A CONTROLLER DEVELOPMENT METHODOLOGY INCORPORATING UNSTEADY, COUPLED AERODYNAMICS AND FLIGHT CONTROL MODELING FOR ATMOSPHERIC ENTRY VEHICLES

A Dissertation
Presented to
The Academic Faculty

By

Zachary J. Ernst

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Daniel Guggenheim School of Aerospace Engineering
Aerospace Systems Design Laboratory

Georgia Institute of Technology

December 2022

A CONTROLLER DEVELOPMENT METHODOLOGY INCORPORATING UNSTEADY, COUPLED AERODYNAMICS AND FLIGHT CONTROL MODELING FOR ATMOSPHERIC ENTRY VEHICLES

Thesis committee:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Mark Costello
School of Aerospace Engineering
Georgia Institute of Technology

Professor Lakshmi Sankar
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Bradford Robertson
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Ashley Korzun
Atmospheric Flight and Entry Systems Branch
NASA Langley Research Center

Date approved: October 25th, 2022

To William Ernst and Monique Petersen

ACKNOWLEDGMENTS

This dissertation marks the culmination of my journey — sometimes taxing, but always fulfilling — as a graduate student at Georgia Tech. Working toward this Ph.D. has shown me the high-water mark of my motivation and discipline, and let me prove to myself that I can rigorously put into practice the scientific method and contribute to the advancement of space exploration. But none of this would have been possible without the support and guidance of a great many people. I would like to acknowledge and thank all of them here.

First and foremost I would like to give my sincere thanks to my advisor and teacher, Professor Dimitri Mavris. Dr. Mavris gave me opportunity to join the Aerospace Systems Design Laboratory as a graduate researcher and take what had been an interest and a hobby and turn it into the beginnings of a career in aerospace engineering. He taught me not only how to understand and analyze problems I've never seen before, but also how to tell the story and convince the experts that I know enough to be dangerous. I will always be grateful to Doc for the time, knowledge, advice, and support that he gave me throughout my time at ASDL, and ultimately for the opportunity he gave me to begin a career at NASA.

I would also like to express my gratitude towards my Ph.D. committee members. I would like to thank Professor Mark Costello for taking the time out of his schedule to serve on my committee, providing expertise and insight in the controls discipline. I would like to thank Professor Lakshmi Sankar for his advice on and scrutiny into my Computational Fluid Dynamics work. I would like to express my great appreciation to Dr. Ashley Korzun for her support and key insights into the NASA Mars entry programs, and helping me make this research useful for future work. I would like to give special thanks to Dr. Bradford Robertson for the considerable time he spent working with me throughout the Ph.D. process. Dr. Robertson has given me the chance to work on some of the most amazing space-related projects at ASDL, and I give him a large portion of the credit for convincing me to stay and complete my Ph.D. His mentorship was instrumental in formulating and scoping this work into a rigorous, novel scientific contribution.

Throughout the Ph.D. process and my graduate education as a whole, I have had the opportunity to work with subject matter experts at NASA on technical aspects of many different disciplines. I would like to thank Paul Tartabini, Anthony Williams, Robert Biedron, Kevin Jacobson, Soumyo Dutta, and Li Wang from NASA Langley Research Center for their support. Without their help I cannot fathom how my code would work at all, much less become a premiere tool used for cutting-edge research beyond this Ph.D. Additionally, I would like to thank Joe Brock, Eric Stern, and Michael Wilder from NASA Ames Research Center for generously sharing their computational mesh and results.

It is also true that my work would not have been as successful or enjoyable without the support of my colleagues and friends from ASDL and Georgia Tech. I would like to thank Brett Hiller, Chelsea Johnson, Alexandra Hickey, Madelyn Drosendahl, and Hayden Dean for the pleasure of working together on the POST2-FUN3D research. I would like to thank Ruxandra Duca for the long hours we spent preparing for the qualifying exams. I would also like to acknowledge the inspiration that Etienne Demers-Bouchard, Stephanie Zhu, Nikki Robertson, Ken Decker, and Manuel Diaz gave me as I followed their work in other exciting areas of research. I am greatly indebted to Tanya and Adrienne for their vital support from the ASDL front office. I would also like to thank Teri Polly, A.J. Schultheis, Kunal Gangolli, Athreya Gundamraj, Mihir Kurande, and everyone else I worked with in the Yellow Jacket Space Program.

I have also greatly relied on my friendships outside of Georgia Tech. I would like to thank Chris, Elijah, William, Lucy, and others who formed a literally worldwide support network that helped keep me sane.

Finally, I would like to thank my parents Daniel Ernst and Angela Petersen-Ernst, my brother Maxwell, and my extended family. They have cheered me on as I grew up, as I moved away for university, and as I have pursued a career doing what I love. I am proud to have such loving, supportive, and encouraging people in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

a_{ref}	reference speed of sound
C_A, C_Y, C_N	axial, side, and normal force coefficients
C_{LL}, C_M, C_{LN}	roll, pitch, and yaw moment coefficients
d	diameter
$\bar{F}$	force vector
$\boldsymbol{I}$	inertia tensor
I_{xx}, I_{yy}, I_{zz}	roll, pitch, and yaw inertia components
L_{grid}	reference length, grid units
L_{ref}	reference length, m
$\bar{M}$	moment vector
M	Mach number
m	mass
P_t	total pressure
P_{ref}	reference static pressure
PR	jet total pressure ratio
Q	freestream dynamic pressure
Q_{ref}	reference dynamic pressure

$\boldsymbol{R}_{ij}$	j to i frame rotation matrix
s	freestream speed
T_t	total temperature
T_{ref}	reference static temperature
t	time
Δt	trajectory propagator physical time step
Δt_{fs}	flow solver physical time step
TR	jet total temperature ratio
$\bar{v}$	velocity vector
$\bar{x}$	vehicle state vector
$\bar{r}$	position vector
$\Delta\bar{r}$	vehicle C.G. translation vector
$\Delta\bar{r}_g$	grid translation vector
α	angle of attack
α_T	total angle of attack
β	sideslip angle
γ	flight path angle
λ, Φ	longitude and geodetic latitude

ϕ, θ, ψ	relative roll, pitch, and yaw Euler angle
ϕ_i, θ_i, ψ_i	inertial roll, pitch, and yaw Euler angle
Ψ	azimuth angle
$\bar{\omega}$	angular velocity vector
$\bar{\omega}_{rel}$	vertical-relative angular velocity vector
ω_{earth}	planetary rotation rate

Subscripts:

B	Body frame
cmd	commanded
E	Earth-centered rotating frame
F	Body reference frame
I	Inertial frame
O	Freestream frame
P	Intermediary frame
$plenum$	at the nozzle plenum
ref	reference parameter
V	Local vertical frame
x, y, z	x-, y-, and z-axis components

SUMMARY

Future missions to the surface of Mars will present new requirements for entry, descent, and landing (EDL) based on the desire to achieve scientific goals and mitigate risk for human explorers. These missions will have payload masses and landing elevations beyond the capability of U.S. mission design heritage, driving the development of new decelerator technologies. Future missions will also require precision landing with an uncertainty less than 100 meters — two orders of magnitude smaller than current U.S. missions.

Landing uncertainty is the result of multiple sources that propagate across the duration of EDL, but aerodynamic and control system uncertainty during entry are two of the largest contributors. Entry vehicle aerodynamics in the hypersonic and supersonic regime are complex and inherently unsteady. This is exacerbated by active control systems which interact with the flow field or the flight dynamics. The interaction can result in diminished attitude control authority or even control reversal. Operation of the entry controller causes a short-term increase in uncertainty even as it works to eliminate error by the end of the entry phase. This coupled, unsteady behavior has affected previous Mars entry vehicles and continues to be a significant design consideration, even with extensive heritage.

The state-of-the-art design process accounts for this behavior indirectly. The aerodynamic and control system models are developed as separate quasi-static databases, ignoring all dynamic effects other than pitch and roll damping. The result is increased model fidelity uncertainty, at the same time as design changes are introducing more volitional and phenomenological uncertainty. By reducing the model fidelity uncertainty, designers can improve the entry controller performance as they evaluate new architectures, thereby reducing the landing uncertainty. The objective of this dissertation is to formulate and implement an entry controller tuning methodology that directly accounts for coupled, unsteady entry vehicle aerodynamic and control system behavior.

Significant progress in modeling entry vehicle aerodynamics and flight dynamics has been made over the past decade by coupling 6 degree-of-freedom rigid body dynamics simulations with computational fluid dynamics flow solvers. The equations of motion are solved simultaneously with the Navier-Stokes equations to produce a time-accurate flight simulation. This work seeks to incorporate guidance and control system modeling into the coupled simulation, allowing high-fidelity evaluation of an entry vehicle control system design and accompanying entry controller.

However, the added cost of coupled simulation makes the state-of-the-art controller tuning methodology impractical. Instead of finding a suitable controller using Monte Carlo simulation, the proposed methodology uses multifidelity optimization. The coupled simulation serves as the high-fidelity model, simulating an entry vehicle with open- or closed-loop control for direct evaluation of a controller. The trajectory and aerodynamic data produced by each run of the coupled simulation is used to train an aerodynamic surrogate for fast, low-fidelity modeling.

The methodology was developed and evaluated through numerical experimentation using the NASA Supersonic Decelerator Flight Test entry vehicle equipped with a one-axis internal moving mass actuator control system, simulating the tuning of an entry controller for said vehicle. Comparisons to simulations using a baseline flight model with aerodynamic database showed that the coupled model satisfactorily captures the unsteady, coupled aerodynamic and control mechanism behavior. Further experimentation investigated the implementation of key components for the multifidelity methodology: the low-fidelity aerodynamic surrogate and the optimization method. It was found that the methodology can use a feed-forward neural network as a low-fidelity aerodynamic surrogate and train it with the time-accurate data output from the coupled model. A comparison was made between two multifidelity optimization methods: a filtering-based method akin to the state of the art, relying on the quasi-hierarchical relationship between the coupled model and the low-fidelity aerodynamic surrogate, and a fusion-based method that directly compared the error between the controller performance predicted by the low-fidelity and coupled models. Based on the similar level of performance between these two methods, the methodology was implemented with a synthesis of the two, combining the stability of filtering-based methods with the exploration and correction capabilities of the fusion-based method. The completed multifidelity controller tuning methodology was successfully executed, yielding an entry controller that performed better than a controller tuned using the state-of-the-art methodology.

CHAPTER 1

INTRODUCTION AND MOTIVATION

1.1 Overview of Atmospheric Entry, Descent, and Landing

Entry, Descent, and Landing (EDL) describes the flight and activities of a vehicle from when it enters a planet's atmosphere from space until it reaches the surface. Griffin and French [1] state that "the atmospheric entry system must provide a controlled dissipation of the combined kinetic and potential energy associated with the vehicle's speed and altitude." The intent for a safe and survivable landing is also a critical part of this definition; they further note that "the entry system must provide suitable provisions for surface contact, usually with constraints on the landing location and vehicle attitude [1]." This distinguishes EDL from deorbit disposal, where a spacecraft enters an atmosphere with the express goal of destroying itself; or from the operation of a re-entry vehicle (RV), which is not intended to survive impact.

Spacecraft have performed atmospheric entry on five bodies in the Solar System [1]. Many Earth orbital missions have returned payload or crew to the surface, and interplanetary probes have used EDL to land on Venus, Mars, Titan, and Jupiter[1].

EDL is perhaps the most challenging and dangerous phases of spaceflight. Vehicles experience massive changes to their energy over a span of minutes. Multiple systems must operate and interact perfectly; no spacecraft has yet flown with the capability of aborting after beginning EDL. The consequences for failure during EDL are also high. To date, three crewed missions (Soyuz 1, Soyuz 11, and *Columbia* STS-107) have suffered fatalities as a result of EDL failures, whereas one mission (*Challenger* STS-51-L) has suffered fatalities during launch [1, 2]. Other crewed missions have suffered "close calls" during EDL, such

[1] The Galileo probe underwent entry and descent at Jupiter, though this spacecraft did not land.

as Voskhod 2 [3], *Columbia* STS-1, and *Atlantis* STS-27 [2]. Multiple robotic missions have suffered failures during EDL, such as the Mars Polar Lander [4] and Genesis sample return capsule [5]. The challenges posed by EDL continue to be a factor in spacecraft design; EDL risk was specifically identified as a factor in the decision to fly Artemis 1, the first circumlunar flight of the Orion capsule, without crew [6].

1.1.1 Mars Mission Entry, Descent, and Landing

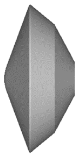
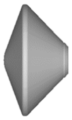
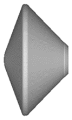
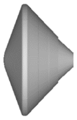
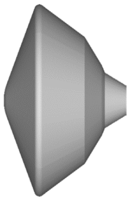
Atmospheric entry at Mars is by far the most challenging out of all the bodies in the Solar System. Mars's thin atmosphere, about 1.6% the density of Earth's atmosphere at the surface [7], barely provides enough drag for spacecraft to slow down from orbital velocities before hitting the surface. However, this is still thick enough to require a thermal protection system (TPS) and complicate the use of propulsion systems for landing and control.

To date, nine U.S. Mars missions have successfully landed on the surface [8]. Table 1.1 lists the key characteristics of their entry vehicles. These vehicles have had a shared and continuing design heritage, to the extent that multiple missions have even reused aeroshell and entry systems designs. Mars InSight and Mars 2020 reused the designs from Mars Phoenix Lander (MPL) and Mars Science Laboratory (MSL), respectively, and the Pathfinder (MPF) and Mars Exploration Rover (MER) capsules are nearly identical.

Entry Vehicle Configuration

The configuration of the MSL/Mars 2020 entry vehicle, shown in Figure 1.1, is representative of U.S. designs [10]. The entry vehicle is a rigid, blunt body with a 70° rounded conical forebody and an afterbody created from conic sections. The vehicle is constructed as an aeroshell consisting of a backshell and heatshield, which encapsulates the payload (in this case the descent stage and rover). The primary decelerator is a disk-gap-band parachute, mounted in the backshell. A cruise/orbiter stage provides power, communications, and propulsion during the flight to Mars.

Table 1.1: U.S. Mars mission entry vehicles (adapted from Dyakonov et al. [9])

	Viking I/II	Pathfinder	MER A/B	MPL/ InSight	MSL/ Mars 2020
					
Launch	1975	1996	2003	2007/2018	2011/2020
Diameter (m)	3.5	2.65	2.65	2.65	4.5
Entry Mass (kg)	930	585	840	602	2,919
Payload Mass (kg)	603	360	539	364	1025
Ballistic Coeff. (kg/m^2)	63.7	62.3	89.8	65.0	126.0
Entry Vel. (km/s)	4.5	7.6	5.5	5.9	5.4
Landing Alt. (km)	-3.5	-1.5	-1.3	-3.5	+1.0
Landing Ellipse (km)	280×100	200×70	150×20	100×20	20×7
Trajectory	Lifting	Ballistic	Ballistic	Ballistic	Lifting
Control	RCS	Spin	Spin	None*	RCS

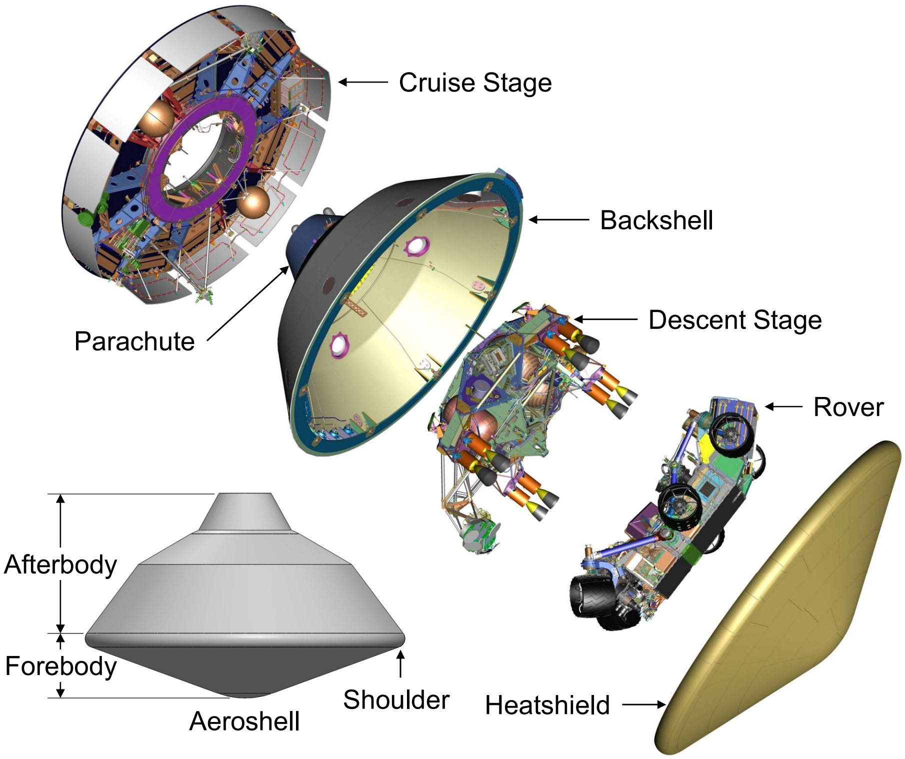
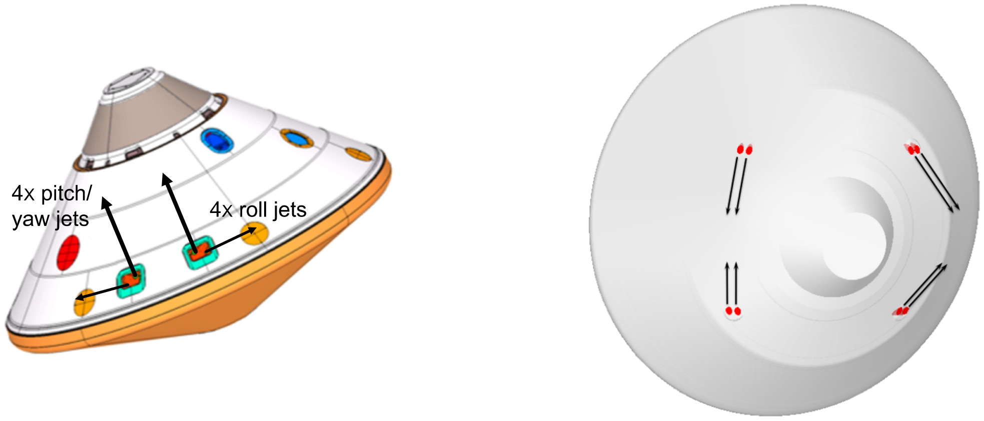


Figure 1.1: Mars 2020 entry vehicle configuration (adapted from Nelessen et al. [10])

Some vehicles have a 3-degree-of-freedom reaction control system (RCS) for active attitude control. RCS jets located on the afterbody generate roll, pitch, and yaw torque. Figure 1.2 demonstrates possible RCS configurations; Phoenix has separate pitch/yaw and roll jets clustered together near the shoulder, whereas MSL has four pairs of combined roll, pitch, and yaw jets constrained further from the shoulder due to packaging concerns.



(a) Phoenix [11]

(b) MSL [12]

Figure 1.2: RCS configuration

Hydrazine monopropellant RCS have thus far been the only active control systems used for Mars entry vehicles. However, studies for future vehicles have proposed other systems, such as aerodynamic control surfaces [13, 14] or internal moving mass actuators (IMMA) [15]. Control surfaces move within the fluid flow in order to change the pressure distribution, and thus the center of pressure, relative to the center of gravity (CG) of the vehicle. Rigid control surfaces have been flown on the Space Shuttle Orbiter [16], X-37 [17], and IXV [18] for Earth entry. Conversely, IMMA systems move the CG relative to the center of pressure [15].

