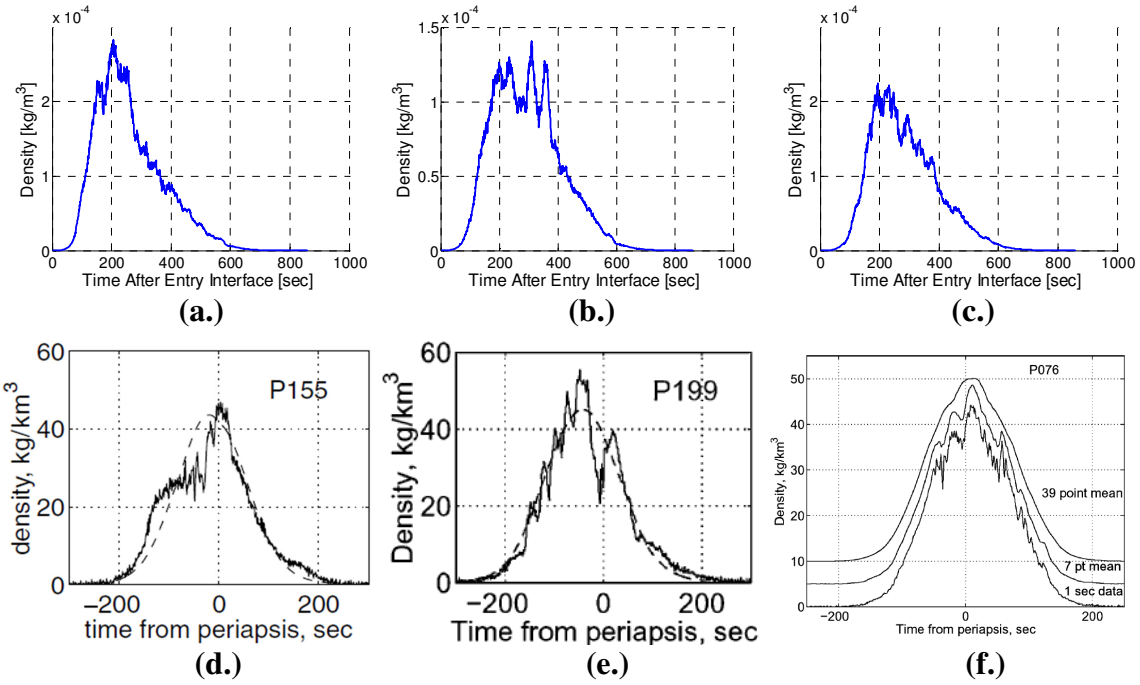


Figure 4.28: Composite Aerocapture Density Perturbations (a-c) Compared with Aerobraking Observations (d-f)

The results of noise component superposition and filtering on a smooth density signal may be seen in Figure 4.28 where several perturbed density models are compared with aerobraking observations. Note that the aerocapture trajectories are not symmetric about periapsis (which occurs at 200 seconds) like an aerobraking trajectory due to the significantly higher velocity changes which occur over an aerocapture. Comparing Figure 4.28.a with Figure 4.28.d reveals that both density signals contain large shears associated with lower levels of background noise. In Figure 4.28.b, large magnitude density potholes

develop in the high density region near periapsis. Comparing this to the observed density trend in Figure 4.28.e reveals that such perturbations are indeed possible in the Martian atmosphere. Finally, comparing Figure 4.28.c with Figure 4.28.f demonstrates that this perturbation technique is also capable of producing noisy density signals with smaller intermixed shears and potholes just as in the true Martian mesosphere. Note that the perturbation wavelengths in the simulated signals are somewhat smaller on average than the associated aerobraking wavelengths. This is appropriate because aerocapture entries occur at higher speeds than aerobraking entries which compresses the perturbations when viewed on a time-relative plot.

In summary, the density signals used for estimator testing are generated by superimposing three perturbation classes on the mean density field produced by MarsGRAM. The three types of perturbations, taken from the literature, are applied over the underlying trend and are filtered to produce qualitative consistency with aerobraking observations. This permits stressful testing scenarios to be developed while ensuring that the real-world atmosphere simulated by MarsGRAM is fully independent of the on-board ensemble developed using the Mars Climate Database.

4.4.3 Comparison to MarsGRAM Perturbations

Various versions of MarsGRAM have been utilized in many Mars aerobraking studies[6,87,89,91,92,97-99] to examine aerocapture vehicles and system requirements. As Cianciolo[98] et al. notes, MarsGRAM has become "a standard accepted throughout the field" and is therefore widely trusted. The well-deserved reputation of MarsGRAM led directly to its use as the "real-world" simulated atmosphere herein. However, as discussed in Section 4.4.1, the perturbation simulator in MarsGRAM has itself undergone significant revisions in recent releases as the communal understanding of the Martian atmosphere evolves. In order to highlight the qualitative changes produced by the perturbation scheme developed in Section 4.4.1-2, it is useful to compare the new

perturbations with those generated by MarsGRAM. Figure 4.29 depicts the differences between the standard and literature-derived perturbation schemes. The upper left plot displays the smooth, unperturbed density trend along the entry corridor for a randomly selected NPC-guided trajectory. As discussed in Section 4.2.2, the mean MarsGRAM density field is perturbed by varying two of the major atmospheric drivers; the amount of dust suspended in the atmosphere and the solar flux incident on the thermosphere. The standard MarsGRAM perturbation model when applied to this mean atmosphere produces the green density signal in the lower left of the figure. The new literature-derived perturbation model is applied to the same mean trend to produce the red density signal in the upper right of the figure.

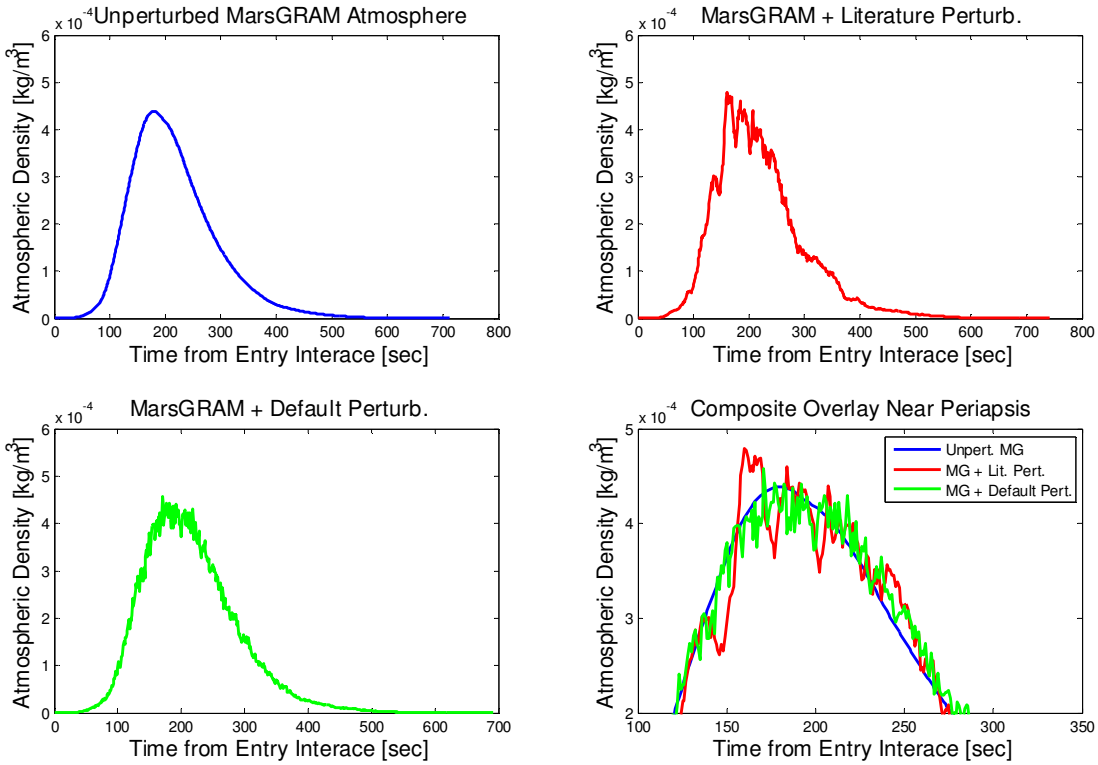


Figure 4.29: Comparison of Literature-Derived Density Perturbations to MarsGRAM Density Perturbations

The lower right plot in the figure offers a composite view which overlays all of the density signals in the critical periapsis region. Note that the choice of the density model does impact the trajectory as is evidenced by slight differences in the simulation ending

times. Also note the differences in both form and magnitude between both perturbation schemes. It may be clearly seen by inspection that the MarsGRAM signal has a relatively high degree of white noise content. Purely random, high frequency density perturbations in which the positive and negative noise components are in balance (i.e. white noise) offers little difficulty for many entry guidance algorithms. This is due to the fact that the differential momentum change imparted by the perturbations remains small in magnitude. Consider the difference between an entry vehicle encountering a short-duration perturbation (say a 2 second pulse of +50% magnitude) and a much longer perturbation (say a 2 minute pulse of the same magnitude). The change in momentum from the first short pulse will be significantly less than the second. Now consider an entire train of white noise density pulses all of short duration and small magnitude. The positive and negative momentum differences begin to have a self-cancellation effect and the entry vehicle is largely unaffected. The situation is not unlike a terrestrial wheeled vehicle running over small imperfections in a roadway surface at high speed. Each imperfection in the road has negligible effect and the result is a relatively smooth ride. Just as in the wheeled vehicle analogy, this situation changes, as the magnitudes and durations of the perturbations become larger. For guided aerocapture entries, the guidance system must continually correct for the momentum errors generated by these perturbations which may stress the guidance into one of the failure modes discussed in Section 1.3 in extreme cases. Even if a failure mode is not entered, these perturbations eventually impact the final apoapsis altitude due to the momentum error imparted over the trajectory. For example, the trajectory using the standard MarsGRAM perturbations achieved an apoapsis altitude which is only 19.7 km different from the smooth/unperturbed trajectory while the trajectory using the literature-derived perturbations is 88.9 km from the unperturbed apoapsis. Given that the literature-derived scheme is in agreement with observed density perturbations, the standard perturbation scheme used in MarsGRAM may be optimistically small. If this is the case, then these perturbations may not replicate

stressful test cases which are most useful for mission/vehicle design and guidance testing. As just demonstrated, the standard perturbations may lead to optimistic designs which are not equipped to survive severe fluctuations present in the Martian atmosphere. This new perturbation scheme, therefore, is designed to offer stressfully realistic density signals in order to evaluate the ability of the PENS algorithm to reduce atmospheric uncertainty and improve aerocapture performance.

CHAPTER 5

GUIDED TRAJECTORY SIMULATIONS

The performance of the PENS-augmented NPC guidance developed and validated in the previous chapter is now examined in detail. Trajectory Monte Carlo simulations which comprise Phase III of the PENS analysis cycle (recall the discussion on the PENS analysis phases in Section 3.3) were conducted on the 200 node i4 Linux cluster located at the NASA Langley Research Center. Over 144,000 aerocapture trajectories have been simulated under a variety of experimental conditions. This large sample size provides a sufficient degree of statistical evidence in order to derive estimated failure rates for all of the density estimators under study, both novel and contemporary.

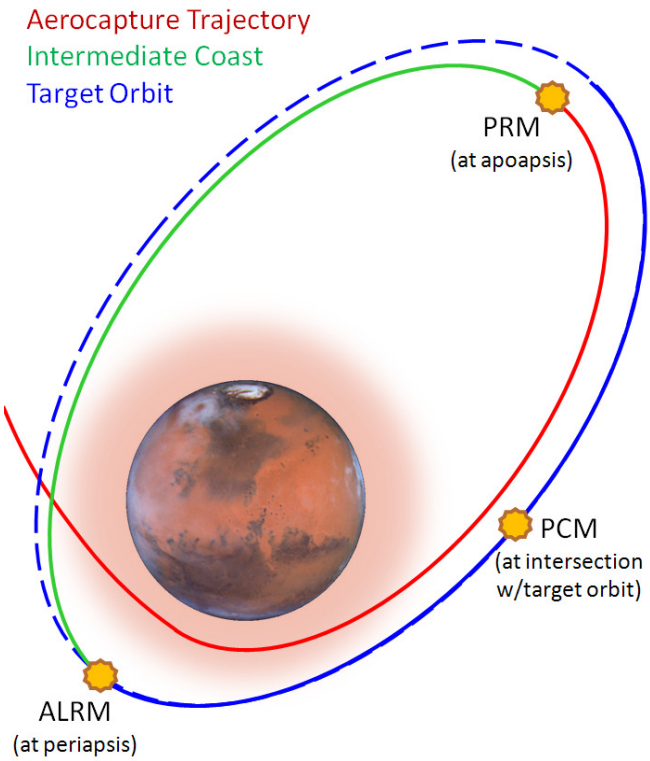


Figure 5.1: Post-Aerocapture In-Space Propulsive Maneuvers to Achieve Target Orbit

Following each atmospheric entry, all the successful cases execute a series of in-space propulsive maneuvers (Figure 5.1) designed to remove any remaining residual errors between the post-aerocapture orbit and the target orbit. These maneuvers may or

may not be autonomous depending on the exact nature of the mission requirements. The three maneuvers proposed herein may also be decomposed into smaller maneuvers after Earth contact has been made and detailed orbit determination conducted. Note, however, that the three proposed maneuvers taken together provide an estimate of the total ΔV necessary to reach a desired target orbit from an immediate post-aerocapture orbit.

The standard POST2 orbital ΔV computation routine is used to create a three-burn correction strategy which begins as the vehicle reaches apoapsis following atmospheric exit. Upon reaching apoapsis, POST2 executes a Periapsis Raise Maneuver (PRM) in the direction of motion (counter clockwise in Figure 5.1) which elevates the periapsis point to 250 km, well above the sensible atmosphere. The vehicle then executes a coast phase to this new periapsis altitude and performs the Apoapsis Lower/Raise Maneuver (ALRM) to correct the achieved apoapsis to the desired target apoapsis. This maneuver may either be in the direction of motion if the achieved apoapsis is smaller than the target or opposite the direction of motion if the achieved apoapsis is larger. At this point, the only remaining errors correspond to a misalignment of the current orbital plane with the target orbit. Another coast phase follows until the vehicle reaches the nearest node crossing at the point where the current and target orbits intersect. At the node, the final burn called the Plane Change Maneuver (PCM) occurs to place the vehicle into the proper orbit. This approach permits four of the six orbital elements to be successfully targeted as discussed in Section 4.3.3.

POST2 uses analytical two-body orbital mechanics to compute the required ΔV for each maneuver. The ΔV for each of the three maneuvers is then summed to determine the required post-aerocapture ΔV_{total};

$$\Delta V_{total} = \Delta V_{PRM} + \Delta V_{ALRM} + \Delta V_{PCM} \tag{5.1}$$

In addition to the computation of the total ΔV, all of the cases are screened to flag any failures that may have occurred. These cases are then used to define the aerocapture failure rate λ;

$$\lambda = (\#\ of\ failed\ cases)/(\#\ total\ cases) \tag{5.2}$$

The aerocapture failures defined here are divided into two general categories; hard and soft failures. Hard failures are those from which the entry vehicle cannot reasonably recover even with the expenditure of very large ΔV. The hard failures for which the trajectory cases are screened include three specific failure types;

1.) Crashes: The vehicle impacts the surface of the planet while completing the entry.
2.) Impending Crashes (rare): The vehicle never re-achieves the entry interface altitude of 128 km. Without the ability to exit the atmosphere, a crash will occur within a very short period without the expenditure of inordinately large ΔV which an aerocapture vehicle is unlikely to carry.
3.) Skip-Out: The vehicle is unable to remain in the atmosphere for sufficient time to reduce its velocity below escape speed.

Unlike hard failures, soft failures occur when larger yet reasonable expenditures of ΔV are required to achieve the target orbit. All soft failures achieve a stable post-aerocapture orbit but the achieved orbit is markedly different from the target orbit.

5.1 Impact of Aerobraking-Derived Density Perturbations

The first series of trajectory simulations are dedicated to examining the impact of the newly proposed density perturbation scheme on aerocapture guidance and performance. Recall that the goal of these new perturbations was to utilize literature-derived perturbation components (potholes, shears, and biases) to create perturbed atmospheres which qualitatively agree with Mars aerobraking density observations. The relevant literature suggests that strong pothole and shear perturbations may cause significant guidance faults and failures to occur (see Sections 1.2-3). Therefore, it is expected that these perturbations will cause an increase in the average ΔV_{total} required for orbit correction and an increase in the number of failure cases. It is further theorized that the PENS algorithm, using a more accurate and more flexible underlying density

prediction model, will be able to better compensate for these perturbations than traditional methods. Figure 5.2 displays the performance of the HYPAS guidance system[10] which employs an analytical predictor-corrector scheme for final state targeting. The data presented in this figure was drawn from a study[99] of aerocapture performance for inflatable decelerators with low ballistic coefficients and low L/D.

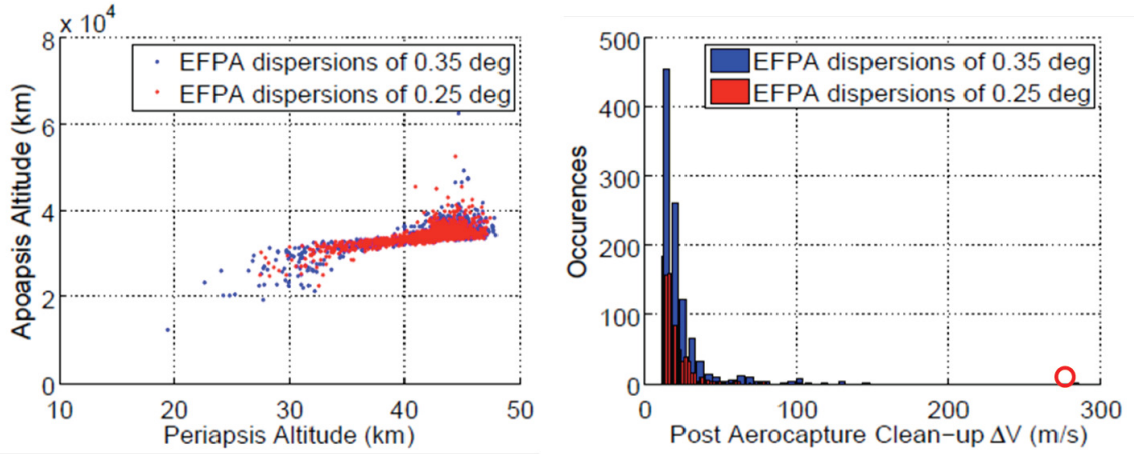


Figure 5.2: HYPAS Aerobraking Performance Using Standard MarsGRAM Perturbations[99]

In order to evaluate the impact of the literature-derived perturbation scheme, an analogous simulation was performed using the NPC guidance scheme developed herein. First, a control experiment was conducted. The results of this simulation (Figure 5.3) were obtained using the standard MarsGRAM perturbation model in order to evaluate the differences between the HYPAS and NPC guidance results. Note that the orbit dispersion plots on the left-hand sides of Figures 5.2 and 5.3 depict the immediate post-aerocapture orbit state, not the orbit state once the corrective ΔV is applied.

The $\rho_{K\text{-}EXP}$ density model described by Equation 3.16 is used in the NPC guidance inner loop rather that the PENS estimator. Because HYPAS uses the same $\rho_{K\text{-}EXP}$ density estimator, this approach permits the NPC scheme to approximate the performance of HYPAS. In addition to the guidance atmosphere model, both simulations utilize the same target orbit (250 km x 1Sol), a similar corrective ΔV strategy, and nearly identical atmospheric dispersions. The HYPAS results in Figure 5.2 contain data for two different

entry flight path angle dispersion ranges; ±0.25° and ±0.35°. Because the entry flight path angle dispersions used for NPC simulations (from Table 4.2) are ±0.25°, the data presented in red in Figure 5.2 are most analogous to the data in Figure 5.3. The primary differences in the two data sets stem from differences in the vehicle characteristics and the number of dispersed variables. Due to its low L/D, the vehicle used in the HYPAS study has less control authority and is therefore more affected by variable dispersions. However, the HYPAS study employed only five dispersed variables as opposed to the nineteen variables dispersed herein.

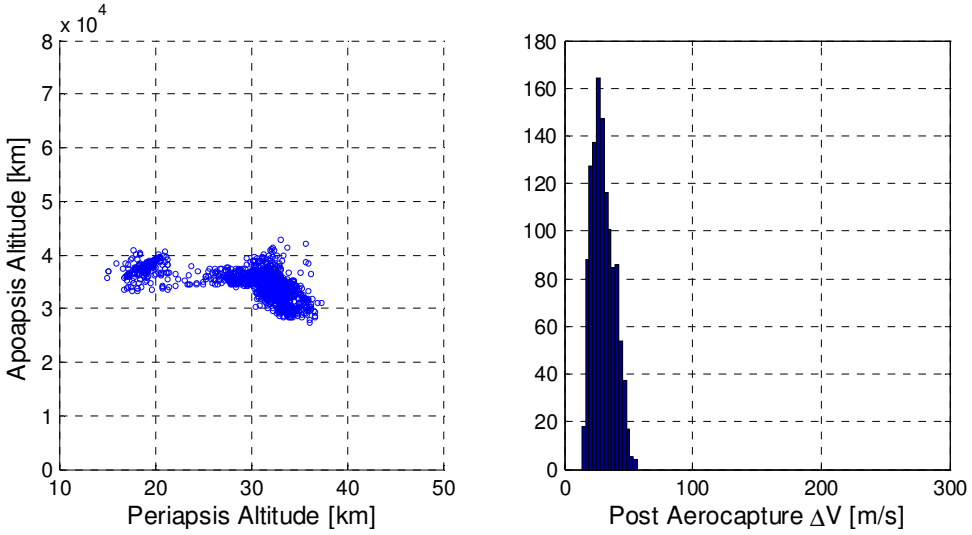


Figure 5.3: Analogous NPC Aerobraking Performance Using Standard MarsGRAM Perturbations

Despite the differences in the vehicle characteristics and number of dispersed variables, some similarity may be seen in an analogous Monte Carlo data conducted using the NPC guidance scheme. Both simulations have similar mean ΔVs and both achieve the desired target apoapsis of 33793 km with relatively small errors. The NPC guidance scheme produces very well contained post-aerocapture orbits due to the accuracy of the numerical inner loop. This observation matches the expectation from the literature (recall Sections 4.1-4.2 and the discussion of the PredGuid NPC guidance used to enhance the landing accuracy of Orion). The HYPAS guidance, using approximated equations of motion and a lower L/D vehicle, produces larger apoapsis and periapsis

dispersions which directly translate into more ΔV outliers. However, despite these differences, the HYPAS and NPC mean ΔVs and orbit characteristics are relatively similar using the standard MarsGRAM perturbations. The same NPC simulation is repeated, only now the literature-derived density perturbations are utilized (Figure 5.4) instead of the MarsGRAM perturbation model. The maximum observed ΔV increases dramatically up to 1163 m/s. While this increase is quite large, it should be noted that only 7% of these new cases have a ΔV greater than the maximum ΔV associated with the original HYPAS data which use the standard MarsGRAM perturbations. Recall that the literature-derived perturbation model exhibits a high degree of qualitative similarity to *in situ* density measurements taken during Mars aerobraking missions (see Section 4.5). The perturbation magnitudes utilized in this simulation, while large, correspond to the observed 6σ density perturbations. Given the prevalence of extreme density variations directly observed in the Martian atmosphere, use of severe yet realistic perturbations as design cases should be given serious consideration. Because PENS was designed to prevent guidance failure under highly perturbed conditions, the algorithm should be tested in harsh environments. The goal of the literature-derived perturbation scheme introduced here is to produce such an environment until additional measured data on the Martian middle atmosphere is available.

The results discussed above still utilize the $\rho_{K\text{-}EXP}$ density model, a common conventional density estimator. The same simulation is once again repeated and now the $\rho_{K\text{-}EXP}$ model in the inner loop is replaced with the PENS estimator. As may be seen in Figure 5.5, the PENS estimator produces noticeable improvements in the performance of the NPC guidance scheme. The PENS algorithm generates a 69% decrease in the maximum ΔV to 359 m/s, the maximum apoapsis error decreases by 63%, and the range of periapsis altitudes also decreases by 18%. Thus, the associative learning process and improved underlying atmospheric physics have their intended effect of reducing large final state errors. Furthermore, the hard failure rate for the NPC guidance using the $\rho_{K\text{-}EXP}$

model was 5.5% but the PENS-augmented NPC guidance produced a lower hard failure rate of 3.8% for the above simulation. Therefore, in addition to improving terminal performance, the PENS algorithm also reduces the frequency of prevalent guidance failure modes leading to irrecoverable trajectory states.

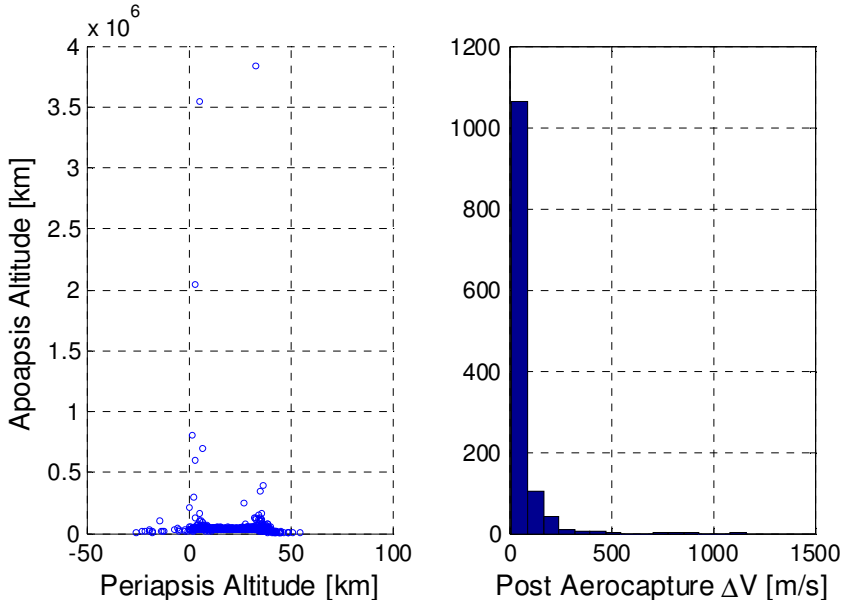


Figure 5.4: NPC Aerobraking Performance Using Literature-Derived Perturbations

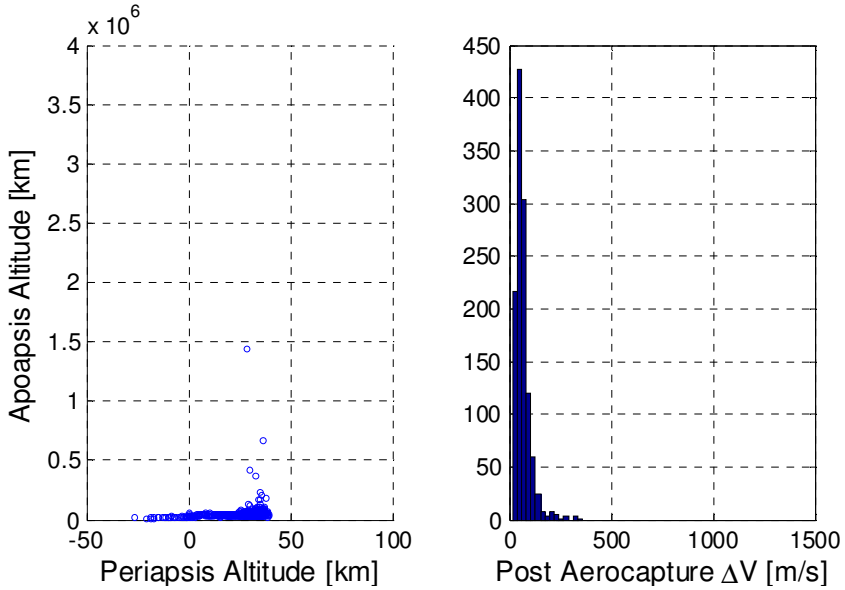


Figure 5.5: PENS-Augmented NPC Aerobraking Performance Using Literature-Derived Perturbations

Now that the basic premise of the PENS algorithm seems to show promise, it is useful to find a method for compactly presenting the information given in the foregoing discussion. Aerocapture is a useful systems-level tool because it can reduce the ΔV required to insert a vehicle into an orbit about a planetary body with an atmosphere. This, in turn, often has beneficial effects on other elements in a mission architecture. For example, a reduction in the ΔV required aboard a Mars-bound vehicle may permit the use of a smaller and more affordable Earth launch vehicle. Because the benefits of aerocapture are often realized at the systems level, it is important to examine the impact of aerocapture enhancement technologies from a system level perspective. How, then, can the system level benefits of an aerocapture enhancement technique be measured in general, independent of the assumptions or requirements of a specific mission architecture? This question may be approached by examining the basic functional requirement of aerocapture. That is, any given aerocapture trajectory must safely deliver its payload to the intended target orbit. The functional decomposition of this requirement then follows naturally. The successful aerocapture trajectory must:

1.) Be Safe: The likelihood of hard and soft failures must be minimized such that the vehicle will successfully complete the entry and exit the atmosphere.
2.) Reach the Intended Target Orbit: The on-board ΔV budget must be sufficient to place the vehicle in the precise target orbit otherwise vehicle/crew safety and mission objectives may be compromised.