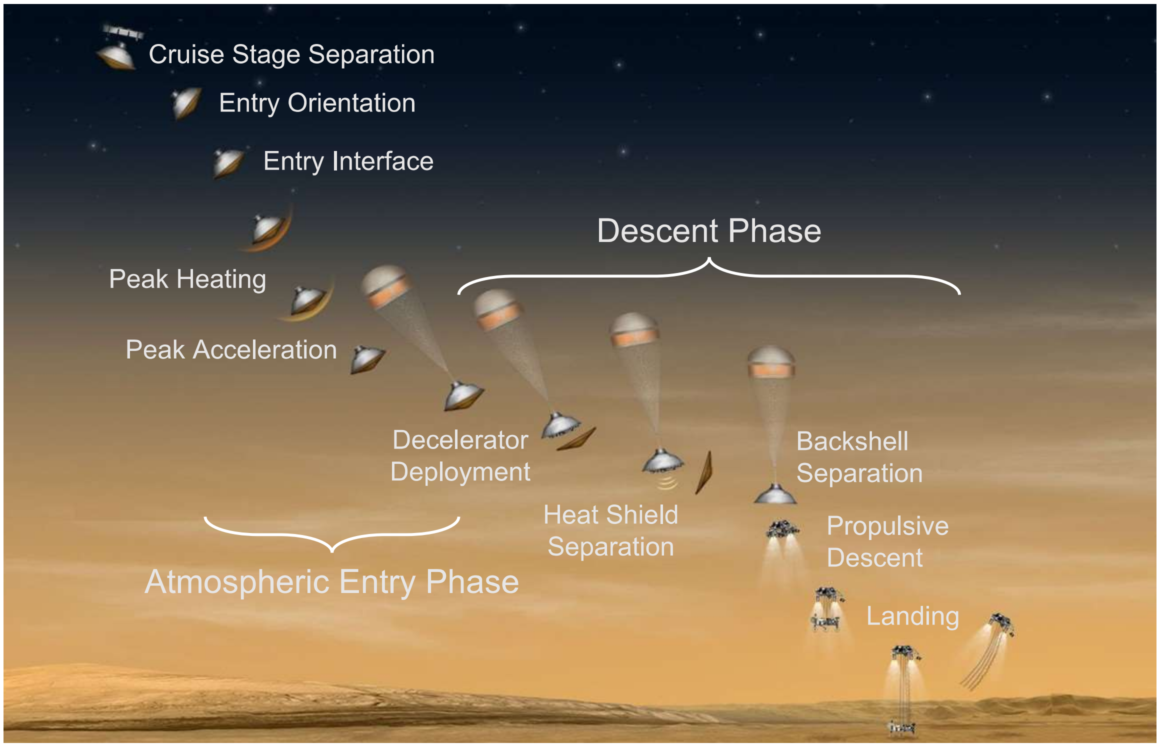


Figure 1.3: EDL concept of operations (adapted from Way et al. [19])

Figure 1.3 illustrates the EDL phases and primary activities for a robotic mission to Mars. Atmospheric entry starts at Entry Interface (EI), a point where the atmosphere is judged to start affecting the vehicle [1]. Prior to EI, the spacecraft may configure itself by jettisoning any cruise stage or service module, and orient itself for atmospheric entry [10]. During entry, vehicles use aerodynamic drag to slow from orbital or interplanetary transfer speeds. They must endure exterme aerodynamic heating and high acceleration as their kinetic and gravitational potential energy is transferred to the atmosphere or converted into heat [1].

The demarcation between the atmospheric entry and descent phases is somewhat arbitrary, and may depend on the specific mission architecture. It could be marked by the deployment of a decelerator (e.g., a parachute) or activation of a propulsive landing system, such that the aerodynamic forces on the vehicle body cease to dominate its flight dynamics. During the descent phase, the vehicle decelerates to a survivable landing speed as it descends to reach the planet surface. Vehicles use propulsion or aerodynamic decelerator systems to control their descent (if the vehicle uses propulsion, descent may be split into “unpowered” and “powered” phases).

The landing (or “touchdown”) is when the vehicle contacts the surface of the planet and reduces its relative velocity to zero.

Entry Trajectory

Entry can be flown along either a ballistic or a lifting trajectory [20]. A ballistic trajectory flies with approximately zero net lift. Without some form of active control or stabilization, the vehicle’s attitude is free to oscillate and drift. Some missions used spin stabilization, rolling the capsule at about 2 rpm, to cancel out lateral forces [21, 22]. Control systems can also be used to damp pitch and yaw oscillation and achieve the same effect. However, these trajectories are still essentially unguided after entry interface.

In a lifting trajectory, the vehicle flies at some angle of attack to produce lift. By modulating the lift vector, the vehicle can control its “downrange” and “crossrange” motion (parallel and perpendicular to the ground track at EI, respectively) [1]. By pointing the lift vector up or down, the vehicle can be made to fly through lower- or higher-density atmosphere, changing the drag and therefore the downrange velocity. Pointing the lift vector to either side changes the crossrange velocity.

Vehicles can be designed with a CG offset to produce a trim angle of attack, minimizing the control authority needed to generate lift. The direction of the lift vector is therefore fixed relative to the vehicle, and its magnitude depends on the flight conditions. Control is achieved by banking to rotate the lift vector as shown in Figure 1.4. The Viking capsules flew a lifting trajectory, but simply maintained orientation to keep the lift vector pointed upward [23]. MSL and Mars 2020 flew lifting trajectories with bank-to-steer to control their descent [24]. Since the crossrange velocity is constantly being changed, pre-planned bank reversal maneuvers are used to manage the crossrange error, as shown in Figure 1.5.

Since this method of control relies on lift, it is most effective during the period of high dynamic pressure.

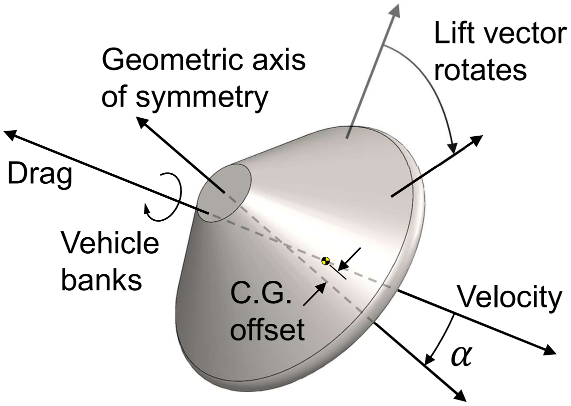


Figure 1.4: Bank-to-steer lift modulation [20]

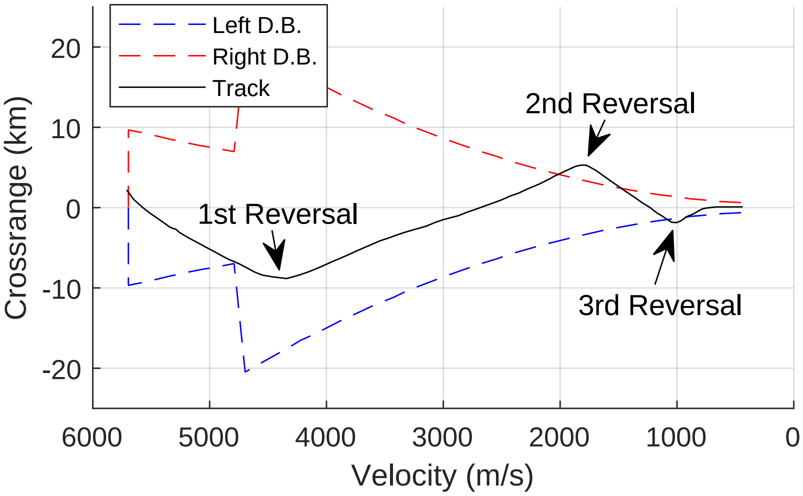


Figure 1.5: MSL bank reversals for crossrange management [24]

Guidance and Control

The entry guidance algorithm seeks to command vehicle motion to satisfy mission constraints. During flight, these constraints include surviving aerodynamic heating and deceleration, as well as specific orientation requirements for sensors, communications, or decelerator systems [20]. The guidance algorithm also seeks to deliver the vehicle to the terminal point with minimal downrange and crossrange error.

MSL was the first mars entry vehicle to fly a lifting trajectory with closed-loop entry guidance, derived from the Apollo entry guidance algorithm [25]. The guidance and control architecture operates in two nested loops as shown in Figure 1.6.

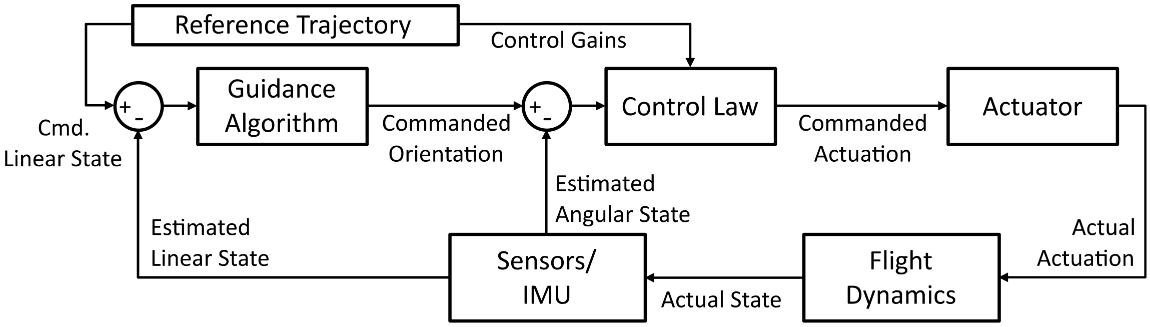


Figure 1.6: Guidance algorithm flowchart (adapted from Mendeck et al. [25])

In the outer loop, the guidance algorithm controls the downrange and crossrange motion, attempting to minimize the error between the actual flight path and a reference trajectory [25]. The algorithm calculates a commanded orientation to produce the desired lift vector. The controller operates over the inner loop, attempting to minimize the error between the actual and commanded orientation [26]. Control surface deflection, jet activation and duration, etc., are determined as functions of the error and current state.

For Mars entry vehicles, the control authority is small compared to the magnitude of the aerodynamic moments [9]. The vehicle is allowed to oscillate about its trim point, with the control system used to damp out angular rates and correct large errors [27].

1.1.2 New Challenges for Mars Entry, Descent, and Landing

The locations of the nine successful U.S. landings are shown in Figure 1.7 on a plot of the Mars Orbiter Laser Altimeter (MOLA) elevations [28]. These landings were restricted to the lower elevations of the northern hemisphere to take advantage of the slightly higher surface atmospheric density and extra few kilometers of distance for decelerating [29]. The only U.S. missions to attempt landing in the southern hemisphere were the relatively lightweight Mars Polar Lander and Deep Space 2 probes (both of which failed during EDL)

[4].

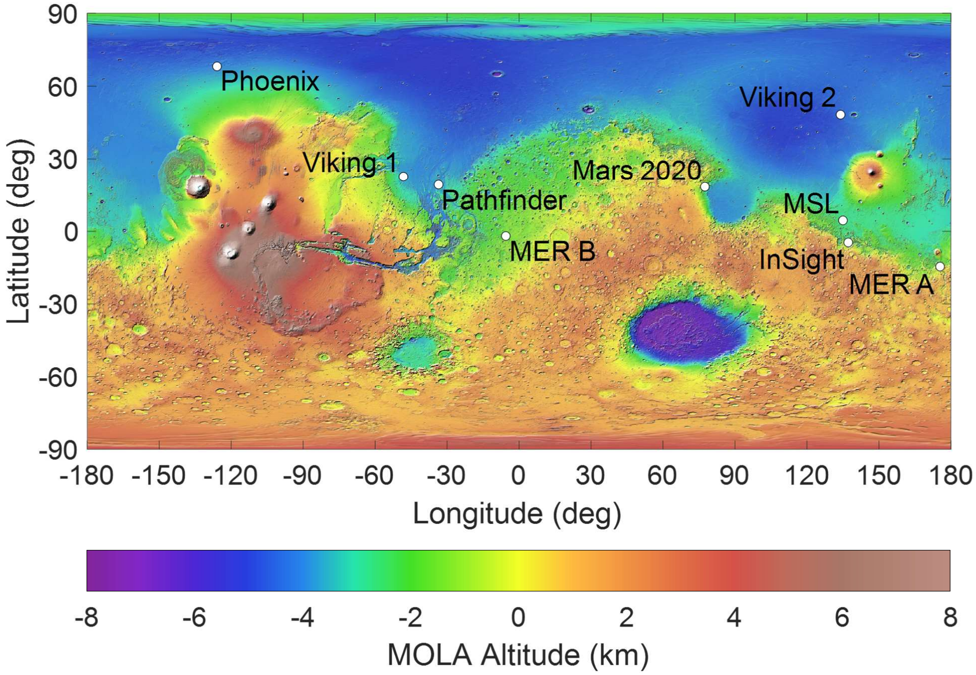


Figure 1.7: Successful U.S. Mars landings

The most recent U.S. mission, Mars 2020, landed the record-breaking 1 metric ton Perseverance rover on the surface [10]. However, Mars 2020 is reaching the limit of payload capability for the Mars EDL heritage [29]. Future missions, especially human exploration of Mars, will present significant new challenges for entry vehicles.

Figure 1.8 illustrates the gap between the current capability of Mars entry vehicles and the desired capability for human missions. The red-shaded area represents the feasible design space for a 4.5m entry vehicle delivered by a currently-available launch vehicle with an entry velocity of 5.5 km/s. For an MSL-class vehicle, this limits the landing site to about 50% of the Martian surface. Human missions will require landers to deliver much higher payload mass — up to 20-30 tons [30]. The desire to explore more of the planet's surface requires landing at elevations up to 2km [30, 29]. This desired capability, shown in the blue-shaded area, would enable exploration of a greater portion of the surface, especially in the southern hemisphere.

These requirements have led to the development of low-density decelerator systems,

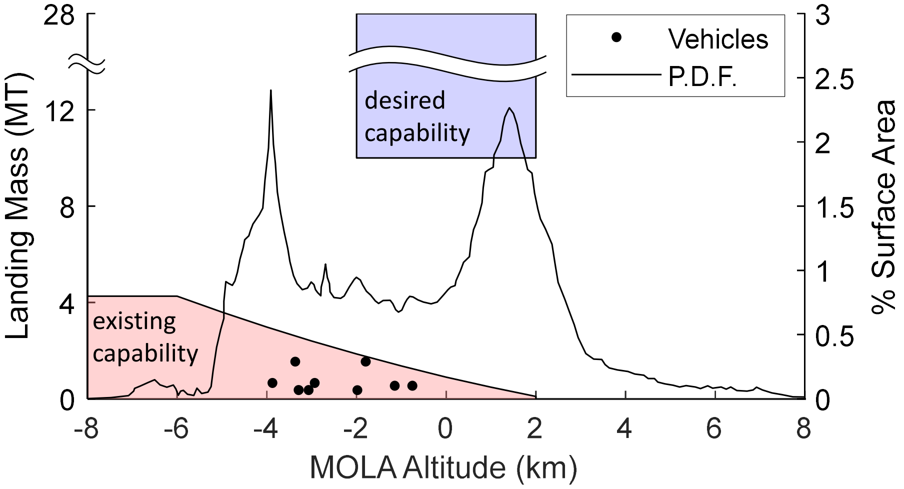
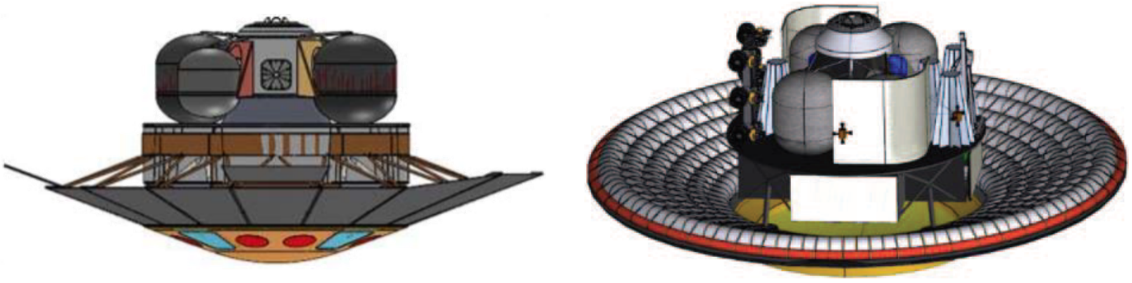


Figure 1.8: Landing mass and land area distribution as functions of altitude

technologies that are designed to decrease the ballistic coefficient by increasing the drag area of the vehicle while minimizing the increase in mass. Many are deployable; packaged within the limited geometry of a fairing during launch, they can be extended while in space, achieving larger diameters than are feasible for rigid aeroshells.



(a) ADEPT [13]

(b) HIAD [31]

Figure 1.9: Decelerator technologies for human Mars landers

One promising technology is the Adaptive Deployable Entry and Placement Technology (ADEPT) [32]. ADEPT is a mechanically-deployed, umbrella-like system with rigid spokes covered with a flexible TPS material. Some configurations include asymmetric spokes for center-of-lift offset, or flaps for aerodynamic control. Another technology is the Hypersonic Inflatable Aerodynamic Decelerator (HIAD) [33]. HIAD is a series of inflatable tauri held in place by tension and covered in a flexible TPS material to form the forebody cone. Human-scale lander concepts using these technologies could have diameters from 16 to 24 meters, over three times the diameter of the MSL entry vehicle [13, 31].

In addition to the change in scale, the configuration of these vehicles are a large departure from Mars EDL heritage. The forebody would be flexible, designed to deform under aerodynamic load. They also lack the regular backshell of heritage vehicles. Other technologies being developed for large entry vehicles, like supersonic retropropulsion [34], may not change the configuration of the vehicle, but they would significantly change how those vehicles operate during entry.

Beyond the mass and elevation requirements discussed previously, human missions will also require much greater landing precision and accuracy. Humans have a smaller acceptable risk than robots for landing in dangerous terrain. Crewed missions will also likely target specific geography to achieve scientific goals [35]. Critically, mission architectures may require landing multiple payload elements in close proximity [36]. Payloads must be landed close enough that humans can assemble a surface base and access their equipment. The human missions being developed could require landing precision of less than 100 meters [31] — also beyond the capability of Mars EDL heritage.

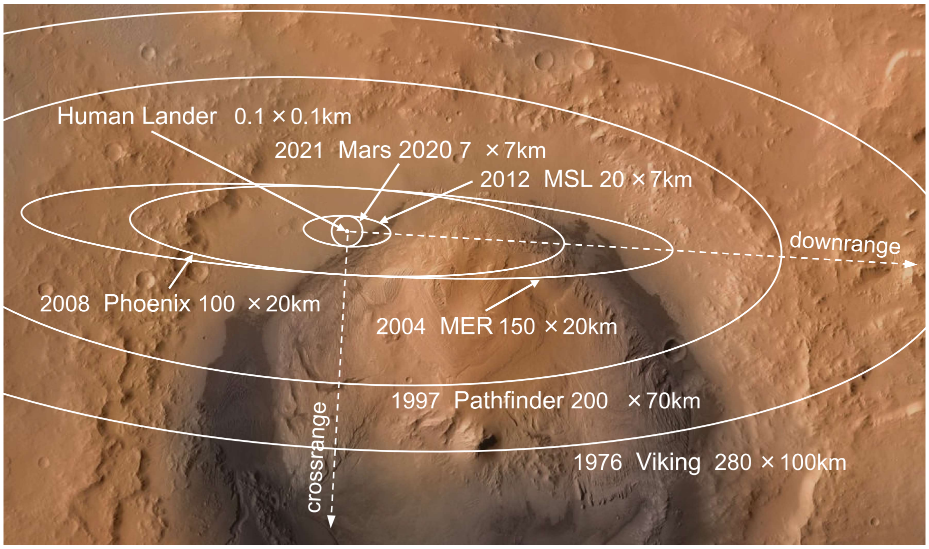


Figure 1.10: Mars mission predicted landing footprint comparison

Figure 1.10 shows the predicted landing footprint for the robotic Mars missions, centered on the landing position of MSL at Gale Crater [20]. The landing footprint is represented as an ellipse, within which the vehicle is predicted to land with a 99% probability. Since 1976, increases in guidance and prediction capability has reduced the landing footprint from hundreds of kilometers to less than 10 km. The largest leap in precision was achieved by MSL due to the use of active guidance during EDL [25], and improvements made between MSL and Mars 2020 resulted in a further 40% reduction in landing ellipse size [10]. However, human missions will require an increase in landing precision by a further two orders of magnitude. For these reasons NASA has defined precision Mars landing as a grand challenge [37]. At a conceptual level, an increase in precision is achieved by decreasing the uncertainty. This forms the motivating objective for this work:

Motivating Objective: Reduce the predicted landing uncertainty for Mars entry, decent, and landing.

In order to further this goal, a detailed examination must be made into the sources and calculation of landing uncertainty.

1.2 Sources of Landing Uncertainty

The landing uncertainty is the distribution of predicted values for the touchdown error. This is the horizontal distance between the landing target and the actual landing position of the vehicle [38]. The landing footprint, described above, is a common figure of merit for the landing uncertainty. This expresses what may be a complex uncertainty distribution as a two-dimensional ellipse for ease of use.

Figure 1.11 categorizes the most important sources of uncertainty during EDL by phase and discipline. For Mars entry vehicles, the uncertainty propagated during entry is the largest contributor to the landing uncertainty [24, 39]. During atmospheric entry, the vehicle is subject to a wide variety of uncertainties from many different disciplines.