These sub-functions may then be readily translated into measurable metrics. The first sub-function, safety, is most readily measured by the probability of failure of the aerocapture (i.e. the total failure rate, λ). The second sub-function is best measured by comparing the required ΔV to the available ΔV aboard the vehicle (i.e. the ΔV budget). Effective aerocapture enhancing technologies are those which reduce the required ΔV budget without significantly raising the failure rate. That is, the systems-level cost of aerocapture (the required ΔV budget) is reduced at an equivalent or smaller level of risk

(the failure rate, λ). This notion of comparing cost and risk for aerocapture is appropriate because aerocapture reduces the operational cost (i.e. ΔV) of orbit insertion by relying on a randomly varying planetary atmosphere which increases operational risk.

Note that other aerocapture cost sources exist (e.g. the mass of the thermal protection needed for entry, etc) and so too do other risk sources (e.g. payload damage due to thermal soak-back, etc). The nature of these other risk and cost sources, however, is often architecture specific. In addition, if the ΔV budget and failure rate, λ are not kept reasonably small, then the use of aerocapture in a given architecture is contraindicated making the other cost and risk sources moot. Thus, the failure rate λ and the ΔV budget are two primal variables for examining aerocapture cost and risk. We therefore seek a means to map the cost, or ΔV, of an aerocapture to its associated level of risk, or failure rate λ.

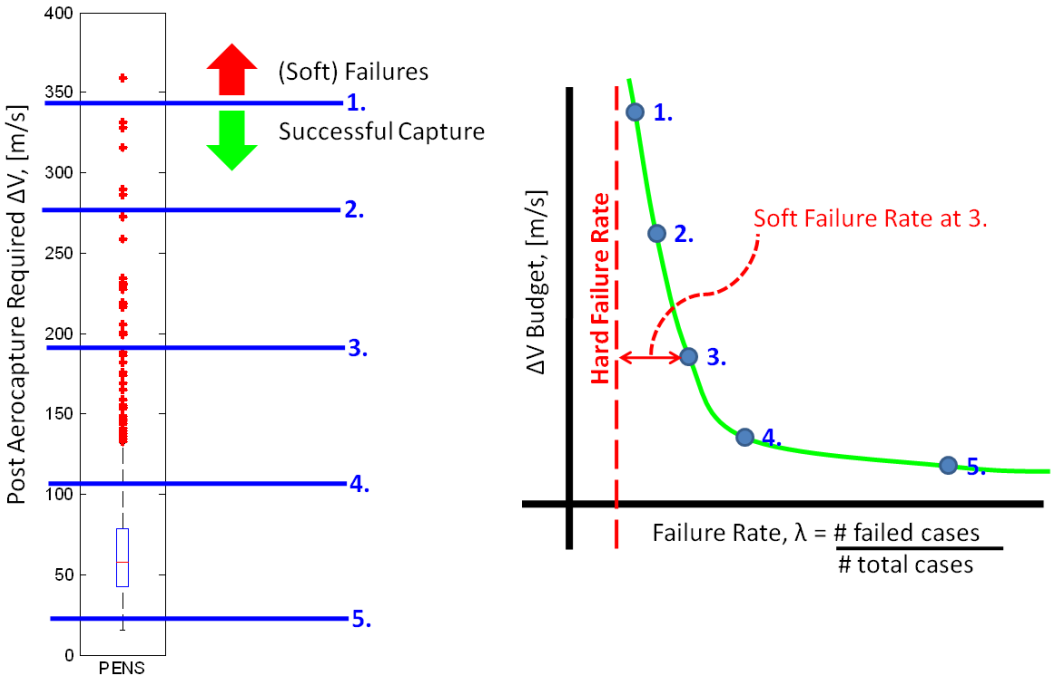


Figure 5.6: Construction of the ΔV-λ Diagram to Map Aerocapture Cost to Risk

Figure 5.6 provides a means for accomplishing this mapping. The left side of the figure displays a box plot of the PENS-augmented ΔV distribution (i.e. the same data

presented in the ΔV distribution in Figure 5.5). Note that hard failures do not appear on the box plot because no reasonable amount of ΔV can save the hard failure cases.

Intuitively, as the post-aerocapture ΔV budget becomes very large, the vehicle will be able to correct for larger and larger atmospheric guidance errors and the soft failure rate will approach zero. Conversely, as the ΔV becomes very small, more and more atmospheric guidance errors will become uncorrectable and the soft failure rate will increase. To formalize this, let the horizontal blue bars in the figure represent a variable ΔV budget. At the bar marked with a one, the ΔV budget is very large and only one case cannot be corrected to the desired target orbit because the required ΔV is larger than the ΔV budget. Therefore, the failure rate is low but the ΔV budget is large. On the right-hand side of Figure 5.6, the point design defined by the first ΔV budget bar is plotted on a set of axes mapping the ΔV budget to the failure rate λ. As the ΔV budget is progressively lowered (i.e. the blue bars marked with 2, 3, etc) the corresponding points on the right-hand plot trace out a tradeoff between the ΔV budget and the failure rate. By the time the budget reaches point 5, the majority of the cases cannot be corrected and the failure rate diverges asymptotically near the horizontal axis of the ΔV-λ diagram. In the reverse situation where the ΔV budget is increased to very large values, the ΔV-λ curve asymptotically approaches the hard failure rate line (red dashed line in Figure 5.6). The horizontal difference between this asymptote and the ΔV-λ curve represents the soft failure rate at the current ΔV budget.

The ΔV-λ diagram is a compact and intuitive way of visually presenting the critical information concerning aerocapture cost and risk. The diagram also provides readily visible context for judging the utility of various density estimators relative to one another. This plot is presented here as it will be heavily used in the following sections to present further results.

5.2 Effect of the Target Orbit

The remaining sections of this chapter, save for the discussion of lidar in Section 5.5, will be dedicated to exploring the impact of the PENS algorithm on both the design and operation of aerocapture systems. The results presented in the previous section utilized only a small number of trajectory simulations (approximately 1200) for proof-of-concept testing. The simulations presented in this section, however, are larger with 6000 simulations per density estimator in order to attain sufficiently large sample sizes to compile adequate failure rate data. In this section, the effect of the target orbit on the performance of the PENS-augmented NPC guidance scheme is explored.

The efficacy of a density estimator is measured by the proximity of its ΔV-λ curve to the origin. Density estimators which provide low failure rates at smaller ΔV budgets clearly deliver superior performance because they reduce both cost and risk. Operation closer to the origin is beneficial on two levels. First, in real time application, the estimator in question directly leads to increases in operational safety during the entry. Second, the density estimator also permits reductions in either the ΔV budget, the level of risk, or both. This increases the system level utility of aerocapture by generating architectural benefits such as mass reductions making aerocapture-based architectures more attractive.

Recall that eight MCD climate models have been summarized as neural networks (NNs) (see Section 2.4). These are now separated into three different climate ensembles in order to evaluate the impact of storing various amounts of information aboard. As previously discussed, these NNs are highly compact equations summarizing thousands of data points with a series of coefficients. The data compression ratio attained for these NNs is roughly 70:1 meaning that 70 atmospheric data points from MCD are replaceable by one NN coefficient. Despite the efficiency of this method, additional NNs translate to larger on-board storage and computational requirements. The number of PENS operations is proportional to $n+1$ where n is the number of ensemble members and the

additional one represents the ensemble echo. Therefore, it is useful to determine the efficacy of the PENS algorithm using smaller ensembles. The three climate ensembles are examined for study:

1.) PENS-8: Consists of eight members corresponding to the eight climate models from the MCD. This ensemble extensively describes the possible conditions in the entry corridor at the time of entry for varying levels of dust and solar flux. By including the three dust storm models, this is the only "complete" ensemble because all climate scenarios, even extreme weather phenomena, are considered.

2.) PENS-5: Consists of all the ensemble members with the exception of the dust storm models. PENS-5 could best be described as a "nearly complete" ensemble because it extensively models the entry corridor without including less likely extreme weather scenarios.

3.) PENS-3: Consists of only three of the PENS-5 models corresponding to the average density and the two extreme density models outside of a global dust storm. PENS-3 is best described as a "bare minimum" ensemble because (a.) no extreme weather phenomena are considered and (b.) the smallest number of models which describe the possible conditions are utilized.

The results from each of the above ensembles are reported separately in the data discussed in this section. However, effects of these various ensembles on guidance performance are examined in greater detail in the next section.

Figure 5.7 depicts the first full Monte Carlo analysis performed using all eleven density estimators of interest (defined in Table 3.1) for the 1 Sol orbit. The total number of trajectory cases completed for each estimator is approximately 6000. The precise number varies by a handful of cases on either side of 6000 because the estimator for a given trajectory is randomly selected by the Monte Carlo simulation.

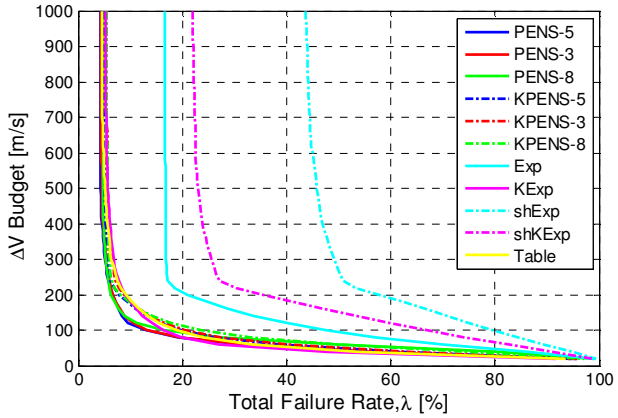


Figure 5.7: ΔV-λ Diagram for 1 Sol Orbit

Each of these cases are subjected to the literature-derived perturbations of Section 4.5 as well as all of the dispersions listed in Table 4.2. As can be immediately seen, both of the short exponential models (labeled *shExp* and *shKExp* in the figure legend) as well as the least squares exponential model (labeled *Exp*) exhibit large hard failure rates under highly perturbed atmospheric conditions. Examining the critical region near the origin in more detail (Figure 5.8), the cost and risk reduction realizable with the PENS algorithm relative to other historical density estimators becomes clear.

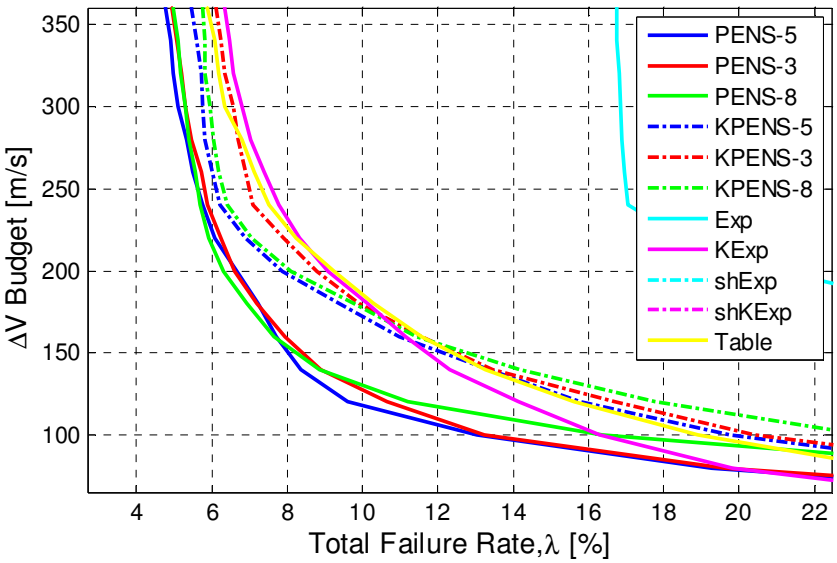


Figure 5.8: ΔV-λ Diagram Near the Origin for 1 Sol Orbit

Selecting design points near the knee of the ΔV-λ curve is advisable because this is the location where the ΔV budget and the failure rate λ are both minimized. Recall that the failure rates used on the diagram correspond to a highly perturbed, 6σ Martian atmosphere and are therefore, desirably conservative. If a 6σ failure rate of 8% is considered programmatically acceptable for design, the best historical estimators are the ρ_{TABLE} model and the $\rho_{K\text{-}EXP}$ model (labeled *Table* and *KExp* respectively in the figures) which require a ΔV budget of over 225 m/s. At an equivalent level of risk, all three PENS estimators require a ΔV of roughly 150 m/s.

Similarly, if a design ΔV budget of 150 m/s is selected, then the best historical models produce a failure rate of roughly 12% whereas the PENS failure rate is only 8%, again a 4% reduction in the total failure rate. Therefore, the PENS algorithm achieves reductions in the required ΔV at a constant failure rate λ or achieves reductions in the failure rate λ at constant ΔV. If operating in the knee of the curve as suggested, then the PENS estimators produce a point design with λ = 8.5% and ΔV = 145 m/s. The knee of the ρ_{TABLE} and $\rho_{K\text{-}EXP}$ models both occur near λ = 10.5% and ΔV = 175 m/s. Therefore, when designing in the critical knee region of the ΔV-λ diagram (for the 1 Sol orbit), the PENS algorithm generates reductions in either the failure rate or the required ΔV.

If designing at very low failure rates to ensure a successful aerocapture even under highly perturbed atmospheric conditions, the PENS algorithm is very beneficial. The lower hard failure rates achieved by PENS shifts their associated ΔV-λ curves horizontally to the left relative to the historical estimators. This, in turn, makes low risk designs more achievable with reasonable ΔV budgets. For example, to achieve a failure rate λ = 4.5%, the ρ_{TABLE} estimator requires a ΔV budget of 960 m/s. The PENS-5 estimator, however, can achieve the same level of risk with a ΔV budget of only 400 m/s, a 58% reduction in required ΔV. Note that, in this case, the $\rho_{K\text{-}EXP}$ model has a hard failure limit of 5.2% and is therefore unable to meet the 4.5% failure rate criterion at all.

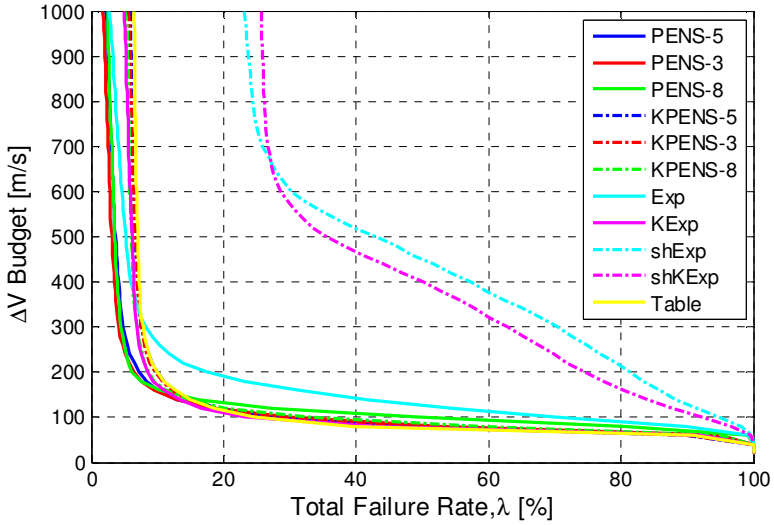


Figure 5.9: ΔV-λ Diagram for 1400 km Orbit

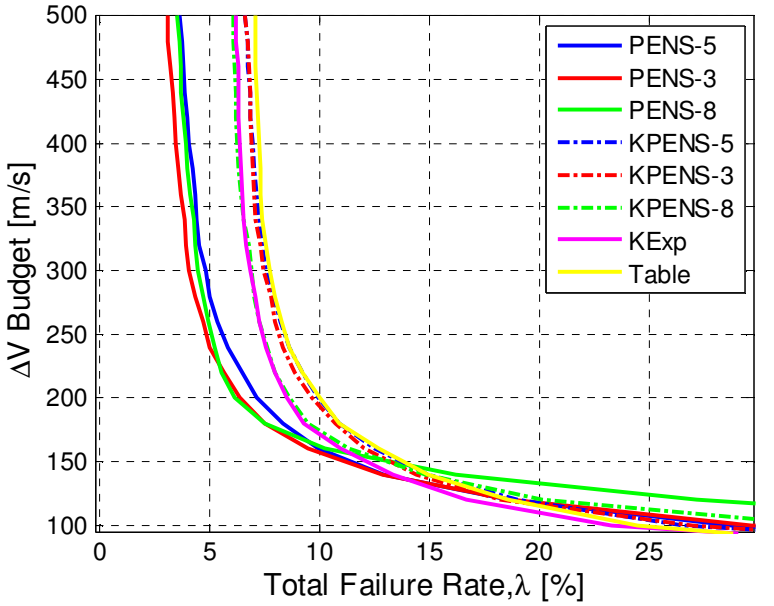


Figure 5.10: ΔV-λ Diagram for 1400 km Orbit Near the Origin

Similar general conclusions may be drawn when examining the ΔV-λ diagram for the 250x1400 km target orbit (Figure 5.9). Again the short models and the pure exponential models (labeled *shExp*, *shKExp*, and *Exp* respectively) are not competitive with the other methods. Figure 5.10 presents the same 1400 km dataset in the critical knee region near the origin. From inspection of both figures, it may be clearly seen that the ΔV-λ behavior of PENS is distinctly different from the 1 Sol orbit. The primary effect

of PENS for the 1400 km orbit is to shift the ΔV-λ curve horizontally to the left relative to the historical estimators. For the 1 Sol orbit, however, the PENS ΔV-λ curves exhibit both vertical and horizontal shifts towards the origin. This disparity is due to the nature of the entry trajectories required to generate the two different orbits as all other variables between both simulations are identical.

The 1400 km orbit is significantly slower than the 1 Sol trajectory which requires the vehicle to spend more time in the atmosphere over a longer downrange distance to achieve the necessary drag losses. In addition, due to its lower apoapsis altitude, the 1400 km trajectory exits the atmosphere at a more shallow flight path angle (see Figure 4.16). These trajectory differences require the 1400 km trajectory to dwell in the highly variable upper atmosphere for a longer timeframe and over longer distances. Recall that the magnitudes of the density perturbations grow exponentially with altitude (see discussion in Section 4.4.1). Because the 1400 km trajectory travels over a longer path length in the highly variable upper atmosphere, the density perturbations have a greater impact on the final state and therefore the improvement offered by the PENS algorithm appears to be reduced relative to the 1 Sol orbit. This is because the random variability in the upper atmosphere cannot be predicted by the ensembles and therefore by PENS. During the slower, longer ascent of the 1400 km trajectory, the vehicle encounters more density perturbations which generate more cumulative apoapsis errors relative to the 1 Sol orbit. The net effect is to slightly erode the ΔV advantage generated by the PENS algorithm preventing significant vertical shifts in the ΔV-λ curves. The 1 Sol orbit, due to its steep exiting flight path angle, ascends through this region quickly such that its path length in the upper atmosphere is significantly smaller. Therefore, the variability of the upper atmosphere has a much smaller impact on the 1 Sol orbit.

The above discussion explains the disparity in the performance of the PENS estimators between the 1400 km and 1 Sol orbits. What remains to be explored is the nature of the performance differences between PENS and the conventional estimators

within the 1400 km orbit. Because the density models generated by the PENS algorithm have a much greater degree of physical fidelity and flexibility than conventional estimators, the long-range predictions are generally more accurate. This increase in general density prediction accuracy relative to other estimators reduces the likelihood of hard failures which shifts the PENS ΔV-λ curves to the left. Conventional density estimators do not enjoy this advantage and, in generating less accurate predictions, permit more guidance faults and therefore larger hard failure rates. This is directly observable in Figure 5.10 by comparing the location of the vertical asymptotes of the PENS estimators relative to the conventional/historical counterparts.

The increase in general prediction accuracy achieved by PENS over long ranges is demonstrated in Figure 5.11 for trajectory #11 from the 1400 km dataset. This trajectory utilized the eight member ensemble and therefore only the PENS-8 and KPENS-8 results are available. In the figure, the density prediction errors from the most competitive density estimators are shown for the final two minutes of atmospheric flight. This window was selected because it demonstrates the performance of the estimators over an extended period in the highly variable upper atmosphere (from roughly 90 km up to 128 km in altitude). The $\rho_{K\text{-}EXP}$ model, based on a simple exponential model, lacks the high fidelity physics to generate accurate predictions over extended ranges. The ρ_{TABLE} and ρ_{KPENS} perform similarly because both are based on MCD climate models corrected with a multiplier K.

The table model is based on the average dust/average solar flux MCD model and KPENS is based on the same eight member ensemble as PENS. As discussed in Section 3.3.7, the purpose of the multiplier K is to correct the instantaneous state, that is, to force the prediction model to match the ADS-measured density at the current vehicle location. The multiplier, therefore, biases the entire prediction model such that greater emphasis is placed on close-in, short-range prediction accuracy. Note that the prediction errors of all the K-based estimators (ρ_{TABLE}, ρ_{KPENS}, and $\rho_{K\text{-}EXP}$) in Figure 5.11 have zero error at the

current vehicle location at 600 seconds. This short-range emphasis, however, is at the expense of long-range accuracy. Note that the K multipliers are updated at the guidance inner loop calling frequency of 2 Hz each time the density estimators are recomputed.

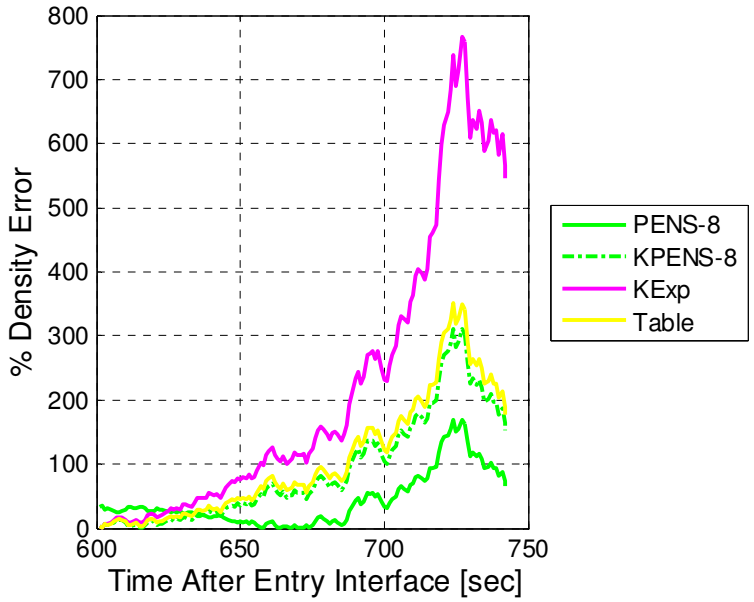


Figure 5.11: Error from a Single Density Prediction in Trajectory #11 for Final Two Minutes of Flight

The table model has only a single degree of freedom (the table multiplier K) and therefore some long-range inaccuracies are expected. For the KPENS model, however, the presence of the multiplier restricts the adaptive flexibility of the algorithm. Recall that the KPENS prediction model may be expressed as:

$$\rho_{KPENS} = K\sum_{i=1}^{n}\rho_i\omega_i = \sum_{i=1}^{n}K\rho_i\omega_i = K\rho_1\omega_1 + K\rho_2\omega_2 + \ldots + K\rho_n\omega_n \qquad (5.3)$$

where ρ_i represents the eight ensemble models described by the NNs, ω_i are the model weights determined by the PENS algorithm, and n is the number of ensemble members. The PENS weights are determined by accounting for both long-range and short-range prediction accuracy (see Section 3.3.5 on multiple prediction horizons). Including K biases the prediction model to favor only short-range accuracy and the model is less able to effectively account for long-range performance. Due to the overriding effect of the

multiplier K, the biased weights $K\omega_i$ applied to the ensemble members ρ_i now no longer have unit sum. This permits the prediction model to generate density predictions which may be physically infeasible over long ranges. In this way the KPENS approach is very similar in concept to the table model (as may be seen in Figure 5.11) because both apply a corrective multiplier K to a high-fidelity density model based on MCD data.

Because the 1400 km orbit requires longer range density predictions due to the shape of the atmospheric trajectory, the larger hard failure rates of the K-based methods are all very similar to one another (see Figure 5.10). This effect is not nearly as pronounced for the 1 Sol orbit which does not require very long range density predictions. Because the locally-biased 1 Sol density predictions are not propagated over long ranges, the density predictions errors do not grow as large. Compare, for example, the density prediction errors of PENS and KPENS at 650 seconds in Figure 5.11 as opposed to 700 seconds.

Because very long range predictions are not required for the 1 Sol orbit, the prediction errors for the K-based estimators do not typically grow to catastrophic levels. This produces low failure rates which are closer to the PENS failure rates. The 1 Sol hard failure rates for the competitive K-based methods are near $\lambda \approx 5\text{-}6\%$ (found by visually determining the vertical asymptote of the K-based methods in Figure 5.8). The 1 Sol hard failure rates of the PENS estimators are relatively similar ($\lambda \approx 5\%$). In contrast, the 1400 km orbit K-based methods have hard failure rates near $\lambda \approx 7\%$ whereas the PENS hard failure rates are near $\lambda \approx 4\%$ (see Figure 5.10).

While, the 1 Sol orbit does not require very long-range predictions, the short-range biasing caused by the multiplier K still generates appreciable errors at the end of the predicted trajectory. These density errors, because they are appreciable yet not catastrophic, cause more apoapsis errors which require greater ΔV to correct. This greater required ΔV vertically shifts the 1 Sol ΔV-λ curves for KPENS up and away from the analogous PENS curves, particularly in the critical knee region illustrated in Figure 5.8.

Because the PENS estimators have the best general performance characteristics, it is useful to compare their impact on both orbits. In contemporary literature, a cost/risk trade in aerocapture orbit selection has been recently discussed. Larger orbits like the 1 Sol orbit require less post-aerocapture ΔV but are inherently more risky. In comparing 1 Sol and 500 km target orbits, Cianciolo et al.[87] observe that "…it is evident a large ΔV savings is achieved by changing to a target orbit with a higher apoapsis altitude but the propellant savings is at the cost of increased risk of skip-out…" If using the most competitive historical density estimators, the conclusions of Cianciolo et al. hold true and the higher orbit is generally more affordable than the lower orbit (see Figure 5.12). The only exception lies in a narrow region between 7.5%< λ <11% wherein the 1400 km orbit becomes marginally more affordable.