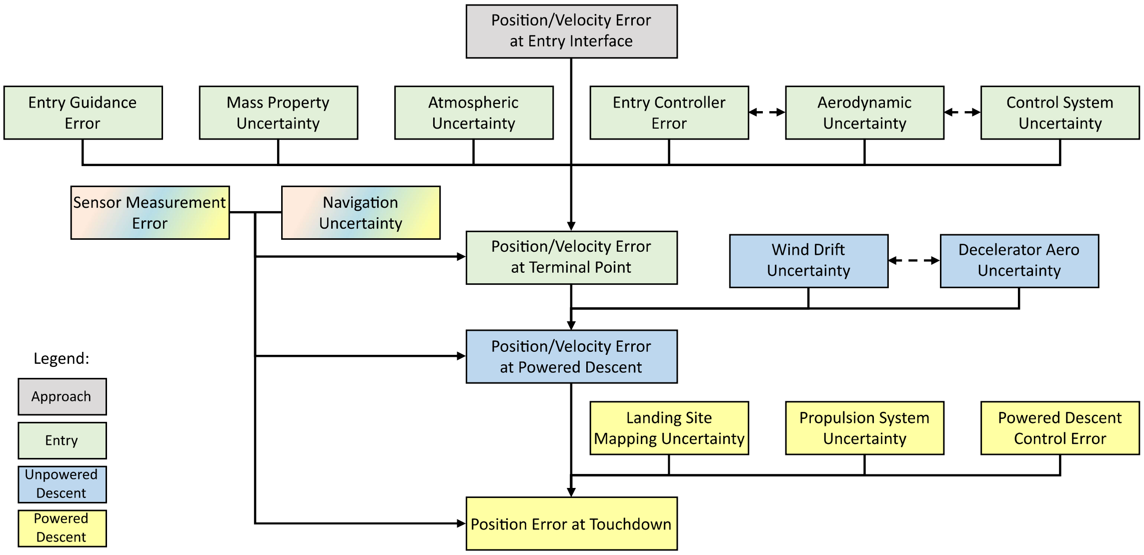

Position/Velocity Error at Entry Interface
Entry Guidance Error
Mass Property Uncertainty
Atmospheric Uncertainty
Entry Controller Error
Aerodynamic Uncertainty
Control System Uncertainty
Sensor Measurement Error
Navigation Uncertainty
Position/Velocity Error at Terminal Point
Wind Drift Uncertainty
Decelerator Aero Uncertainty
Position/Velocity Error at Powered Descent
Legend:
Approach
Entry
Unpowered Descent
Powered Descent
Landing Site Mapping Uncertainty
Propulsion System Uncertainty
Powered Descent Control Error
Position Error at Touchdown

Figure 1.11: Sources of landing uncertainty (adapted from Wolf et al. [39])

Aerodynamics

Aerodynamic uncertainty concerns the magnitude of aerodynamic force and moment coefficients acting on the entry vehicle [40].

As the vehicle flies through entry, it first experiences non-continuum flow at the edge of the atmosphere. It then flies through the hypersonic and supersonic regimes, and may reach transonic or subsonic speeds during entry depending on the conditions for decelerator deployment. The flow features of an entry vehicle in high-speed flow are shown in Figure 1.12. The flow is dominated by complex shock and expansion layer geometry [41]. As the flow passes from the forebody to the afterbody, it might separate or stay attached depending on the flight conditions and the geometry. The flow might attach and separate as the vehicle oscillates in pitch, greatly affecting the pressure distribution on the afterbody [42]. Even if the orientation of the vehicle remained constant over time, the aerodynamics would still be unsteady. Separated flow around the vehicle creates a region of low-pressure, subsonic flow on the afterbody [43]. Unsteadiness in the recirculation is the primary contributor to the uncertainty on the afterbody [43], and changes in the wake can have a significant effect on the stability of the vehicle in the low supersonic and transonic regimes [44, 45]. The fundamentally unsteady nature of the aerodynamics is a significant contributor to the uncertainty, and the uncertainty of individual coefficients can vary widely with flight regime and vehicle state.

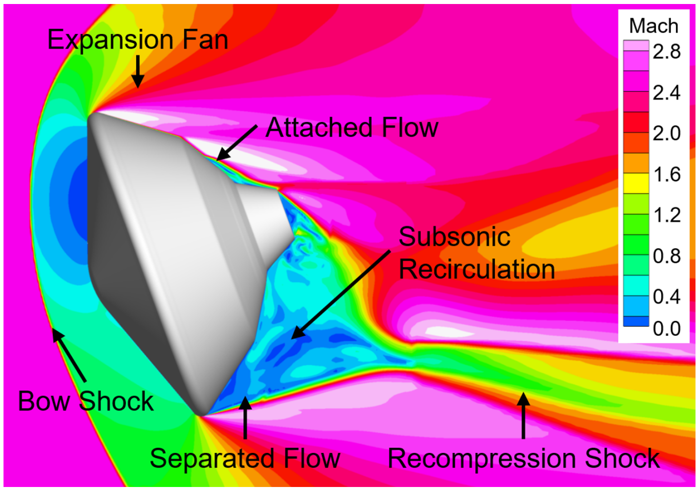


Figure 1.12: Entry vehicle flow features (adapted from Stern et al. [41])

Control Systems

Control system uncertainty represents uncertainty in the behavior of the physical control systems [39]. This includes errors between the commands from the entry controller and the actual authority, responsiveness, and state (e.g., control surface deflection, propulsive thrust coefficient).

For an RCS, the direct effect of the jet due to the momentum transfer with the vehicle will have some uncertainty, but uncertainty is largely driven by the jets' interaction with the aerodynamics. When a jet is activated, a large volume of flow is quickly ejected into the wake of the vehicle [9]. The plume can disrupt the flow around the vehicle, causing an attached flow to separate. Under other circumstances, however, the plume could cause a separated flow to reattach [11]. Since the near portion of the wake is subsonic, the plume can be communicated across the whole afterbody, significantly changing the pressure distribution even on the opposite side of the afterbody [12]. For all the same reasons that the subsonic recirculation contributes to the aerodynamic uncertainty, the RCS plume contributes to the control system uncertainty. In turn, the expansion of the plume is affected by the flight conditions and the flowfield of the vehicle [9]. Two (or more) members of a system are considered to be "coupled" when they have mutual interaction [46, 47]. Thus, the operation of an RCS is fundamentally coupled with the aerodynamics.

Aerodynamic control surfaces may also interact with the aerodynamics of the rest of the vehicle. Shock-shock interactions may form between the control surface and the body, changing with the deflection of the surface [48].

Based on the literature presented in this and the preceding section, the following observation can be made:

Observation 1.1: Entry vehicle aerodynamic behavior is inherently unsteady. For an entry vehicle with an active control system that affects the fluid flow, the interaction between the aerodynamics and control system is inherently coupled and unsteady.

Entry Guidance and Controller

Entry guidance uncertainty consists of targeting uncertainty and termination uncertainty [39]. Targeting uncertainty is the result of errors in the vehicle's understanding of the desired entry trajectory. Termination uncertainty arises due to error in the deployment of decelerator systems, determining when the entry phase ends [10].

Entry controller error, also referred to as tracking error, is the failure of the entry controller to hold the vehicle at the commanded attitude [39].

Other Sources

Uncertainty in the mass properties can arise due to errors in the mass, center of gravity, and inertia of the vehicle [39]. This can be the result of differences between the vehicle as built and the design configuration, or from change in mass during flight (e.g., propellant usage).

Atmospheric uncertainty is the lack of knowledge about density, temperature, chemical composition, etc., of the atmosphere at the time of entry [49]. These properties vary over time, by day and by season, and have different variations at different altitudes. Atmospheric uncertainty affects the flight of the vehicle throughout EDL, but it is most important during entry, where high dynamic pressure has a multiplicative effect on the variation.

Sensor measurement error occurs as the vehicle measures its acceleration, angular velocity, pressures, and temperatures at the vehicle's surface [39]. Navigation uncertainty tracks the knowledge error of the vehicle's initial state, especially attitude, and the integration of that state over time [50]. The Inertial Measurement Unit (IMU) integrates the forces and moments as measured onboard and calculates the vehicle's state. Any uncertainty will be propagated forward in time as the calculations are updated. Sensor measurement and IMU propagation uncertainty affect the vehicle across all phases of EDL [39].

1.3 Uncertainty Quantification

Uncertainty quantification (UQ) is the process of determining the relationship between the true parameters of a system and its approximate representation in the model(s) [51]. UQ is used to predict the landing footprint at various points during vehicle design and during flight prior to EDL [52]. For this purpose a flight simulation framework is constructed to simulate the vehicle as it flies through EDL, incorporating vehicle, planetary, and atmospheric models [53]. Monte Carlo analysis is performed on this simulation, stochastically dispersing input variables to represent the uncertainties in the system. By running thousands of simulations allowing the uncertainties from various sources to affect the vehicle and propagate over time, designers can generate a distribution of touchdown points, from which the footprint is calculated.

Successful Guidance: Mars Science Laboratory

Figure 1.13 shows a prediction for the landing footprint of MSL made the day before landing. In addition to calculating the touchdown footprint, UQ can be used to identify the

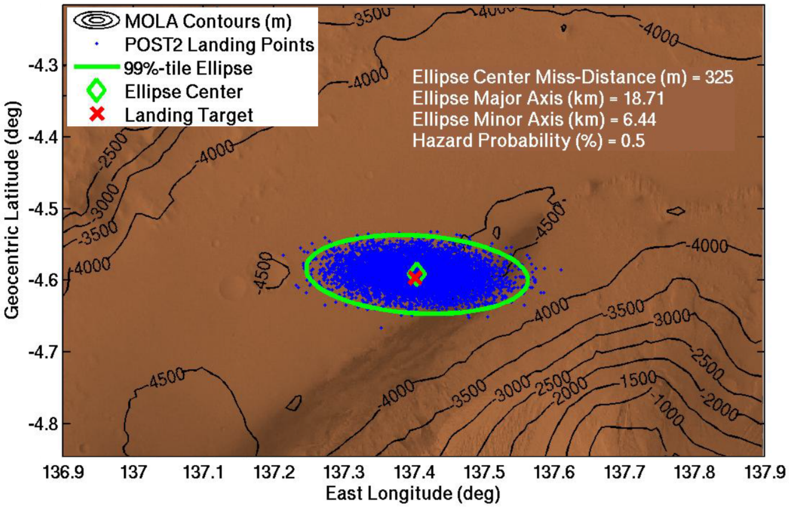
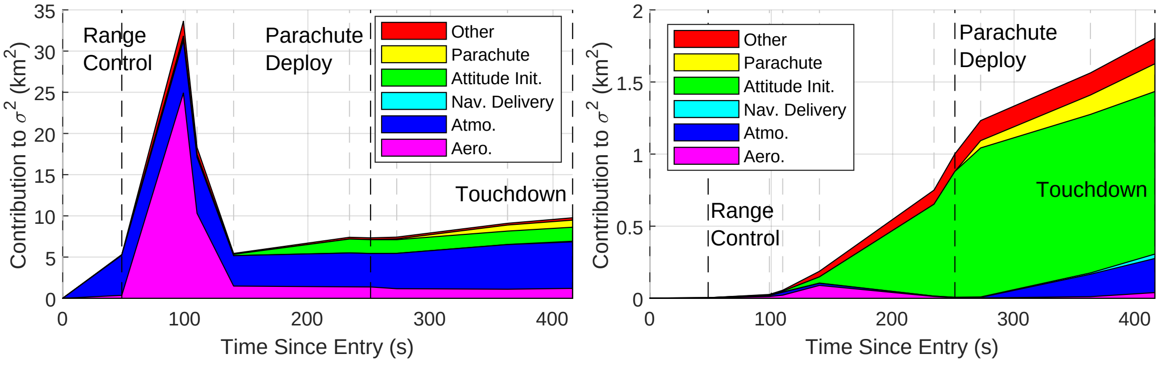


Figure 1.13: MSL landing footprint prediction (adapted from Davis et al. [52])

contribution of each source of uncertainty over time. Figure 1.14 shows the contribution to downrange and crossrange variance for MSL from entry to touchdown [50][2]. Based on the results from MSL and the supporting evidence in Section 1.2, the following observation can be made:

Observation 1.2: Aerodynamic and control system uncertainty are primary contributors to entry uncertainty.



(a) Downrange

(b) Crossrange

Figure 1.14: Contribution to MSL landing variance (adapted from Dutta et al. [50])

[2] The results for MSL using the velocity trigger are considered here over the results for Mars 2020's range trigger since MSL serves as the best example of a new vehicle configuration, with all of the risk-averse decision-making that such a design process entails.

The uncertainty growth over time can be examined in light of the physics and behavior of the system as described above. The downrange variance is much larger than the crossrange variance since the velocity is much higher in this direction. A large spike in the downrange variance occurs during entry. The timing coincides with the period of high dynamic pressure. The rise in aerodynamic uncertainty corresponds to the start of the range control phase of active guidance, where the guidance algorithm starts commanding bank reversals to control the crossrange position.

As an illustrating example, consider a vehicle flying along its reference trajectory when the guidance system decides that it has exceeded the deadband for crossrange error. If it was flying up to this point along the reference trajectory and the pitch and yaw oscillations were within the deadband, the control system would not be actuated. Thus, the only contribution to the uncertainty is due to the aerodynamics of the capsule. However, when the guidance system commands a bank, suddenly the control system must be activated to roll the vehicle, introducing control system uncertainty. Actuation of the control system might affect the pressure distribution in the wake, altering the aerodynamic moments and changing the oscillation of the vehicle. This could cause the controller to actuate the pitch and yaw jets as well. This is borne out by the RCS activity of MSL during flight, where the actuation of the pitch and yaw jets mostly coincide with the bank reversal maneuvers [54]. As a result, multiple jets are firing in close proximity, in short-duration pulses, as the vehicle attitude is rapidly changing, all of which has a significant effect on the uncertainty propagation.

The active guidance limits the growth of uncertainty due to atmospheric and aerodynamic variation. However, actuating the control system introduces uncertainty as the plumes interact with the flow field. The system accepts the temporary increase in downrange variance in order to control the crossrange variance. When the bank reversals are complete, the guidance algorithm can again focus on controlling the downrange position.

Observation 1.3: Active control exacerbates aerodynamic/control system uncertainty growth in the short term.

Unsuccessful Guidance: Mars Phoenix Lander

Phoenix was developed with affordability and low risk in mind [55], reusing the entry vehicle geometry from MER and the control system from the canceled Surveyor mission [56]. It was also intended to be the first Mars surface mission since Viking to carry RCS for control during entry. The budget and time constraints meant that, aside from historical data and ground testing for other programs, analysis of the aerodynamics and control system would be limited to computational experiments. During development, Computational Fluid Dynamics (CFD) simulations were used to predict the interaction of the RCS jets with the flow field around the vehicle in the hypersonic and supersonic regimes [11]. However, these simulations resulted in large uncertainties, to the extent that the net sign of the effective control moment could not be determined for some orientations. If the jets were commanded to pitch the vehicle up for example, designers could not eliminate the possibility that they might actually cause the vehicle to pitch down.

Given this high uncertainty of the CFD and the lack of physical testing for validation, mission planners decided to increase the deadbands of the controller during flight, to prevent the jets from firing for all but the most extreme movement. This effectively disabled control during the entry phase, making Phoenix the first unguided, uncontrolled Mars entry vehicle. Phoenix landed at the very edge of the 99th-percentile landing ellipse [57]. The Mars InSight lander, with a near-copy of the Phoenix entry vehicle, was flown in 2018 using the same relaxed deadbands [58].

Sensitivity to Aerodynamics, Guidance, and Control System Uncertainty

Uncertainty propagation reveals that entry uncertainty is usually the largest contributor to landing uncertainty. Furthermore, the uncertainty due to the aerodynamics and control system are usually the largest contributors to entry uncertainty. The interaction between the entry controller, aerodynamics, and control system has previously been identified, lending importance to examining these sources together.

Entry guidance corrects for error and mitigates uncertainty, but it relies on knowledge of the aerodynamic and control system behavior of the vehicle. Figure 1.15 illustrates the twofold effect that aerodynamic and control uncertainty have on the landing uncertainty.

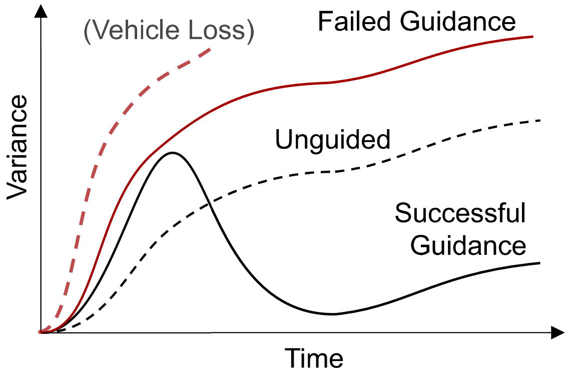


Figure 1.15: Notional uncertainty growth during EDL

The solid black line represents cases like MSL, where the active guidance is successful in limiting the growth of variance over time. The dashed line represents how a vehicle without active guidance would have lower variance growth during the early stages of EDL, but end up with a significantly increased downrange variance. The solid red line shows a case where the guidance is unsuccessful, such as if Phoenix had flow without reduced deadbands. Not only would an increase in these uncertainties be propagated forward, it would also result in reduced effectiveness of guidance and control, compromising the ability to correct for uncertainty. This illustrates one reason why the controller deadbands were increased for Phoenix: the use of the guidance was judged to result in higher uncertainty than simply flying an unguided entry.

Observation 1.4: Entry controller success relies on accurate knowledge (i.e., low uncertainty) of the aerodynamics and control systems.

Phoenix and MSL represent the most recent two unique Mars entry vehicle designs, with heritage going back to the Viking landers. Phoenix suffered from uncertainty arising from the control/aerodynamic interaction which essentially nullified the effectiveness of the entry guidance. While Phoenix serves as an example of the deleterious effect of uncertainty on guidance, MSL demonstrates the effort required to eliminate these effects. Even with the extensive heritage of Mars entry vehicles, aerodynamic/control interaction problems persist and continue to affect entry guidance.

Observation 1.2 indicates that examining the aerodynamic and control system uncertainty growth during entry could be a fruitful strategy for increasing landing precision. Observations 1.3 and 1.4 describe a feedback cycle that could lead to failed guidance as shown in Figure 1.15. Examining the converse of this scenario leads to the first statement of the motivating hypothesis:

Motivating Hypothesis, Part 1: If the aerodynamic and control system uncertainty during atmospheric entry can be reduced, the performance of the entry controller, and therefore landing precision, can be improved.

This statement suggests investigating the process of accounting for and delivering the aerodynamic and control system knowledge and uncertainty.

1.3.1 Disciplinary Models and Uncertainty

The disciplinary models for the aerodynamics and control system define the values and uncertainties used during UQ. These models are constructed as databases of aerodynamic coefficients as a function of the local conditions [42, 58, 56].

These models are traditionally provided as independent, self-contained modules which are surrogates for one or more higher-fidelity models [53]. For example, the aerodynamics model may provide the force and moment coefficients as a database [59, 42]. Interpolation is used to calculate the aerodynamic coefficients at off-design points. These models achieve a low computational cost by a reduction in fidelity, necessary due to the large number of flight simulations that must be performed. Each model may be constructed by different organizations at different stages of the design process [53].

The UQ methodology for Mars entry vehicles follows the pattern of continuous development from a common heritage. This heritage only extends back to Pathfinder, since the methods for UQ and risk assessment used for Viking were poorly documented [40]. Korzun et al. [58] describe the process of UQ for the Mars InSight capsule, taking into account “differences in fidelity and uncertainty in the underlying tools and methods used to generate the nominal aerodynamic coefficients, as well as engineering judgement and post-flight reconstruction on prior Mars EDL missions.” For vehicles which shared an aeroshell geometry, the uncertainty model was used with no modification from the first vehicle [58].

Robertson [60] developed a taxonomy of uncertainty in the development of space and launch vehicles which can be applied here to examine the components of uncertainty affecting guidance development. This taxonomy, shown in Figure 1.16, can be considered for each disciplinary uncertainty identified in Figure 1.11, such as aerodynamic uncertainty.

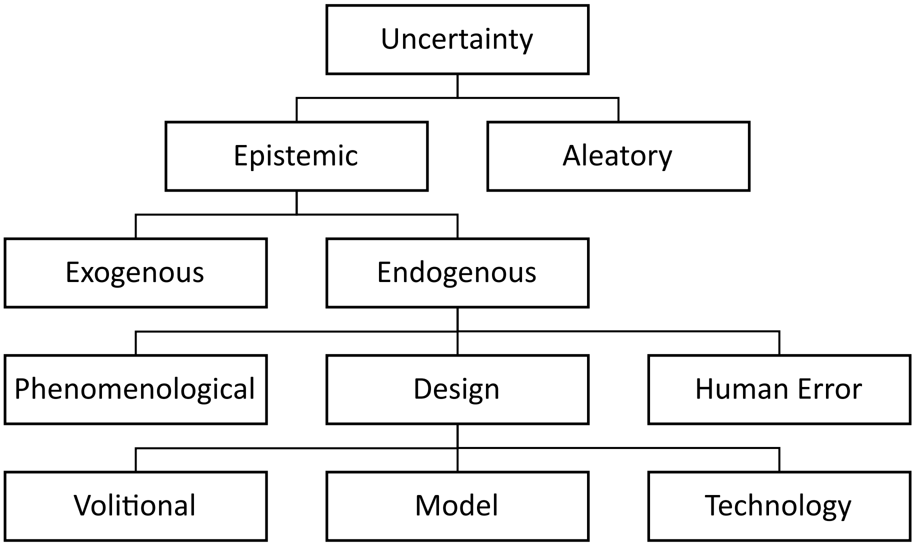


Figure 1.16: Taxonomy of uncertainty (adapted from Robertson [60])

Epistemic uncertainty is the lack of knowledge about the system, whereas aleatory uncertainty is due to the inherent randomness in the system. Within the epistemic error, sources that are within the control of designers are termed “endogenous”. Phenomenological uncertainty results from a lack of knowledge about physical phenomena [60]. These are the “unknown unknowns” that are present even at full design maturity. The Space Shuttle Orbiter notably encountered differences in between predicted and actual hypersonic pitching moment as a result of unanticipated Mach number and real gas effects during entry [61, 62]. Design uncertainty results from lack of knowledge during the design process [60]. Model uncertainty describes the lack of fidelity in the analysis tools used during design. This includes decisions made about the use of the model during design, such as the selection of model inputs and outputs, and the use of surrogate models.

Whereas determining the measurement error is a fully quantitative process, state-of-the-art[3] accounting for phenomenological and model uncertainty is a partly qualitative process, using "engineering judgment" based on the state of the art and historical missions. Evaluation of model uncertainty involves the comparison of the different sources of data and evaluating their relative fidelity. In an evaluation of CFD sources, variability is assessed based on the use of different codes, governing equations, turbulence models, grid resolutions, etc., but this does not capture the limitations of the models themselves [12]. Their fidelity is compared to physical experiments, where the "model" is a subscale object in real, physical flow, with all the phenomenological uncertainty that entails. However, there are still large differences in the model represented by a subscale wind tunnel experiment performed in Earth atmosphere and full-scale flight conditions at Mars.

This phenomenological uncertainty remains a primary concern for designers. The designers for MSL adopted a policy or robustness for setting the aerodynamic uncertainty bounds [40]. They started with the chosen values for MER and Pathfinder, and adjusted them based on engineering judgement of the differences in flight regime, the lifting trajectory, and vehicle geometry. Decisions were made conservatively, such as the choice to use the dynamic coefficient model that produced the most destabilizing behavior [63].

This database is considered to be quasi-static, since the only dynamic coefficient it contains is pitch damping. This means that the uncertainty must account for the model fidelity uncertainty due to ignoring the other unsteady behavior. The models for the aerodynamics, control system, and guidance & control are also constructed independently. This means

[3] The term "state of the art" in this dissertation refers to U.S. Mars entry vehicle programs.

that the interactions between these disciplines must also be accounted for using the uncertainty. This is an indirect method of accounting for the coupled, unsteady behavior, as opposed to a direct method that would include these effects within the model.

Observation 1.5: The state-of-the-art methodology indirectly accounts for coupled, unsteady behavior through model uncertainty.

For future Mars missions, the interplay becomes more complex. The increased landing precision necessary for human missions will require lower overall uncertainty. New technological uncertainty is introduced by the use of low-density decelerator technologies. The change in OML, TPS, etc., could also introduce phenomenological uncertainty. With the phenomenological and technological uncertainty increasing at the same time that overall uncertainty must decrease, the only remaining area that can be managed is the model uncertainty.

Observation 1.6: Overall uncertainty can be managed by reducing the model fidelity uncertainty.

1.4 Thesis Objectives and Outline

Future Mars missions will require increased landing precision, which can be accomplished by limiting the growth of uncertainty during EDL. This can be most effectively achieved by using active guidance, but this also imposes requirements on the aerodynamic and control system uncertainty during atmospheric entry. The uncertainty must be low enough that the controller can effectively maintain the vehicle attitude, and that the uncertainty induced by the active guidance does not outweigh its benefits.

Guidance development for Phoenix was hampered by model uncertainty. A lack of fidelity in the CFD models meant that model uncertainty was too high to verify correct behavior of the control system. Rather than attempting to reduce this uncertainty, designers opted to change the requirements, accepting a higher landing uncertainty.

The development of MSL demonstrates the effort necessary to meet these requirements. MSL was a deviation from the Viking design heritage - incurring increased volitional uncertainty by flying at a higher trim angle of attack and changing the position of the RCS jets, even though it retained many similarities to the OML. In order to compensate, a rigorous testing regime was run with physical and computational experiments to reduce the model uncertainty. The landing of Mars 2020 demonstrates that the desired level of precision may be within reach for the current Mars entry vehicle heritage, with 10 landing attempts over the past 46 years.

However, future Mars missions will have more stringent payload and landing elevation requirements, beyond the capabilities of the current heritage. New vehicle designs will increase the volitional uncertainty, and new technologies will increase the technological and phenomenological uncertainty at the same time as overall uncertainty must be reduced. This leaves model uncertainty as the only other epistemic source of uncertainty within the designers' control. The best recourse is to reduce uncertainty by increasing the aerodynamic, flight dynamic, and control model fidelity, and adapt the guidance development to leverage that reduction in uncertainty. This was expressed in the first statement of the Motivating Hypothesis, repeated below:

Motivating Hypothesis, Part 1: If the aerodynamic and control system uncertainty during atmospheric entry can be reduced, the performance of the entry controller, and therefore landing precision, can be improved.

Observation 1.1 notes that the behavior of entry vehicle aerodynamics and control systems are fundamentally unsteady. These have affected Mars entry vehicles in the past and continue to be a critical design consideration, even with a common design heritage. However, Observation 1.5 finds that the state-of-the-art methodology uses the model fidelity uncertainty to indirectly account for the effects of unsteady control/aerodynamic/flight dynamics interaction. As noted in Observation 1.6, model fidelity uncertainty is the best available tool for managing the overall uncertainty. A reduction could be achieved by ameliorating the compromise made in the state-of-the-art methodology. This forms the second statement in the Motivating Hypothesis:

Motivating Hypothesis, Part 2: If the process of entry controller tuning directly accounts for coupled, unsteady behavior, then the aerodynamic and control system uncertainty during atmospheric entry can be reduced.

These two statements form a hypothetical syllogism relating back to the Motivating Objective. The whole Motivating Hypothesis leads to the Research Objective for this work:

Research Objective

Develop an entry controller tuning methodology that directly accounts for coupled, unsteady entry vehicle aerodynamic and control system behavior.

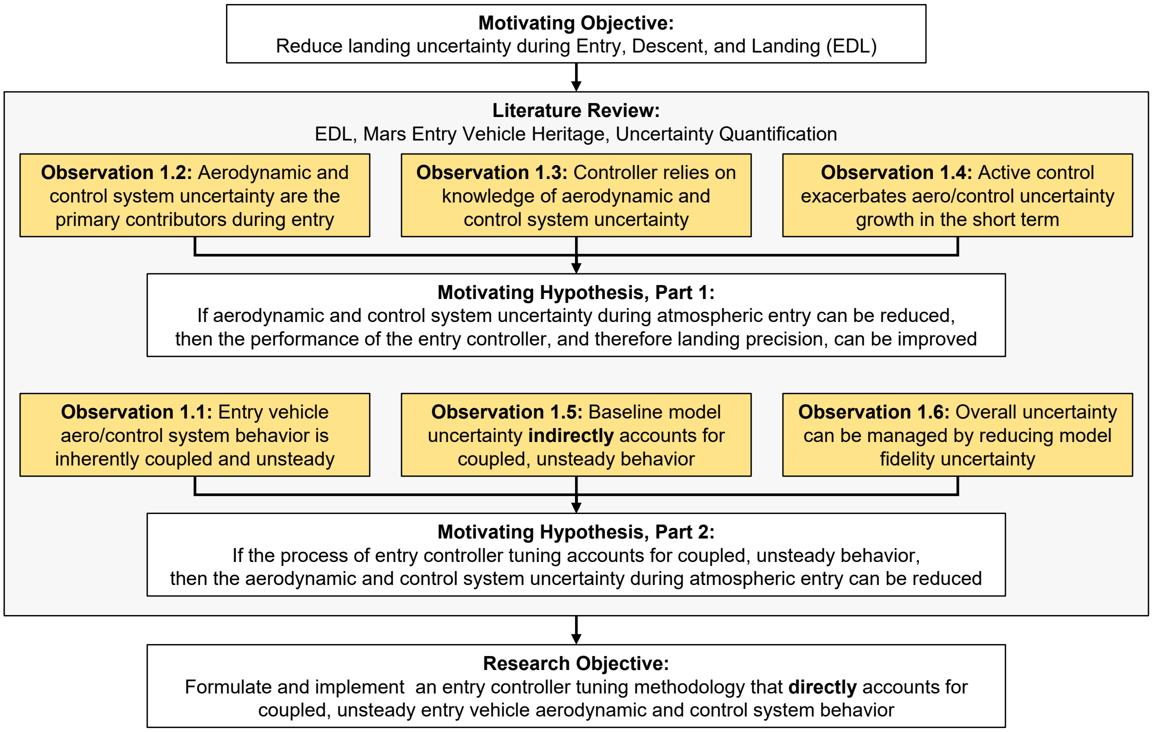
Figure 1.17 summarizes the motivation presented in this chapter that led to the Research Objective.

The remainder of this dissertation is organized as follows:

- Chapter 2 presents a literature review of tools required for controller tuning as used in historical missions and for other vehicles, along with alternative methods used in similar problems.

- Chapter 3 presents an examination of the state-of-the-art controller tuning process, and investigates how it could be adapted to use the model alternatives identified in Chapter 2.

- Chapter 4 formulates research questions, hypotheses, and experiments to support the investigation of the overarching research objective.



Motivating Objective: Reduce landing uncertainty during Entry, Descent, and Landing (EDL)

Literature Review: EDL, Mars Entry Vehicle Heritage, Uncertainty Quantification

Observation 1.2: Aerodynamic and control system uncertainty are the primary contributors during entry

Observation 1.3: Controller relies on knowledge of aerodynamic and control system uncertainty

Observation 1.4: Active control exacerbates aero/control uncertainty growth in the short term

Motivating Hypothesis, Part 1: If aerodynamic and control system uncertainty during atmospheric entry can be reduced, then the performance of the entry controller, and therefore landing precision, can be improved

Observation 1.1: Entry vehicle aero/control system behavior is inherently coupled and unsteady

Observation 1.5: Baseline model uncertainty **indirectly** accounts for coupled, unsteady behavior

Observation 1.6: Overall uncertainty can be managed by reducing model fidelity uncertainty

Motivating Hypothesis, Part 2: If the process of entry controller tuning accounts for coupled, unsteady behavior, then the aerodynamic and control system uncertainty during atmospheric entry can be reduced

Research Objective: Formulate and implement an entry controller tuning methodology that **directly** accounts for coupled, unsteady entry vehicle aerodynamic and control system behavior

Figure 1.17: Summary of motivation and Research Objective

- Chapter 5 documents the technical approach, including the development of a coupled modeling framework.

- Chapter 6 presents the selection of a vehicle, control mechanism, and control system for the problem of interest.

- Chapters 7 through 10 document the experimentation, results, and findings.

- Chapter 11 presents the conclusions, primary contributions, and discusses opportunities for future work.

CHAPTER 2
MODELING AND SIMULATION

The overarching research objective presented at the end of Chapter 1 can be considered in two parts. The first, and the subject of this chapter, is the "direct" accounting for coupled, unsteady behavior. As is understood from Section 1.3.1, this occurs in the construction and use of the disciplinary models. A series of guiding questions can be posed to direct the literature search:

Guiding Question 1: What is the state-of-the-art methodology for creating the flight dynamics, aerodynamic, and control system models used for entry controller tuning?

A literature review for each of these disciplines will be presented, encompassing the underlying physics, the experimental models, and the delivery of this data for flight simulation. The perspective of this chapter primarily follows the state of the art in entry vehicles for NASA missions. Studies for historic and contemporary entry vehicles are also considered. In addition to surveying the methods themselves, an evaluation must be made of their performance:

Guiding Question 2: What are the limitations of this methodology with respect to the physics of actively-controlled entry vehicle flight?

When examining the methodologies used for individual activites, a philosophy of investigating similar processes in other disciplines is applied to find opportunities for cross-pollination of ideas:

Guiding Question 3: What alternative methods exist that could ameliorate these limitations?

The literature search will be used to make broad and specific observations that serve to answer these guiding questions.

2.1 Flight Dynamics Model

The flight dynamics of historical Mars entry vehicles have been modeled according to rigid body dynamics (RBD) [57, 19]. The vehicle is treated as a rigid body (i.e., its OML does not deform) subject to external forces and moments including gravity, aerodynamics, propulsion, etc., that are calculated as a function of the current vehicle state. In response, the vehicle can move in six degrees of freedom (6DOF): three angular degrees (roll, pitch, and yaw rotation), and three linear degrees (translation).

Equations of Motion

Etkin [64] gives the equations of motion for a rigid body with respect to an inertial reference frame. The inertial frame I and body frame B are defined as shown in Figure 2.1. The body frame is fixed to the vehicle with its origin at the center of mass, the x-axis pointing towards the front of the vehicle, and the y- and z-axes forming a right-handed reference frame. The vehicle state $\bar{x}$ consists of: $\bar{r}_I$, its position vector of the center of gravity in the inertial frame; $\bar{v}_B$ and $\bar{\omega}_B$, its body-frame linear and angular velocity vectors, respectively; and ϕ_i, θ_i, and ψ_i, the roll, pitch, and yaw Euler angles, respectively.

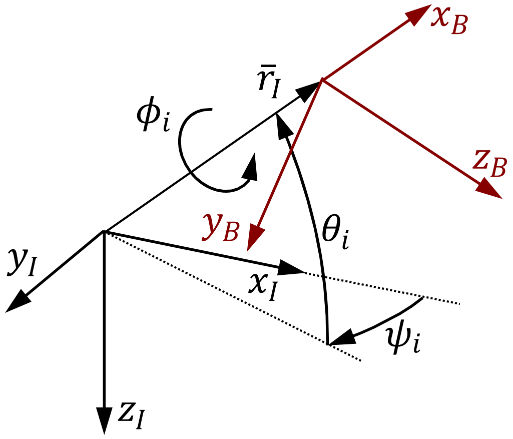


Figure 2.1: Inertial and body frames

The equations of motion govern changes to the vehicle state over time [64]:

$$\dot{\bar{v}}_B = \frac{1}{m}(\bar{F}_B - \dot{m}\bar{v}_B) - \bar{\omega}_B \times \bar{v}_B \tag{2.1}$$

$$\dot{\bar{r}}_I = \boldsymbol{R}_{IB}\bar{v}_B \tag{2.2}$$

$$\dot{\bar{\omega}}_B = \boldsymbol{I}_B^{-1}(\bar{M}_B - \bar{\omega}_B \times (I_B\bar{\omega}_B) - \dot{\boldsymbol{I}}_B\bar{\omega}_B) \tag{2.3}$$

$$\begin{bmatrix} \dot{\phi}_i \\ \dot{\theta}_i \\ \dot{\psi}_i \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi_i \tan\theta_i & \cos\phi_i \tan\theta_i \\ 0 & \cos\phi_i & -\sin\phi_i \\ 0 & \sin\phi_i \sec\theta_i & \cos\phi_i \sec\theta_i \end{bmatrix} \bar{\omega}_B \tag{2.4}$$

$\bar{F}_B$ and $\bar{M}_B$ are the external force and moment vectors, respectively. The mass properties of the vehicle are its mass m and inertia tensor $\boldsymbol{I}_B$. A dot over the variable represents its derivative with respect to time. $\boldsymbol{R}_{IB}$ is the rotation matrix defining the transformation between the body and inertial frames:

$$\boldsymbol{R}_{IB} = \begin{bmatrix} \cos\psi_i \cos\theta_i & \cos\psi_i \sin\phi_i \sin\theta_i - \cos\phi_i \sin\psi_i & \sin\phi_i \sin\psi_i + \cos\phi_i \cos\psi_i \sin\theta_i \\ \cos\theta_i \sin\psi_i & \cos\phi_i \cos\psi_i + \sin\phi_i \sin\psi_i \sin\theta_i & \cos\phi_i \sin\psi_i \sin\theta_i - \cos\psi_i \sin\phi_i \\ -\sin\theta_i & \cos\theta_i \sin\phi_i & \cos\phi_i \cos\theta_i \end{bmatrix} \tag{2.5}$$

Together, Equations 2.1 to 2.4 form a set of 12 ordinary differential equations.

Numerical Integration

Numerical integration is used to solve the equations of motion and propagate the vehicle state forward in time according to the initial value problem in Equation 2.6 [65].

$$\dot{\bar{x}} = f(\bar{x}, t), \quad \bar{x}(t_0) = \bar{x}_0 \tag{2.6}$$

A variety of numerical integration schemes are available depending on the class of problem to be solved [66]. The Runge-Kutta 4th-order integration scheme (RK4) is widely used in trajectory propagation [67, 68]. RK4 is a single-step method, using only the information from the previously-calculated step. This integration scheme has been found to be suitable for both smoothly- and non-smoothly-varying problems [67]. The RK4 scheme calculates the update to the vehicle state based on a series of substeps [65]:

$$\bar{x}_{t+\Delta t} = \bar{x}_t + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.7}$$

where Δt is the physical time step, and k_1-k_4 are the derivatives at each of the substeps:

$$\begin{aligned} k_1 &= f(\bar{x}, t) \\ k_2 &= f(\bar{x} + \Delta t\frac{k_1}{2}, t + \frac{\Delta t}{2}) \\ k_3 &= f(\bar{x} + \Delta t\frac{k_2}{2}, t + \frac{\Delta t}{2}) \\ k_1 &= f(\bar{x} + \Delta t k_3, t + \Delta t) \end{aligned} \tag{2.8}$$

Trajectory Simulation Environment

Since the flight dynamics model requires the inputs and outputs for the disciplinary models and generates the updated vehicle state, it is natural to make this the core of a trajectory simulation environment [69]. Verification and validation of Mars EDL relies almost entirely on computer simulation. Therefore, the state of the art for Mars vehicle development includes the construction of a complete end-to-end (E2E) simulation environment for the EDL activities. These environments can run continuously from entry interface all the way to touchdown.

Way et al. provide a detailed overview of the E2E environment constructed for MSL [19]. The simulation environment is constructed around a trajectory propagator that uses numerical integration to solve the equations of motion. The propagator calls disciplinary models to determine vehicle properties and manage simulated activities. Figure 2.2 shows the disciplinary models present in the MSL E2E environment.