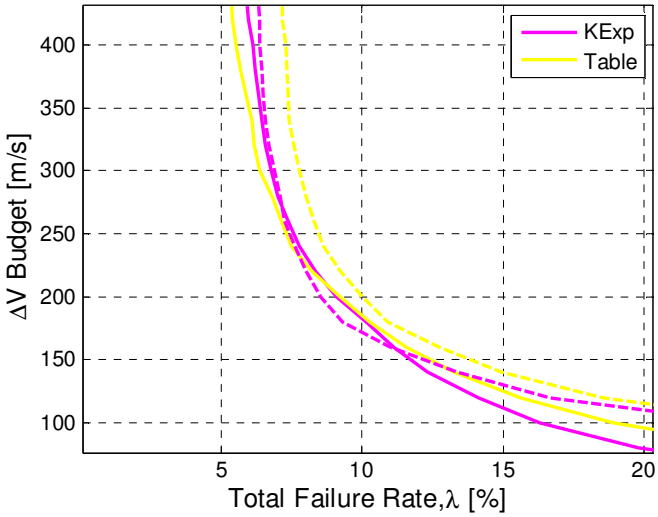


Figure 5.12: Comparison of the 1 Sol (solid) and 1400 km (dashed) ΔV-λ Trends Using Historical Density Estimators

Using the PENS algorithm, however, further information may be obtained on this cost/risk trade inherent in aerocapture orbit selection. Figure 5.13 depicts the ΔV-λ curves from all of the PENS estimators for both the 1 Sol orbit (solid lines) and the 1400 km orbit (dashed lines).

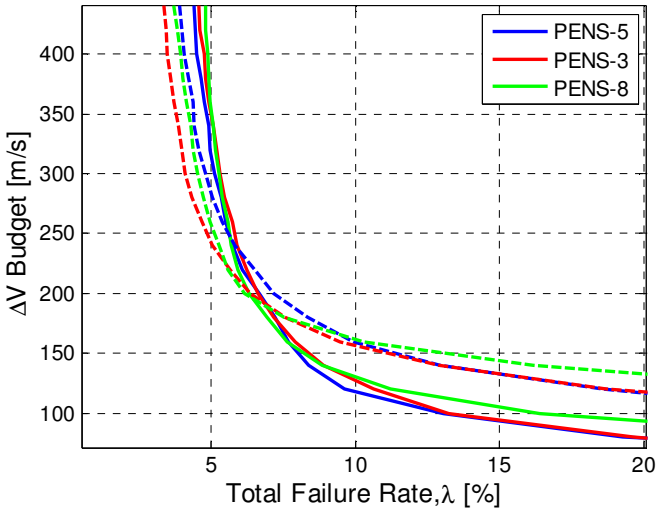


Figure 5.13: Comparison of the PENS 1 Sol (solid) and 1400 km (dashed) ΔV-λ Trends

It is clear from inspection of the figure that the ΔV-λ curves intersect in the region near λ = 7%. It is also immediately evident that, in general, the ΔV associated with the 1 Sol orbit is smaller compared to the 1400 km orbit. This corresponds well to expectations from the literature. For PENS-augmented guidance, however, the 1400 km orbit actually becomes more affordable between failure rates λ < 7% but prior to the vertical asymptote near 4%. Because the ΔV-λ trends are diverging toward the vertical at this point, the ΔV savings associated with the 1400 km orbit at low failure rates may be quite appreciable. For example, in the 1400 km orbit the PENS-8 estimator requires approximately 250 m/s ΔV at a failure rate λ = 5%. At the same level of risk, the 1-Sol orbit ΔV requirement for PENS-8 is 40% greater at 350 m/s. In contrast to the PENS results, the best historical estimators only reach this level of risk with a ΔV budget in excess of 530 m/s. Therefore, the lower 1400 km orbit actually becomes significantly more affordable than the 1 Sol orbit at low failure rates for the PENS estimator. This is quite notable because the PENS estimator, in producing solutions with the best cost and risk, precludes the cost/risk tradeoff for aerocapture orbit selection cited in the literature. If the PENS estimators are used, the low-altitude orbit becomes the most affordable and least risky option for design failure rates λ < 7% but prior to the hard failure rate asymptote near λ = 4%.

Because all failure rates given herein correspond to the observed 6σ density perturbations in the Martian atmosphere, it may be reasonable to select higher failure rates depending on the programmatic risk posture. If a failure rate beyond the 7% boundary is selected, the 1 Sol orbit becomes more affordable as expected.

Therefore, for $4\% < \lambda < 7\%$, the 1400 km orbit has been shown to be more affordable than the 1 Sol orbit when employing the PENS estimator in the guidance scheme. This is in contrast to the historical estimator results and the results from contemporary literature which suggest that the more risky 1 Sol orbit is more affordable. However, even if a 1 Sol orbit is of interest for architectural reasons, then the PENS algorithm can still offer significant risk and cost reduction options.

5.3 Effect of Ensemble Size & Scope

As may be seen from the figures presented in the last section, the size of the ensemble appears to have relatively little impact on the performance of the PENS estimator. As will be shown, however, the size of the ensemble does have a quantifiable effect on PENS estimator performance. This effect may be discerned by re-computing the ΔV-λ diagrams several times using subsets of the available data. Recall that approximately 6000 trajectory cases are available for each density estimator in a given orbit. In the previous section, all 6000 cases were used to compute the ΔV-λ trends. Now, one sixth of the data (approximately 1000 cases) are used to create six ΔV-λ curves for each density estimator. An example result of these computations is shown for the 1400 km orbit PENS-3 estimator in Figure 5.14. While the figure contains six distinct ΔV-λ curves, it may be seen that they all are similar in form to one another.

The change in the location of these ΔV-λ trends is due only to the random atmospheric and entry state perturbations described in Table 4.2 and Section 4.4. Therefore the magnitude of these ΔV-λ variations provides an indicator of the remaining uncertainty which the guidance and density estimators are unable to attenuate. This

remaining uncertainty, by directly impacting final state of the trajectory, also impacts the location of the ΔV-λ curve.

The approach just described (i.e. subsets of 1000 cases each) was selected over alternate approaches (i.e. subsets of 2000 cases or 500 cases each, etc.) because it offers a good compromise between statistical confidence and conservative uncertainty estimation. As more and more cases are used in a given subset, the effects of the random perturbations (which generate the differences in the ΔV-λ curves) begin to average out and the ΔV-λ curves become more similar.

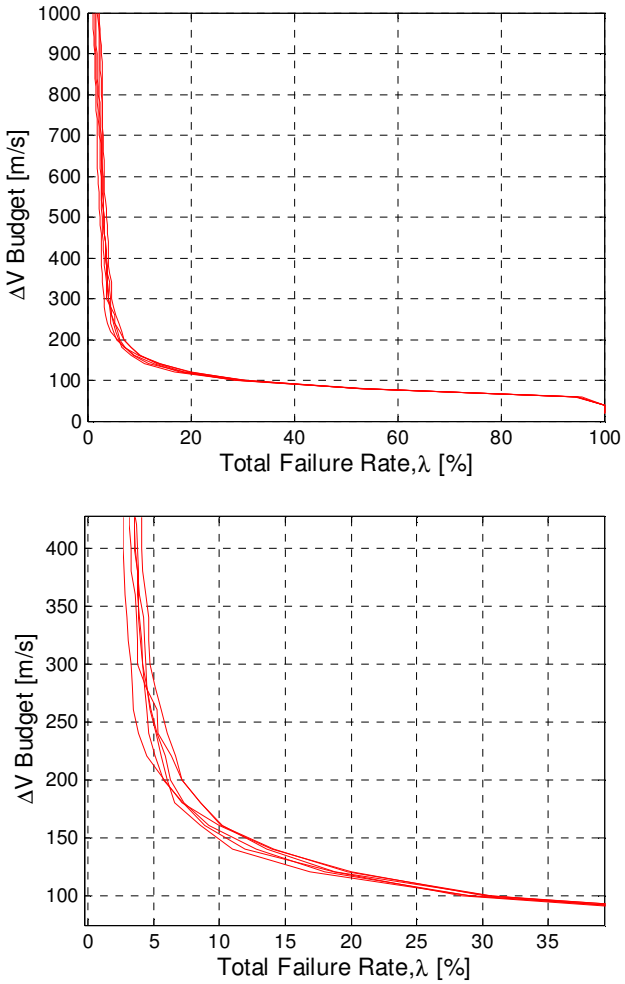


Figure 5.14: Variation in the ΔV-λ Diagram for the PENS-3 Estimator: Total Variation (Top Plot) and Variation near the Origin (Bottom Plot)

As fewer and fewer cases are used in each data subset, the resulting ΔV-λ curves become ill-defined and jagged as gaps and holes develop in the subset ΔV distributions used to generate the ΔV-λ curves. Extracting conclusive information useful for design with defensible statistical confidence is no longer possible. At this point, the effect of individual outliers begins to drive the behavior of the curves and the subsets suffer from data starvation. Therefore, at 1000 cases each, the subsets have statistically useful ΔV distributions and also produce markedly different ΔV-λ trends.

The extreme bounds of the six ΔV-λ curves created for each estimator are determined (see Figure 5.15). These extreme bounds define a variation window which is an approximation of the remaining uncertainty due to atmospheric perturbations and random initial state variations. Composite ΔV-λ diagrams are now created using these variation windows as uncertainty approximations (Figure 5.16).

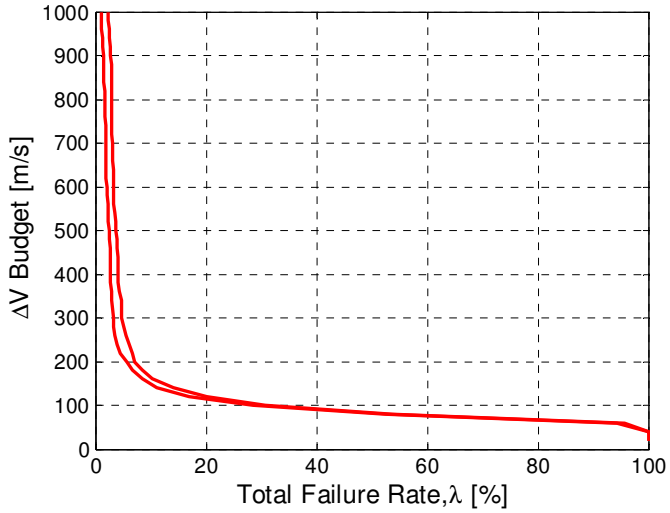


Figure 5.15: ΔV-λ Variation Window for the PENS-3 Estimator

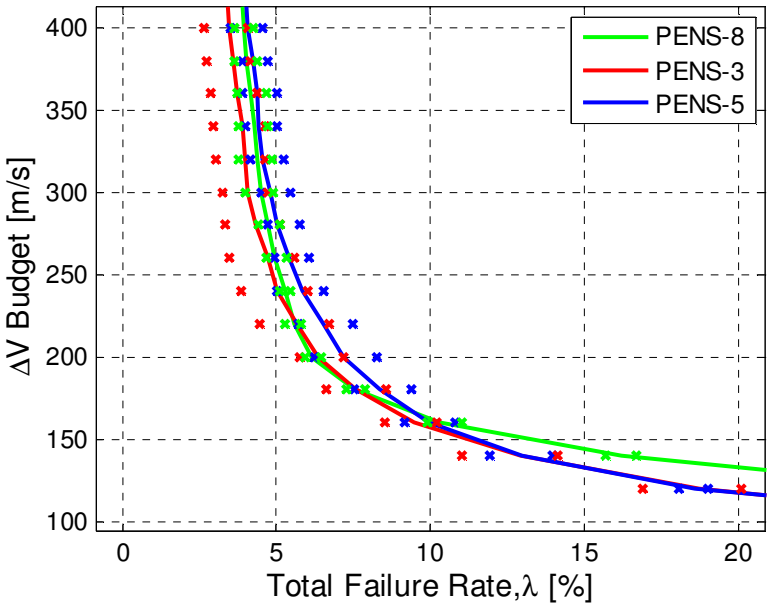


Figure 5.16: Composite ΔV-λ Diagram for 1400 km PENS Estimators Including Both Average Performance (Solid Lines) and Uncertainty Window (X Markers)

With the inclusion of the variation window in the composite ΔV-λ diagram, the effect of ensemble size becomes clear. As may be seen by inspection, the uncertainty of the eight member and five member PENS estimators (PENS-8 and PENS-5) is less on average than the uncertainty of the three member ensemble (PENS-3). The higher level of uncertainty associated with PENS-3 is unsurprising given the limited amount of atmospheric information stored aboard the vehicle. PENS-3 has only three models to compare against the ADS measured density trend.

Note that PENS-3 has better average performance in many regions of the design space and that its lower uncertainty bound has superior performance throughout most of the design space. This can only be explained by the presence of fewer on-board models because this is the only difference between the various PENS estimators. With fewer on-board models, the parameter estimation problem is somewhat simpler because each of the three models are markedly different from one another. These large differences simplify both the model pruning and the associative learning processes by making it far less likely

that an incorrect model will be trusted. With larger ensembles, however, the reverse is true.

Larger ensembles require more atmospheric exposure but, once enough learning has occurred, they are equipped to produce more accurate predictions. More accurate predictions lead to more successful apoapsis targeting and greater consistency of results (i.e. lower uncertainty). However, depending on the atmospheric perturbations, an estimator with a large ensemble may permit the guidance to trust an inappropriate ensemble member at a critical moment while the estimator is still learning. The possibility of trusting incorrect ensemble members leads to slightly higher failure rates for larger, more complex ensembles like PENS-8 (e.g. compare PENS-8 and PENS-3 hard failure rates in Figure 5.16).

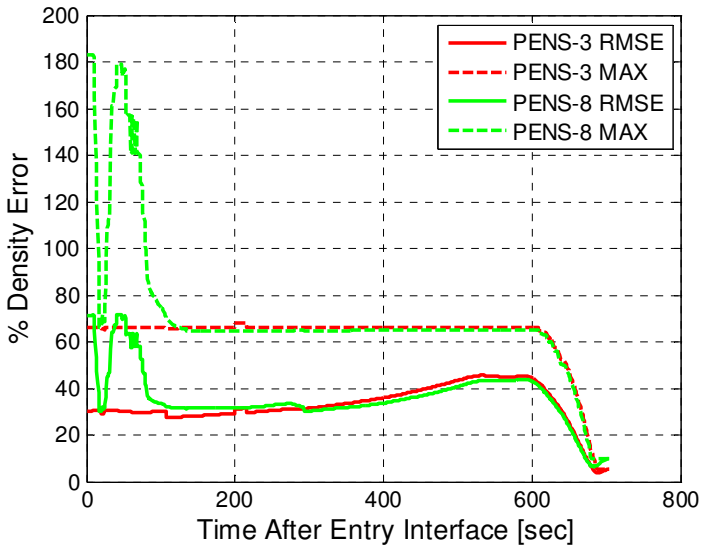


Figure 5.17: Ensemble Complexity vs. Accuracy Tradeoff for a 1400 km Trajectory

This tradeoff between ensemble complexity and long-term accuracy is illustrated in Figure 5.17. The maximum density prediction error is displayed as a dashed line while the average prediction error is solid. The larger eight member ensemble must be exposed to the atmosphere over a broad altitude range, nearly up to periapsis before it can begin to

make distinctions among the various ensemble members. Note, however, that the prediction accuracy of the eight member ensemble eventually becomes more accurate after significant atmospheric exposure has occurred.

Therefore, the simulation statistics indicate that larger ensembles inhibit uncertainty growth while smaller ensembles have the potential to achieve higher performance but at a greater level of uncertainty. The PENS-5 estimator would seem to offer an ideal compromise by using an intermediate number of members. However, the members of PENS-5 are more closely spaced and include no dust storm models. Note that because there are upper and lower limits on feasible atmospheric densities, larger ensembles will have more closely spaced members. It is highly likely, then, that larger ensembles will require more atmospheric exposure in order to successfully learn the differences among the various ensemble members. Future work should therefore examine smaller ensembles with distinct, widely spaced members that include all possible weather conditions, even dust storms. Such an ensemble may inhibit uncertainty growth while still requiring smaller periods of atmospheric exposure to successfully accomplish associative learning.

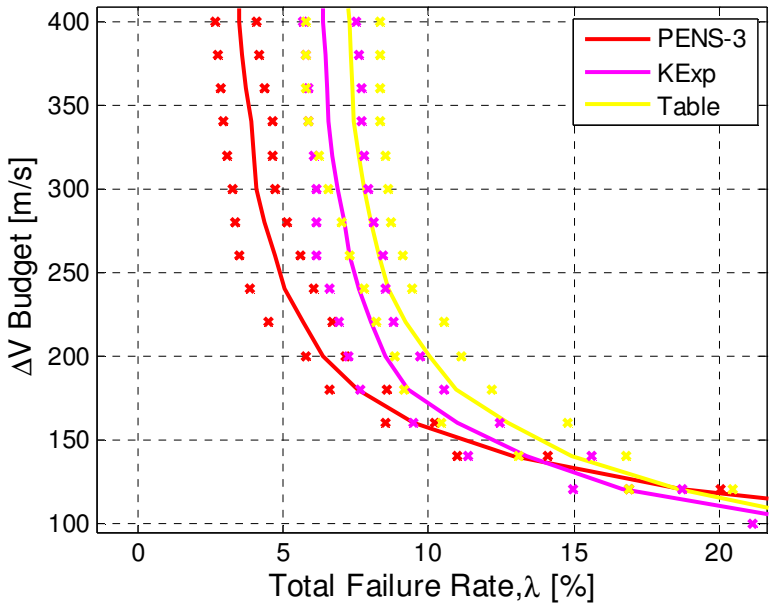


Figure 5.18: Average Performance (Solid) & Uncertainty (X) Comparison of PENS-3 to Competitive Historical Estimators for the 1400 km Trajectory

Figure 5.18 illustrates the performance improvement realizable with the PENS method relative to the best historical alternatives (the ρ_{TABLE} and $\rho_{K\text{-}EXP}$ models). Even though the PENS-3 estimator has the greatest degree of uncertainty of the three PENS estimators, its uncertainty is comparable to the most competitive of the historical methods. The average performance of PENS-3 (i.e. the average location of the ΔV-λ curve), however, is far better as it offers significantly lower risk design options than the historical methods.

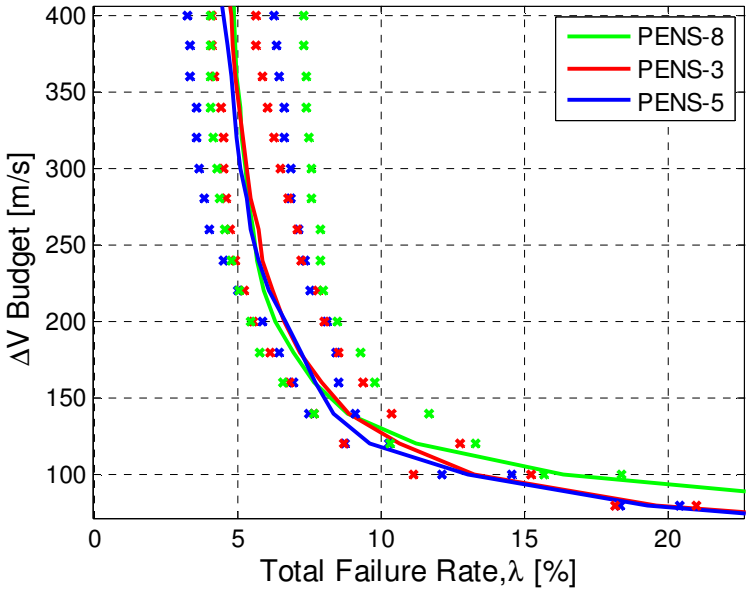


Figure 5.19: Composite ΔV-λ Diagram for 1 Sol PENS Estimators Including Both Average Performance (Solid) and Uncertainty Window (X)

Interesting new trends begin to emerge when performing a similar analysis on the fast 1 Sol trajectory. Figure 5.19 displays the uncertainty windows around the three PENS estimators which are now markedly larger for this trajectory. In order to explore the changes in the uncertainty windows of the density estimators between the 1400 km and 1 Sol trajectories, the average window width was determined (Table 5.1). This average, taken over all ΔV budgets up to 1000 m/s, is an indicator of the expected variation in the failure rate λ for a given estimator and orbit type. For example, the

PENS-3 estimator for the 1 Sol orbit has an average failure rate λ uncertainty of 2.05%. This indicates that a given ΔV budget, the horizontal location of the PENS-3 ΔV-λ curve varies by 2% on average. From Figure 5.19, for a ΔV budget of 300 m/s, uncertainty window extends from λ = 4.5% to λ = 6.4% for a total uncertainty of 6.4% -4.5% = 1.9% which roughly agrees with the average uncertainty of 2%.

The third column of the table presents the average change in uncertainty associated with a given estimator when the trajectory is changed from the 1400 km orbit to the 1 Sol orbit. Note that the uncertainty of all of the PENS estimators increases when changing to the 1 Sol orbit. This observation is revealing when considering the nature of the 1 Sol orbit. Recall from Section 4.3.3 that the atmospheric trajectory which produces the 1 Sol orbit is significantly faster than the 1400 km trajectory. The atmospheric flight time of the 1 Sol trajectory is approximately half that of the 1400 km trajectory with a significantly earlier peak dynamic pressure. This ultimately presents the 1 Sol trajectory with a very narrow time window in which to make use of the lessons learned from atmospheric observation.

Table 5.1: Average Estimator Failure Rate Uncertainty (Δλ)

	1400 km	1 Sol	Difference
PENS-3	1.4397	2.0498	0.6101
PENS-5	0.9351	2.4936	1.5585
PENS-8	0.9055	3.2213	2.3158
KPENS-3	2.0947	1.3443	-0.7504*
KPENS-5	3.2344	2.2045	-1.0299*
KPENS-8	1.9455	2.0368	0.0913*
Exponential	2.3951	3.2035	0.8084
K-Exponential	1.8295	1.9044	0.0749*
Exponential (short)	2.9414	4.1965	1.2551
K-Exponential (short)	3.4366	2.4663	-0.9703*
Table	2.3199	1.3502	-0.9697*
Minimum Uncertainty	*PENS-8*	*KPENS-3*	--

**indicates a model with a short-range corrective multiplier K*

On average, the 1 Sol trajectory has enough time to make gross trajectory adjustments to attain an orbit close to the desired exit state. However, the dynamic

pressure rapidly falls away after periapsis as the vehicle leaves the atmosphere much more quickly than the 1400 km orbit (recall the nominal trajectories in Figure 4.16). Therefore, very little control authority is available for the short outbound leg. The 1400 km orbit, on the other hand, has a much longer outbound leg while retaining a modicum of control authority. Even though the dynamic pressure in the outbound leg is, on average, much smaller than the peak pressure, it persists over a longer timeframe for the 1400 km orbit. This permits the vehicle to use the small resulting lift vector over time to perform fine trajectory adjustments which ultimately improve the final orbital state. Because the 1 Sol trajectory does not have this opportunity, the increase in uncertainty relative to the 1400 km orbit is expected for the PENS estimators.

As noted earlier, a tradeoff exists between ensemble complexity and prediction accuracy. Smaller ensembles are not as complex, and therefore, the PENS algorithm can successfully assign appropriate weights to the prediction model soon after entry. Larger ensembles, however, require more atmospheric exposure over broad altitude ranges in order to eliminate closely spaced members. Observe in Table 5.1 that the 1 Sol uncertainty increases with ensemble size for the PENS estimators. For the 1 Sol trajectory, as soon as the larger ensembles are exposed to the altitude range of interest, the dynamic pressure begins to rapidly decrease and the vehicle is already on its way out of the atmosphere. The smaller PENS-3 ensemble requires less atmospheric exposure and is therefore able to make more consistent predictions than the larger ensembles in the 1 Sol trajectory.

For some of the historical estimators with limited learning capability or simplistic model physics, the shorter prediction ranges associated with the 1 Sol trajectory reduces the difficulty of the density prediction problem. Some historical models are therefore able to generate more consistent predictions and their associated uncertainty decreases when switching to the 1 Sol orbit. This is especially true for models with K multipliers which bias the prediction model to favor short-range performance. Observe that the uncertainty

of all of the models utilizing a short-range corrective multiplier K (marked with an * in Table 5.1) either decreases or remains approximately the same when changing to the 1 Sol orbit.

Note that the KPENS estimators enjoy both accurate model physics and a short-range corrective multiplier *K*. It seems reasonable then to expect these models to produce superior performance in the 1 Sol orbit where the average prediction ranges are smaller. Indeed, the KPENS-3 estimator achieves the lowest uncertainty of all the estimators for the 1 Sol orbit. However, while this estimator is very consistent (i.e. has a small uncertainty), the average performance (i.e. location of the ΔV-λ curve) is similar to historical estimators. This is again due to the effect of the multiplier *K* on the PENS prediction model (see the discussion supporting Equation 5.3 in Section 5.2). In biasing the prediction model to emphasize short-range performance, the long-range accuracy resulting from the PENS associative learning process is curtailed. The model produces very consistent results because the associative learning process, the main source of adaptive freedom for PENS, is restrained. As mentioned in Section 5.2, the inhibition of the KPENS learning and adaptation mechanisms forces KPENS to behave much like the table model. Note in Table 5.1 that KPENS-3 has very similar average uncertainties to the table estimator for both the 1 Sol and 1400 km orbits.

Figure 5.20 illustrates the uncertainty and performance differences between the most competitive estimators for the 1 Sol orbit. Note that, for clarity, the KPENS-3 is plotted in blue rather than the usual red. Observe that, as expected, the KPENS-3 and the table models are nearly identical in both performance and uncertainty windows. While, the historical models become more competitive at lower failure rates, the PENS-3 estimator exhibits the best performance throughout the design space.

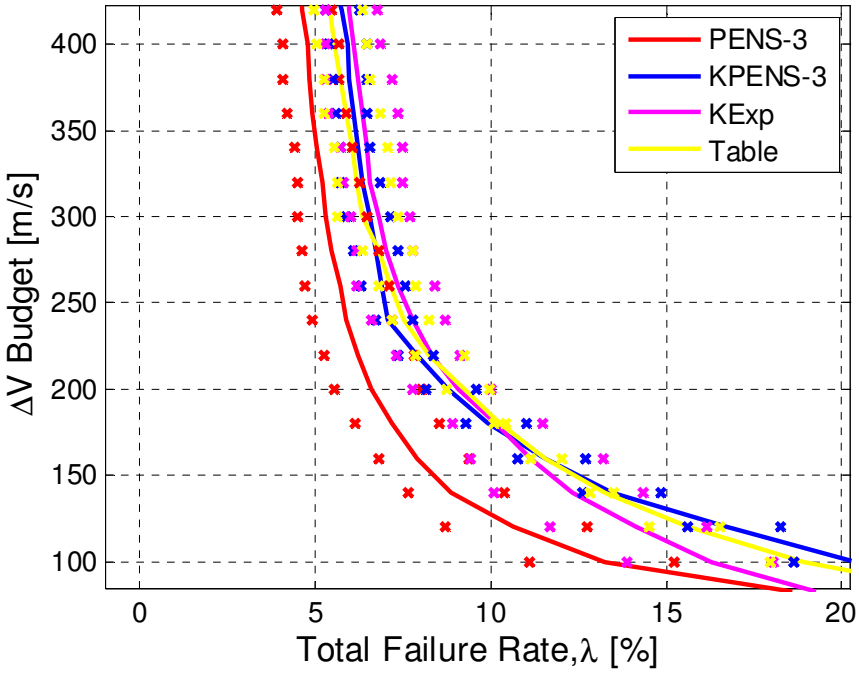


Figure 5.20: Average Performance (Solid) & Uncertainty (X) Comparison of Competitive Historical & PENS Estimators for the 1 Sol Trajectory

5.4 Impact of On-Board Remote Sensing on Aerocapture Guidance

The final series of guided trajectory simulations examine the impact of atmospheric remote sensing in an attempt to provide context for the performance improvements attained with the PENS algorithm. The remote sensing architecture adopted herein is based on the use of lidar (light detection and ranging) technology. Lidar sensor equipment has been tested and developed extensively for decades resulting in applications to both ground-based[124,125] and space-based[126] sensing platforms. Lidar systems are a form of laser-based remote sensing which have the potential to permit direct measurement of atmospheric information in real time. Such systems utilize a laser to excite the constituent gases in an atmospheric volume and the excited gases then re-radiate the absorbed laser energy. The re-radiated photons which are backscattered directly toward the emitter are detected with a telescope and used to determine information such as constituent gas concentration or aerosol profiles[126].