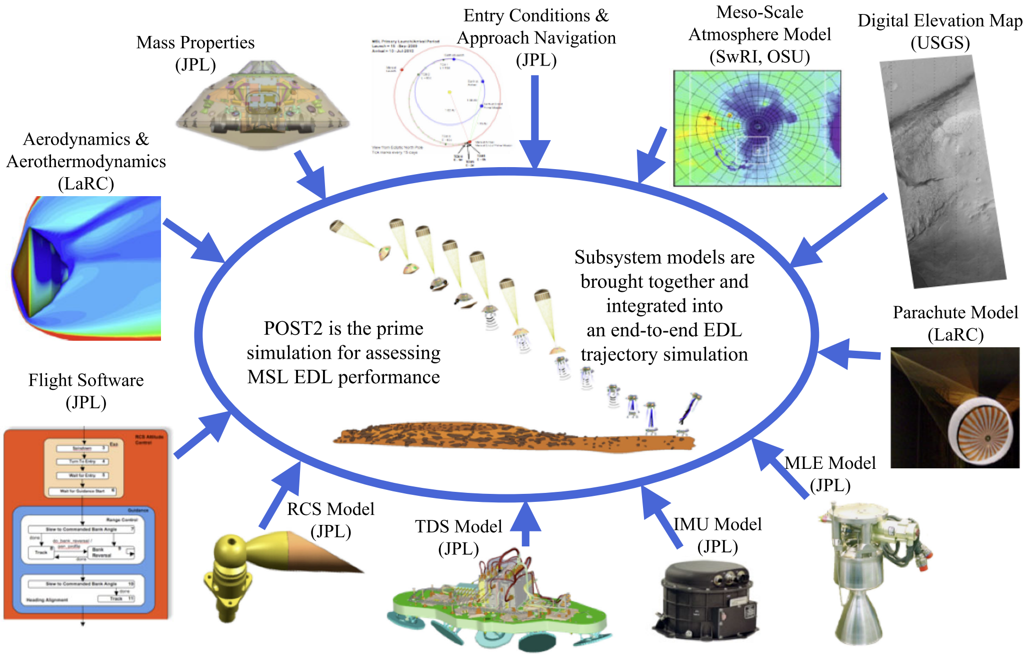

Mass Properties (JPL)
Entry Conditions & Approach Navigation (JPL)
Meso-Scale Atmosphere Model (SwRI, OSU)
Digital Elevation Map (USGS)
Aerodynamics & Aerothermodynamics (LaRC)
Subsystem models are brought together and integrated into an end-to-end EDL trajectory simulation
POST2 is the prime simulation for assessing MSL EDL performance
Parachute Model (LaRC)
Flight Software (JPL)
MLE Model (JPL)
RCS Model (JPL)
TDS Model (JPL)
IMU Model (JPL)

Figure 2.2: MSL E2E simulation environment [52]

The disciplines include physical models such as the mass properties, aerodynamics, propulsion, atmosphere and terrain, etc. The E2E simulation also includes simulations of the flight software, sensors, and data collection. Communications to, from, and between these models are handled by the trajectory propagator.

Predictions of landing uncertainty are made by performing Monte Carlo simulations using the E2E environment. Each disciplinary model also includes uncertainty parameters which are randomly dispersed. The MSL simulations, for example, contained 682 dispersions [19].

The E2E environment is also used for designing and validating the guidance algorithm and control law. Monte Carlo simulations are run over a shorter duration, with dispersions on the uncertainty parameters and a design of experiments for exploring the space of control gains.

The execution time for a 6DOF trajectory simulation is low enough that it can be used directly in Monte Carlo simulations [38, 50]. While reduced-order models for trajectory simulation might be used, they result in reduction of execution time by 1-2 orders of magnitude [70].

Observations

The flight simulation framework is designed to handle general communication between disciplinary models, and it is agnostic to the type of disciplinary models being used. The communication is processed at each time step, meaning that if the disciplinary models have the capacity, state-of-the-art flight dynamics models are capable of time-accurate communication with aerodynamic, guidance, and control system models.

Furthermore, the execution time necessary for the numerical integration is low. Even adding the computation time necessary purely for communication with (but excluding the execution times of) the disciplinary models does not significantly increase the cost.

Therefore, there is no reason that the state-of-the-art flight dynamics cannot be used, provided that the rigid body assumption still holds valid. Further review will thus focus on the aerodynamic and control system models.

2.2 Aerodynamic Models

An aerodynamic model provides an estimate of the effect of the fluid flow on the vehicle surface. This can be found as a distribution of pressure and shear forces over the whole surface, or integrated into forces and moments acting on a reference point. The force and moment coefficients acting on the vehicle are defined in Figure 2.3.

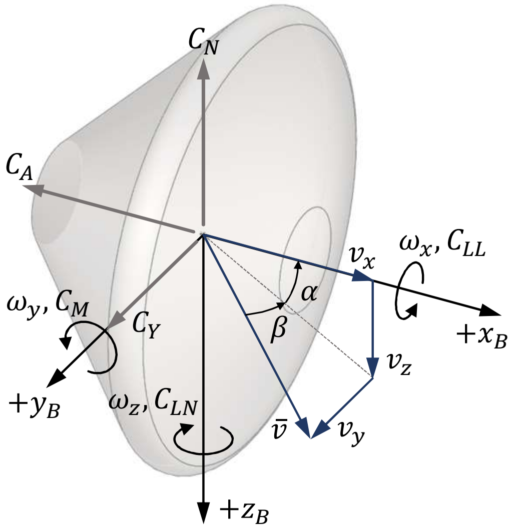


Figure 2.3: Aerodynamic coefficients

This section details the flight regimes that are applicable during entry and the multiple individual models that are used within these regimes (or a subset thereof).

2.2.1 Flight Regimes

As the vehicle enters the atmosphere at five to eight kilometers per second, it encounters a wide variety of flight regimes as shown in Figure 2.4. The vehicle first experiences non-continuous flow in the rarified atmosphere at high altitudes [71]. At lower altitudes and higher densities, the flow transitions to continuous. The focus of this section is on the hypersonic and supersonic regimes, where the dynamic pressure is high enough for the aerodynamic and control uncertainty to produce significant dispersion, and where guidance is most effective.

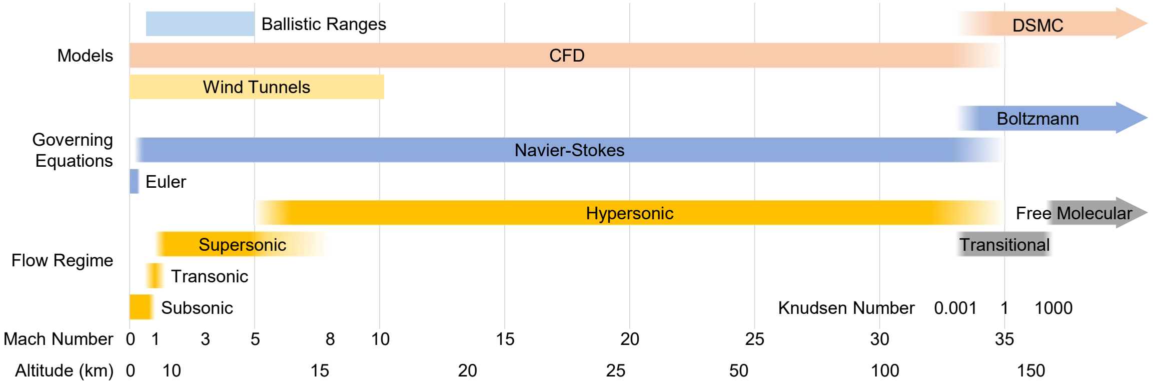


Figure 2.4: Atmospheric entry flight regimes

Hypersonic Flow

The hypersonic regime is defined on the upper end of speed and altitude by the transition to continuous flow, where Knudsen number $Kn \ll 1$, equivalent to about Mach 35 on Mars. The lower end of the scale is somewhat arbitrarily based on behavior and flow features, with sources defining the transition to supersonic flow anywhere from Mach 15 [71] to Mach 5, with some hypersonic behavior observed as low as Mach 3 [20]. Flight in the hypersonic regime is dominated by the complex flow features shown in Figure 2.5. The MSL entry vehicle is shown flying at an angle of attack α relative to horizontal freestream flow. The forebody causes a bow shock, with expansion fan and lip shocks at the interface with the afterbody [42]. The sonic line marks the boundary of the subsonic flow on the surface of the forebody. The size of the shock layer is dictated by the vehicle geometry and speed. The behavior of the flow is also specific to Mars' CO2 atmosphere. Transition across the shock and dissipation of kinetic energy by friction results in a high temperature and pressure boundary layer [72]. These high temperatures can cause the flow to be in thermochemical nonequilibrium due to the disassociation and ionization of atmospheric gasses. The high temperature boundary layer is the reason for the use of TPS. With an ablative TPS, the energy of the flow is absorbed into pyrolysis of the ablative material, resulting in outgassing. The ablation causes additional chemical processes and nonuniform changes to the shape of the forebody surface over time [73].

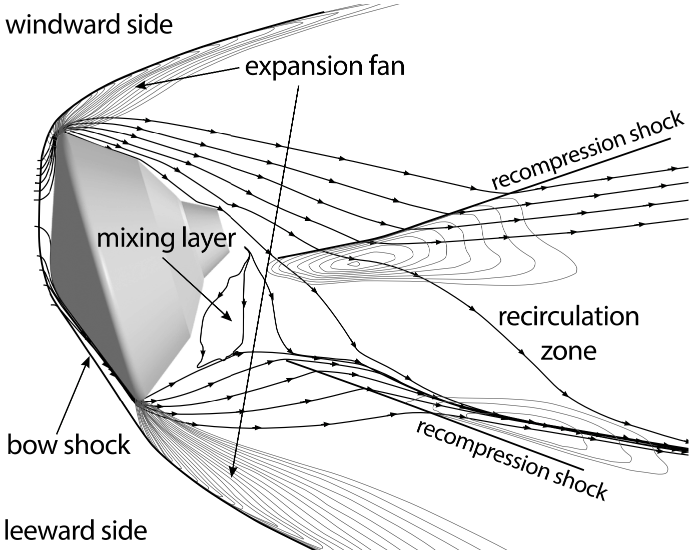


Figure 2.5: Hypersonic flow features on MSL (adapted from [42])

The wake of the vehicle is the region downstream of the forebody within the bow shock and expansion fan. Within the wake, the flow diverted by the forebody travels across the afterbody. Changes in α cause different behavior on the windward and leeward side, with flow separation and reattachment [74]. In the region of separated flow, a mixing layer of lower-velocity, higher-pressure flow exists near the afterbody surface. The flow can meet into a recompression shock with a central recirculation zone.

During the hypersonic regime, the pressure on the forebody is significantly higher than on the afterbody [56]. Pressure forces also dominate over shear forces [42]. Thus, the lift and drag forces and the pitching moment are almost entirely driven by forebody pressure distribution. A negative α produces positive lift through the cosine of the axial force. The magnitude of lift is much smaller than drag, with the MSL entry vehicle achieving a lift-to-drag ratio of 0.24 at Mach 6 [40].

Unsteady effects occur as the vehicle oscillates and changes Mach number, driven by real gas effects. Static aerodynamic instability occurs as the sonic line shifts across the forebody, especially when it interacts with changing geometry at the nose and shoulder [75].

Supersonic Flow

Flight in the supersonic regime begins around Mach 5 to 8 and ends at about Mach 1.2. As the vehicle transitions into the low hypersonic and the supersonic regimes, the wake has an increasing affect on the aerodynamics. Figure 2.6 shows a visualization of the flow in the supersonic regime, demonstrating the extent of the subsonic region of the wake in comparison to the size of the MSL entry vehicle.

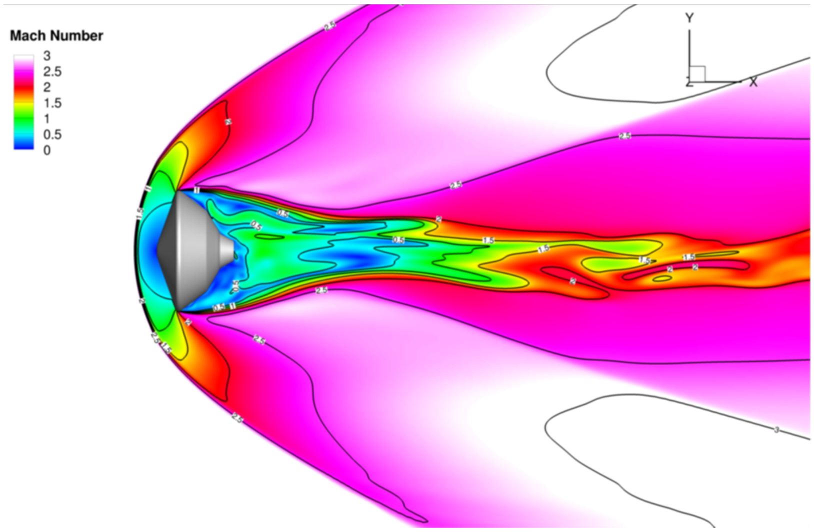


Figure 2.6: MSL supersonic flow visualization [41]

Subsonic recirculation allows for fluid communication across the wake and interaction with the afterbody geometry. Subsonic flow can extend for multiple diameters aft of the vehicle. The vehicle generates turbulence and 3D vorticial structures [9]. Afterbody pressure distribution becomes significant at high supersonic speeds, primarily contributing to the pitching moment. While pressure forces continue to dominate, the shear forces are not negligible [42].

Capsule-shaped entry vehicles are generally known to be dynamically unstable in the low supersonic and transonic regimes [76, 59, 77, 75]. Dynamic instabilities are driven by the wake, particularly at low supersonic speeds [20]. Unsteady flow separation and attachment as the vehicle oscillates continues to affect dynamics. It is important to note that the areas closest to the shoulder where attachment changes have the largest moment arm about the center of gravity.

The aerodynamics of an entry vehicle are inherently unsteady, with highly sensitive flow features. Figure 2.7 shows data from a ballistic range test of the MSL entry vehicle as it decelerates through the supersonic regime. The plot of total angle of attack α_T shows the unsteady interaction between the vehicle's aerodynamics and flight dynamics as it oscillates about its trim angle of attack. Note that the oscillation frequency and trim point are changing with Mach number. Overlaid on the figure is a plot of pressure at a location on the afterbody. The pressure is much more unsteady, operating with a time constant much smaller than the dynamics.

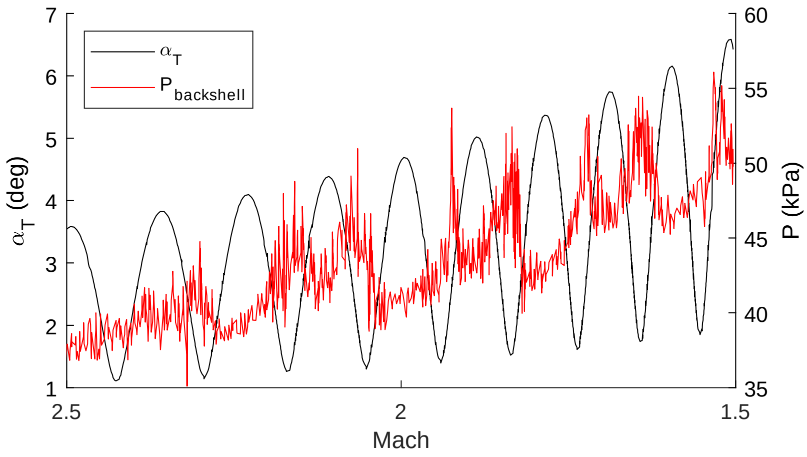


Figure 2.7: MSL ballistic range test (adapted from Schoenenberger et al. [78])

2.2.2 Model Nomenclature

Aerodynamic models are classified in literature based on their properties. The broadest classification is the accounting of vehicle motion relative to the freestream. In a static model, the vehicle does not rotate or accelerate, so it has a fixed attitude with respect to the freestream. In a dynamic model the vehicle is free to move relative to the fluid freestream, in one or more dimensions, over time.

Models can be classified based on time variance of the fluid flow variables. A time-accurate model allows for unsteady flow, where the flow field variables measured at a fixed point can change with time [79]. Conversely, a steady model assumes the flow field variables at a fixed point do not change with time. By necessity the fluid flow in a dynamic model must be time-accurate.

Models can also be classified based on their domain: physical or computational. Physical experiments are performed in the real world, physically manipulating a fluid and vehicle to produce the aerodynamic effects. Computational experiments use a mathematical representation of the fluid and vehicle, using numerical methods to obtain a flow solution. All physical models inherently have time-accurate flow. Thus only computational aerodynamic models can have steady flow[1]. Figure 2.8 provides a taxonomy of models with this nomenclature, including examples of the types of models that can exist.

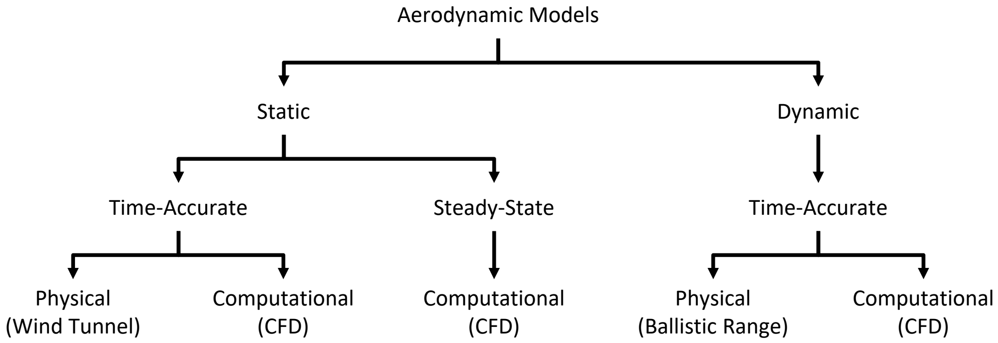


Figure 2.8: Taxonomy of Aerodynamic Models

2.2.3 Static Models

Static aerodynamic models represent the most basic accounting of aerodynamic effects.

Wind Tunnel Experiments

For the Viking missions, launched in 1976, static aerodynamic modeling relied solely on subscale, physical experiments. By using physical fluid flow, these models are inherently time-accurate. Wind tunnel tests were performed with sting-mounted models at 2.75% to

[1] In the computational aerodynamics literature, the word "static" is sometimes used as the opposite of "time-accurate" to refer to static, steady-state models. In order to eliminate confusion, models which can be properly termed "time-accurate static" can be referred to as "time-accurate fixed-attitude."

8% scale, in both air and CO_2 [80, 81, 59]. These tests collected lift and drag forces and pitching moments from Mach 0.4 to 10.5 with angles of attack up to 25°. Wind tunnel tests can collect integrated forces and moments on the model through a sting balance, and pressure ports can be installed on the model surface to record pressure on the surface at key locations, such as the backshell.

Wind tunnel tests of the Orion capsule are illustrative of the measurement methodology [82]. Measurements were taken at 30 Hz and averaged over a two-second window. This time span is longer than the time constants in the flow, so the measurements will not distinguish unsteady aerodynamic effects. The range of these measurements, along with the hysteresis and repeatability, form the basis of the uncertainty [83].

Present-day vehicles continue to use these physical experiments [42, 12, 63]. The experimental methods have remained mostly unchanged, except for updates in measurement systems and the introduction of refinements such as pressure sensitive paint [84].

Steady Computational Fluid Dynamics Models

For the vehicles that followed Viking, physical experiments have also been supplemented by computational experiments. The flow of viscous, Newtonian fluids is described by the Navier-Stokes equations [72]. The Navier-Stokes equations are written using the arbitrary Lagrangian-Eulerian (ALE) formulation in Equation 2.9 [85, 86]:

$$\frac{\partial}{\partial t}\int_{\mathcal{V}} \boldsymbol{q} d\mathcal{V} + \oint_{\partial\mathcal{V}} (\boldsymbol{F}^{*} - \boldsymbol{F}_{\mathcal{V}}) \cdot \hat{n} dS = 0 \tag{2.9}$$

where $\boldsymbol{q}$ represents the conserved variables and $\mathcal{V}$ is the control volume, bounded by control surface $\partial\mathcal{V}$, with local control volume face velocity $\bar{W}$. $\boldsymbol{F}^{*}$ and $\boldsymbol{F}_{\mathcal{V}}$ are the convective and diffusive fluxes of $\boldsymbol{q}$, respectively. The flux through a moving control volume must account for augmented or reduced flux through the control surface due to the local control surface speed:

$$\boldsymbol{\mathcal{F}}^{*} = \boldsymbol{\mathcal{F}} - \boldsymbol{q}\bar{W}^{T} \tag{2.10}$$

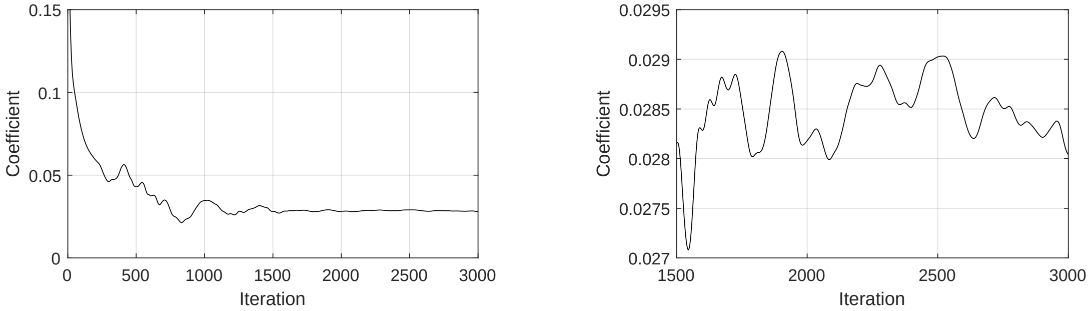
The ALE formulation allows for the prediction of unsteady aerodynamics as the vehicle moves in time. A computational fluid dynamics (CFD) flow solver is used to solve these equations. The volume is spatially discretized into a grid, with the convective fluxes computed using a flux-splitting scheme [87]. Time-accurate models use temporal discretization in both physical and pseudophysical time steps [88]. Steady calculations are achieved by only advancing the pseudophysical time step.