Recent research[127] has suggested that a 2-mircon coherent differential absorption lidar (DIAL) system has the capability to measure the atmospheric density on Mars. This development has spawned interest in the application of such a system aboard an entry vehicle for real time atmospheric measurement. This would allow the spacecraft to directly measure the upcoming atmospheric density trends along the predicted flight path. Thus the goals of this discussion are twofold;

1. To provide an optimistic best-case scenario as context for the performance improvements achieved with the PENS estimator and;
2. To examine the impact of on-board laser-based remote sensing on aerocapture guidance, focusing on the formation of fundamental system requirements.

Because the goal of this analysis is to provide a best-case context, the assumptions concerning the lidar system and its integration into both the vehicle and guidance scheme are intended to produce optimistic results. Future work should focus on testing the feasibility of the assumptions discussed in the following paragraphs. This will, in most cases, require the assumption or specification of more detailed vehicle/lidar system parameters. This is purposely avoided here, however, to ensure that the fundamental requirements generated from this study are generally applicable to a wide range of atmospheric entry vehicles and scenarios.

The vehicle is assumed to carry at least one laser-based remote sensor such as a lidar system which is used to obtain atmospheric density data. During atmospheric flight, the lidar is used by the NPC guidance scheme in an interactive fashion. The NPC guidance numerically integrates the equations of motion to construct a predicted trajectory. During the integration process, the total predicted acceleration vector is computed using information such as the local atmospheric density provided by the guidance sub-models. In the case of remote sensing, it is assumed that the lidar is capable

of rapidly scanning the spatial region in front of the entry vehicle to provide information to the atmospheric sub-model.

For example, before the vehicle reaches periapsis, it must construct a trajectory prediction from its current location, through periapsis, and up to atmospheric exit. As the on-board trajectory integration process reaches the predicted periapsis location, the iteration scheme requires an estimate of atmospheric density at the spatial location (altitude, latitude, and longitude) of the predicted periapsis. The lidar system must scan this location, determine an atmospheric density, and pass this information back to the NPC guidance so the integration process may continue. It is further assumed that the lidar is capable of scanning the correct altitude, longitude, latitude requested by the NPC guidance with negligible position error. This scheme, while challenging to implement due to the fast required lidar scanning cycles, provides an optimal, best-case configuration which is of most use to the NPC guidance.

Because one of the main goals of this discussion is to determine requirements for future lidar systems, two of the most basic variables, the range and measurement error of the lidar, are chosen for study. Unlike most of the assumptions discussed in this section which are intended to be optimistic and in favor of the lidar, these two trade variables are deliberately varied over broad ranges in order to stress the combined lidar/guidance system. This approach is adopted to permit useful range and measurement error requirements to be obtained from the guided trajectory data.

The operational concept of the atmospheric remote sensing architecture is visually illustrated in Figure 5.21. In this figure, the lidar rapidly scans ahead of the vehicle from its current location in order to provide the guidance with the requisite information to construct a predicted trajectory (red and blue lines). The lidar is used to measure atmospheric density up to a user-defined maximum range (green point on the trajectory in Figure 5.21). Beyond the maximum lidar range (red curve), the guidance obtains density estimates from the table look-up model ρ_{TABLE}. This model was chosen because it is one

of the most competitive of the historical estimators and also requires little computational effort. This offsets some of the complexity of the remote sensing scheme. The table multiplier K is computed to ensure a smooth transition between the lidar density measurements and the table model at the maximum lidar range (the green point in the figure). The table multiplier K is therefore determined by the ratio of the lidar density to the table-predicted density at the maximum range or $K = \rho_{lidar,max\_range} / \rho_{table}$. This multiplier is only computed once per predicted trajectory at the point of transition from the lidar to the table model.

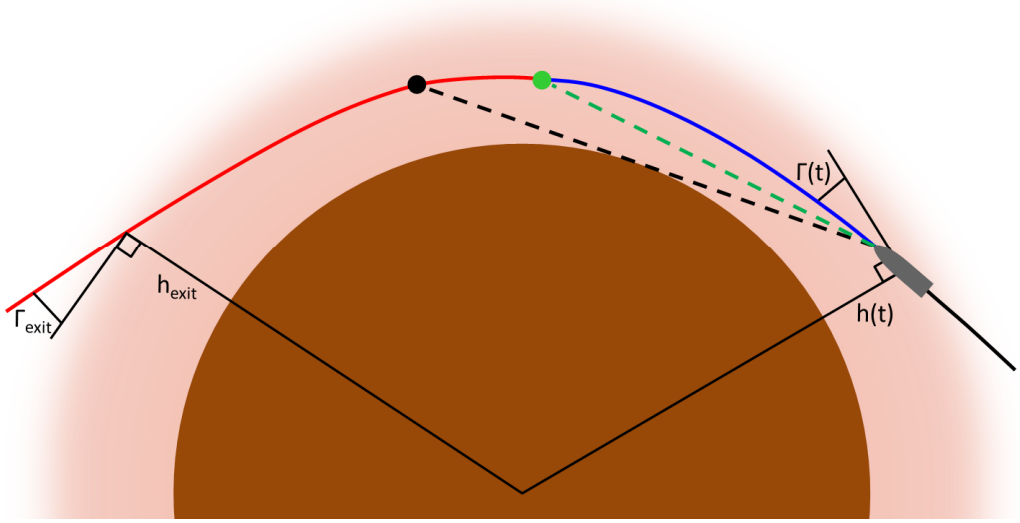


Figure 5.21: Remote Sensing Operations Concept & Line of Sight Limits

Within the NPC simulation environment developed in Section 4.2, the lidar is simulated by querying the MarsGRAM climate model at each point of the predicted trajectory. Because the same MarsGRAM model is now used by both the predicted and real world trajectories, the density "measured" by the lidar subroutine exactly matches the real world density. (Note that the user-defined lidar measurement error must be set to zero for the lidar density to exactly match the real world density.) The simulation tracks the elapsed distance traveled along the predicted trajectory which in Figure 5.21 is the arc length of the blue curve. Once this distance exceeds the user-defined maximum lidar

distance, then the table model replaces the lidar in the inner loop for the remainder of the predicted trajectory.

Both the maximum range and measurement error of the lidar are treated as discreet random variables in order to permit failure statistics to be collected. Using continuous distributions would complicate the compilation of failure rate statistics because all cases would have slightly different performance characteristics. Discreet distributions, however, permit the data to be sorted into bins where each case in the bin has precisely the same range or error. The lidar range variable is simulated by a discreet uniform distribution with a lower limit of 200 km and an upper limit of 1600 km in increments of 200 km. The lower boundary was selected as it was hypothesized that shorter range systems would not be of significant utility. Note that the atmospheric backscatter lidar systems on the CALIPSO and ICESat Earth science spacecraft operate at orbital altitudes near 650 km. Their lasers are capable of penetrating through most of the terrestrial thermosphere and middle atmosphere to reach their respective target regions[126]. Thus, it appears to be highly likely that a 200 km lidar system suitable for use on an entry vehicle is possible with current methods and materials.

The maximum range was determined by computing the maximum line-of-sight available to the entry vehicle for the original reference trajectory presented in Section 2.7. The maximum range line-of-sight (LOS) runs tangent to the planet's surface and intersects the outbound leg of the trajectory (see the black dashed line in Figure 5.21). From the perspective of the entry vehicle, this max range LOS is the point where the predicted trajectory sets over the local horizon. It is impossible for the entry vehicle to see any more of the remaining trajectory because it would require the lidar to look through the planet. Note that this is a very conservative case as the lidar laser must penetrate the densest region of the atmosphere near the surface in order to measure the furthest segment of the observable trajectory. In addition, the lidar telescope must be capable of detecting the returning atmospheric backscatter at very long ranges through a

potentially dusty lower atmosphere. The lidar is assumed to be able to resolve density at all locations up to and including the maximum range with no intermediate zones of exclusion. Lastly, it should be noted that the true maximum LOS varies throughout the entry as higher altitudes offer a better vantage point and therefore a longer maximum LOS. However, for the purposes of this fundamental analysis, the maximum LOS is assumed to be fixed at a given value.

The discreet distribution of error values selected range from -12% to +12% from the true atmospheric density in increments of 3%. This distribution is quite broad and larger than what contemporary remote sensing literature suggests for possible error levels. This broad distribution was dual-purpose:

1. As previously mentioned, it permits the point at which the lidar error significantly affects the guidance to be identified. This allows useful design requirements to be derived from the guided trajectory data.
2. Because a lidar system capable of performing the described mission role does not yet exist, unforeseen design or integration challenges may reduce the accuracy of the system.

The assumed lidar error is treated as a constant bias from the true atmospheric density over the entire field of view up to the maximum range. This constant error could have also been modeled as a continually varying random variable with its own distribution. However, a random or Gaussian white noise measurement error is highly likely to self attenuate. This conclusion is based on the observations in Sections 4.4 and 5.1 wherein the high frequency white noise of the standard MarsGRAM density perturbation scheme was shown to have relatively little impact on aerocapture guidance. The high frequency white noise signal imparts as many positive momentum changes as it does negative momentum changes and the resulting trajectory remains nearly the same. The same principle applies here. The noisy measured density signal produced by the lidar would be used to construct a predicted trajectory and the effects of white noise would self

attenuate. A biased measurement error, however, has a much different effect. The guidance scheme is unaware of the measurement error and the resulting trajectory predictions will differ from the real world behavior. Therefore, because a constant bias is a more dangerous and stressful case, the error is assumed to be constant over the entry.

To evaluate the effectiveness of on-board remote sensing, 6000 additional aerocapture trajectories were simulated for each of the two target orbits. The resulting data offers insight into the stochastic performance of the lidar-augmented guidance as the range and measurement error vary. The average performance and uncertainty window of the lidar-augmented guidance for both target orbits is presented in Figure 5.22. The analogous PENS results are also presented for comparison. Note that both PENS and the lidar-augmented systems perform similarly at high failure rates but the lidar-augmented guidance has a clear performance advantage at low failure rates.

The lidar effectively protects against hard failures and the resulting low hard failure rates shifts the lidar ΔV-λ curve far to the left. These results were achieved by averaging all of the trajectory data irrespective of the lidar range or level of measurement error. All 6000 cases were used to generate the average (solid line) trend while six bins of 1000 cases each were compared to generate the uncertainty window. These results, therefore, indicate that even with uncertainty in the maximum range and measurement error, lidar generally produces superior performance under the best-case assumptions of this study. However, because the performance of long range, high accuracy lidar systems tends to balance the behavior of shorter range, lower accuracy systems, this conclusion is somewhat conditional as will be discussed.

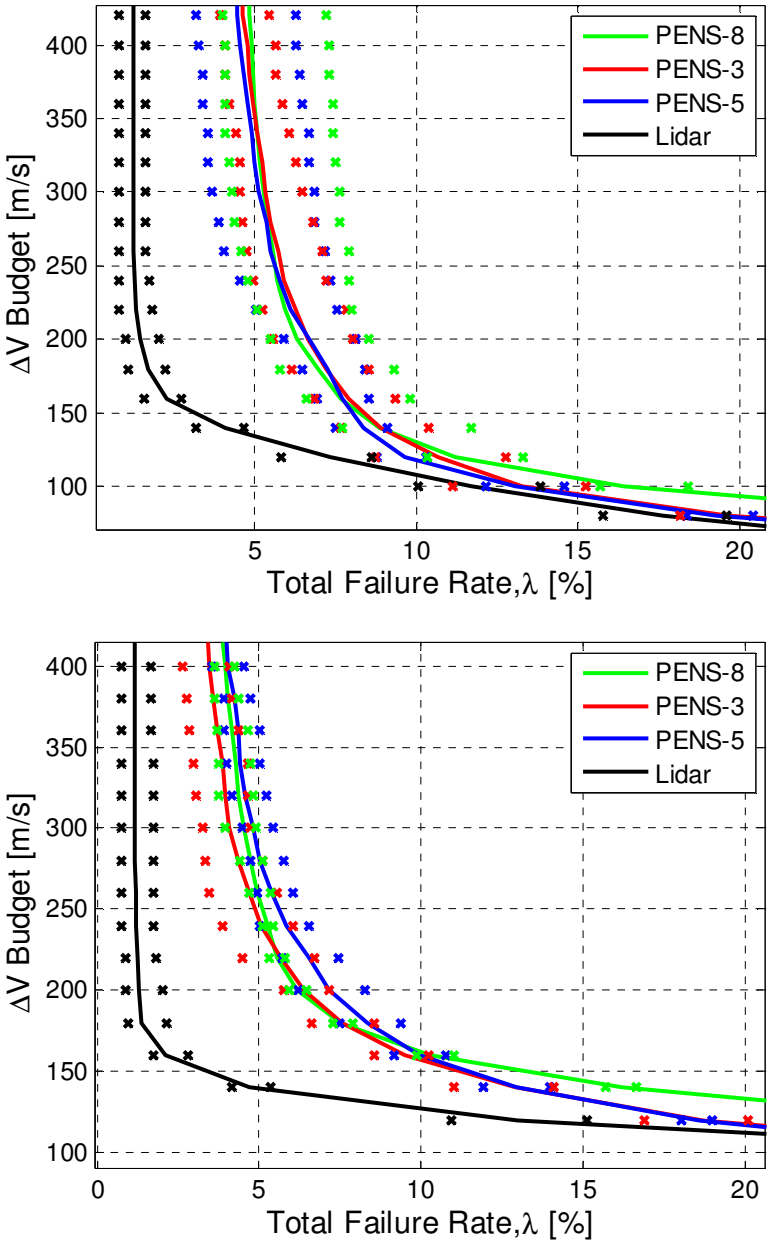


Figure 5.22: 1 Sol (top) and 1400 km (bottom) General Lidar Performance (Solid) and Uncertainty Window (X)

These results provide the desired context for the performance improvements achieved by the PENS estimator. The PENS algorithm has been shown to outperform all other existing conventional methods. However, in the theoretical limit, general lidar systems have the ability to yield larger performance gains than the PENS algorithm if the assumptions of this study hold.

The 6000 cases for each orbit are now separated into bins according to maximum range and measurement error. The ΔV-λ diagrams are again constructed with this new data grouping scheme (Figures 5.23 and 5.24). Note that the ΔV-λ curves for a given lidar range contain data at all levels of measurement error. Conversely, ΔV-λ curves for a given lidar measurement error contain data at all maximum ranges. This approach illustrates the performance uncertainty of the lidar-augmented guidance system as a function of a single variable. Note in Figure 5.23 that variations in the maximum lidar range have relatively little impact on the performance results with the exception of the 200 km case for the 1 Sol orbit. Here, at failure rates $\lambda > 10\%$, the PENS estimators begin to outperform the lidar-augmented guidance.

This is not surprising due to the sensitivity of 1 Sol orbit to the level of atmospheric knowledge available to the NPC guidance. Recall from Sections 5.2 and 5.3 that the 1 Sol trajectory exits the atmosphere in very short order and that some of the PENS estimators scarcely have the time to apply the information learned through atmospheric observation. At short maximum ranges, a similar situation occurs for the lidar-augmented guidance. The vehicle does not attain the requisite atmospheric knowledge with enough time to prevent apoapsis errors from accruing. These errors raise the required post-aerocapture ΔV and shift the ΔV-λ curve upward relative to the other lidar systems. In addition, the range-limited systems also produce slightly elevated hard failure rates as may be seen in Figure 5.23.

It is interesting to observe that the ΔV-λ curves of many longer range lidar systems exhibit marginally poorer performance than shorter range systems in the 1 Sol orbit. For example, compare the results of the 1600 km lidar (black dashed line in Figure 5.23) to the 1000km lidar. This is again due to the sensitivity of the 1 Sol trajectory to the quality and quantity of available atmospheric knowledge. Recall that each ΔV-λ curve corresponding to a given lidar range contains cases spanning the full range of lidar measurement errors.

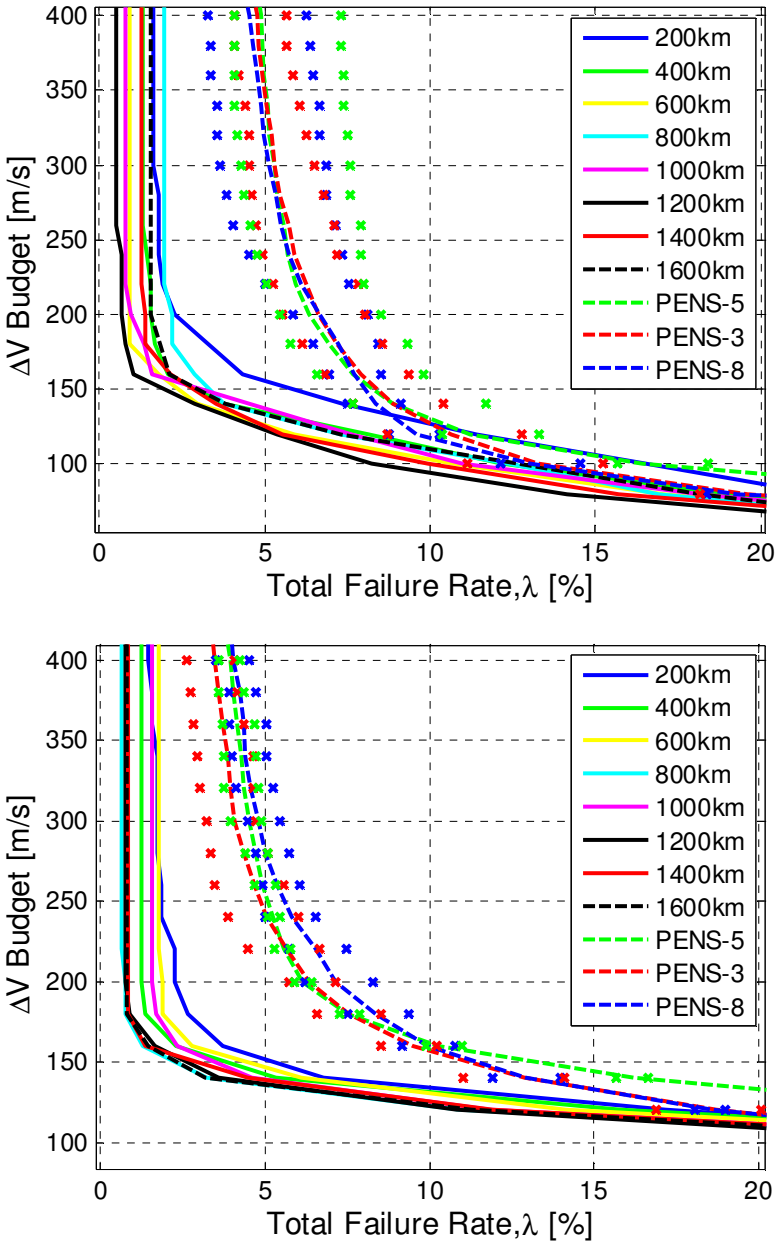


Figure 5.23: 1 Sol (top) and 1400 km (bottom) Lidar Performance as a Function of Maximum Range

If the measurement error of a lidar system is appreciable, the trajectory predictions based on the lidar measurements will contain significant errors. This problem is exacerbated as the maximum range of the lidar grows because the appreciable measurement error is propagated over longer ranges. Because the guidance makes

decisions based on these predictions, inaccurate long range lidar systems produce greater failure rates and higher ΔV requirements.

The system uncertainty due to variations in measurement error is illustrated in Figure 5.24. As is readily visible, the impact of measurement error has a more profound effect on the performance of the lidar-augmented guidance. Again, due to the sensitivity of the 1 Sol orbit to the level of available atmospheric knowledge, the PENS estimators begin to outperform several lidar-augmented systems above a failure rate $\lambda > 7.5\%$. However, also observe that the same is true for the 1400 km orbit above a failure rate $\lambda > 10\%$. This highlights the sensitivity of the lidar-augmented system to large negative measurement errors.

It is unsurprising to observe that large negative measurement errors produce much higher ΔV requirements than large positive errors. Negative measurement errors, indicating that the measured density is smaller than the true value, trick the guidance into believing that lower densities lie ahead. Consequently, the guidance commands the vehicle to remain at lower altitudes for longer periods to dissipate excess momentum before beginning the outbound leg. During the outbound leg the density is much larger than predicted which causes the vehicle to exit the atmosphere below the target velocity. The derivative of the energy equation from two-body orbital mechanics may be expressed as:

$$\varepsilon = \frac{V^2}{2} - \frac{\mu}{r} \Rightarrow \frac{d\varepsilon}{dV} = V \quad \text{or} \quad d\varepsilon = VdV \tag{5.4}$$

where V is the velocity of the vehicle, μ is the gravitational parameter, r is the orbit radius, and ε is the orbital specific energy. Consequently, a vehicle traveling at a slower velocity must expend a larger ΔV to accomplish a given change in energy. Therefore, lidar systems which chronically under predict the true density require significantly higher ΔV budgets.

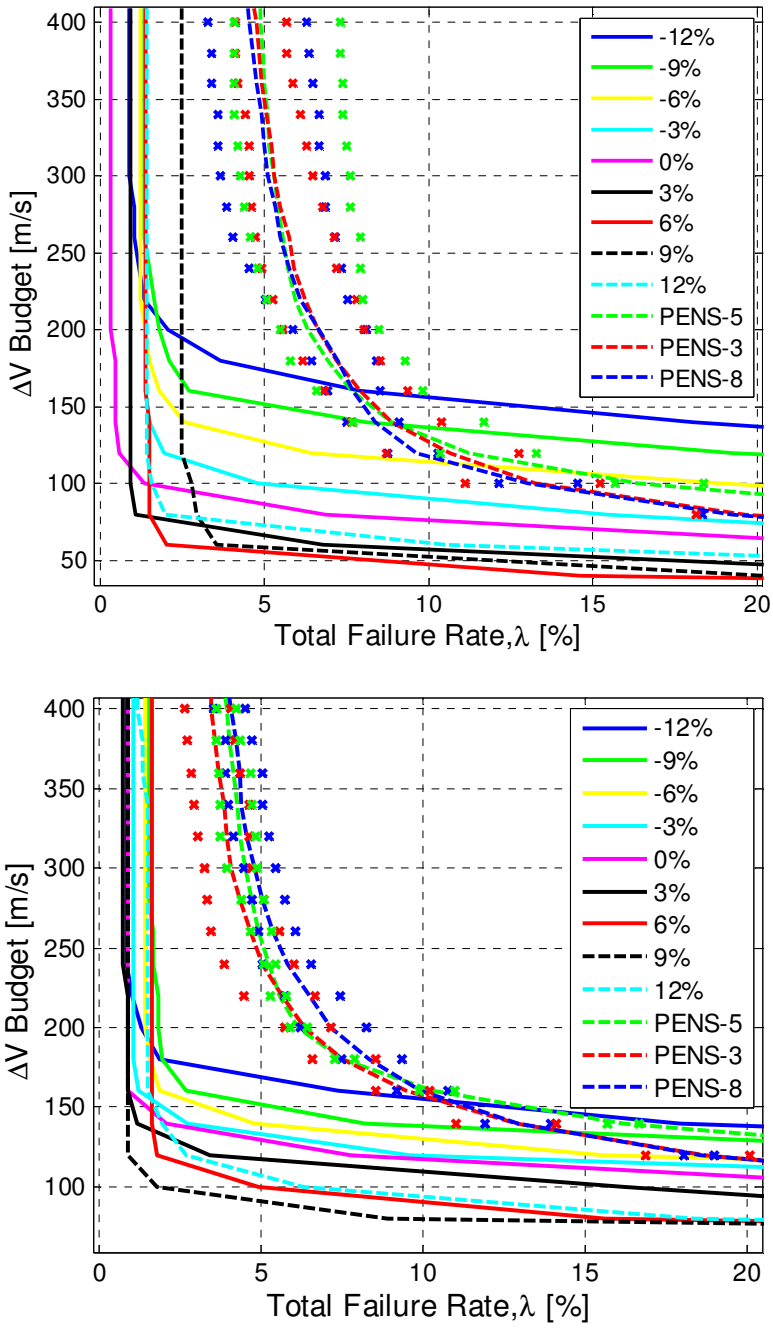


Figure 5.24: 1 Sol (top) and 1400 km (bottom) Lidar Performance as a Function of Measurement Error

Conversely, if the lidar over predicts the density (i.e. has a positive measurement error), then the reverse situation is true. The vehicle exits the atmosphere above the target velocity and the ΔV required to correct to the target orbit is much smaller owing to the high vehicle velocity. Exiting the atmosphere above the target velocity, while generating

lower ΔV requirements, is not an advisable strategy. Maintaining excess energy increases the likelihood that the vehicle will skip-out of the atmosphere if it is unable to rapidly compensate for the lidar error. This explains why many of the lidar systems with large positive errors have higher hard failure rates even though they have very low ΔV requirements.

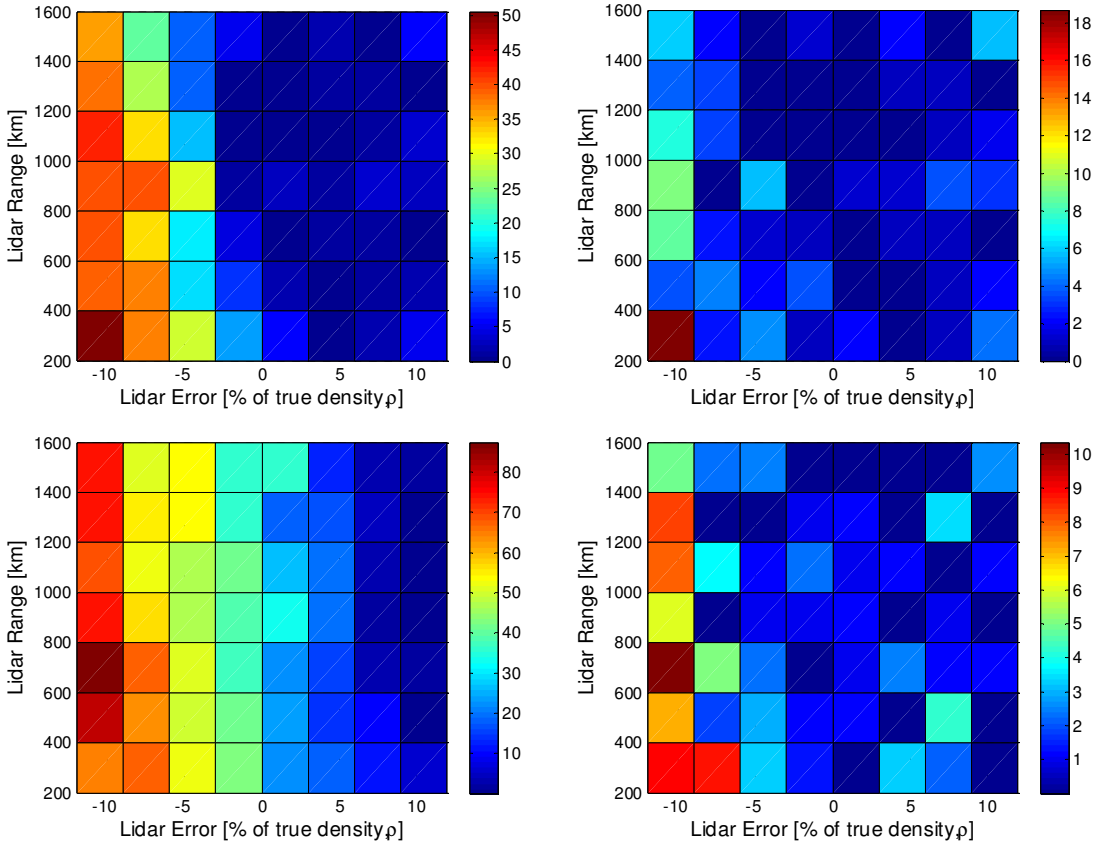


Figure 5.25: Lidar Range/Error Failure Rate Maps for the 1 Sol (Top Row) & 1400 km (Bottom Row) Orbits for 100 m/s (Left Column) & 160 m/s (Right Column)

Figure 5.25 illustrates the total failure rate of the lidar-augmented guidance as a function of both range and error. These "checkerboard" plots decompose all 6000 cases for each orbit into 72 range/error bins (8 lidar ranges x 9 measurement error levels). Consequently, each range/error bin contains 6000/72 ≈ 83 cases. Due to the relatively small sample size, the statistical confidence of these plots is lower than the preceding results. However these checkerboard plots are useful because they illustrate the correlation between range and error as well as their joint impact on the total failure rate.