For Pathfinder in 1996, CFD was used to calculate the pressure distribution over the forebody [75]. Designers for each subsequent vehicle have made greater use of CFD as flow solver capabilities have improved. For each vehicle since Pathfinder, CFD flow solvers have been the primary modeling tool for static aerodynamics [89, 56, 42]. Multiple flow solvers are used covering different flight regimes and areas of flow [40]. Validation of the static aerodynamic model relies on physical experiments and the close heritage of each capsule [42, 12, 63].

For computational experiments, the measurement error is based on the variability of the coefficients by iteration [11]. Figure 2.9 demonstrates this variability: as the coefficients converge with increasing iterations, the change in the value of the coefficient (and in the residuals) each iteration will decrease until it meets some threshold where the solution is considered to be converged. However, the coefficients will continue to change with each iteration due to numerical precision. The uncertainty also incorporates differences between the results from different CFD codes, the use of different flow solver settings, and comparison to wind tunnel data.

2.2.4 Dynamic Models

Entry vehicle development has primarily relied on physical experiments to characterize dynamic behavior.



(a) Solution convergence

(b) Variation of converged solution

Figure 2.9: Steady-state coefficient solution history

Ground-based Experimentation

The Viking program used ballistic range and forced oscillation experiments to characterize the dynamic behavior of the capsule, namely the pitch damping coefficient [90, 91]. Forced and free oscillation tests are performed in a wind tunnel with the model on a pivot. This provides one degree of freedom (1DOF) of angular motion, usually either in pitch or yaw.

Ballistic range tests model the free-flight behavior of scale models shot from a gun into quiescent fluid [92]. A ballistic range model is free to move in six degrees of freedom (6DOF) for full translation and rotation. The model is measured from fixed stations as it flies down range, capturing the vehicle trajectory. Modern ballistic range tests can also measure trajectory with an inertial measurement unit (IMU) embedded in the model [93]. Ballistic range tests eliminate interference due to a wind tunnel sting balance, but the duration is limited by the length of the range and the drop-off of the model due to gravity. The models also generally use smaller scales than the static experiments; the Viking ballistic range models used models at a scale of only 0.29% [90] compared to 2.75% for the smallest wind tunnel tests.

Reconstruction techniques such as parameter estimation are used to determine the aerodynamic characteristics that would fit the trajectory [94]. Parameter estimation (or identification) is the process of fitting an aerodynamic surrogate model from a trajectory [95]. A 6DOF flight dynamics model is constructed by integrating the equations of motion shown in Equations 2.1-2.4. The external forces and moments are given a functional form using aerodynamic coefficients. This model is fitted to the trajectory data, using the aerodynamic coefficients as the unknowns. The aerodynamic model can then be used as a surrogate (details of the model form are discussed in Section 2.4.3). Parameter estimation is commonly used to construct an aerodynamic model from ballistic range tests [96, 97, 78, 98]. In these cases, the damping coefficients may be used in an aerodynamic database. Measurement error for ballistic range tests includes error in the graphical determination of the position and orientation of the model at each station [96]. Reducing the trajectory measurements to aerodynamic coefficients also introduces model fit error [94].

Since Viking, innovation in dynamic testing has enabled 6DOF motion in wind tunnels. A magnetic balance can be used to suspend the model for motion subject to fluid flow, gravity, and the magnetic field [99]. Free-flight drop tests release a stationary model to fall through a transverse flow (akin to an inverse of the ballistic range) [100]. These experiments yield substantially similar results to ballistic range tests, albeit with differences in cost and duration.

Flight Testing

Flight testing measures the characteristics of the vehicle while in real-world flight conditions. For vehicles like fixed-wing aircraft, these tests might use the actual vehicle flying in the design flight regimes, but for Mars entry vehicles, full-scale flight testing is infeasible. Compromises are often made to use a subscale or boilerplate vehicle, or to fly through only a subset of the design flight regimes. Flight tests have been performed for new decelerator technologies, such as the Supersonic Inflatable Aerodynamic Decelerator (SIAD) [101] and the Inflatable Reentry Vehicle Experiment (IRVE) [102], though these tests entered at suborbital speeds, not from orbital or interplanetary speeds. Furthermore, these tests can only be done on Earth, which has different atmospheric properties from Mars.

The only ’flight tests’ for Mars entry vehicles are the actual flights at Mars. The Viking landers, MSL, and Mars 2020 carried sensors to directly measure the aerodynamic environment around the vehicle [103, 104, 10]. This data has been used to validate Earth-based experimentation [105] and provide correction factors for forebody-only CFD experiments [42].

Coupled CFD-RBD Models

Up through the development of MSL, modeling dynamic behavior still relied primarily on physical experiments and historical data [98]. However, several factors are driving the investigation of time-accurate CFD simulations as suitable replacements: the desire to reduce experimental costs, declining availability of facilities, increases in CFD flow solver fidelity, and increasing computational power [106, 107].

Time-accurate simulation is achieved by the flow solver executing dual time stepping: the outer time step advances the simulation forward in time, while the inner time step. The continuous integration of the flow field preserves flow history, which means that unsteady aerodynamic effects are modeled.

While these simulations can be run with the vehicle state held constant, dynamic simulations can be executed by including vehicle motion. Dynamic, time-accurate CFD models are created by coupling the flow solver with a rigid body dynamics (RBD) simulation [67, 108, 109]. The governing equations for fluid flow are solved simultaneously with the equations of motion as the simulation steps forward in time. The physical time step of the RBD model is chosen to be equivalent to (or an integer multiple of) the outer time step of the CFD model. Figure 2.10 illustrates the high-level operation of the model. Aerodynamic forces and moments calculated by the CFD are used to drive the motion of the vehicle, which in turn changes the fluid flow around the vehicle.

The CFD flow solver must incorporate linear and angular velocity terms in the governing equations for time-accurate simulation in a non-inertial frame. A number of flow solvers incorporate this capability, either strictly through rigid mesh movement or by gen-

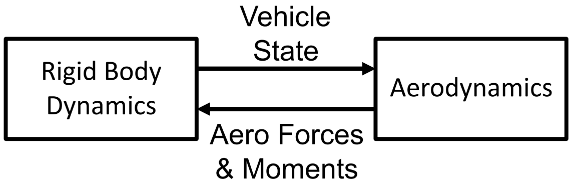


Figure 2.10: CFD-RBD flowchart

eralized mesh movement and deformation. Accurately resolving the flow requires high mesh density both near the vehicle surface and in the wake. State-of-the-art grids for entry vehicles can have on the order of 20 million cells [41, 110]. The simulations must use a very fine discretization in time to achieve time-accurate fluid flow. As a result, coupled CFD-RBD runs for entry vehicles are typically less than one second in simulated duration. However, advances in computing power and flow solver capabilities are continuously increasing the feasibility of CFD-RBD simulation.

The use of free-flight CFD-RBD simulations is prevalent in other disciplines, such as the analysis of aircraft stores and projectiles. Store separation relies heavily on CFD-RBD simulation to determine whether a projectile or other object released by an aircraft might be at risk of recontact [111]. Meakin developed some of the earliest simulations for time-accurate Navier-Stokes coupled with 6DOF for modeling store separation [112].

CFD-RBD simulations are used regularly in the field of subsonic and supersonic projectile design. Coupled simulations described as “virtual fly-out” have been used to estimate aerodynamic coefficients [113, 114, 115]. In particular, they are useful for determining pitch damping, roll damping, and magnus coefficients [116, 117]. Similarly to ballistic range tests, parameter estimation has also been used to extract data from CFD-RBD simulations [118, 110, 114]. These simulations have the advantage of directly measuring forces and moments, making them available to be used in the parameter estimation.

In the field of aircraft and rotorcraft design, the CREATE program has led to the development of a high-fidelity aircraft design toolset including Kestrel: a multidisciplinary, multiphysics flight simulation tool[119, 120, 121]. Morton et al. [122] demonstrate Kestrel’s

CFD-RBD capability linked to the kAVUS flow solver for a gravity munition. Roget et al. [123] demonstrate a similar CFD-RBD capablity in CREATE-AV Helios for rotorcraft.

Murman and Aftosmis modeled entry vehicles in CFD-RBD simulations with 1DOF pitch oscillation [124]. Murman also expanded this work to simulate an inflatable decelerator and the Orion crew module [125]. Their CFD-RBD models were developed with the OVERFLOW solver, using overset grids for translation and rotation [126]. These 1DOF simulations are the equivalent to the free- and forced-oscillation wind tunnel tests. Through this and subsequent work, Murman demonstrated the sensitivity of CFD predictions to the pitch rate of the vehicle [125].

Expanding on the 1DOF simulations, 6DOF simulations model flight dynamics in translation and rotation. The 6DOF simulations are a virtual equivalent to ballistic range experiments. Stern et al. [108] constructed a CFD-RBD model using the US3D flow solver, which they used to predict dynamic properties of the MSL entry vehicle. This model was also used to predict surface pressure at sample points on the afterbody in order to provide a comparison to flight instrumentation [41]. Brock et al. [110] used the same US3D-based model to simulate the ballistic range tests for the Supersonic Flight Dynamics Test (SDFT). These models used detached eddy simulation (DES) [127] to accurately resolve the turbulence behavior for the subsonic wake. They were able to achieve excellent agreement with the experimental trajectory and aerodynamic data. The US3D framework uses grid deformation to achieve the angular degrees of freedom. An inner region of the grid is rotated with fixed orientation to the surface, with an algebraic grid deformation transitioning to the fixed outer region. This model has been validated for multiple entry vehicles [110, 128]

2.2.5 Observations

Between the various fundamental models for static and dynamic aerodynamics used in the state-of-the-art methodology, the physics of unsteady fluid flow are accurately modeled. The ballistic range experiments on their own provide a model that only significantly differs from flight in terms of the scale and atmospheric conditions. The difficulty arises in extracting information from this model. Measurement of the trajectory effectively provides a measurement of the integrated forces and moments, from which the instantaneous aerodynamics must be calculated, incurring uncertainty propagation.

For this reason (among others such as cost), the ballistic range tests are supplemented by the static models. Wind tunnel tests are easier to measure, but the measurement techniques average out any unsteady fluid behavior and include the variation in the uncertainty. CFD is used as the primary model for steady-state aerodynamics, though CFD is capable of modeling time-accurate flow. Thus, the unsteady fluid behavior is inherently modeled, but not sufficiently measured.

Observation 2.1: The fundamental aerodynamic models capture unsteady fluid behavior, but model choice and the measurement process cause a reduction in fidelity.

As a result, the state-of-the-art methodology lacks instantaneous measurement for unsteady fluid flow around an unsteady vehicle. However, in the same way that physical, static aerodynamic models have been augmented by the use of CFD, research is ongoing to augment physical, dynamic models with dynamic CFD models. This is due to two primary factors: their recent introduction (and the need to validate these models), and their computational cost. The execution time for a CFD-RBD simulation may be orders of magnitude higher than the cost of a static CFD solution.

Observation 2.2: Numerical, dynamic entry vehicle aerodynamic (CFD-RBD) models exist, but are not yet used in the state-of-the-art methodology due to their computational cost.

2.3 Control System Models

The control system model provides an estimate of the effect of the control system on the vehicle. This is found as a sum of forces and moments. The total contribution of an RCS jet to the vehicle moment is modeled according to Equation 2.11:

$$\bar{M}_{jet,net} = \dot{m}_{jet} s_{jet} \bar{r}_{jet} \times \hat{n}_{jet} + \bar{M}_{jet,interaction} \tag{2.11}$$

Similarly, the force is:

$$\bar{F}_{jet,net} = \dot{m}_{jet} s_{jet} \hat{n}_{jet} + \bar{F}_{jet,interaction} \tag{2.12}$$

where $\dot{m}_{jet}$ and s_{jet} are the jet's mass flow rate and exhaust speed, respectively. $\bar{r}_{jet}$ is the moment arm of the jet about the center of mass and $\hat{n}_{jet}$ is the thrust direction vector. This first term is the contribution due to the momentum of the exhaust gasses, which is easier to model since it is relatively insensitive to the aerodynamics. Therefore this discussion will primarily be concerned with modeling the interaction term. The second term, $\bar{M}_{jet,interaction}$, is the contribution due to the interaction between the jet plume and the vehicle aerodynamics. This is defined as:

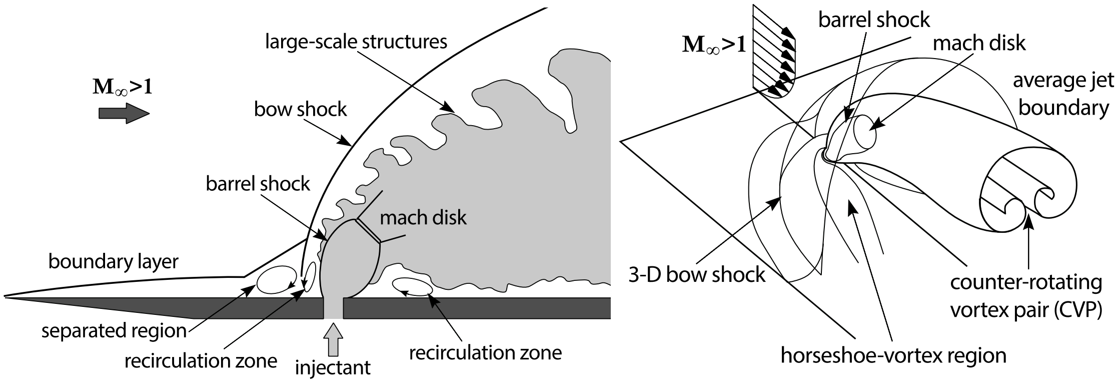
$$\bar{M}_{jet,interaction} = \bar{M}_{jet,on} - \bar{M}_{jet,off} \tag{2.13}$$

where $\bar{M}_{jet,on}$ and $\bar{M}_{jet,off}$ are the net aerodynamic moments with the jet on and off, respectively (note that this is a summation over the entire vehicle, including the nozzle plenum). For an aerodynamic control surface, the interaction would be the only term, since there is no net momentum transfer. Since the control surface is not removed from the vehicle, the equivalent moments would be for the control surface in a deflected versus neutral position.

2.3.1 RCS/Aerodynamic Interaction

The direct effect of the RCS jets is typically only a small fraction of net aerodynamic moments [40]. As such, their authority in pitch and yaw is limited to rate damping. However, the plume of an RCS jet interacts with the aerodynamic flow field around the vehicle. Whereas the direct effect of the RCS is calculated by the pressure distribution and momentum transfer from within the nozzle, jet interaction causes changes to the pressure distribution over the rest of the vehicle surface [12].

Figure 2.11 illustrates the flow features of an RCS jet interacting with attached, supersonic flow. Even though the jets are located on the afterbody of the vehicle, the high angle of attack of the freestream means that the windward side may still experience attached flow. Entrainment and recirculation near the exit of the jet causes a local decrease in the surface pressure distribution [12]. Horseshoe shocks can cause an area of higher-pressure stagnated flow upstream of the thruster. The jet flow lifts up the oncoming stream, increasing the extent of separation. For jets that are more parallel to the surface, impingement can cause a local increase in the pressure distribution. The effluent can interact with the vorticial structures already present in the wake [9]. These effects are as dependent on the flow history of the vehicle as on the current state. Interactions can have different behavior in attached and separated flow.

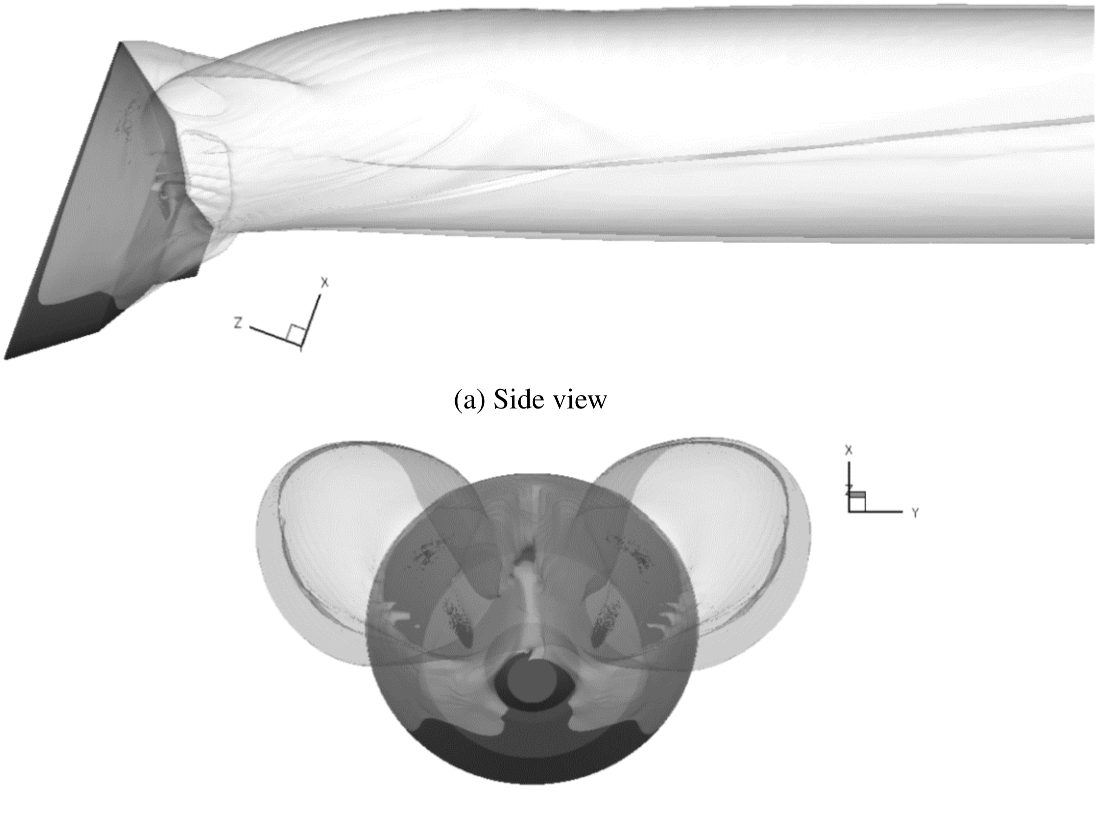


(a) Side view

(b) Orthographic view

Figure 2.11: RCS interaction features in attached flow (adapted from Ben-Yakar et al. [129])

Figure 2.12 shows how the exhaust can affect a large portion of the afterbody surface since the wake is subsonic and the jets are under-expanded. The jet can also cause flow separation and reattachment. These global changes can affect the surface pressure distribution in ways that are very sensitive to freestream conditions and attitude. Figure 2.13 shows how the surface pressure changes when the jets are turned on. In addition to the net changes, the plume recirculation can impact the surface on the opposite side of the capsule.



(a) Side view

(b) Aft view

Figure 2.12: Iso-surface of 1% exhaust in MSL wake with leeward engines firing. Mach 2.5, 19.7° angle of attack (adapted from Dyakonov et al. [12]).

For historical Mars entry vehicles, the aerodynamic moment caused by the interaction can be on the same order of magnitude as the direct RCS effect. Depending on the pressure distribution and moment arm on the vehicle, the interaction could yield increased or reduced effectiveness, or even control reversal. The interaction can also be cross-channel (e.g., actuation of the pitch jets could cause a yaw moment). Since the RCS interaction af-

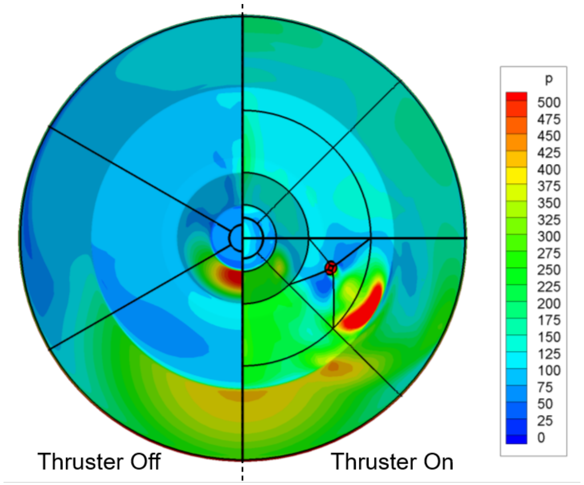


Figure 2.13: RCS interaction surface pressure [20]

fects the afterbody pressure distribution, its impact is greatest when the afterbody pressure is highest: during high dynamic pressure. This is also when aerodynamic uncertainty is most significant to landing uncertainty, and when entry guidance is most effective.