The left hand checkerboard plots correspond to a ΔV budget of 100 m/s while the right hand plots present data for a ΔV budget of 160 m/s.

As expected, the lidar systems with large negative errors produce the highest failure rates (see Figure 5.25, left column). In addition, as discussed earlier, lidar systems with long ranges and large positive or negative measurement errors have elevated failure rates. For example, examine 160 m/s checkerboards for both orbits. Compare the failure rates in the upper right hand corners between 1200-1600 km and errors between 9-12% (i.e. the four squares in the upper right corner) to the failure rates with the same ranges and no measurement error. It is also readily observable that the largest failure rates occur in the lower left hand corner of the plots where the lidar systems have large negative measurement errors and short ranges.

The results from the 12,000 lidar trajectory cases presented in the last three figures may be used to derive fundamental lidar system requirements for future use. The major quantitative and qualitative requirements may be summarized as follows:

- Lidar measurement error is a major performance driver whereas the range of the lidar is not as significant.
- Lidar measurement errors should be contained in the range [-3%,+6%] in order to circumvent the greatest amount of aerocapture risk at reasonable ΔV budgets. For very small ΔV budgets (< 100-130 m/s for the orbits examined herein), avoid negative lidar error.
- Positive measurement errors (over-estimation) are significantly more cost effective than negative measurement errors as the vehicle exits the atmosphere at higher speeds. However, this is at the cost of a marginal increase in failure rate.
- For lidar measurement errors bounded between ±3%, all lidar ranges exhibit similar failure rate performance.

- If larger measurement errors (>3%) are unavoidable, attempt to maximize the effective range. Lidar ranges greater than 1000 km appear to be best but the failure rate sensitivity to range is quite small compared to measurement error.

5.5 Guided Trajectory Results Summary

The key results from the analysis of the PENS estimator and the lidar-augmented guidance may be summarized as follows:

- The PENS estimator was proven to outperform all conventional density estimators examined in this study even in the presence of large atmospheric perturbations.
- The enhanced physical fidelity and flexibility of the PENS estimator leads to significant reductions in failure rate for a given ΔV or the required ΔV for a given failure rate.
- The PENS estimator leads to performance improvements when targeting both high-altitude, high-energy orbits as well as low-altitude, low-energy orbits.
- A tradeoff exists between ensemble complexity and prediction performance. Larger ensembles produce smaller performance uncertainties (in both the failure rates and required ΔV for aerocapture) with sufficient atmosphere exposure. The large quantity of atmospheric information available in a larger ensemble enables the Hebbian learning process implemented in the ensemble echo to produce very consistent performance by learning from atmospheric observations.
- Smaller ensembles simplify the associative learning process and often offer better performance. However, this performance gain is commonly associated with higher performance uncertainties due to the limited amount of knowledge contained in a smaller ensemble.

- Future studies should focus on smaller ensembles where each of the 3-5 members are distinct and widely spaced from one another. These models should span the range of conditions which may be encountered during the entry, including even extreme weather phenomena like dust storms.
- The PENS-augmented guidance alters the cost/risk trade inherent in aerocapture orbit selection. Higher orbits such as the 1 Sol orbit have traditionally been found to produce small ΔV requirements but at an increased risk of aerocapture failure. The same trend is confirmed herein using conventional density estimators. However, when the PENS estimator is employed, the lower 1400 km orbit is shown to be more affordable (require less ΔV) than the 1 Sol at equivalent or lower level of risk for certain regions of the trade space. Thus, the PENS estimator has the potential to alter the cost and risk trade of aerocapture to the point where the low-altitude orbit may become the lowest cost option at commensurate or slightly lower levels of risk.
- Under the conditions and assumptions of this study, laser-based remote sensing from on-board the entry vehicle in real time has been shown to produce superior performance to the PENS estimator. Future studies should focus on determining whether these assumptions hold for specific mission architectures.
- Even a perfect lidar system (no measurement error with practically no sensor range limit) can still fail to complete a successful aerocapture, though the failure rates are quite small.
- The lidar sensor range is not a major driver of aerocapture performance within the ranges examined (200–1600 km).

- The lidar measurement error is a major performance driver. Underestimation of density by the lidar produces a significantly elevated failure risk relative to overestimation.

CHAPTER 6

CONCLUSIONS, CONTRIBUTIONS, & RECOMMENDATIONS

The PENS density estimator developed herein generates accurate and adaptable atmospheric density models throughout the entry. This estimator has been designed to replace the simplistic and inaccurate atmospheric density estimators utilized in modern guidance methods. Such estimators, like the common exponential density model, have little predictive power over large spatial volumes, particularly in highly variable environments such as the Martian atmosphere. When implemented into a numerical guidance scheme, the PENS estimator produces significant performance improvements, reducing both the likelihood of irrecoverable failures and the required post-aerocapture ΔV. The PENS estimator achieves these performance gains by combining accurate physical models stored aboard the vehicle with atmospheric observations.

6.1 Contributions of PENS to the State-of-the-Art

A series of innovations was required in order to create the PENS algorithm and expand the current state-of-the-art of adaptive density forecasting. These innovations provide solutions to the supporting research questions discussed in Section 1.5 and contribute to the overall solution of the primary research question. Recall the primary question under study: "Can an improved atmospheric density estimator be created that reduces the impact of atmospheric dispersions on the performance of aerocapture guidance?" In order to approach this question, the problem was decomposed into several supporting research questions with a more limited scope. This decoupled the prime problem into smaller, more readily conquerable components.

The first supporting research question sought to examine how the atmospheric state uncertainty may be reduced throughout the entry. This issue was approached with ensemble forecasting techniques commonly used in terrestrial applications. By employing

a series of climate models spanning the range of anticipated atmospheric conditions, ensemble techniques are useful for reducing the uncertainty of atmospheric forecasts in highly variable environments. However, these analyses are often complex and computationally expensive, requiring significant time and computational resources to complete. By employing the ensemble neural architecture and the ADALINE gating logic introduced in Sections 2.4-5, it was possible to adapt ensemble techniques for use aboard an entry vehicle.

The second supporting research question sought to determine a means for utilizing the ensemble to form density forecasts which are consistent with observed flight data. The ability to use a density ensemble on-board an entry vehicle (an innovation provided by the first supporting question) enabled the creation of a data assimilation problem. Data assimilation is the process of combining information from a numerical climate model with measured/observed weather data while maintaining consistency in both datasets. Common in meteorology and the climate sciences, classical data assimilation problems are also computationally complex and require significant resources and time to complete. However, a new approach is used herein to reduce this complexity and still achieve the same end goal. The classic data assimilation problem is converted into a parameter estimation problem. Parameter estimation is quite common in real time applications as the current literature reveals (see Section 2.6). This conversion is accomplished by introducing the ensemble linear combination strategy, the weighting of the ensemble members based on their average errors, and the ensemble pruning scheme (see Sections 2.5-6 and 3.2). Combining the information stored in the ensemble with the measured/observed atmosphere density permits efficient use of all available atmospheric information in order to produce an effective and adaptable forecast model.

As discovered in Section 3.3.1, the estimator architecture created above, while effective, had no means of remembering good solutions or learning from atmospheric experience. It was conjectured that a novel associative learning mechanism could be

created in order to improve forecast performance. This led directly to the third supporting research question which sought to identify methods for leveraging atmospheric prediction experience to improve future forecast models. To answer this question, an associative learning mechanism known as an ensemble echo is introduced herein (see Section 3.3). This construct provides the means to selectively store past density prediction models which have been found to be effective and consider them again for future use. By only storing a limited set of the top performing models, the system continues to learn and evolve throughout the entry. The prediction models resulting from the echo selectively absorb critical atmospheric information and reject spurious trends and noise. This results in an effective and efficient PENS forecasts with a high degree of accuracy and flexibility.

The last supporting question sought to determine how a guidance scheme could be created which leverages the information produced by the new PENS density estimator. Contemporary guidance schemes were examined and it was determined that a numerical predictor corrector (NPC) offered the best option for properly using the estimator information throughout the entry (see Sections 4.1-2). In using simplistic atmospheric density models, most modern guidance schemes are forced to compensate for atmospheric dispersions after they have already affected the trajectory. This directly leads to prevalent guidance failure modes identified in the literature (see Section 1.3). Because the primary research question is concerned with reducing the impact of atmospheric uncertainty on aerocapture guidance, preventing or reducing the occurrence of these atmosphere-induced guidance failure modes is critical. Proper implementation of the PENS estimator into the NPC guidance scheme is crucial. The need to continually correct for atmospheric dispersions overwhelms many modern guidance schemes leading to incorrect or untimely guidance decisions and ultimately complete failure of the entry. Thus, it was determined that the PENS estimator should operate in parallel to the guidance scheme rather than in subordination to it (see Figure 3.7 in Section 3.3). This

permits the PENS estimator to unburden the guidance scheme by absorbing both the atmospheric noise and atmospheric state uncertainty. The prediction models resulting from the PENS estimator are linear combinations of the noiseless ensemble members and are, therefore, noise free. The guidance scheme may then plan maneuvers and base guidance decisions on smooth (i.e. noise free) yet accurate average density forecasts. This results in better guidance decisions which reduce the likelihood that an irrecoverable state will be reached leading to guidance failure.

As mentioned, the innovations discussed above each answer one of the supporting research questions. These unique innovations, when combined, produce the novel PENS approach to adaptive density estimation and atmospheric uncertainty mitigation. The efficacy of these innovations at answering the prime research question is demonstrated through guided entry simulations in Chapter 5. The performance of the PENS estimator has been tested with over 144,000 guided trajectory simulations in order to fully characterize its impact under a variety of situations. These simulations reveal that the PENS estimator is effective in all of the examined scenarios. Reductions of 33% in both the average post-aerocapture ΔV and entry failure rates relative to the best contemporary density estimators are routinely observed. Much larger performance increases are also commonly observed. In one Monte Carlo simulation, for example, the PENS estimator decreased the maximum observed apoapsis error by 63%, the maximum required post-aerocapture ΔV by 69%, and reduced the range of periapsis altitudes by 18%. Thus, the PENS algorithm successfully addresses the primary research question by reducing the impact of atmospheric dispersions on the aerocapture guidance process.

In addition, the PENS estimator is free of arbitrary tuning and start-up gains. At guidance activation, PENS is able to produce a density model as soon as the first measured density data point is collected. As the vehicle descends towards periapsis, the primary goal of the guidance is to ensure a successful capture. Detailed atmospheric information is not yet needed giving PENS time to learn from the atmosphere and adapt.

By the time the vehicle nears periapsis, the experience gained by PENS from atmospheric exposure produces more refined density prediction models suitable for accurate path prediction and control.

Lastly, PENS has no need for *a priori* assumptions about the atmospheric noise or atmospheric dynamics. The only recommended practice is to develop ensembles which span the possible day-of-flight conditions such that the measured density will remain bounded by the ensemble. However, even this is not a strict limitation. The PENS-3 estimator which was shown to have excellent performance among the three ensembles examined included no dust storm models. The test simulations, however, included dust-storm conditions with very high levels of suspended dust present in many cases. Limited ensembles which do not span the entire range of possible atmospheric densities may still performed well if the extreme density members are sufficiently close to actual extremes encountered in flight.

6.2 Significance of the PENS Results

6.2.1 Improved Aerocapture Risk-Adjusted Benefit

The reduction in required ΔV achievable with aerocapture often has beneficial effects on multiple architecture elements by reducing propellant and engine mass requirements. However, the price of this ΔV reduction is the need to rely on a variable and uncertain planetary atmosphere instead of a traditional propulsive system. Because the majority of the world's space faring nations have the capacity to produce reliable in-space propulsive systems, traditional propulsive orbit insertion techniques are often viewed as less risky. Therefore, the architectural benefits of utilizing aerocapture on past missions have often been contradicted by the perceived risk and fiscal cost increases. For example, the Mars Odyssey mission originally planned to use aerocapture but this was changed to traditional propulsive capture in the early mission design phases[128]. Hence,

the risk-adjusted benefit of aerocapture has classically been considered too high to warrant its use to date. The PENS estimator, however, has the demonstrated capacity to improve the risk-adjusted benefit of aerocapture by simultaneously reducing the inherent operational risks and increasing the achievable ΔV savings. These benefits come at the cost of relatively benign programmatic risks as discussed in Section 6.2.3.

Enhancing the safety and utility of aerocapture is significant for two reasons. First, it permits current robotic missions in the same class as Odyssey or MRO to be conducted more efficiently with smaller propellant requirements and therefore larger payload fractions. Second, for high-mass robotic or human scale architecture components, the addition of even small quantities of ΔV may require large design changes and generate considerable mass growth. High-mass architectures often involve many more elements than contemporary missions, many of which act as payloads for other elements. Mass reduction techniques such as aerocapture are therefore increasingly important in larger architectures in order to constrain flow-down mass growth. For this reason, the Office of the Chief Technologist (OCT) at NASA recently highlighted the importance of aerocapture as a "crosscutting" capability which has the potential to yield large mass savings in future Mars exploration architectures[129]. By increasing the risk-adjusted benefit of aerocapture, the PENS algorithm has the potential to fulfill this requirement from the OCT and make future exploration architectures more affordable.

6.2.2 Alternate Approach for Atmospheric Uncertainty Mitigation

For many missions to Mars requiring an atmospheric entry, the variability in the atmosphere is a major source of design uncertainty which is often mitigated by the addition of large design safety margins[5]. These margins often generate large increases in the system mass by favoring overdesign at the system level. This mass growth, in turn, may generate other negative consequences such as the need for a larger launch vehicle or aerodynamic decelerator system. This current paradigm may lead to non-trivial mass

growth in larger robotic or human scale exploration architectures. The variability in the Martian atmosphere will likely impact an entry mission in some tangible manner and this impact may be observed at both the operational and design levels. For example, atmospheric uncertainty often generates the need for large safety margins during the design process. Even with these margins, however, the effects of atmospheric dispersions still may generate large final state errors under operational conditions.

The significance of the PENS algorithm partly lies in its approach to atmospheric uncertainty mitigation. As mentioned above, the PENS estimator has been developed to reduce the impact of atmospheric uncertainty throughout the entry and therefore operates in parallel to the guidance system.

This not only transfers the task of uncertainty mitigation from the guidance system to PENS, it also transfers uncertainty mitigation from the design level to the operational level. This is a novel alternate approach to atmospheric uncertainty mitigation for Mars entry vehicles. By incorporating PENS into the GNC framework, the impact of the atmospheric uncertainty may be managed throughout the entry. The management of atmospheric variability during operations, using purpose-built mitigation mechanisms like PENS reduces uncertainty at the design level. Therefore the size and scope of the necessary design safety margins are also reduced leading to the development of a more efficient vehicle and exploration architecture. A clear example of this may be seen in comparing the ΔV-λ diagrams for any of the PENS estimators to an analogous historical density estimator. Because the PENS-augmented guidance produces lower required ΔV at an equivalent failure rate, the necessary ΔV safety margin is be reduced. (See, for example, the discussion of Figure 5.8 in Section 5.2).

6.2.3 Comparison of PENS to Atmospheric Remote Sensing

Alternate means of operational atmospheric uncertainty mitigation also exist. The conceptual laser-based remote sensing system (i.e. lidar) examined in Chapter 5 fills the

same uncertainty mitigation role as the PENS estimator. Because the lidar remote sensing system produced greater performance gains than the PENS estimator, it would seem natural to question why on-board density estimators are necessary. By choosing to further develop laser-based remote sensing, it would seem that almost all uncertainty and risk due to atmospheric perturbations could be completely eliminated. However, the cost-benefit relationships of the various approaches must be examined. This is necessary to prevent the substitution of atmospheric uncertainty for dangerous increases in design uncertainty, vehicle cost, and programmatic risk.

Lidar systems have existed for decades and are quite reliable in many applications. For entry vehicles and space applications, however, a significant number of developmental and operational challenges exist. Indeed, with sufficient time and developmental funding, laser-based remote sensing will likely become a higher performing option than conventional on-board density estimation. However, in the near-term, many technology hurdles and integration considerations must be examined.

Singh, Heaps, and Komar note that technical problems with several NASA lidar systems in the 1990's led to the formation of an external review panel which recommended that NASA work to lower the technology risk of lidar missions[130]. The risks and technical challenges of operating a lidar system on even an orbital platform are significant. The vibratory and acceleration stresses of launch, the high required laser output power, thermal energy management, contamination of optical surfaces, material outgassing, radiation exposure, electro-optic component reliability, and mass/volume constraints are just a few of these challenges[130].

In response to the recommendations made by the external review panel, NASA initiated the Laser Risk Reduction Program (LRRP). The goals of this program focused on enhancing lidar long term reliability, development of new space-qualified lasers, development of enabling hardware (detectors, receiver systems, frequency converters, etc.), and management of knowledge advancements[130]. In short, concerted technology

incubation and process development programs were required to generate the successful lidar instruments utilized aboard the spacecraft in the first decade of this century. It should be reiterated that the majority of these instruments are located on orbital platforms. The Mars Phoenix lander, however, is equipped with an upward-looking meteorological lidar which was activated after landing[126]. This system is designed to measure aerosol and dust particulates in the Martian atmosphere up to a few kilometers above the surface. While this system did survive the entry environment as a stowed payload, it was inactive during atmospheric flight and has a very limited range relative to the applications discussed herein.

For operation during the entry, several additional challenges must be faced. As discussed in Section 5.4, the remote sensor used here was assumed to:

(1.) have a high degree of pointing and tracking accuracy,

(2.) have a constant error/bias,

(3.) be capable of very rapid scanning patterns at the pace of the on-board numerical integration process, and

(4.) have a sufficient field of view to always observe the necessary trajectory segment even during maneuvers such as bank reversals.

The veracity of these assumptions for specific lidar systems and mission architectures must be examined in future analyses. If any of these assumptions do not hold, then the results discussed in Section 5.4 are likely to be optimistic and not fully reflective of achievable performance. In addition to these assumptions, several other developmental and integration issues exist which may increase the programmatic cost and/or risk. The remote sensing system will likely need to operate behind the thermal protection system of the vehicle. This will require the inclusion of one or more transparent viewports in the high temperature windward side of the vehicle. While temperature resistant transparent materials exist, a window of any appreciable size may add significant weight to the vehicle. In addition, thermal radiation may be able to penetrate this window making

thermal management of the lidar system critical, perhaps requiring active cooling. An articulation system must be available to accurately slew the lidar emitter(s) in the direction of intended scanning with sufficient pointing accuracy. The effects of vibration and acceleration loading on the active lidar system must be mitigated. The system must be supplied with sufficient power by the spacecraft bus in order to resolve density features at long ranges.

The lidar must also be capable of operating behind a dynamically varying shock-layer. At sufficiently high entry velocities, radiative heating may become appreciable and generate added optical noise which the lidar must attenuate. In addition, the higher densities near the shock-layer may generate a high degree of backscatter from the layer. This would likely require a large increase in laser power in order to successfully penetrate this obstruction.

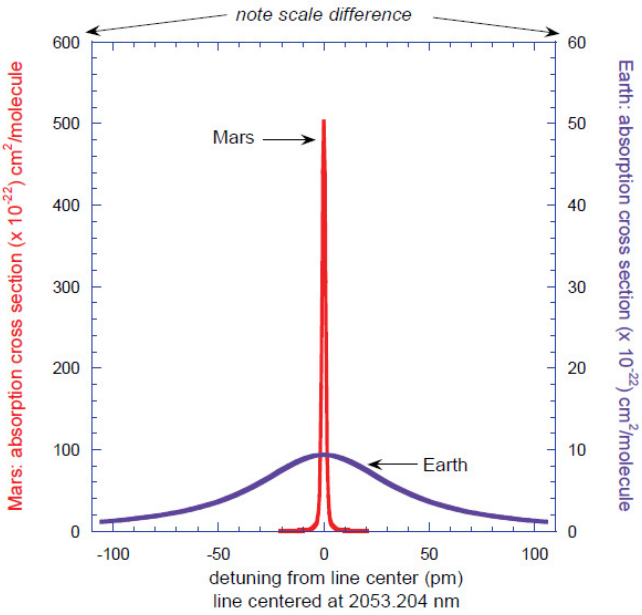


Figure 6.1: Absorption Cross Sections for a 2-Micron Lidar at Earth and Mars[131]

Lastly, the absorption cross section at Mars is very narrow due to the composition of the atmosphere (primarily CO_2) at very low pressures. The absorption cross section refers to the range of wavelengths over which a target atmospheric gas absorbs the laser

energy emitted by the lidar. If the laser is not tuned to produce photons in this band of wavelengths, then the probability of absorption and subsequent backscattering by atmospheric molecules becomes uselessly small. If this were to occur, the lidar would effectively be blind as it would receive no return data from the atmosphere. Figure 6.1 illustrates the absorption cross sections for both Earth and Mars for a candidate 2-mircon laser system capable of resolving atmospheric density on Mars[131].

Note the very narrow region of wavelengths in which the Mars lidar laser must function. This narrow absorption cross section for Mars led Singh et al. to observe that "Control of the laser spectrum becomes critical for the Mars application.[131]" Laser refraction across the shock structure, vibration, thermal effects, and many other factors may cause significant laser detuning making laser spectrum control a significant challenge.

All of the technical issues discussed above may impede the effectiveness of the lidar relative to the results obtained herein and will likely increase the cost and weight of the entry vehicle. In addition, significant technology development and evaluation programs likely would be needed to implement such a system in the short term, adding to the programmatic risk and costs. In time, many of these technical issues may be solved independently for other applications. However, the vehicle integration issues discussed above will likely persist until entry vehicle lidar technology is flight proven.

In contrast to laser-based remote sensing, on-board density estimators such as the PENS algorithm rely on proven methods and systems. The atmospheric density estimates required by PENS are obtainable with an entry air data system (ADS). Such systems are common and have been used for atmospheric data collection on entry vehicles from the X-15[31-33,105] to the Space Shuttle[29,106-108] to MSL[30,109]. Therefore, vehicle integration considerations, design of the sensors, data collection computers, and heat shield penetrations are well understood. In addition, many aerobraking missions have used only the information from accelerometers and gyros in the inertial navigation system (INS) to

estimate atmospheric density[36-39]. This approach, while simple and effective, has no means of separating atmospheric and aerodynamic uncertainties and is therefore less accurate.

The PENS algorithm would also require an additional yet limited development effort to ensure fault tolerant operation and computational efficiency. Lastly, the PENS algorithm would require computational resources aboard the entry vehicle. However, it should be noted that lidar systems also would require both computational resources and fault tolerance measures.

In summary, the potential benefits of remote sensing systems are quite appreciable but significant development and integration issues must be considered. The large reduction in atmospheric uncertainty afforded by lidar systems must be weighed against the increases in programmatic cost, risk, and vehicle design uncertainty. In contrast, the PENS algorithm offers a viable alternative in the near term. While less effective than lidar, the appreciable performance gains produced by PENS must be balanced against more benign expenditures such as greater on-board computational capability and the need for an ADS. In addition, future improvements to the PENS architecture may generate additional performance gains, further narrowing the gap between on-board estimators and remote sensing systems.

6.3 Recommendations for Future Study and Implementation

Several subjects of interest to the development of future adaptive density estimators have been identified during the PENS development and testing process. These subjects warrant future study either because they may improve the PENS estimator in its current form or offer promising new directions for future adaptive estimators. Future research efforts should:

(1.) Examine small ensembles containing 3-5 distinct, widely space members. As currently defined, the parameter estimation scheme may incorrectly trust an

inappropriate ensemble model when the members are closely spaced (i.e. have very similar density trends). In addition, larger ensembles require more atmospheric exposure in order to prune members and assign weights (see Chapter 5). The PENS-3 estimator demonstrated excellent performance with limited data. Therefore, one promising ensemble based on an expanded PENS-3 architecture would include average, low, and dust storm density models as well as an intermediate density model (between the average and dust storm models). Such an ensemble may generate excellent performance with very low performance variability/uncertainty.

(2.) Examine improvements to the parameter estimation scheme. The approach used herein is to weight the models by the inverse of their average (RMSE) errors. This approach is designed to be as simple as possible while still maintaining a high degree of effectiveness. Least squares fitting was found to produce poor results due to the effects of collinearity (see Section 3.2.2). Other techniques exist however, such as partial least squares[77] and principle component regression[78] which remove the similarity between the ensemble members before the application of a least squares algorithm. While more computationally demanding, these algorithms may produce enough performance gains to justify their use. Also hybrid approaches could be examined wherein the inverse RMSE scheme is used at high frequency and a more detailed regression is performed only occasionally throughout the entry.

(3.) Examine extensions to the machine learning elements utilized in the PENS algorithm. This is a diverse and fascinating field and useful improvements may be produced by:

a. Replacing the pruning process with a classifier neural network which would compute the probability that a given ensemble member is useful.

b. Examining the use of multiple adaptive linear neurons (MADALINE). The single ADALINE architecture used here is powerful, but additional learning and decision making prowess is undoubtedly attainable.

c. Considering the addition of fuzzy logic to the decision making algorithm. Fuzzy controllers are powerful because they can mimic human-like reasoning like neural networks and are readily understandable. For this reason fuzzy systems are often combined with neural networks (creating a so-called neuro-fuzzy system). The weights derived by the parameter estimation scheme are reminiscent of fuzzy membership functions. This may be a natural avenue to improve the performance of parameter estimation without the need for complex and expensive regression algorithms.

(4.) Further examine lidar and remote sensing techniques for real time entry vehicle applications. The significant performance gains achievable with remote sensing make it an attractive option for future exploration even in light of the possible technical hurdles. Future research should focus on the interaction of the lidar laser with the flow field and shock structure near the vehicle. Also of interest are the effects of vibration and shock-layer thermal radiation on the laser and on the resulting data return signal.

(5.) Examine extensions of PENS-augmented entry guidance for landing missions such as MSL and future human exploration architectures. This will likely require consideration of winds as well as density. As seen in Section 4.2.3, density estimation error due to winds increases as vehicle velocity decreases. In addition, many types of landing vehicles can be strongly affected by winds, especially systems relying on parachutes. The PENS method could be applied several times; one PENS system would be responsible for density estimation as described herein. In addition, two more PENS estimators could also be used to estimate

zonal (east/west) and meridional (north/south) wind components. This would require the extension of the ensemble neural system to include models for average wind speeds over the range of possible atmospheric conditions. This would also require a means for estimating the wind velocity which can be achieved by combining information from the INS and ADS[109].

For near-term applications, programs seeking to deploy the PENS algorithm aboard a guided entry vehicle should consider several developmental tasks. Aside from examining the subjects just highlighted for future study, a notional PENS development process is briefly outlined. Future missions should consider the following PENS development cycle:

(1.) Software-in-the-Loop Simulations: These simulations include detailed and mission specific models for actuators, controllers, the ADS, INS, and other systems which are necessary to support the guidance system and PENS. These simulations should be conducted to search for mission-specific issues, anomalies, and algorithmic faults resulting from the interactions among the various software components.

(2.) Bench Top Testing: Conduct tests with breadboard versions of the mission hardware. This phase would permit the refinement of fault tolerance measures, drive out system integration roadblocks, and ensure the proposed mission hardware has the requisite computational ability for the required load.

(3.) Piggyback Payload Test Flight (Optional): Similar to Phases I and II of the testing process herein (see Section 3.3), the PENS algorithm would be integrated into the mission flight software but be operated in an open loop. In this manner, the PENS algorithm has no ability to influence guidance decisions. Such a test would give confidence that the PENS algorithm is a flight-ready technology, is able to perform its intended purpose, and reduce future flight risks.

(4.) Hardware-in-the-Loop Testing: Conduct tests of the actual flight hardware to be used on the maiden test flight of the closed loop PENS system. Fault detection and mitigation, fault tolerance measures, and mission specific hardware interactions with the PENS algorithm should be of prime interest in these tests.

(5.) Operational Test Flight: The first flight of a PENS-augmented guidance scheme should have the computational and memory capacity to save (or be able to reconstruct at a later time) the computational steps which actually occurred aboard the vehicle. This information may be invaluable for future algorithm improvements and for solving future integration issues.