The RCS design for MSL went through four major iterations due to persistent challenges with control/aerodynamic interaction [63]. Design deviation from the Viking entry vehicle heritage OML and RCS was sufficient to require extensive validation using both physical and computational experiments [40]. The resulting interaction uncertainty was large enough to raise concerns about negation of control authority and excess aerothermodynamic heating.

Control System Operation

The mode of control system operation also contributes to the unsteady behavior of aerodynamics and control system interaction. RCS jets are operated in short-duration pulses, with a discrete startup time and commanded duration. These pulses can cause sudden changes in pressure distribution [9]. Jet thrust is not a perfect step response; Figure 2.14 plots the thrust response for an Apollo capsule RCS thruster, demonstrating the unsteady characteristics of a jet pulse [130]. Starting from the time t_{on} when the jet is commanded to activate, the thrust can have a rise time t_{rise} to nominal thrust and a fall time t_{fall} after commanded shutoff time t_{fall}. Thrust might overshoot the nominal value on startup, and trail off after shutoff. Figure 2.15 shows the activity of the roll RCS during MSL entry. MSL rcs had most pulses between 50 and 150 milliseconds, clustered around the bank reversals [54].

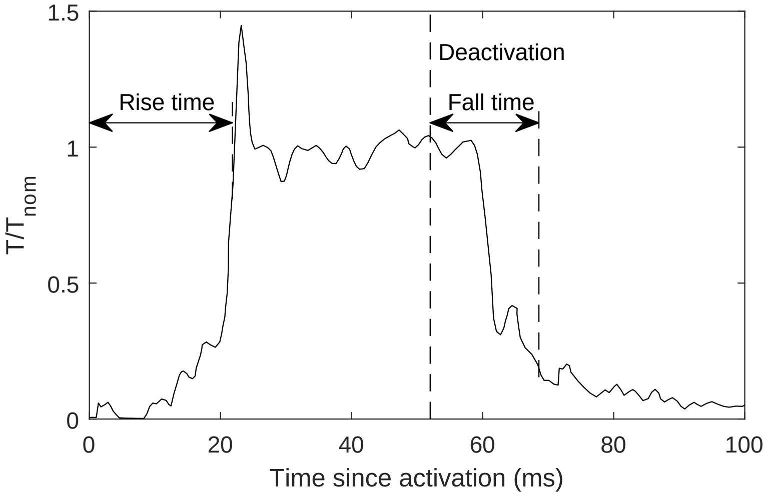


Figure 2.14: RCS thrust response (adapted from Foote et al. [130])

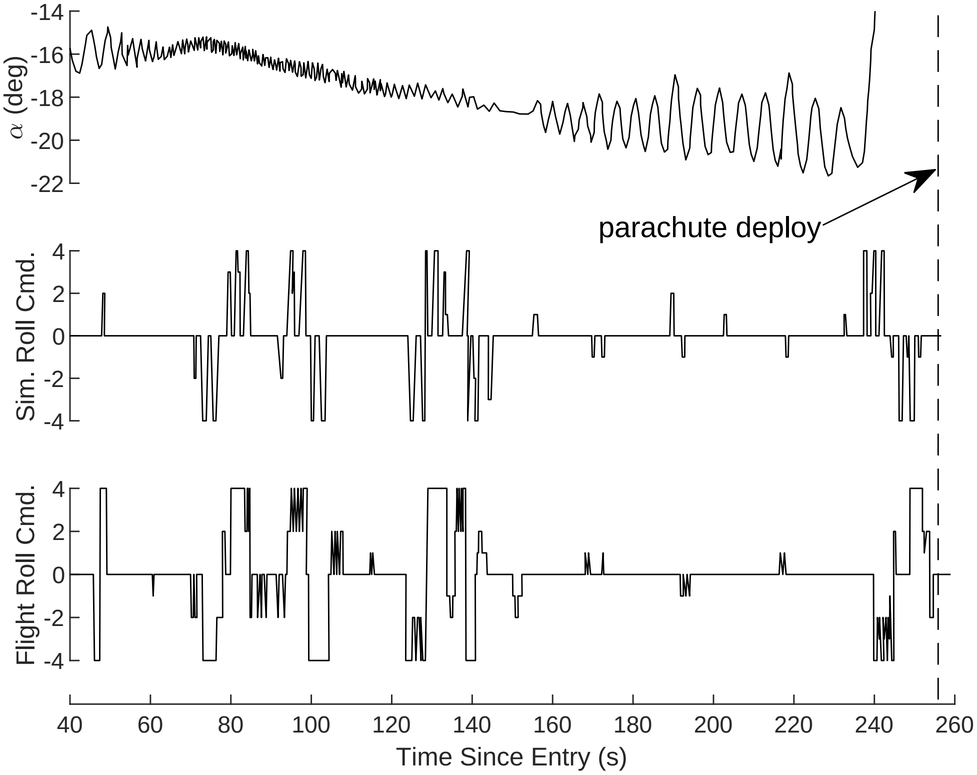


Figure 2.15: Predicted and actual MSL roll RCS activity during entry (adapted from Schoenenberger et al. [54])

2.3.2 Static Models

The definition of the interaction in Equation 2.11 is indicative of how it is modeled: the aerodynamic forces and moments are measured with the RCS jet (or combination of jets) on, and the difference is taken with the measurements taken when the jet is off.

Wind Tunnel Experiments

Wind tunnel tests for RCS interaction are performed with scaled jet nozzles integrated into the model geometry [131]. Simulated exhaust gasses are fed through the sting balance to be ejected through the scaled nozzles (hence ballistic range tests and magnetic levitation tests are infeasible). Concessions must also be made to scalability and manufacturability for these tests. Wind tunnel tests for the MSL RCS used simplified capsule outer mold line (OML) geometry [63]. The nozzles were simplified to a cylindrical throat and axisymmetric conical nozzle, resulting in a change of area ratio. The scaling of the nozzles was chosen to match the plume shape and momentum ratio expected at flight conditions, but at a different exit Mach number. Additionally, pure Nitrogen gas was used as the exhaust for the nozzles instead of Hydrazine combustion products. Modifying the sting balance to maintain the desired fluid conditions of the N_2 resulted in increased interference with the capsule body [63].

Static CFD Experiments

Since the exhaust from an RCS jet is also fluid flow, it can also be modeled using CFD. RCS jets are modeled as an inflow boundary condition to the volume. This boundary is placed in a location where the flow properties of the jet are known and well-behaved, such as the nozzle plenum [12]. The pressure, temperature, mass flow rate, and chemical composition of jets might be specified to match flight conditions [12] or to match wind tunnel tests for validation [63].

The Phoenix lander was the first entry vehicle for which CFD was used to predict the interaction between RCS and the aerodynamics [11]. Analysis of the Phoenix RCS relied on computational techniques and some ground testing from MSL development. Steady-state CFD solutions were calculated for Phoenix with RCS on and off in the rarified, hypersonic, and supersonic regimes. The desire to model RCS drove the need to include afterbody and vehicle wake within the flow volume [9]. Analysis in the hypersonic regime was performed using LAURA, a 3D Navier-Stokes flow solver that can solve laminar and turbulent flows with thermochemical nonequilibrium [132]. Solutions in the supersonic regime were made with FUN3D, another 3D Navier-Stokes flow solver [133]. Due to time constraints, there were no purpose-built physical experiments for RCS validation [11].

Figure 2.16 shows the iteration history for afterbody moment in the presence of the pitch jets, compared to the baseline afterbody moment and the ideal jet authority. The flow in the wake is unsteady, with resulting oscillation in moment coefficient. This oscillation is due to numerical effects as the solution converges.

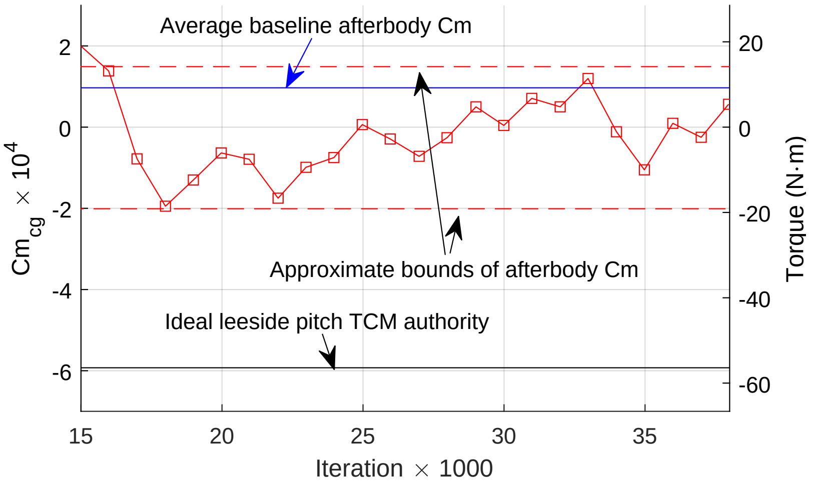


Figure 2.16: LAURA CFD iteration history of Phoenix afterbody moment, Mach 18.8 (adapted from Dyakonov et al. [11])

The variability by iteration is significant compared to the ideal authority of the jets, especially in yaw. Figure 2.17 shows the interaction in yaw for the FUN3D solutions at Mach 3, near peak dynamic pressure, with error bars representing the solution variability.

Over the entire range of sideslip angles simulated, the error bars are significant compared to the ideal moment of the yaw thrusters. At sideslip angles of 0° and 2°, the mean coefficients predict a control deficit larger than the yaw jet authority. This would result in control reversal, where a commanded yaw in one direction would result in motion in the opposite direction. The CFD solutions also predicted cross-channel interaction, between pitch and yaw and between roll and pitch/yaw.

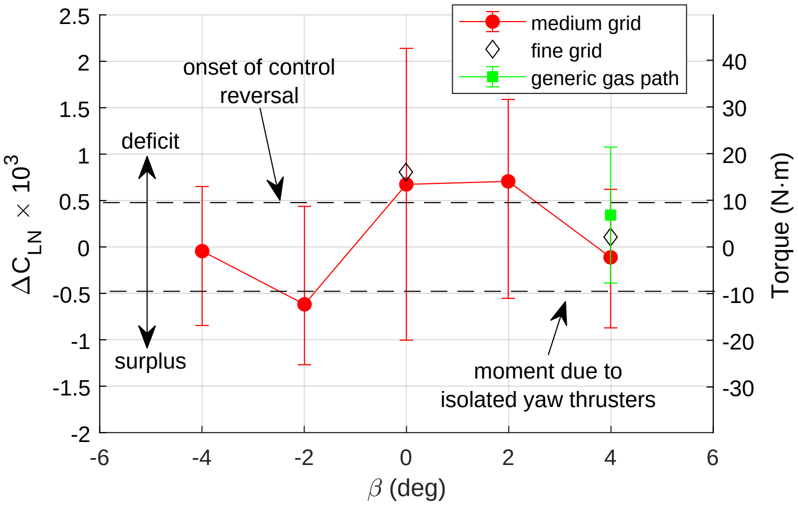


Figure 2.17: FUN3D prediction of Phoenix yaw RCS interaction at Mach 3 (adapted from Dyakonov et al. [11])

The MSL program used CFD as the primary tool for RCS development, with wind tunnel tests for validation [63]. CFD experiments were able to ameliorate the scalability and manufacturability issues with physical experiments. RCS jet configurations were evaluated with the LAURA, DPLR, FUN3D, and OVERFLOW grids [12]. However, the CFD flow solvers used were still subject to shortcomings in model fidelity for gas chemistry and wake flows. Schoenenberger observes that the interaction results were sensitive to changes to the grid and numerical schemes [63].

2.3.3 Dynamic Models

Dynamic models for RCS interaction have not been used for previous Mars entry vehicles. Purpose-built physical, dynamic models for RCS interaction are not constructed for entry vehicles due to the manufacturability and scalability issues noted above for wind tunnel tests. Due to the small scale of the physical models, the propellants cannot be stored on-board; jets must be fed through a mounting point, as is done with wind tunnel RCS testing. The results from flight data are limited to "qualitative assessments" [54] for RCS performance due to the high measurement uncertainty.

However, CFD simulations are capable of modeling jets in time-accurate flow. Korzun et al. [134] modeled jets using time-accurate CFD to study supersonic retropropulsion for future Mars entry vehicle concepts. Given that Dyakonov et al. [9] found that the jet interactions were strongest when the exhaust direction was pointed against the oncoming flow, the success in modeling time-accurate jets in these conditions supports its use in modeling RCS jets in crossflow.

CFD-RBD-FCS Models

The "virtual fly-out" method introduced in Section 2.2.4 used for projectiles has been expanded to include control system modeling. Sahu [135] simulated a rolling projectile with open-loop control of a synthetic jet at the base of a projectile. The numerical simulation demonstrated that the jet interacted with the separated flow region resulting in a change to the dynamic behavior. Sahu et al. [136] investigated the same interaction for a jet in crossflow.

Sahu et al. [137] developed the capability to add a flight control system model into the CFD-RBD simulation, which was demonstrated by simulating a canard-controlled projectile during closed-loop roll and crossrange control maneuvers. Canard deflection was modeled using overset or "chimera" meshing. Sahu and Fresconi validated simulations of closed-loop roll control [138] and open-loop pitch control [139] against wind tunnel experiments.

Sahu and Fresconi [140, 118] used the CFD-RBD-FCS model to investigate the development of flight control algorithms. A canard-controlled projectile was simulated with open-loop control to generate aerodynamic data for parameter estimation. The simulations captured interactions between the vortical structures shed from the canards on the fins. The controller was then validated using closed-loop three-axis control simulations.

The capabilities of Kestrel have also been expanded to include control surface deflection through mesh deformation [121, 141].

2.3.4 Observations

The most salient observations about control interaction models is that whole areas of modeling are omitted:

Observation 2.3: The state-of-the-art methodology does not include dynamic control system interaction or time-accurate control system actuation.

As noted in the above section, this deficit in physical experimentation is due to limitations in manufacturability. As with the aerodynamic models, this suggests exploring computational experimentation.

Examining the state-of-the-art modeling methods for control system interaction and comparing it to the methods described in Section 2.2, it is interesting to note that the models for aerodynamics and for control system interaction are the same. This is true in regards to their actual utilization as well as to their capabilities and limitations. Comparing the static models, the values are found from steady solutions, with time-accurate effects accounted for in the model uncertainty. Indeed, comparing CFD-RBD simulations to CFD-RBD-FCS simulations shows that the only difference in these numerical models is the inclusion of logic to drive the control system (i.e. guidance and controller modeling), and the exercising of a capability that already exists in CFD to model inflow boundary conditions. The literature demonstrates this for projectiles, leading to the observation about applying this method to entry vehicles:

Observation 2.4: It is possible to expand entry vehicle CFD-RBD models to include guidance and control system modeling.

2.4 Surrogate Modeling

While the fundamental disciplinary models for aerodynamics and control systems have been identified, these models are not used directly in the flight simulation framework. That would be both cost- and time-prohibitive, and each individual model may not cover the entire regime required for the flight simulation.

Instead, surrogate models must be constructed to combine the disparate models identified in the above sections. Queipo et al. [142] define a surrogate model as "[the] inverse problem for which one aims to determine a continuous function of a set of design variables from a limited amount of available data." Giselle Fernandez-Godino et al. [143] note that "[m]ost surrogates are algebraic models that approximate the response of a system based on fitting a limited set of computationally expensive simulations in order to predict a quantity of interest." Given the wide range of definitions found in the literature, the term 'surrogate' as used in this dissertation will be defined thusly:

Definition 1: Surrogate Model: An approximation that predicts the response of a model continuously over the design space of the input variables, that is trained or fitted from a limited number of executions of the original model.

Instead of evaluating the true response y, a surrogate model calculates the predicted response $\tilde{y}$. The use of these surrogates allows for rapid evaluation of the disciplines during flight simulation. Whereas the measurement of a single data point in a wind tunnel experiment might take between 2-30 seconds [144, 82], evaluation of a surrogate model might take less than a millisecond. Factoring in the overhead for setup, the aerodynamic and control interaction surrogates might achieve a reduction of at least 4-6 orders of magnitude.

A wide variety of surrogate model forms have been developed and applied to aerodynamic modeling [142]. Hiller [145] identifies a set of classifiers that can be applied to surrogate models:

> Intrusive vs. Nonintrusive: ... Intrusive methods involve manipulation of a system's governing equations in identifying its dynamics, whereas nonintrusive methods rely on system identification techniques to replace the original computational model with a smaller size system.
>
> Parametric vs. Nonparametric: A parametric method assumes an explicit functional form for the model of a system's behavior and proceeds to define the coefficients of that particular model. A nonparametric method uses input-output mappings to develop the best functional representation of a system.
>
> Linear vs. Nonlinear: Linear methods assume that a system's response is linearly dependent on the system's inputs. Nonlinear methods assume a nonlinear dependency on the inputs with the potential for cross-coupling of terms. The distinction between linear and nonlinear models exists for both parametric and nonparametric methods.

The following sections discuss the surrogates used in the state-of-the-art entry vehicle models, and identifies alternatives that are used in similar aerospace design problems.

2.4.1 State-of-the-Art Models

Aerodynamic data from the disparate modeling sources is compiled into a single surrogate model for use during development. The most common form of surrogate used for entry vehicles is a database of look-up tables [89, 56, 42, 58]. Databases are nonintrusive and nonparametric, and, depending on their dimensionality, they may be nonlinear.

Aerodynamic Database

Table 2.1 demonstrates the general format of a database, where each row is an experimental design point. The vehicle attitude is given in terms of angle of attack α and sideslip angle

β, or in terms of the total angle of attack α_T. In continuum flow, vehicle speed is given either in absolute magnitude or by local Mach number M. The dependent atmospheric properties are given in terms of altitude h or density ρ. The aerodynamic characteristics are presented as coefficients for axial force (C_A), normal force (C_N), pitching moment about a reference point (C_M), and pitch damping (C_{M_q}). The database may include yaw moment and damping coefficients (C_{LN} and C_{LN_r}), or if the capsule geometry is approximately axisymmetric, the aerodynamics in pitch and yaw may be assumed to be identical [42, 56]. The damping coefficients usually calibrated to non-dimensional angular rates, with the non-dimensional pitch rate $\hat{q}$ calculated as an example:

$$\hat{q} = \frac{q\, l_{ref}}{2s} \tag{2.14}$$

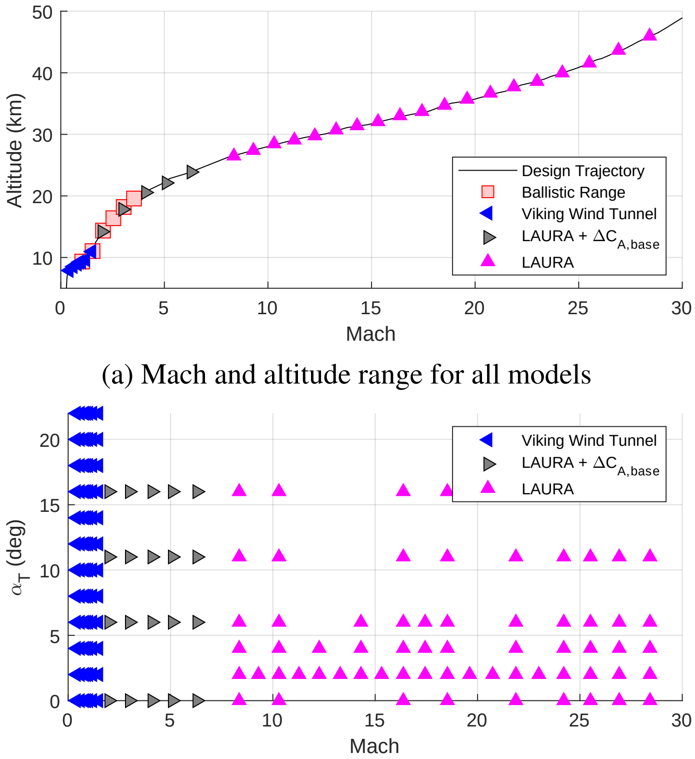
Table 2.1: Notional entry vehicle aerodynamic database

M	h	α	β	C_A	C_N	C_M	C_{m_q}
x	x	x	x	x	x	x	x
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x	x	x	x	x	x	x	x

Interpolation is used to calculate the aerodynamic coefficients at off-design points. An efficient design of experiments (DoE) can be used to populate the database for the expected flight regime with the minimum computational and physical cost. Figure 2.18 plots the population of design points for the Phoenix aerodynamic database [56, 80]. Figure 2.18a shows how the design points are placed along the design trajectory. Information from physical and computational experiments are used together to populate the database along the trajectory. This is a basic form of data fusion, where the sources do not overlap. Different models are used with different experimental designs for different flight regimes. Note also that the static database includes data from Viking, which has a different OML to Phoenix.