APPENDIX A

NEURAL NETWORK PRIMER, EQUATIONS, & FITTING STATISTICS

An artificial neural network (NN) is a metamodel similar in function to a response surface equation (RSE) or other regression models. The purpose of an artificial NN is to compactly represent large amounts of data in equation form. The theory underpinning NN design and creation stems from the study of biological nervous systems. As Rojas observes, artificial NNs attempt to model the information processing capabilities of nervous systems[79]. The methods and tools used to create NN equations are designed to mimic the intuitive and associative learning processes observed in animals with complex nervous systems. Therefore, NNs are adept at modeling complex patterns in datasets which regression techniques are unable to capture with high fidelity.

The general architecture of an artificial NN is presented in Figure A.1. This general architecture consists of successive layers of nodes where arithmetic operations are performed. (These nodes are sometimes referred to as neurons in the literature.) Recall that every NN equation is a function mapping one or more input variables, p_i to an output variable, q (i.e. $q = f(p_1, p_2, \ldots, p_n)$ where n is the total number of inputs to the NN equation). Thus the black nodes in the input layer each represent an input variable p_i to the NN function. Each of the training cases in the training dataset provides one input to each of the nodes in the input layer. For the application herein, each training case consists of the altitude (p_1), latitude (p_2), and longitude (p_3) of a point in the Martian atmosphere for which MCD density information is available.

At each of the hidden nodes, an arithmetic operation is performed. For example, the output of the i^{th} hidden node could be the sine, square, or tangent of its input. This value would then be output to every other node which is connected to the i^{th} node. The data received at the output layer (far right layer in the figure) after all arithmetic

operations have been performed represents the NN function output q. Note in Figure A.1 that multiple output variables exist. However, for all NNs represented herein, only one output q is generated. This output q is the atmospheric density predicted by the NN given the three input variables describing a spatial location in the Martian atmosphere; altitude (p_1), latitude (p_2), and longitude (p_3).

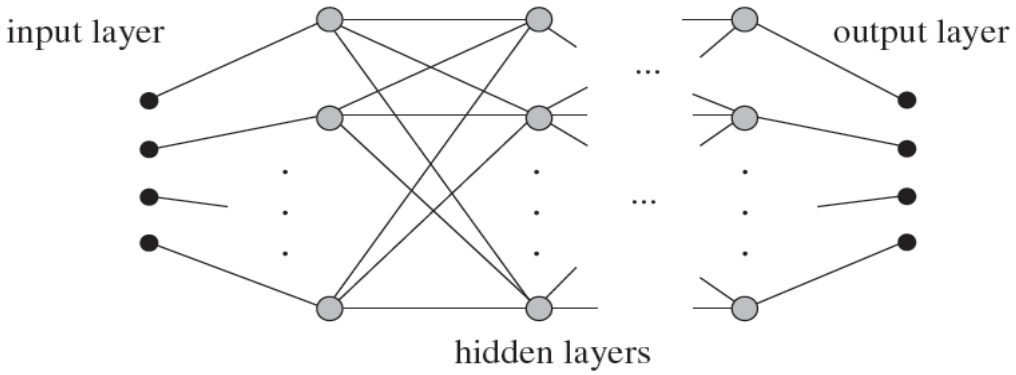


Figure A.1. General Neural Network Architecture[79]

The basic process for creating an artificial NN first requires the acquisition of a training dataset representing the input variables, p_i. In this case, the Mars Climate Database (MCD) was used to generate tables of input data which describe the atmospheric density in the entry corridor at the time of the entry for various atmospheric conditions of interest (e.g. varying levels of dust optical depth, solar flux, etc.).

Next, a training algorithm iteratively examines the NN output and modifies the architecture of the hidden layers until the output (i.e. the atmospheric density) matches the known output from the training data. The training algorithm may alter the hidden layer architecture by changing the weights applied to incoming or outgoing data from each of the nodes. It may change the nature of the arithmetic operation performed inside each node. It may change how the nodes are connected to adjacent nodes. Because many training methods are available, specific training algorithms are not discussed here. For a more complete introduction to NN training, including the biological origin of network

training as well as popular training algorithms, the introductory text of Rojas[79] is recommended.

Following NN training, a model validation measure of some form is often implemented. This validation procedure is necessary to ensure a good model fit and to confirm that smooth interpolation between members of the training dataset is possible. Once the NN is created, the arithmetic operations represented by each of the NN nodes may be collected and summarized as a single equation in order to simplify integration into larger algorithms.

The thirty two NNs used herein to form the neural ensemble were created in the statistical software package JMP using single hidden layer perceptron NN models with logistic activation. Note that a perceptron is a NN wherein the connections between the nodes are weighted. Therefore, only one layer of nodes separates the input nodes (latitude, longitude, and altitude) from the output node (density). The number of nodes in the single hidden layer depends on the nature of the training data which the NN is attempting to describe. In general, more complex datasets require more nodes to generate adequate model fits. Because different atmospheric conditions create differing levels of atmospheric instability, the NNs created herein have differing numbers of hidden nodes. The basic arithmetic operation performed at each node is to apply the logistic function:

$$f(x) = \frac{1}{1+e^{-x}} \tag{A.1}$$

The arithmetic operation employed in each of the hidden nodes is commonly referred to as the activation function. The logistic activation function is commonly used in NN modeling packages because it introduces nonlinearity into the model but always maintain a positive derivative. As Rojas[79] explains, maintaining a positive, finite derivative is useful in many popular training algorithms using gradient search methods. The results of this operation at each of the hidden nodes are then summed at the output layer to form the basis of the NN equation. Thus, each NN is a sum of logistic equations of the form:

$$\rho = \left[C_0 + \sum_i C_i f(x_i) \right] A + B \tag{A.2}$$

where A, B, C_0, and C_i are constants which are determined in the training process by JMP and the index i represents the i[th] node in the hidden layer. The logistic function $f(x)$ implemented at each hidden node is a function of the three input variables of altitude, latitude, and longitude. For all the NNs herein, the logistic activation function is of the form

$$f(x_i) = \frac{1}{1 + e^{-x_i}} \tag{A.3}$$

where

$$x_i = I_i + \frac{ALT_i(h - ALT_A)}{ALT_B} + \frac{LAT_i(\theta - LAT_A)}{LAT_B} + \frac{LONG_i(\varphi - LONG_A)}{LONG_B}$$

The constants of the above equation (I, ALT_A, ALT_B, ALT_i, LAT_A, LAT_B, LAT_i, $LONG_A$, $LONG_B$, & $LONG_i$) are determined by the NN training process in JMP and the index i represents the i[th] node in the hidden layer. Note that the logistic function $f(x_i)$ is a function of the three NN input variables (altitude h, latitude θ, and longitude φ). Thus the arithmetic operation at each of the hidden nodes transforms the three NN inputs into a component of the density equation A.2. JMP was utilized to create the thirty two NNs which form the onboard ensemble. The JMP fitting process requires both numerical and visual analysis to ensure sufficiently small modeling error. The NN model fitting statistics are presented in Table A.1 while the coefficients of the NN equations are presented in Tables A.2-4. The first three labeled columns of Table A.1 (Scenario, Region, and Altitude) describe the data set being modeled. The number of nodes describes the number of hidden neurons necessary to achieve the stated modeling statistics. The coefficient of determination (R^2) and the coefficient of determination following cross validation (CV R^2) numerically quantify the general quality of the fit where better models have R^2 values closer to unity. Cross validation is a process of

creating a NN model using only a portion of the available training data with the remaining data (1/3 of the data in this case) withheld. The withheld data is then used to test the model to ensure sufficiently high predictive accuracy. If the model is valid, the cross validation R^2 should be close in value to the initial R^2 as is the case for all the models herein. The actual-predicted and residuals-predicted columns are the results of visual inspections of two types of error plots. The metric labeled γ results from numerical analysis of the residuals-predicted plot. It should be generally less than 5-6% to ensure a good model fit. After the NN model is created, the original data is tested in the model and the resulting errors are all well described by Gaussian distributions. The last two columns (mean and std dev) describe the means and standard deviations respectively of these Gaussian error distributions expressed in percentage error.

Table A.1. Ensemble NN Fitting Statistics

							Actual -	Residuals-		Model Error	
NN	Scenario	Region	Altitude	# nodes	R^2	CV R^2	Predicted	Predicted	γ, %	mean%	std dev%
1	Average	North	Low	13	0.99995	0.99994	OK	OK	4	0.0325	2.54
2	Average	North	High	10	0.99967	0.99968	OK	OK	1.25	0.055	3.38
3	Average	South	Low	9	0.99996	0.99996	OK	OK	5	0.023	2.048
4	Average	South	High	12	0.99977	0.99972	OK	OK	2	0.014	3.163
5	Minimum	North	Low	11	0.99994	0.99994	OK	OK	4.5	0.0163	2.27
6	Minimum	North	High	8	0.99974	0.99975	OK	OK	3.13	0.0479	3.154
7	Minimum	South	Low	9	0.99995	0.99995	OK	OK	5	0.022	2.102
8	Minimum	South	High	15	0.99976	0.99976	OK	OK	1.67	0.049	3.13
9	Maximum	North	Low	11	0.99994	0.99993	OK	OK	4.5	0.039	2.792
10	Maximum	North	High	11	0.99959	0.99955	OK	OK	1.56	0.074	3.818
11	Maximum	South	Low	9	0.99995	0.99994	OK	OK	5	0.0217	2.138
12	Maximum	South	High	10	0.99971	0.99969	OK	OK	1.83	0.052	3.327
13	Warm	North	Low	7	0.99995	0.99994	OK	OK	4.5	0.028	2.338
14	Warm	North	High	13	0.99954	0.99953	OK	OK	1.67	0.0725	3.847
15	Warm	South	Low	9	0.99994	0.99994	OK	OK	4	0.027	2.278
16	Warm	South	High	11	0.99965	0.99964	OK	OK	1.73	0.066	3.653
17	Cold	North	Low	6	0.99994	0.99993	OK	OK	6	0.038	2.827
18	Cold	North	High	9	0.99979	0.99977	OK	OK	1.25	0.0424	2.901
19	Cold	South	Low	7	0.99996	0.99996	OK	OK	4.5	0.025	2.218
20	Cold	South	High	14	0.99978	0.99975	OK	OK	1.25	0.0472	3.006
21	Dust Avg	North	Low	8	0.9999	0.99989	OK	OK	1.875	-0.0131	3.304
22	Dust Avg	North	High	13	0.99966	0.99963	OK	OK	0.909091	0.0623	3.539
23	Dust Avg	South	Low	9	0.99989	0.99988	OK	OK	1.5625	0.0487	3.1049
24	Dust Avg	South	High	10	0.99968	0.99968	OK	OK	1.285714	0.063	3.636
25	Dust Low	North	Low	9	0.99995	0.99995	OK	OK	1.25	0.0241	2.191
26	Dust Low	North	High	13	0.99961	0.99964	OK	OK	0.909091	0.0745	3.821
27	Dust Low	South	Low	6	0.99994	0.99994	OK	OK	1.40625	-0.028	2.373
28	Dust Low	South	High	6	0.99968	0.99964	OK	OK	1.136364	0.0547	3.669
29	Dust High	North	Low	8	0.99989	0.99989	OK	OK	1.4375	0.0518	3.21
30	Dust High	North	High	8	0.99954	0.99954	OK	OK	1.071429	0.0746	3.881
31	Dust High	South	Low	8	0.99988	0.99988	OK	OK	1.625	0.0527	3.236
32	Dust High	South	High	10	0.99962	0.99963	OK	OK	1.190476	0.0725	3.86

γ = scaled residuals span = span of residuals / minimum of predicted
1/3 of training data withheld for cross-validation purposes

Table A.2. Model Coefficients for Neural Networks #1-8

NN	1	2	3	4	5	6	7	8
Nodes	13	10	9	12	11	8	9	15
I_1	0.621049	-1.61551	1.765673	-0.27647	0.282786	-0.47402	-2.80192	0.676045
I_2	-2.67946	0.486871	-0.51849	-0.41327	-0.28278	-0.72331	-0.90433	0.997213
I_3	0.322455	0.971849	0.126708	-0.11385	-0.28284	-1.26071	-0.30419	-0.13142
I_4	-0.18146	-0.72349	-1.10613	0.025198	-0.28293	1.066214	-0.29965	0.13141
I_5	-0.18502	-0.45788	-0.47636	0.396961	1.4932	-0.96667	-0.96207	-0.13143
I_6	-0.18191	0.394205	-0.35022	-0.435	0.402463	-0.32075	1.080283	-1.22928
I_7	0.182086	-1.22971	-3.00015	0.259056	-0.28395	-0.25251	-0.19545	0.131332
I_8	1.236334	0.624164	-1.03035	1.894586	-1.54841	1.785239	0.241379	0.059375
I_9	0.203247	0.03756	0.317826	0.995049	-0.28278	0	-0.14856	0.131381
I_{10}	-0.1821	-0.55206	0	0.288187	0.517579	0	0	0.148745
I_{11}	0.18215	0	0	-0.51222	1.397028	0	0	-0.13162
I_{12}	0.318908	0	0	-0.31354	0	0	0	-0.13143
I_{13}	-0.07752	0	0	0	0	0	0	0.131402
I_{14}	0	0	0	0	0	0	0	-0.13142
I_{15}	0	0	0	0	0	0	0	-0.13143
ALT_1	1.406308	-1.06854	1.175109	-1.04606	-0.66751	-2.23082	1.266049	-2.06511
LAT_1	0.197957	-0.12167	-0.60121	1.273755	-0.04083	-0.41191	0.09836	-0.45411
$LONG_1$	0.118708	-0.05678	0.075877	0.075971	-0.01384	-0.46339	0.086931	-0.11961
ALT_2	1.265871	0.613923	-0.86021	0.140647	0.667572	0.411548	1.105676	0.036668
LAT_2	0.237236	-0.48893	-1.32815	0.758958	0.040791	-0.17967	-0.87583	-1.03926
$LONG_2$	0.169452	-1.14077	-0.0109	0.619392	0.014111	0.163757	0.073089	-0.4238
ALT_3	0.524618	-0.13983	0.244415	0.406646	0.666745	0.691111	0.474168	-0.39709
LAT_3	-0.48065	-0.51953	0.577782	0.259778	0.041255	0.81037	0.125379	0.056449
$LONG_3$	-0.40707	-0.41435	-0.23447	-0.00485	0.010514	-0.46417	-0.13357	-0.02048
ALT_4	0.485108	-2.51866	1.561499	-0.13716	0.665861	0.085517	0.055692	0.396368
LAT_4	0.016943	-0.32806	-1.08181	-1.58186	0.041757	-0.48616	1.057293	-0.05639
$LONG_4$	-0.07769	-0.44117	0.076818	0.144719	0.006599	-0.39539	-0.06959	0.020072
ALT_5	0.48616	0.882974	-0.14602	0.456265	1.841951	0.788213	-0.85796	-0.39711
LAT_5	0.017579	-0.39847	1.010319	0.287609	-0.36065	-0.25612	-0.33203	0.056451
$LONG_5$	-0.08062	0.164065	0.060247	0.886834	-0.05452	-0.34342	-0.05576	-0.02049
ALT_6	0.485514	-1.38209	1.979407	2.358308	0.113477	-0.17863	0.485894	-0.7718
LAT_6	0.016715	0.038999	0.030618	0.440128	0.776066	0.397244	-0.53961	0.160468
$LONG_6$	-0.07729	0.356066	0.028067	0.199473	-0.10677	1.147388	0.128272	0.003479
ALT_7	-0.48559	0.584863	1.417541	0.642419	0.656506	-0.556	1.964533	0.395177
LAT_7	-0.01672	0.743833	-0.02009	-0.20695	0.047152	-0.32956	0.116515	-0.05629
$LONG_7$	0.077365	-0.47714	0.081166	0.044276	-0.03513	-0.26033	0.052145	0.019428
ALT_8	1.404825	0.849655	-0.85401	1.169773	2.026612	1.116995	0.800641	0.012272
LAT_8	-0.48225	-0.13341	-0.56667	-0.15656	0.160711	0.049808	1.222198	-1.71727
$LONG_8$	-0.16777	-0.13231	-0.1019	-0.05616	0.213409	-0.02552	-0.00374	0.25198

Table A.2. (continued)

ALT_9	-0.49383	-2.22367	0.263337	-0.98327	0.667526	0	0.469532	0.397632
LAT_9	-0.01864	0.256216	0.40583	0.232795	0.040817	0	0.11021	-0.05649
$LONG_9$	0.089624	0.052405	0.496517	0.455878	0.013906	0	-0.09065	0.020802
ALT_{10}	0.485338	0.831018	0	-0.63134	0.594415	0	0	0.140761
LAT_{10}	0.01701	0.827597	0	-0.59881	-0.26652	0	0	0.669623
$LONG_{10}$	-0.0781	0.96383	0	0.813388	-0.21862	0	0	0.586576
ALT_{11}	-0.48534	0	0	0.012973	-0.94709	0	0	-0.39513
LAT_{11}	-0.01703	0	0	-1.59136	-0.20448	0	0	0.056323
$LONG_{11}$	0.078172	0	0	-0.49512	-0.16245	0	0	-0.01929
ALT_{12}	0.125538	0	0	0.611891	0	0	0	-0.39687
LAT_{12}	0.698899	0	0	-0.4255	0	0	0	0.056432
$LONG_{12}$	-0.14298	0	0	0.343382	0	0	0	-0.02035
ALT_{13}	0.243577	0	0	0	0	0	0	0.397356
LAT_{13}	-0.21823	0	0	0	0	0	0	-0.05647
$LONG_{13}$	-0.6998	0	0	0	0	0	0	0.020637
ALT_{14}	0	0	0	0	0	0	0	-0.39727
LAT_{14}	0	0	0	0	0	0	0	0.056462
$LONG_{14}$	0	0	0	0	0	0	0	-0.02058
ALT_{15}	0	0	0	0	0	0	0	-0.39683
LAT_{15}	0	0	0	0	0	0	0	0.056429
$LONG_{15}$	0	0	0	0	0	0	0	-0.02032
C_0	0.389822	1.071988	2.18454	3.003378	-0.00298	2.64255	1.937184	-2.17331
C_1	-1.12745	1.911941	-1.36662	-0.42849	1.105512	0.786673	-2.2094	0.641256
C_2	-1.88581	0.547173	-0.60785	-0.96793	-1.10551	-1.01756	-0.46091	0.779168
C_3	-0.62149	-1.19179	-0.63384	-0.96985	-1.10551	-0.63666	-1.03964	0.638639
C_4	-0.71493	0.836888	-0.50556	0.836477	-1.10555	-1.1575	-0.69695	-0.63869
C_5	-0.7175	-0.82203	-0.68214	0.540095	0.713954	-1.16457	1.333422	0.638639
C_6	-0.71513	0.941143	-0.89901	-0.67523	0.478812	-0.55512	-1.08905	1.372017
C_7	0.715245	-0.64398	-2.54187	-1.07994	-1.10607	0.998841	-0.66978	-0.6387
C_8	0.616575	-1.19922	1.301796	-2.0202	0.768244	-1.83313	0.62434	0.481085
C_9	0.729772	-0.73579	0.292919	1.13943	-1.10551	0	-0.99929	-0.63855
C_{10}	-0.71537	-0.51676	0	-0.44356	-0.79585	0	0	0.436031
C_{11}	0.715417	0	0	-0.53063	1.657747	0	0	0.638929
C_{12}	0.578129	0	0	-1.02155	0	0	0	0.638664
C_{13}	0.448435	0	0	0	0	0	0	-0.6386
C_{14}	0	0	0	0	0	0	0	0.638618
C_{15}	0	0	0	0	0	0	0	0.63867
A	3.727505	2.010381	3.424234	1.995987	3.532479	2.126388	3.42971	2.080528
B	-9.05645	-19.1232	-8.64415	-18.4791	-8.7254	-19.253	-8.64681	-18.5653
ALT_A	43.83333	128	43.83333	128	43.83333	128	43.83333	128
ALT_B	31.54537	15.49745	31.54537	15.4948	31.54739	15.49745	31.54537	15.49767
LAT_A	45	22.5	-45	-45	22.5	22.5	-45	-45

Table A.2. (continued)

LAT_B	27.04021	14.03008	27.04021	27.04334	14.02696	14.03008	27.04021	27.04835
$LONG_A$	70.30833	70.30833	70.30833	70.30833	70.30833	70.30833	70.30833	70.31667
$LONG_B$	19.41374	19.4193	19.41374	19.41598	19.41498	19.4193	19.41374	9.630858

Table A.3. Model Coefficients for Neural Networks #9-16

NN	9	10	11	12	13	14	15	16
Nodes	11	11	9	10	7	13	9	11
I_1	-0.12868	-0.18723	-1.11229	-1.07554	-0.13147	-1.25539	-2.26826	0.039154
I_2	-0.12842	-0.79532	-0.59524	-0.65195	-1.74013	0.077074	-0.07887	-0.03945
I_3	0.128844	-1.04943	0.554184	-0.06649	0.641765	-1.26148	0.050861	-1.27644
I_4	0.128728	0.729315	1.687753	-0.55361	-1.29992	0.076095	-0.13398	-0.03918
I_5	-0.59892	-0.31033	-1.0678	1.191368	-0.70054	-0.18565	-0.33596	1.811116
I_6	-0.18659	-0.01596	-0.47797	0.164058	0.829186	-0.07751	-0.85853	-0.70011
I_7	2.682039	0.813585	0.313963	0.680353	-1.00314	0.07757	-0.11211	0.039177
I_8	0.329819	-1.11131	0.327439	1.178184	0	-0.86845	-0.39151	0.597307
I_9	0.767521	-0.01597	-2.9875	2.154479	0	-0.0773	0	0.039174
I_{10}	-0.55609	-0.74157	0	0.335847	0	0.81219	0	-0.2032
I_{11}	-0.30204	2.2436	0	0	0	0.122703	0	-0.03914
I_{12}	0	0	0	0	0	-0.16117	0	0
I_{13}	0	0	0	0	0	-0.83956	0	0
I_{14}	0	0	0	0	0	0	0	0
I_{15}	0	0	0	0	0	0	0	0
ALT_1	0.457673	-0.47974	1.55357	0.423955	-1.97504	-0.76702	1.333303	0.403256
LAT_1	0.070367	-0.99703	-1.06996	0.221518	0.055104	0.802223	0.287452	-0.26275
$LONG_1$	-0.0608	0.170522	0.098617	-0.5027	-0.24553	0.575702	0.004858	0.091392
ALT_2	0.459274	0.712033	0.498897	0.52034	1.197051	0.475741	-0.90904	-0.40303
LAT_2	0.069208	0.771447	0.106051	0.911044	-0.281	0.022454	0.244672	0.261746
$LONG_2$	-0.06148	1.205745	-0.153	-0.75837	-0.22539	-0.00519	-0.07111	-0.09193
ALT_3	-0.456	-2.82936	0.143921	0.153915	-0.21094	-1.53294	-0.86267	0.819734
LAT_3	-0.07152	-0.52297	-1.08035	1.779501	-0.95251	-0.12543	0.204236	1.15827
$LONG_3$	0.060059	-0.56056	-0.02424	-0.66041	0.089463	-0.13413	0.002909	-0.04024
ALT_4	-0.45731	-0.37034	1.108636	0.737543	-0.43932	0.479337	0.077027	-0.40311
LAT_4	-0.07062	0.377969	-0.64565	0.552119	0.159522	0.022857	1.078702	0.262852
$LONG_4$	0.060645	-0.3353	0.091943	0.254623	0.519576	-0.00734	0.006569	-0.0907
ALT_5	-1.38222	0.769228	-0.91415	-1.22624	-0.96349	0.630539	-0.73998	1.16661
LAT_5	-0.25949	-0.20739	-0.56621	0.357636	-0.11206	-0.03155	-1.21724	0.44777
$LONG_5$	-0.15638	-0.37884	-0.07802	0.141392	-0.12225	0.229607	-0.03427	-0.05859
ALT_6	0.025011	-0.37727	-0.82354	-2.76749	-0.82803	-0.47541	-0.70941	0.438845
LAT_6	-0.71534	-0.12965	-1.25117	-0.39635	0.078276	-0.02251	-0.39214	-0.43508
$LONG_6$	0.156242	0.079905	-0.0327	-0.15417	-0.38678	0.005159	0.032928	-0.21342
ALT_7	-1.29069	-0.84013	0.171386	-0.45363	0.807565	0.473763	0.850417	0.403128
LAT_7	-0.25489	-0.74098	0.543614	0.199691	0.417855	0.022228	-0.15817	-0.26285

Table A.3. (continued)

$LONG_7$	-0.17434	0.224159	-0.16885	-0.38296	0.023413	-0.00401	-0.01363	0.09078
ALT_8	0.169135	-0.16199	-2.0791	0.298613	0	-0.4497	-2.21997	0.630266
LAT_8	-0.57623	0.503364	-0.0677	-1.61119	0	-0.34814	0.410686	-1.48396
$LONG_8$	-0.27751	0.586662	-0.05664	-0.54823	0	-0.78923	-0.05612	0.131668
ALT_9	-0.88038	-0.37725	1.386224	1.414069	0	-0.47505	0	0.403137
LAT_9	0.248782	-0.12966	-0.0461	-0.13381	0	-0.02239	0	-0.26286
$LONG_9$	0.221397	0.079939	0.102297	0.014914	0	0.004803	0	0.090809
ALT_{10}	-0.73414	-0.72807	0	0.533501	0	0.40323	0	2.461975
LAT_{10}	0.728574	0.654818	0	0.624523	0	0.175579	0	0.346805
$LONG_{10}$	-0.01683	1.207516	0	0.740819	0	-0.36351	0	0.005649
ALT_{11}	0.35584	1.317564	0	0	0	0.712742	0	-0.40315
LAT_{11}	-0.33686	0.201062	0	0	0	-0.49107	0	0.262911
$LONG_{11}$	-0.70409	0.1289	0	0	0	-1.00335	0	-0.09078
ALT_{12}	0	0	0	0	0	-0.60038	0	0
LAT_{12}	0	0	0	0	0	-1.19426	0	0
$LONG_{12}$	0	0	0	0	0	0.171189	0	0
ALT_{13}	0	0	0	0	0	-0.45732	0	0
LAT_{13}	0	0	0	0	0	-1.09635	0	0
$LONG_{13}$	0	0	0	0	0	-0.45622	0	0
ALT_{14}	0	0	0	0	0	0	0	0
LAT_{14}	0	0	0	0	0	0	0	0
$LONG_{14}$	0	0	0	0	0	0	0	0
ALT_{15}	0	0	0	0	0	0	0	0
LAT_{15}	0	0	0	0	0	0	0	0
$LONG_{15}$	0	0	0	0	0	0	0	0
C_0	-2.42509	0.259878	1.354096	0.220064	-0.27571	0.62911	0.035109	1.917819
C_1	-0.76335	-0.43705	-0.53508	-1.04856	-0.77979	1.034701	-1.93759	-0.62534
C_2	-0.76315	-0.49144	-0.94866	0.703742	-1.69951	-0.77788	1.365605	0.624232
C_3	0.763473	0.681655	0.623628	-0.45308	-0.43089	1.666516	1.33384	-0.84451
C_4	0.763388	0.734947	-1.31139	-1.03685	0.972745	-0.77736	-0.67046	0.625388
C_5	1.136354	-0.94587	1.078663	1.014906	1.816812	-0.89928	-0.61161	-1.8373
C_6	-0.59757	0.637478	-0.66828	0.653754	1.47043	0.777956	1.246312	-0.8249
C_7	2.076519	1.004047	-0.70785	0.894957	-1.94579	-0.77806	-1.31916	-0.6254
C_8	-0.62334	1.418766	0.713098	0.46676	0	-0.55107	-0.5751	0.707708
C_9	1.048629	0.637478	-2.3609	-2.17982	0	0.777957	0	-0.62541
C_{10}	-0.45803	-0.569	0	0.480087	0	-1.07278	0	-0.64329
C_{11}	0.344856	-1.99371	0	0	0	0.986866	0	0.625464
C_{12}	0	0	0	0	0	-0.88254	0	0
C_{13}	0	0	0	0	0	0.761355	0	0
C_{14}	0	0	0	0	0	0	0	0
C_{15}	0	0	0	0	0	0	0	0
A	3.52577	1.950763	3.424273	1.952028	3.451546	1.85154	3.253927	1.999472

Table A.3. (continued)

B	-8.72103	-19.074	-8.64229	-18.4536	-8.57536	-18.8804	-8.41218	-18.2296
ALT_A	43.83333	128	43.83333	128	43.83333	128	43.83333	128
ALT_B	31.54739	15.49745	31.54537	15.4948	31.54739	15.49745	31.54844	15.49882
LAT_A	22.5	22.5	-45	-45	22.5	22.5	-45	-45
LAT_B	14.02696	14.03008	27.04021	27.04334	14.02696	14.03008	27.04284	27.05035
$LONG_A$	70.30833	70.30833	70.30833	70.30833	70.30833	70.30833	67.5	67.5
$LONG_B$	19.41498	19.4193	19.41374	19.41598	19.41498	19.4193	7.977544	7.979762

Table A.4. Model Coefficients for Neural Networks #17-24

NN	17	18	19	20	21	22	23	24
Nodes	6	9	7	14	8	13	9	10
I_1	0.068859	-0.50902	-2.96637	0.206482	0.612022	-0.03639	1.532815	-2.17623
I_2	0.08896	0.910966	-0.45347	0.206028	-0.8223	-0.54193	3.959451	0.040869
I_3	-0.15075	-0.41537	0.169347	-0.20586	-0.10457	-0.03637	0.241988	-0.56396
I_4	2.873016	0.548708	-0.46987	-0.87814	0.103154	-0.03637	0.802489	0.041084
I_5	0.900633	0.422053	0.130223	-0.20643	3.669213	0.524198	-0.02718	-0.38296
I_6	0.879811	-0.57037	0.025573	-0.2098	0.104715	0.674197	0.026361	-0.27831
I_7	0	0.706559	-0.90478	-0.4518	-1.50782	-0.03651	0.613304	-0.04086
I_8	0	-1.05407	0	0.20942	-0.30769	0.036379	-0.30177	0.041068
I_9	0	-0.99295	0	0.371436	0	-0.03637	1.902856	0.040812
I_{10}	0	0	0	-0.0047	0	-0.03678	0	-1.17748
I_{11}	0	0	0	0.206303	0	-1.1472	0	0
I_{12}	0	0	0	-0.20637	0	0.036371	0	0
I_{13}	0	0	0	-0.77487	0	0.036377	0	0
I_{14}	0	0	0	0.20641	0	0	0	0
I_{15}	0	0	0	0	0	0	0	0
ALT_1	-2.6789	0.626728	1.501989	0.437857	-0.01629	-0.42858	1.462442	-1.21247
LAT_1	0.156893	-0.32382	-0.03285	-0.12825	-0.51537	-0.12103	-0.2676	-0.19492
$LONG_1$	0.007694	-0.28312	0.025134	0.036195	-0.27198	-0.02558	0.014322	0.052659
ALT_2	0.480134	0.186453	-0.72578	0.437555	1.080556	-0.39687	-2.42821	0.402088
LAT_2	-0.08569	-0.6986	-1.15756	-0.12847	0.560605	0.914996	-0.03696	-0.03095
$LONG_2$	-0.08957	-1.33733	-0.1355	0.036319	0.230757	0.169693	-0.05517	0.00352
ALT_3	-0.40564	1.391074	0.729869	-0.43745	0.281032	-0.42783	0.835585	-0.17128
LAT_3	-0.749	0.501866	-0.60485	0.128557	-0.14401	-0.12062	-0.62446	0.061087
$LONG_3$	0.015805	0.56838	-0.05133	-0.03637	-0.13912	-0.02662	0.159998	-0.43094
ALT_4	-1.42256	-0.6638	-0.29698	-0.71857	0.912924	-0.42824	-0.5372	0.3965
LAT_4	-0.01015	-0.44474	-0.51236	0.147426	-0.75991	-0.12087	0.900472	-0.02147
$LONG_4$	-0.05099	0.335804	0.039663	-0.00601	-0.10487	-0.02608	-0.02163	-0.01039
ALT_5	-0.66581	-0.71161	-2.04689	-0.43783	-2.33418	0.778415	-0.28168	0.144236
LAT_5	-0.40423	0.382537	-0.16826	0.128275	-0.10065	1.174777	-0.21281	1.200976
$LONG_5$	-0.00862	-0.27883	0.001107	-0.03621	-0.09093	0.437089	0.100694	0.063011
ALT_6	1.43965	-1.19835	-0.04241	-0.43668	-0.2802	-0.08847	0.281102	2.205385

Table A.4. (continued)

LAT_6	0.206931	-0.10855	0.907538	0.124401	0.143724	0.821265	0.212849	0.572687
$LONG_6$	0.048958	-0.1197	0.121699	-0.03152	0.137991	0.481913	-0.10221	0.091985
ALT_7	0	0.8472	-1.34026	0.253957	-1.40167	-0.42802	-0.00361	-0.40241
LAT_7	0	0.177707	0.14375	-0.29546	-0.16433	-0.1206	-0.77807	0.03151
$LONG_7$	0	0.048649	-0.02651	-0.4398	-0.06742	-0.02639	-0.0616	-0.00433
ALT_8	0	2.240615	0	0.435405	0.183884	0.428055	-0.37561	0.396952
LAT_8	0	0.251929	0	-0.12382	-0.65877	0.120752	-0.91665	-0.02222
$LONG_8$	0	0.410068	0	0.03044	0.042661	0.026339	-0.13896	-0.00929
ALT_9	0	0.118311	0	0.39016	0	-0.42798	-1.86075	0.403867
LAT_9	0	0.47454	0	-1.30676	0	-0.12071	0.693109	-0.03404
$LONG_9$	0	0.542102	0	-0.01813	0	-0.02643	-0.05026	0.008018
ALT_{10}	0	0	0	2.248972	0	-0.42933	0	0.632344
LAT_{10}	0	0	0	0.362834	0	-0.12095	0	-0.48516
$LONG_{10}$	0	0	0	0.06878	0	-0.02422	0	-0.18713
ALT_{11}	0	0	0	0.437736	0	-1.36918	0	0
LAT_{11}	0	0	0	-0.12834	0	-0.054	0	0
$LONG_{11}$	0	0	0	0.036241	0	-0.00139	0	0
ALT_{12}	0	0	0	-0.43778	0	0.428022	0	0
LAT_{12}	0	0	0	0.128309	0	0.120739	0	0
$LONG_{12}$	0	0	0	-0.03622	0	0.026381	0	0
ALT_{13}	0	0	0	0.537903	0	0.427988	0	0
LAT_{13}	0	0	0	0.979343	0	0.120712	0	0
$LONG_{13}$	0	0	0	0.033398	0	0.026425	0	0
ALT_{14}	0	0	0	0.437809	0	0	0	0
LAT_{14}	0	0	0	-0.12829	0	0	0	0
$LONG_{14}$	0	0	0	0.036213	0	0	0	0
ALT_{15}	0	0	0	0	0	0	0	0
LAT_{15}	0	0	0	0	0	0	0	0
$LONG_{15}$	0	0	0	0	0	0	0	0
C_0	-1.54028	-0.93352	-0.07116	0.371036	-1.73328	-1.78969	-0.48694	1.496927
C_1	0.484409	-0.92723	-2.36049	-0.655	-0.77817	0.618118	-1.93485	1.876292
C_2	-0.89279	0.534561	-0.45849	-0.65459	-1.0014	1.019726	2.036199	-0.55129
C_3	-0.63007	-1.10464	-0.73045	0.654441	-0.57999	0.618231	-0.6498	0.564344
C_4	2.292475	0.925773	0.774273	1.178682	0.57317	0.618206	-0.93556	-0.55222
C_5	1.236218	0.828595	0.920743	0.654956	2.337024	0.616218	0.508698	-0.9888
C_6	-1.35219	1.5773	-0.7059	0.658231	0.580099	-0.66815	-0.50826	-0.81277
C_7	0	-1.43211	1.30185	-0.51651	2.035941	0.618227	0.904349	0.55123
C_8	0	0.774294	0	-0.65797	-0.74278	-0.61823	-0.52169	-0.55216
C_9	0	1.303778	0	0.987387	0	0.618232	0.987632	-0.55093
C_{10}	0	0	0	-0.49259	0	0.617637	0	-1.28126
C_{11}	0	0	0	-0.65484	0	1.870136	0	0
C_{12}	0	0	0	0.654893	0	-0.61823	0	0

Table A.4. (continued)

C_{13}	0	0	0	-0.65776	0	-0.61823	0	0
C_{14}	0	0	0	-0.65493	0	0	0	0
C_{15}	0	0	0	0	0	0	0	0
A	3.742515	2.093029	3.593269	2.064919	3.249709	1.883784	3.04986	2.076872
B	-8.96338	-19.6658	-8.81648	-18.925	-8.40308	-18.798	-8.21917	-18.0984
ALT_A	43.83333	128	43.83333	128	43.83333	128	43.83333	128
ALT_B	31.54739	15.49745	31.54537	15.49882	31.55161	15.50298	31.5484	15.49767
LAT_A	22.5	22.5	-45	-45	22.5	22.5	-37.5	-45
LAT_B	14.02696	14.03008	27.04021	27.05035	14.02883	14.03509	22.70723	27.04835
$LONG_A$	70.30833	70.30833	70.30833	67.5	70.31667	70.31667	70.31667	70.31667
$LONG_B$	19.41498	19.4193	19.41374	7.979762	9.629863	9.634155	9.628883	9.630858

Table A.5. Model Coefficients for Neural Networks #25-32

NN	25	26	27	28	29	30	31	32
Nodes	9	13	6	6	8	8	8	10
I_1	0.337035	0.712334	-0.20012	0.425896	0.68871	0.128817	3.92033	1.89444
I_2	0.278487	-0.58327	2.506259	-1.88326	-1.48398	0.15148	-0.24817	-0.1211
I_3	-0.80487	-0.08602	-0.61764	-0.60667	0.939135	0.164572	-0.33949	-0.10794
I_4	-1.21446	-0.43663	-1.43786	-1.38205	-0.11136	-0.80906	-0.57826	-0.10795
I_5	-0.2019	-0.47479	1.036723	-0.9585	0.334249	1.129886	-0.1128	-0.1078
I_6	-0.00818	0.091552	-1.77021	0.258528	-0.11158	-0.69053	0.790002	-0.12073
I_7	0.276906	0.661609	0	0	3.802754	0.484341	1.525775	0.141877
I_8	0.273899	0.731669	0	0	0.070367	-0.54282	1.887156	0.107922
I_9	0.279082	-0.0952	0	0	0	0	0	-1.20795
I_{10}	0	-0.09099	0	0	0	0	0	-0.10797
I_{11}	0	0.148469	0	0	0	0	0	0
I_{12}	0	-0.10088	0	0	0	0	0	0
I_{13}	0	0.477006	0	0	0	0	0	0
I_{14}	0	0	0	0	0	0	0	0
I_{15}	0	0	0	0	0	0	0	0
ALT_1	-0.03488	1.243876	-0.20913	0.129227	-0.07614	0.572902	-2.45696	1.09004
LAT_1	-0.66844	0.235255	1.212975	0.518521	-0.4879	-0.04772	-0.03799	0.166303
$LONG_1$	-0.07732	0.100528	0.06438	0.597244	-0.2731	-0.06436	-0.04717	-0.00291
ALT_2	-0.67465	0.51068	-1.53118	-1.22492	-1.36377	0.584443	-0.86393	2.356636
LAT_2	-0.08329	0.11713	-0.3603	-0.05593	-0.19875	-0.04592	0.646757	0.540731
$LONG_2$	-0.02468	0.418454	-0.01474	-0.05934	-0.0772	-0.05724	-0.16372	0.09985
ALT_3	0.840706	-0.30258	-0.66527	1.613495	-1.16223	-0.85942	-0.35976	-0.41934
LAT_3	0.26549	0.011288	-1.33289	0.225494	-0.60334	-0.41094	-0.93788	-0.06365
$LONG_3$	0.047067	0.097396	-0.10095	0.058762	-0.24426	-0.3932	-0.13438	-0.05056
ALT_4	1.060679	-0.65125	1.413554	0.807883	0.248579	0.072424	-0.00987	-0.41934
LAT_4	-0.33513	0.117404	-0.43808	-0.80291	-0.20677	-0.89045	0.766266	-0.06365
$LONG_4$	0.012971	0.034134	0.007078	0.032933	-0.13979	-0.32605	0.053127	-0.05057

Table A.5. (continued)

ALT_5	0.607212	0.011881	0.838677	0.494532	-0.15756	1.278307	-0.40147	-0.41945
LAT_5	0.018984	1.065914	0.506349	0.576643	0.56227	0.059056	-0.23046	-0.06355
$LONG_5$	-0.07011	0.046875	0.002748	0.137733	-0.02932	0.008862	0.11083	-0.05004
ALT_6	1.989186	0.309635	-1.30905	-0.38348	0.247921	-0.93872	-0.54729	-0.42632
LAT_6	-0.0984	-0.01401	0.458707	1.578924	-0.20648	0.058875	0.894632	-0.05615
$LONG_6$	0.022052	-0.09426	-0.00095	0.024722	-0.13881	-0.0439	-0.02555	0.028416
ALT_7	-0.66761	0.779827	0	0	-2.49197	0.707981	1.469248	-0.20394
LAT_7	-0.0761	-0.01851	0	0	-0.11438	1.113692	-0.2643	-1.13898
$LONG_7$	-0.01332	-0.02811	0	0	-0.10501	0.645978	0.009626	-0.05306
ALT_8	-0.66014	-0.2816	0	0	0.915236	-0.34376	-1.89363	0.419364
LAT_8	-0.06835	0.846098	0	0	-0.70823	0.838541	0.726521	0.063634
$LONG_8$	-0.00202	0.272253	0	0	-0.14871	0.137279	-0.04928	0.050483
ALT_9	-0.67993	-0.31426	0	0	0	0	0	0.428303
LAT_9	-0.08845	0.015802	0	0	0	0	0	-0.70261
$LONG_9$	-0.03376	0.092282	0	0	0	0	0	-0.13682
ALT_{10}	0	-0.30892	0	0	0	0	0	-0.41932
LAT_{10}	0	0.013733	0	0	0	0	0	-0.06368
$LONG_{10}$	0	0.094576	0	0	0	0	0	-0.05067
ALT_{11}	0	0.379573	0	0	0	0	0	0
LAT_{11}	0	-0.04155	0	0	0	0	0	0
$LONG_{11}$	0	-0.06915	0	0	0	0	0	0
ALT_{12}	0	-0.32144	0	0	0	0	0	0
LAT_{12}	0	0.018592	0	0	0	0	0	0
$LONG_{12}$	0	0.089313	0	0	0	0	0	0
ALT_{13}	0	0.658871	0	0	0	0	0	0
LAT_{13}	0	-0.11417	0	0	0	0	0	0
$LONG_{13}$	0	0.00562	0	0	0	0	0	0
ALT_{14}	0	0	0	0	0	0	0	0
LAT_{14}	0	0	0	0	0	0	0	0
$LONG_{14}$	0	0	0	0	0	0	0	0
ALT_{15}	0	0	0	0	0	0	0	0
LAT_{15}	0	0	0	0	0	0	0	0
$LONG_{15}$	0	0	0	0	0	0	0	0
C_0	-1.20114	1.783021	0.017215	0.979016	-2.64803	0.271715	-0.5457	0.493921
C_1	-0.58243	-1.43398	-0.51311	0.435608	-0.83969	-0.83379	2.055688	-1.82438
C_2	1.056194	-0.54759	1.843073	2.112642	2.142904	-0.84444	0.590821	-0.62588
C_3	-1.38453	0.423526	-0.51991	-1.34788	1.021655	0.848584	-0.49067	0.62657
C_4	-1.35697	0.841223	-1.5518	-1.40411	-0.65238	0.886242	-0.89471	0.626574
C_5	-0.99666	0.800198	-1.43847	-1.50966	0.741493	-1.79148	0.757244	0.626447
C_6	0.933833	-0.42913	1.315338	-0.52942	-0.65247	1.308233	-0.93866	0.636327
C_7	1.051486	-1.04583	0	0	2.148999	0.618289	-1.9283	0.903025
C_8	1.045972	-0.73465	0	0	0.54392	0.735213	0.920791	-0.62655

Table A.5. (continued)

C_9	1.059494	0.432905	0	0	0	0	0	-1.25296
C_{10}	0	0.428552	0	0	0	0	0	0.626597
C_{11}	0	-0.49279	0	0	0	0	0	0
C_{12}	0	0.438909	0	0	0	0	0	0
C_{13}	0	-0.87786	0	0	0	0	0	0
C_{14}	0	0	0	0	0	0	0	0
C_{15}	0	0	0	0	0	0	0	0
A	3.417015	1.946775	3.238627	2.068362	3.260688	1.861445	3.067066	2.039114
B	-8.55499	-18.8507	-8.4082	-18.2226	-8.41282	-18.7753	-8.23112	-18.1252
ALT_A	43.83333	128	43.83333	128	43.83333	128	43.83333	128
ALT_B	31.55161	15.50298	31.54756	15.49767	31.55161	15.50298	31.5484	15.49767
LAT_A	22.5	22.5	-45	-45	22.5	22.5	-37.5	-45
LAT_B	14.02883	14.03509	27.04208	27.04835	14.02883	14.03509	22.70723	27.04835
$LONG_A$	70.31667	70.31667	70.31667	70.31667	70.31667	70.31667	70.31667	70.31667
$LONG_B$	9.629863	9.634155	9.628628	9.630858	9.629863	9.634155	9.628883	9.630858

APPENDIX B

INTERPRETATION OF RMSE/MAX DENSITY ERROR PLOTS

The goal of the density estimators tested herein is to observe the free stream atmospheric density along the entry flight path and use this information to construct successive density prediction models throughout the entry. Because the estimator makes multiple predictions during the entry and because each of these predictions originates at a different point in the trajectory, plotting density error over time for each prediction quickly becomes cumbersome. Therefore, it is useful to develop a plot convention which permits multiple predictions from various points throughout the trajectory to be visually represented on a single plot. The convention introduced here accomplishes this goal and permits the results of hundreds of estimator predictions to be presented on a single plot. Consider a single entry trajectory wherein an entry vehicle must make successive density predictions throughout the entry. The entry begins at some initial time t_0 and ends at some final time t_f. From t_0 until some arbitrary time after t_0, the density estimator observes the atmosphere. At this arbitrary time t_1, the atmospheric data gathered from t_0 to t_1 is used to create a density prediction model (Figure B1). The performance of this model is assessed by computing the maximum and average (RMS) percent density errors from t_1 to t_f. In order to stressfully test the prediction efficacy of a given density estimator, it is assumed that no further density measurements are gathered after time t_1. Note that the density error is defined as the difference between the density prediction model and the true atmospheric density. In addition, the absolute value of the maximum error is computed. These two numbers, the RMSE and the maximum error, compactly summarize the critical performance information from this single prediction and are useful for judging prediction estimator prediction efficacy.

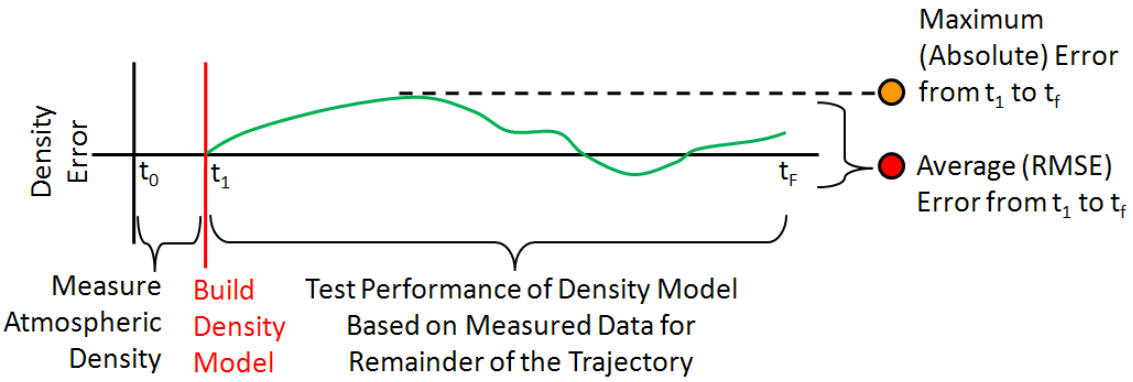


Figure B.1: Performance of a Single Density Model At Time t_1

If a different arbitrary model generation time was chosen, the results would have likely been different because the density estimator would have a different amount of information to construct the prediction. Assume that another prediction was made at a later time $t_2 > t_1$ (Figure B.2).

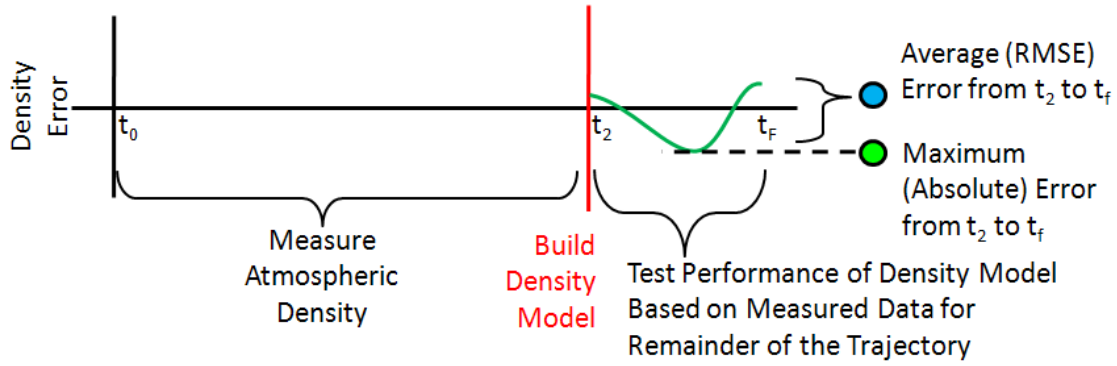


Figure B.2: Performance of a Single Density Model At Time t_2

Again the maximum and average (RMS) percent errors are computed and stored at this new arbitrary time t_2. If the information from these two prediction models is plotted, useful information illustrating the behavior of the estimator at two distinct points during the entry can be summarized on a single plot (Figure B.3).

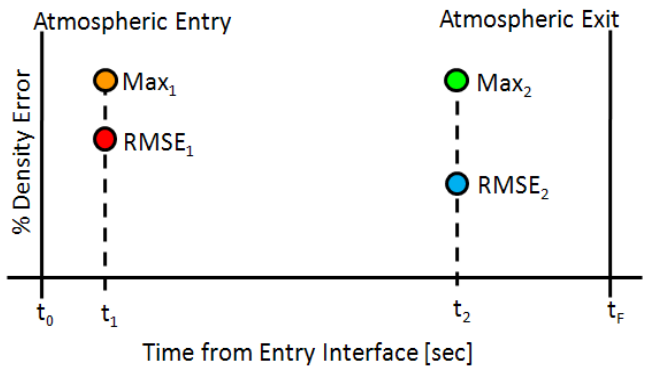


Figure B.3: Single Plot Illustrating the Maximum Error

Now allow this process to be repeated many times for different arbitrary points throughout the entry (Figure B.4). The arbitrary points are always selected in the direction of increasing time starting at t_0 and ending at time t_f. This approximates the conditions that a density estimator would face in flight. Note that the maximum and RMSE trend lines are influenced by differences in the atmospheric physics at varying altitudes. Often, the predictions made near the end of the trajectory have higher error than those made near periapsis despite the fact that more atmospheric information is available as the vehicle approaches t_f. This error at high altitude is due to (a) the greater level of density uncertainty at higher altitudes (which lead to larger magnitude density fluctuations) and (b) differences between the on-board models (i.e. the ensemble) and the true atmosphere which are often more significant at high altitudes.

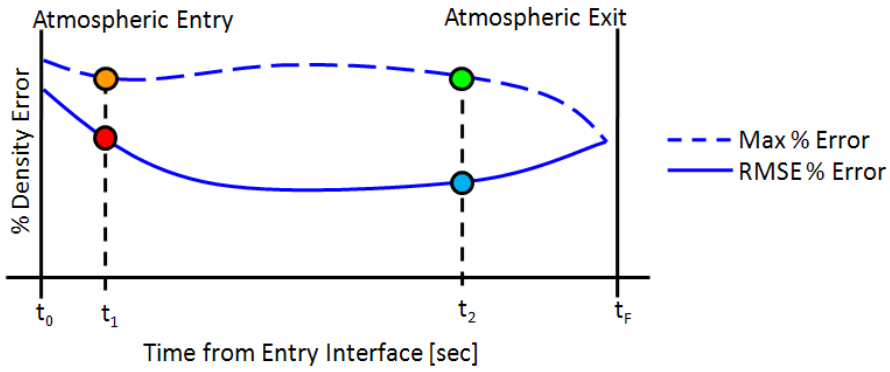


Figure B.4: RMSE and Maximum Prediction Error History Throughout the Entry

Prior to the integration of the various density estimators examined herein with a guidance system, the above approach offered a compact and simple method for assessing the performance of each estimator. This approach separates the main functions of the estimator (observe the atmosphere, build a prediction model, apply the prediction model to test performance) which is convenient for development and analysis purposes. Note that, once integrated with a guidance system, the density estimator must perform all three functions simultaneously. For guided simulations, the ΔV-λ plots developed in Chapter 5 are more appropriate as they illustrate the effect of density estimator on the system-level performance.

REFERENCES

[1] Knocke, P.C., Wawrzyniak, G. G., Kennedy, B. M., Desai, P. N., Parker, T. J., Golombek, M. P., Duxbury, T. C., Kass, D. M., "Mars Exploration Rovers Landing Dispersion Analysis." AIAA-2004-5093. August, 2004.

[2] Fritts, D. C., L. Wang, and R. H. Tolson (2006), "Mean and Gravity Wave Structures and Variability in the Mars Upper Atmosphere Inferred from Mars Global Surveyor and Mars Odyssey Aerobraking Densities," *Journal of Geophysical Research*, 111, A12304, doi:10.1029/2006JA011897.

[3] Fritts, D., Gordley, L., Dissly, R., and Johnson, B., "Atmospheric Dynamics of Mars: What Can Be Learned With High Resolution Limb and Sub-Limb IR Measurements?" *LPI Contributions*, 1353:3330–+, July, 2007.

[4] Wang, L., D. C. Fritts, and R. H. Tolson, (2006) "Nonmigrating tides inferred from the Mars Odyssey and Mars Global Surveyor aerobraking data," *Geophysical Research Letters* , 33, L23201, doi:10.1029/2006GL027753.

[5] Engelund, W. C., Powell, R. W., and Tolson, R. H., "Atmospheric Modeling Challenges and Measurement Requirements for Mars Entry, Descent and Landing." *LPI Contributions*, 1447:9025–+, November 2008.

[6] Masciarelli, J. P., Rousseau, S., Fraysse, H., E. Perot, "An Analytic Aerocapture Guidance Algorithm for the Mars Sample Return Orbiter," AIAA-2000-4116.

[7] Knocke, P.C. and Desai, P. N., "Mars Exploration Rovers Entry, Descent, and Landing Trajectory Analysis." AIAA-2004-5092. August, 2004.

[8] Golombek, M. P., et al., "Selection of the Mars Exploration Rover Landing Sites," *Journal of Geophysical Research*, 108(E12), 8072, doi:10.1029/2003JE002074, 2003.

[9] Kass, D. M., J. T. Schofield, T. I. Michaels, S. C. R. Rafkin, M. I. Richardson, and A. D. Toigo, "Analysis of Atmospheric Mesoscale Models for Entry, Descent, and Landing," *Journal of Geophysical Research*, 108(E12), 8090, doi:10.1029/2003JE002065, 2003.

[10] Cerimele, C. J. and Gamble, J. D., "A Simplified Guidance Algorithm for Lifting Aeroassist Orbital Transfer Vehicles," AIAA-1985-348. January, 1985.

[11] Fuhry, D.P., "A Design Study of Onboard Navigation and Guidance During Aerocapture at Mars," M.S. Dissertation, Aeronautics and Astronautics Dept., Massachusetts Institute of Technology, Boston, MA, 1988.

[12] Vijayaraghavan, A., "Aerocapture and Aeromaneuvering at Mars," AIAA-1992-4586. August 1992.

[13] Bryant, L.E., Tigges, M.A., and Ives, D.G., "Analytic Drag Control for Precision Landing and Aerocapture," AIAA-1998-4572. August, 1998.

[14] Carman, G.L., Ives, D.G., and Geller, D.K., "Apollo-Derived Mars Precision Lander Guidance," AIAA-1998-4570.

[15] Tigges, M., and Ling, L., "A Predictive Guidance Algorithm for Mars Entry," AIAA-1989-632.

[16] Tu, K.Y., Munir, M.S., Mease, K.D., and Bayard, D.S., "Drag-Based Predictive Tracking Guidance for Mars Precision Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 4, 2000, pp. 620–628. *–or-* AIAA-1998-4573.

[17] Mendeck, G.F. and Carman, G.L., "Guidance Design for Mars Smart Landers Using The Entry Terminal Point Controller," AIAA-2002-4502. August, 2002.

[18] Ess, R.H., "Atmospheric Effects on Martian Aerocapture," AIAA-1990-2818.

[19] Skalecki, L. M., Cerimele, C. J., and Gamble, J. D., "Meteorological Accuracy Requirements for Aerobraking Orbital Transfer Vehicles," AIAA-1984-0030. January, 1984.

[20] Cianciolo, A.D., Way, D.W., and Powell, R.W., "Effects of Atmospheric Phenomena on Mars Science Laboratory Entry Performance," AIAA-2008-6427. August, 2008.

[21] Findlay, J. T., Kelly, G. M. Troutman, P. A. "Final Report–Shuttle Derived Atmospheric Density Model: Part 1." NASA-CR-171824. Dec. 1984.

[22] Findlay, J. T., Kelly, G. M. Troutman, P. A. "Final Report–Shuttle Derived Atmospheric Density Model: Part 2." NASA-CR-171824. Dec. 1984.

[23] Anderson, J. L., "Outer Atmospheric Research," AIAA-1988-0686. January, 1988.

[24] Casoliva, J., Lyons, D.T., Wolf, A.A., and Mease, K.D., "Robust Guidance via a Predictor-Corrector Algorithm with Drag Tracking for Aero-Gravity Assist Maneuvers," AIAA-2008-6818. August, 2008.

[25] Perot, E., and Rousseau, S., "Importance of an On-Board Estimation of the Density Scale Height for Various Aerocapture Guidance Algorithms," AIAA-2002-4734. August, 2002.

[26] Harpold, J.C. and Graves C.A., “Shuttle Entry Guidance.” NASA-TM-79949 (aka JSC-14694). February, 1979.

[27] Bogner, I., “Description of Apollo Entry Guidance.” NASA-TM-66-2012-2. August, 1966.

[28] Lickly, D.J., Morth, H.R., and Crawford B.S., “Apollo Reentry Gudiance.” NASA-CR-52776. July, 1963.

[29] Pruett, C.D., Wolf, H., Heck, M. L., and Siemers, P.M. III, “Innovative Air Data System for the Space Shuttle Orbiter,” *Journal of Spacecraft and Rockets*, Vol. 20, No. 1, 1983, pp. 61–69.

[30] Gazarik, M., Wright, M., Little, A., Cheatwood, F., Herath, J., Munk, M., Novak, J., and Martinez, E., “Overview of the MEDLI Project,” IEEE Aerospace Conference, Big Sky, MT, IEEE paper No. 1510, 2008.

[31] Webb, L.D., “Characteristics and Use of X-15 Air-Data Sensors.” NASA TN D-4597. June, 1968.

[32] Wolowicz, C.H. and Gossett, T.D., “Operational and Performance Characteristics of the X-15 Spherical, Hypersonic Flow-Direction Sensor.” NASA TN D-3070. November, 1965.

[33] Cary, J.P. and Keener E.R., “Flight Evaluation of the X-15 Ball-Nose Flow-Direction Sensor as an Air-Data System.” NASA TN D-2923. July, 1965.

[34] Dutta, S. and Braun, R.D.; "Mars Entry, Descent, and Landing Trajectory and Atmosphere Reconstruction," AIAA 2010-1210, 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, FL, January 2010.

[35] Christian, J.A.; Verges, A.M.; and Braun, R.D.; "Statistical Reconstruction of Mars Entry, Descent, and Landing Trajectories and Atmospheric Profiles," AIAA 2007-6192, AIAA Space 2007 Conference and Exposition, Long Beach, CA, September 2007.

[36] Tolson, R. H., Keating, G. M., Cancro, G. J., Parker, J. S., Noll, S. N., Wilkerson, B. L., “Application of Accelerometer Data to Mars Global Surveyor Aerobraking Operations.” *Journal of Spacecraft and Rockets*. Vol. 36, No. 3, May–June 1999. pp. 323-329.

[37] Tolson, R., Bemis, E., Hough, S., Zaleski, K., Keating, G., Shidner, J., Brown, S., Brickler, A., Scher, M., Thomas, P., “Atmospheric Modeling Using Accelerometer Data During Mars Reconnaissance Orbiter Aerobraking Operations.” *Journal of Spacecraft and Rockets*. Vol. 45, No. 3, May–June 2008.

[38] Tolson, R. H., Keating, G. M., Zurek, R. W., Bougher, S. W., Justus, C. G., Fritts, D. C., "Application of Accelerometer Data to Atmospheric Modeling During Mars Aerobraking Operations." *Journal of Spacecraft and Rockets*. Vol. 44, No. 6, November–December 2007, pp. 1172–1179.

[39] Tolson, R. H., Dwyer, A. M., Hanna, J. L., Keating, G. M., George, B. E., Escalera, P. E., Werner, M. R., "Application of Accelerometer Data to Mars Odyssey Aerobraking and Atmospheric Modeling." *Journal of Spacecraft and Rockets*. Vol. 42, No. 3, May–June 2005.

[40] Forget, F., Millour, E., Gonzalez-Galindo, F., Lebonnois, S., Madeleine, J-B., Meslin, P-Y., Montabone, L., Spiga, A., Hourdin, F., Lefevre, F., Montmessin, F., Lewis, S.R., Read, P., Lopez-Valverde, M.A., andGilli, G. "Modeling the Martian Atmosphere with the LMD Global Climate Model." *LPI Contributions*, 1447:9054–+, November 2008.

[41] Millour, E., Forget, F., González-Galindo, F., Spiga, A., Lebonnois, S., Montabone, L., Lewis, S.R., Read, P.L., López-Valverde, M.A., Gilli, G., Lefèvre, F., Montmessin, F., Desjean, M.-C., Huot, J.-P., "The Latest (Version 4.3) Mars Climate Database," *LPI Contributions*, 1447:9029–+, November 2008.

[42] Lewis, S.R., Collins, M., Read, P.L., Forget, F. , Hourdin, F., Fournier, R., Hourdin, C., Talagrand, O., Huot, J.-P. "A Climate Database for Mars," *Journal of Geophysical Research – Planets.* Vol. 104, No. E10, 1999, pp. 24,177-24,194.

[43] Forget, F., Millour, E., and Lewis, S.R. *Mars Climate Database v4.3 User Manual*. Laboratoire de Météorologie Dynamique, Paris & The Open University, Milton Keynes. April, 2008.

[44] Justus, C.G., Johnson, D.L. *Mars Global Reference Atmospheric Model 2001: Users Guide*. NASA/TM-2001-210961.

[45] Leith, C.E., "Theoretical Skill of Monte Carlo Forecasts," *Monthly Weather Review*, Vol. 102, No. 6, June, 1974, pp. 409-418.

[46] Jin, L., Yao, C., and Huang, X.-Y., "A Nonlinear Artificial Intelligence Ensemble Prediction Model for Typhoon Intensity," *Monthly Weather Review*, Vol. 136, No. 12, December, 2008, pp. 4541-4554.

[47] Zhou, X., and Chen, J., "Ensemble Forecasting of Tropical Cyclone Motion Using a Baroclinic Model," *Advances in Atmospheric Sciences*. Vol. 23, No. 3, 2006, pp. 342-354.

[48] Vukicevic, T., Jankov, I., and McGinley, J., "Diagnosis and Optimization of Ensemble Forecasts." *Monthly Weather Review*, Vol. 136, No. 3, 2008, pp. 1054-1074.

[49] Seidman, A.N., "Averaging Techniques in Long-Range Weather Forecasting," *Monthly Weather Review*, Vol. 109, No. 7, July, 1981, pp. 1367-1379.

[50] Molteni, F., R. Buizza, T. N. Palmer, and T. Petroliagis, "The ECMWF Ensemble Prediction System: Methodology and Validation," *Quarterly Journal of the Royal Meteorological Society, Vol.* 122, Issue 529, January 1996, pp. 73–119.

[51] Hamill, T. M., and S. J. Colucci, "Verification of Eta–RSM Short-Range Ensemble Forecasts." *Monthly Weather Review, Vol.* 125, No. 6, 1997, pp. 1312–1327.

[52] Rogberg, P., Read, P. L., Lewis, S. R., and Montabone, L., "Assessing Atmospheric Predictability on Mars Using Numerical Weather Prediction and Data Assimilation," *LPI Contributions*, 1447:9062–+, November 2008.

[53] Simpson, T., Mistree, F., Korte, J., Mauery, T., "Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization," AIAA-1998-4755, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, September 1998.

[54] Simpson, T.W., Peplinski, J. D., Koch, P. N., and Allen, J. K., "On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments." *Proceedings of DETC'97 1997 ASME Design Engineering Technical Conferences*. September, 1997. Sacramento, California.

[55] Merwe, R. vd., Leen, T.K., Lu, Z., Frolov, S., Baptista, A.M., "Fast Neural Network Surrogates for Very High Dimensional Physics-Based Models in Computational Oceanography," *Neural Networks*, *Volume 20, No. 6*, *August 2007*, *pp. 462-478*.

[56] Loyola R., D.G., "Applications of Neural Network Methods to the Processing of Earth Observation Satellite Data," *Neural Networks*, Vol. 19, No. 6-7, July-August 2006, pp. 168-177.

[57] Maqsood, I., Khan, M.R., and Abraham, A., "An Ensemble of Neural Networks for Weather Forecasting," *Neural Computing & Applications*, Vol. 13, No.2, May, 2004, 112-122, DOI: 10.1007/s00521-004-0413-4.

[58] Krasnopolsky, V.M., Fox-Rabinovitz, M.S., and Belochitski, A.A., "Decadal Climate Simulations Using Accurate and Fast Neural Network Emulation of Full,

Longwave and Shortwave, Radiation," *Monthly Weather Review*. Vol. 136, No. 10, October, 2008, pp. 3683-3695.

[59] Han G., and Shi, Y., "Development of an Atlantic Canadian Coastal Water Level Neural Network Model," *Journal of Atmospheric and Oceanic Technology.* Vol. 25, No. 11, November, 2008, pp. 2117-2132.

[60] Hsieh, W. W., and Tang, B., "Applying Neural Network Models to Prediction and Data Analysis in Meteorology and Oceanography," *Bulletin of the American Meteorological Society*. Vol. 79, No. 9, May 1998, pp.1855-1870.

[61] Tang, Y., and Hsieh, W. W., "Coupling Neural Networks to Incomplete Dynamical Systems via Variational Data Assimilation," *Monthly Weather Review*. Vol. 129, No. 4, April 2001, pp. 818-834.

[62] Sharkey, A. J. C. "On combining artificial neural nets." *Connection Science: Special Issue on Combining Artificial Neural Networks.* 1996. pp. 299–314.

[63] Maclin, R., & Optiz, D. "An empirical evaluation of bagging and boosting." *Proceedings of the 14th national conference on artificial intelligence*, Providence, RI, 1997. pp. 546–551.

[64] Strausberg, M. J., H. Wang, M. I. Richardson, S. P. Ewald, and A. D. Toigo (2005), "Observations of the Initiation and Evolution of the 2001 Mars Global Dust Storm," *Journal of Geophysical Research*, 110, E02006, doi:10.1029/2004JE002361.

[65] Lewis, S.R., Read, P.L., and Collins, M., "Martian atmospheric data assimilation with a simplified general circulation model : orbiter and lander networks." *Planetary & Space Science*. Vol. 44, No.11, 1996, pp 1395 – 1409.

[66] Lewis, S.R. and Barker, P.R. "Atmospheric tides in a Mars general circulation model with data assimilation." *Advances in Space Research*. Vol. 36, No.11, 2005, pp 2162 – 2168.

[67] Hoffman, M.J., Greybush, S.J., Wilson, R.J., Gyarmati, G. , Hoffman, R.N., Kalnay, E., Ide, K., Kostelich, E.J., Miyoshi, T., Szunyogh, I. "An ensemble Kalman filter data assimilation system for the martian atmosphere: Implementation and simulation experiments." *Icarus*. Vol.209, No.02, Oct. 2010, pp 470-481.

[68] Crassidis, J., and Junkins, J., Optimal Estimation of Dynamic Systems. Chapman & Hall/CRC, 2004, Boca Raton, FL.

[69] Johnson, E. and Oh, S. "Adaptive Control Using Combined Online and Background Learning Neural Network," 43rd IEEE Conference on Decision & Control. December, 2004.

[70] Chowdhary, G. and Johnson, E. "Adaptive Neural Network Flight Control Using both Current and Recorded Data," AIAA Paper 2007-6505, August 2007.

[71] Chowdhary, G. and Johnson, E. "Theory and Flight Test Validation of Long Term Learning Adaptive Flight Controller," AIAA Paper 2008-6781., August 2008.

[72] Widrow, B., Lehr, M.A., Beaufays, F., Wan E., and Bilello, M., ``Adaptive Signal Processing," *Proceedings of the World Conference on Neural Networks*, IV-548, Portland, July 1993.

[73] Widrow, B., and Hoff, M.E., Jr., ``Adaptive Switching Circuits," *IRE WESCON Convention Record*, 4:96-104, August 1960.

[74] Striepe, S.A., Powell, R.W., Desai, P.N., Queen E.M., Program to Optimize Simulated Trajectories (POST II) Volume II: Utilization Manual. Version 1.1.6.G, January 2004.

[75] Wagner, J.J., Wilhite, A.W., Stanley, D.O., and Powell, R.W., "An Adaptive Real Time Atmospheric Prediction Algorithm for Entry Vehicles," AIAA-2011-3200, 3rd AIAA Atmospheric & Space Environments Conference, Honolulu, HI, June 2011.

[76] Farrar, D.E. and Glauber R.R. "Multicollinearity in Regression Analysis: The Problem Revisited," *The Review of Economics and Statistics*. Vol. 49, No.1, February, 1967, pp. 92-107.

[77] De Jong, S., "SIMPLS: An Alternative Approach to Partial Least Squares Regression," *Chemometrics and Intelligent Laboratory Systems*. Vol. 18, No. 3, March 1993, pp.251-263.

[78] Geladi, P., and Kowalski, B.R., "Partial Least Squares Regression; A Tutorial," *Analytica Chimica Acta*. Vol. 185, No. 1, January 1986, pp. 1-17.

[79] Rojas, R. *Neural Networks: A Systematic Introduction*. Springer-Verlag, Berlin, 1996.

[80] Zhou, Z. H., Wu, J., and Tang, W., "Ensembling Neural Networks: Many Could Be Better Than All." *Artificial Intelligence.* Vol. 137, No. 1-2, May, 2002, pp. 239–263.

[81] Chowdhary, G. and Johnson, E. "Flight Test Validation of a Neural Network based Long Term Learning Adaptive Flight Controller," AIAA Paper 2009-5854., August 2009.

[82] Paulsen, O., and Sejnowski, T.J., "Natural Patterns of Activity and Long-Term Synaptic Plasticity," *Current Opinion in Neurobiology*. Vol. 10, No. 2, April 2000, pp. 172-180.

[83] Turrigiano, G.G., "Homeostatic Plasticity in Neuronal Networks: The More Things Change, the More They Stay the Same." *Trends in Neurosciences*. Vol. 22, No. 5, May, 1999, pp. 221-227.

[84] Kempter, R., Gerstner, W., and Van Hemmen, J.L., "Hebbian Learning and Spiking Neurons," *Physical Review E*. Vol. 59, No. 4, April 1999, pp. 4498-4514.

[85] Hebb, D.O. "Distinctive features of learning in the higher animal". In Brain Mechanisms and Learning, J. F. Delafresnaye (Ed.). Oxford University Press, London. 1961.

[86] Millour, E., Forget, F., and Lewis, S.R., *Mars Climate Database v4.3 Detailed Design Document*., May 2008, ESTEC Contract 11369/95/NL/JG.

[87] Dwyer Cianciolo A.M., et al. "Entry, Descent, and Landing Systems Analysis Study: Phase 2 Report on Exploration Feed-Forward Systems," NASA/TM-2011-217055. February, 2011.

[88] Kluever, C. A., "Entry Guidance Performance for Mars Precision Landing," *Journal of Spacecraft and Rockets,* Vol. 31, No 6, 2008, pp. 1537-1544.

[89] Powell, R. W., "Numerical Roll Reversal Predictor Corrector Aerocapture and Precision Landing Guidance Algorithms for the Mars Surveyor Program 2001 Missions," AIAA-1998-4574.

[90] Xue, S., and Lu, P., "Constrained Predictor-Corrector Entry Guidance," AIAA-2009-5767. AIAA Guidance, Navigation, and Control Conference, Chicago, IL, August, 2009.

[91] Braun, R. D., and Powell, R.W., "Predictor-Corrector Guidance Algorithm for Use in High-Energy Aerobraking System Studies," *Journal of Guidance, Control, and Dynamics,* Vol. 15, No. 3, May-June 1992, pp. 672 – 678.

[92] Powell, R.W., and Braun, R. D., "Six-Degree-of-Freedom Guidance and Control Analysis of Mars Aerocapture," *Journal of Guidance, Control, and Dynamics,* Vol. 16, No. 6, November-December 1993, pp. 1038 – 1044.

[93] Putnam, Z.R., Neave, M.D., and Barton, G.H., "PredGuid Entry Guidance for

Orion Return from Low Earth Orbit," IEEEAC #1571, January, 2009.

[94] Bairstow, S.H., and Barton, G.H., "Orion Reentry Guidance with Extended Range Capability Using PredGuid," AIAA-2007-6427. AIAA Guidance, Navigation, and Control Conference, Hilton Head, South Carolina, August, 2007.

[95] Gamble, J. D., Cerimele, C. J., Moore, T. E., and Higgins, J., "Atmospheric Guidance Concepts for an Aeroassist Flight Experiment," *Journal of Astronautical Sciences*, Vol. 36, No. 1/2, Jan.- June 1988, pp. 45-71.

[96] Striepe, S.A., et al. "Program to Optimize Simulated Trajectories (POST II): Volume II, Utilization Manual," Version 1.1.6.G, January 2004.

[97] Davis, J.L., Dwyer Cianciolo A.M., Powell, R.W., Shidner, J.D., and Garcia-Llama, E., "Guidance and Control Algorithms for the Mats Entry, Descent, and Landing Systems Analysis," AIAA-2010-7972. AIAA/AAS Astrodynamics Specialist Conference, Toronto, Ontario, Canada, August, 2010.

[98] Dwyer Cianciolo A.M., et al. "Entry, Descent, and Landing Systems Analysis Study: Phase 1 Report," NASA/TM-2010-216720. July, 2010.

[99] Zumwalt, C. H., Sostaric, R.R., Westhelle, C.H., and Dwyer Cianciolo A.M., "Aerocapture Guidance and Performance at Mars for High-Mass Systems," AIAA-2010-7971. AIAA/AAS Astrodynamics Specialist Conference, Toronto, Ontario, Canada, August, 2010.

[100] Sostaric, R. R., Zumwalt, C., Garcia-Llama, E., Powell, R., Shidner, J., "Trajectory Guidance for Mars Robotic Precursors: Aerocapture, Entry, Descent, and Landing," JSC-CN-23876. Presented at The International Planetary Probe Workshop (IPPW-8), June, 2011, Portsmouth, VA.

[101] Dwyer-Cianciolo, A. , Zang, T. A., Sostaric, R. R., Mcguire, M. K., "Overview of the NASA Entry, Descent, and Landing Systems Analysis Exploration Feed-Forward Study." Presented at The International Planetary Probe Workshop (IPPW-8), June, 2011, Portsmouth, VA.

[102] Dwyer-Cianciolo, A., "Entry Descent, and Landing Systems Analysis: Exploration Feed Forward Internal Peer Review Slide Package," NASA/TM-2011-217050. February, 2011.

[103] Stewart, J., *Calculus: Early Transcendentals*, Brooks/Cole: Pacific Grove, CA, 2001.

[104] Rousseau, S., Etienne, P., Graces, C., Masciarelli, J.P., and Queen, E., "Aerocapture Guidance Algorithm Comparison Campaign," AIAA-2002-4822. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Monterey, CA, August, 2002.

[105] Landis, T.R. and Jenkins, D.R., "*Hypersonic: The Story of the North American X-15*," Specialty Press, 2008, North Branch, MN.

[106] Wolf, H., Henry, M.W., and Siemers, P.M. III, "Shuttle Entry Air Data System (SEADS): Optimization of Preflight Algorithms Based on Flight Results," AIAA-1988-2053. May, 1988.

[107] Henry, M.W., Wolf, H., and Siemers, P.M. III, "An Evaluation of Shuttle Entry Air Data System (SEADS) Flight Pressures: Comparisons with Wind Tunnel and Theoretical Predictions," AIAA-1988-2052. May, 1988.

[108] Siemers, P.M. III, Wolf, H., and Flanagan, P.F. "Shuttle Entry Air Data System Concepts Applied to Space Shuttle Orbiter Flight Pressure Data to Determine Air Data – STS 1-4" AIAA-1983-0118. January, 1983.

[109] Karlgaard, C.D. , Becky, R.E., O'Keefe, S.A, Siemers, P.M., III, White, B.A. , Engelund, W.C., and Munk, M.M., "Mars Entry Atmospheric Data System Modeling and Algorithm Development." AIAA-2009-3916. June, 2009.

[110] Compton, H.R., Blanchard, R.C., and Findlay, J.T., "Shuttle Entry Trajectory Reconstruction Using Inflight Accelerometer and Gyro Measurements," AIAA-1979-0257. 17th Aerospace Sciences Meeting, New Orleans, LA, January, 1979.

[111] Blanchard, R.C., and Rutherford, J.F., "The Shuttle Orbiter High Resolution Accelerometer Package Experiment: Preliminary Flight Results," AIAA-1984-0490. AIAA 22nd Aerospace Sciences Meeting, Reno, NV, January, 1984.

[112] Blanchard, R.C., Hinson, E.W., and Nicholson, J.Y., "Shuttle High Resolution Accelerometer Package Experiment Results: Atmospheric Density Measurements Between 60 and 160 km," *Journal of Spacecraft and Rockets*, Vol. 26, No. 3, 1989, pp. 173-180.

[113] Braun, R.D., and Manning, R. M., "Mars Exploration Entry, Descent, and Landing Challenges," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, 2007, pp. 310-323.

[114] Lafleur, J. M. and Cerimele, C.L., "Mars Entry Bank Profile Design for Terminal State Optimization," *Journal of Spacecraft and Rockets*, Vol. 48, No. 6, 2011, pp. 1012-1024.

[115] D'Amario, L.A., "Mars Exploration Rovers Navigation Results," AIAA-2004-4980. August, 2004.

[116] Thomson, W.T., Introduction to Space Dynamics, Dover Publications, Inc., 1986, Mineola, N.Y.

[117] Moreno, R., Lellouch, E., Encrenaz, T., Forget, F., Chassefiere, E., Hourdin, F., Guilloteau, S., "Wind Measurements in Mars' Middle Atmosphere at Equinox and Soltice: Iram Pleteau De Bure Interferometric CO Observations," 2nd Workshop on Mars Atmosphere Modeling and Observations, March 2006, Granada, Spain.

[118] Moreno, R., Lellouch, E., Forget, F., Encrenaz, T., Guilloteau, S., Millour, E., "Wind Measurements in Mars' Middle Atmosphere: IRAM Plateau de Bure Interferometric CO Observations," *Icarus,* Vol. 201, No. 2, 2009, doi: 10.1016/j.icarus.2009.01.027.

[119] Crowley, G., and Tolson, R.H., "Mars Thermospheric Winds from Mars Global Surveyor and Mars Odyssey Accelerometers," Journal of Spacecraft and Rockets, Vol. 44, No. 6, 2007, pp. 1188-1194.

[120] McKenzie, R. L., "Method of Atmospheric Density Measurements During Shuttle Entry Using Ultraviolet-Laser Rayleigh Scattering," *Journal of Spacecraft and Rockets*, Vol. 26, No. 1, 1988, pp. 56-64.

[121] Fritts, D. C., Wang, L., and Tolson, R. H., "Mean and Gravity Wave Structures and Variability in the Mars Upper Atmosphere Inferred From Mars Global Surveyor and Mars Odyssey Aerobraking Densities," *Journal of Geophysical Research*, Vol. 111, A12304, 2006, doi:10.1029/2006JA011897.

[122] Creasey, J. E., Forbes, J. M., and Hinson, D. P., "Global and Seasonal Distribution of Gravity Wave Activity in Mars' Lower Atmosphere Derived From MGS Radio Occultation Data", *Geophysical Research Letters*, Vol. 33, L01803, 2006, doi:10.1029/2005GL024037.

[123] Justus, C. G. and James, B.F., "Mars Global Reference Atmospheric Model 2000 Version (Mars-GRAM 2000) Users Guide," NASA/TM-2000-210279. May, 2000.

[124] FAA Testing New Microburst Detection System in Las Vegas. Aviation Week. Online: http://www.aviationweek.com/aw/generic/story_generic.jsp?channel=busav&id=news/RADA12187.xml&headline=FAA%20Testing%20New%20Microburst%20Detection%20System%20In%20Las%20Vegas

[125] Amzajerdian, F., Kavaya, M., Singh, U., and Yu, J. "2-micron Coherent Doppler Lidar for Space-Based Global Wind Field Mapping," Geoscience and Remote Sensing Symposium, Proceedings of IGARSS 2003. Vol. 1, pp. 515-517.

[126] Amzajerdian, F. "The Expanding Role of Lidar at NASA." *Photonics Spectra*, July 2008, pg 50 – 55.

[127] Singh, U. N., Emmitt, G. D., Levine, J. S., Engelund, W.C., Yu, J., Koch, G. J., and Kavaya, M. J., "A Mars-Orbiting 2-Micron Lidar System to Monitor the Density, Winds, and Dust of the Atmosphere of Mars." Presented at the 4th International Workshop on the Mars Atmosphere: Modeling & Observation. Paris, Fr., Feb., 2011.

[128] Percy, T.K., Bright, E., and Torres, A.O., "Assessing the Relative Risk of Aerocapture Using Probabilistic Risk Assessment," AIAA-2005-7107. JPC 2005 Conference, July 2005, Tuscon, AZ.

[129] Braun, R.D., "NASA Innovation and Technology Preliminary Planning," Meeting Notes from March 9th, 2010. Office of the Chief Technologist, National Aeronautics and Space Administration, Washington, D.C.

[130] Singh, U.N., Heaps, W.S., and Komar, G.J., "Laser/Lidar Technologies for NASA's Science and Exploration Mission's Applications," AIAA-2005-6773. Space 2005, September 2005, Long Beach, CA.

[131] Singh, U. N., Emmitt, G. D., Levine, J. S., Engelund, W.C., Yu, J., Koch, G. J., and Kavaya, M. J., "A Mars-Orbiting 2-Micron Lidar System to Monitor the Density, Winds, and Dust of the Atmosphere of Mars," The Fourth International Workshop on the Mars Atmosphere: Modelling and Observation. February, 2011, Paris, France. Published online at http://www-mars.lmd.jussieu.fr/ paris2011/program.html, pp.56-59.