The dynamic coefficients are a special case: these values are already calculated from a surrogate model. Parameter identification was used to fit a parametric functional form for the aerodynamic coefficients based on the trajectory results [146]. The polynomial fit for the damping coefficient was evaluated at the desired points to populate the database [56]. Though this surrogate also produces static coefficients, these are not used in the overall surrogate model [146].

Figure 2.18b shows the sweep in α_T at each of the Mach conditions. Experiments are performed at each of these conditions for a sweep of α_T. The bounds of α_T are chosen based on the vehicle's trim angle of attack.



(a) Mach and altitude range for all models

(b) Mach and α_T range for static models

Figure 2.18: Aerodynamic model design conditions for Phoenix database [56]

In the context of entry vehicle aerodynamic models, the uncertainty is delivered as a set of bounds on the aerodynamic coefficients. The Aerodynamic uncertainty is itself a model, with inputs from the vehicle state. Schoenenberger et al. document the static aerodynamic uncertainty model for MSL in [40]. Uncertainty for the aerodynamic coefficients is implemented as a set of "adders" and "multipliers" as a function of Mach or Knudsen number. The magnitudes of these values represent the 3σ limits of the nominal values, where σ is the standard deviation. The static aerodynamic coefficient uncertainty model for MSL is shown in Table 2.2. The uncertainty of the axial force coefficient is given as a single multiplier, and the other force coefficients include an adder. The pitch and yaw moment coefficients use adders and multipliers, but the roll moment uncertainty is given as a constant adder applied to the roll torque. MSL was the first Mars entry vehicle to have quantified roll moment dispersions [40].

Table 2.2: MSL static aerodynamic coefficient uncertainties [40]

Regime	C_A	C_N, C_Y	C_M	C_{LN}	C_{LL}
$\text{Kn} > 0.1$	$\pm 5\%$	$\pm 0.01, \pm 10\%$	$\pm 0.005, \pm 20\%$	$\pm 0.005, \pm 10\%$	0.0005
$M > 10$	$\pm 3\%$	$\pm 0.01, \pm 10\%$	$\pm 0.006, \pm 20\%$	$\pm 0.003, \pm 10\%$	0.000219
$M < 5$	$\pm 10\%$	$\pm 0.01, \pm 10\%$	$\pm 0.005, \pm 20\%$	$\pm 0.005, \pm 10\%$	0.00023

Uncertainties are applied to the coefficients in the Body frame. When the model is used, a random value is chosen within the bounds from a Gaussian distribution, which is used throughout the simulation. The dispersion force coefficients are found using the following equations:

$$C_{A,disp} = C_{A,nom}(1 + U_{C_A}^M) \tag{2.15}$$

$$C_{Y,disp} = [C_{Y,nom} + U_{C_Y}^A](1 + U_{C_Y}^M) \tag{2.16}$$

$$C_{N,disp} = [C_{N,nom} + U_{C_N}^A](1 + U_{C_N}^M) \tag{2.17}$$

where $U_{C_x}^A$ and $U_{C_x}^M$ are the adder and multiplier terms. The dispersion of the moment coefficients includes the uncertainty in the moments, as well as accounting for the offset in C.G.

$$C_{M,cg,disp} = [C_{M,MRP} + \frac{\Delta x}{d} C_N - \frac{\Delta z}{d} C_A + U_{C_M}^A](1 + U_{C_M}^M) \tag{2.18}$$

$$C_{LN,cg,disp} = [C_{LN,MRP} + \frac{\Delta x}{d} C_Y + \frac{\Delta y}{d} C_A + U_{C_{LN}}^A](1 + U_{C_{LN}}^M) \tag{2.19}$$

$$C_{LL,cg,disp} = \frac{\Delta y}{d} C_N - \frac{\Delta z}{d} C_Y + U^A_{C_{LL}} \tag{2.20}$$

Schoenenberger et al. document the dynamic coefficient uncertainty model in [98]. In the supersonic regime, the equations for pitch and yaw damping coefficient take the form:

$$C_{M_q,disp} = (1 + U^M_q)(C_{M_q,nom} + 0.5) - 0.5 + U^A_q \tag{2.21}$$

where the multiplier and adder dispersion bounds are:

$$U^M_q = 1 \pm 0.5, \quad U^A_q = 0.05 \pm 0.05 \tag{2.22}$$

The multiplier and adder are selected using a Gaussian and uniform distribution, respectively.

Control Interaction Database

The control interactions are delivered as a database of coefficients [11, 40]. These force and moment coefficients are a correction to the static aerodynamic database. In addition to the freestream properties and orientation inputs present for the aerodynamic database, the interaction adds the inputs for control system actuation. Control surface actuation can be given in terms of deflection angle δ, and RCS actuation can be given in terms of a thrust coefficient C_T. Table 2.3 demonstrates the general format of an RCS interaction database.

Table 2.3: General entry vehicle control interaction database

M	h	α	β	$C_{T,1}$	$C_{T,2}$	$\cdots$	$C_{T,n}$	ΔC_A	ΔC_N	$\Delta C_{m,cg}$
x	x	x	x	x	x	$\cdots$	x	x	x	x
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x	x	x	x	x	x	$\cdots$	x	x	x	x

Figure 2.19 shows the CFD run matrix used for testing the MSL RCS interaction [63]. Wind tunnel tests were performed at a subset of the Mach 10 points, centered around $\beta =$

0°, with some tests at $\beta = \pm 5°$.

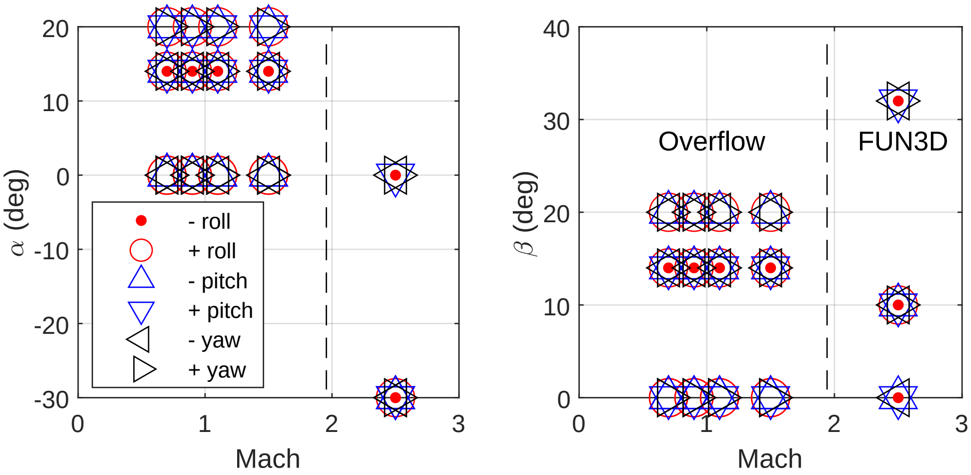


Figure 2.19: MSL RCS CFD run matrix

Schoenenberg et al. document the uncertainty for control interactions in [63]. The interaction model calculates the dimensional torque vector $\boldsymbol{\tau}$ as:

$$\boldsymbol{\tau} = dA_0 F_{RCS} \tag{2.23}$$

where d is the vehicle diameter, F_{RCS} is a vector of forces commanded for each jet, and A_0 is a matrix of nondimensional interaction moments. Each item in A_0 is calculated as:

$$A_{0,ij} = a_i \frac{l_{ij}}{d} R \tag{2.24}$$

where l_{ij} is the nondimensional moment arm of the jet, and a_i is the 3σ limit for the interaction magnitude in roll, pitch, or yaw. R is selected randomly from a Gaussian distribution with a mean of zero. For MSL, the bounds for pitch and yaw were $\pm 40\%$, and the bounds for roll were $\pm 20\%$. These bounds are constant across Mach number, slip angles, and backshell pressure.

2.4.2 Observations

Quasi-static databases are the state-of-the-art for entry vehicle aerodynamic and control interaction surrogate models. This is partly a result of the common heritage of Mars entry vehicles from Viking all the way to Mars 2020. The MSL database included a pressure correction term derived from the Viking flight data, and the Phoenix database included Viking wind tunnel data. Since Mars 2020 has substantially the same OML as MSL, the newest aerodynamic and control interaction databases date back to at least 2011. This helps to explain why databases continue to be used even with a variety of alternative surrogate methods.

The review of state-of-the-art surrogate models leads to observations about methodological gaps that can arise for future missions. In the process of compiling the data measured from the fundamental disciplinary models, the dynamic behavior is condensed to the uncoupled angular rate damping coefficients. This ignores dependency of these coefficients on other variables, such as the difference between C_{m_q} and $C_{m_{\dot{\alpha}}}$. This also ignores cross-coupling stability derivative terms such as for the rolling moment, which could be significant due to the noted sensitivity of the MSL controller to extraneous roll torque [40]. This is in addition to the reduction in fidelity in the fundamental disciplinary models, such as the lack of measurement for damping coefficient in the control interaction models. Indeed, current NASA research goals are aimed at constructing models that achieve a better understanding of the underlying physics of entry vehicle dynamics, especially with respect to pitch damping [147].

Observation 2.5: The state-of-the-art databases have reduced dynamic behavior model fidelity compared to the fundamental disciplinary models.

In addition to Observation 1.5 which previously noted that the uncertainty takes on the burden of the reduction in model fidelity, another gap concerns the construction of the uncertainty model. While the magnitude of the unsteady behavior is represented in the uncertainty, it does not account for the variation of that behavior with time.

Observation 2.6: The state-of-the-art control interaction and aerodynamic uncertainty models ignore the behavior's time-accuracy.

Aside from these gaps, more general observations can be made about the process of creating the surrogate models. Multiple fundamental disciplinary models cover different subsets of the flight regime with a greater or lesser degree of overlap. Designers must consider each model's fidelity to the physics of the problem during construction of the surrogate models. In most instances for the state-of-the-art methodology, models of higher fidelity (e.g., wind tunnel experiments compared to static CFD) are used for validation and uncertainty quantification. The state-of-the-art surrogate models are a fusion of data from multiple models at multiple levels of fidelity. However defaulted the process is, designers must be aware of the differences in fidelity driven by the choice of model, geometry, etc.

2.4.3 Alternative Surrogate Models

While historical Mars entry vehicles have all used databases as the form of the aerodynamic and control interaction surrogate models, alternative model forms are available. In light of the observations made about the capability gaps of the state-of-the-art surrogate models, a literature search was conducted on surrogate model forms that are applied to similar problems of interest. A particular focus was placed on identifying models that can be trained using time-series data, and models that can predict dynamic, time-accurate behavior.

Parametric Models

Parametric models are constructed using an explicit functional form with parameters or coefficients that must be determined from the source models. One of the main advantages of the parametric model is its predictable behavior in extrapolation. The prescribed nature of the model form can also be a double-edged sword: terms can be intelligently added to the model based on expected behavior, but this also introduces the possibility that unexpected behavior will be poorly captured.

Polynomial regression is a common form of parametric model that is often used in parameter estimation. In a polynomial regression, the predicted response can be modeled as a combination of a series of basis functions z with coefficients a [148]:

$$\tilde{y} = \sum_{j=1}^{n} a_j z_j \tag{2.25}$$

The form of the regression used can vary in complexity. Brown et al. [96] document the aerodynamic data reduction from ballistic range testing for an early OML of MSL. The polynomial expansion used for the aerodynamic coefficients and their 3-σ uncertainty have the general form shown in Equation 2.26:

$$C = a_0 + \sum_{i=1}^{3} a_i \sin^i \alpha + b \sin \beta \tag{2.26}$$

where a_0, a_i and b are the regression coefficients for some aerodynamic coefficient C. The coefficients may have one or more of these terms defaulted to zero. It was assumed that the coefficients did not vary with Mach number over the span of Mach numbers in the ballistic range test.

Wilder et al. [97] used parameter estimation to fit a regression to an entry vehicle ballistic range test. The regression used was a linear aerodynamic model calculated in the CADRA program [94]. The aerodynamic coefficients were found as nonlinear polynomials of Mach number and α_T. Calculation of the drag coefficient is shown as an example in Equation 2.27:

$$\begin{aligned} C_D =& [a_0 + a_1(M - M_{ref}) + a_2(M - M_{ref})^2] \cos \alpha_T \\ &+ [a_3 + a_4(M - M_{ref}) + a_5(M - M_{ref})^2] \sin^2 \alpha_T \\ &+ [a_6 + a_7(M - M_{ref}) + a_8(M - M_{ref})^2] \sin^4 \alpha_T \end{aligned} \tag{2.27}$$

where each regression coefficient has a value and uncertainty. The form of the pitch moment and damping coefficients included higher-order and nonlinear terms:

$$\begin{aligned} C_M = &\{[a_1 f + a_2(1-f)]g + (a_3 + a_4 f)(1-g)\} \sin\alpha_T, \\ &f = e^{a_5(M-M_{ref})}, \quad g = e^{a_6 \sin^2\alpha_T} \end{aligned} \tag{2.28}$$

The pitch damping coefficient was fitted with multiple models since the form of each model was not sufficiently validated [97]. One form was a function of the non-dimensional pitching rate q^*:

$$\begin{aligned} C_{M_q} =& g e^{-f q^{*2}} + a_1(1 - e^{-f q^{*2}}), \\ & f = e^{a_4(M-M_{ref})}, \quad g = h e^{-(a_2 q^*)^2} + a_3(1 - e^{-(a_2 q^*)^2}), \\ & h = \max[a_5 + a_6(M - M_{ref}),\ a_7,\ a_8 + a_9(M - M_{ref})] \end{aligned} \tag{2.29}$$

The other form was a function of q^* and α_T:

$$\begin{aligned} C_{M_q} =& g e^{-f q^{*2}} + a_1(1 - e^{-f q^{*2}}), \\ & f = e^{a_2(M-M_{ref})}, \quad g = h e^{-a_3^2 \sin^2\alpha_T} + j(1 - e^{-a_3^2 \sin^2\alpha_T}), \\ & h = \max[a_4 + a_5(M - M_{ref}),\ a_6,\ a_7 + a_8(M - M_{ref})] \\ & j = a_9 + a_{10}(M - M_{ref}) + a_{11}(M - M_{ref})^2 \end{aligned} \tag{2.30}$$

Parametric models can also be constructed on include control system actuation. Sahu and Fresconi [118] constructed a regression for a projectile with canard control. The model was split into the contribution from the rigid and the four moving surfaces:

$$C_A = a_0 + \sum_{i=1}^{4} a_i \sin^i \alpha_T + \sum_{j=1}^{4} \left(b_0 + \sum_{i=1}^{4} b_i \sin^i \alpha_{T,j} \right) \tag{2.31}$$

where $\alpha_{T,j}$ is the local total angle of attack on surface j. Each of the regression coefficients a_i and b_i were fitted as a function of Mach number.

Cheng et al. [149] performed parameter identification for an entry vehicle with RCS using maximum likelihood identification. The RCS contribution was modeled as a correction to the coefficients:

$$C_N = C_{N,aero} + a_0 \frac{r_{jet}\delta_{jet}}{QS_{ref}L_{ref}\cos\theta_{jet}} \tag{2.32}$$

where a_0 is the coefficient relating the direct jet moment to the interference, θ_{jet} is the angle between the jet thrust and the moment arm vectors, and δ_{jet} is the jet amplitude, calculated according to a trapezoidal function:

$$\delta_{jet}(t, t_{on}, t_{off}) = \begin{cases} (t - t_{on})/t_{rise}, & t_{on} \leq t < t_{on} + t_{rise} \\ 1, & t_{on} + t_{rise} \leq t < t_{off} \\ 1 - (t - t_{off})/t_{fall}, & t_{off} \leq t < t_{off} + t_{fall} \\ 0, & \text{otherwise} \end{cases} \tag{2.33}$$

where t_{on} and t_{off} are the values associated with the next firing of the jet. Hong et al. [150] used a similar model for the RCS, except with the addition of a delay time t_d shifting the entire trapezoidal function. These models were successfully used to extract the RCS contribution to the aerodynamic coefficients.

Neural Networks

Artificial Neural Networks (ANN) are surrogates that are designed to mimic the construction of the organic brain [151]. ANNs are nonintrusive, nonparametric, and nonlinear. The networks are constructed out of neurons which receive signals, integrate them and send them out to other neurons [151]. The parameters of the network are determined by “training” the network using example sets of data, whereupon it can be used to “recall” the generalized behavior for new inputs.

Many different ANN architectures have been developed, classified by the connectivity between the neurons [151]. In a feedforward neural network (FFNN), the connections pass data in one direction from the inputs to the outputs. An example single-layer FFNN architecture is shown in Figure 2.20. The design of a particular network can be varied by changing the number of hidden layers, and changing the number of nodes in each hidden layer.

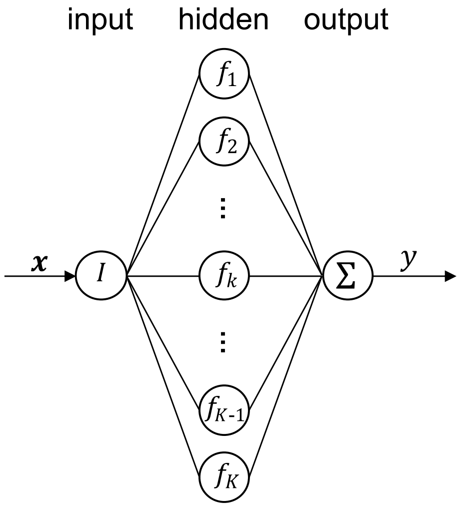


Figure 2.20: FFNN architecture

The architecture for a single perceptron neuron is shown in Figure 2.21. A set of signals S_j are input into the neuron, where they are multiplied by weight coefficients w_{ik} and summed. The activation function f_k is evaluated with the sum and a neuron bias b_k. The output ϕ_k is passed to the next layer. The sigmoid function is one of the most commonly used nonlinear activation functions [152]:

$$f_k(x) = \frac{1}{1 + e^{-x}} \tag{2.34}$$

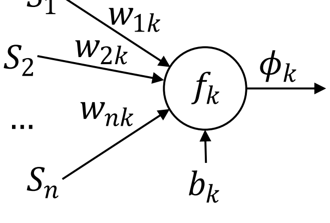


Figure 2.21: Artificial neuron

Linse and Stengel [153] trained an FFNN to model the normal force coefficient of an aircraft from simulated flight data. They found that the feed-forward models were sufficient to capture the coefficient using components of the vehicle and control state as inputs. Linse and Stengel emphasized the importance of selecting the input data to span the design space, as well as avoiding correlation between the input components. However, they did note that the feed-forward networks had difficulty at high angles of attack, where the behavior of the vehicle grew more nonlinear.

Singh and Ghosh [154] performed parameter estimation of a missile using an FFNN trained using simulated longitudinal flight data. The state variables α, α^2, q, and α_T^2, along with control deflection δ, were used as inputs for C_N and C_M responses. They experimented with training these responses concurrently or separately, with the separate training being more successful.

A time-delay neural network (TDNN) differs from the FFNN in that it includes recurrence of the inputs delayed over time. The response is a function of the input vector at the current time step $\boldsymbol{x}(t)$ and its history over delays $\boldsymbol{x}(t-\Delta_1), ..., \boldsymbol{x}(t-\Delta_j)$. The addition of this delay layer is shown in Figure 2.22. TDNNs have the advantage of being universal approximators for nonlinear, dynamical systems [155]. The memory of the past inputs effectively gives the model the ability to use high-order derivatives of the inputs. Choosing the number and values of the delays massively increases the hyperparameter space of the model. While the choice of delay is arbitrary, in practice for discrete data the delays are often chosen as integer multiples of the time step to avoid interpolation. Each neuron can use the same architecture and activation functions as described for the FFNN.

The connectivity of the network can be expanded to include recurrence of the output to form a recurrent neural network (RNN). The response is now also a function of the output vector history over delays $y(t-\Delta_1), ..., y(t-\Delta_j)$. Including recurrence in the TDNN forms a specific RNN architecture called a Nonlinear AutoRegressive model with eXogenous variables (NARX) [156]. This model architecture is shown in Figure 2.23.

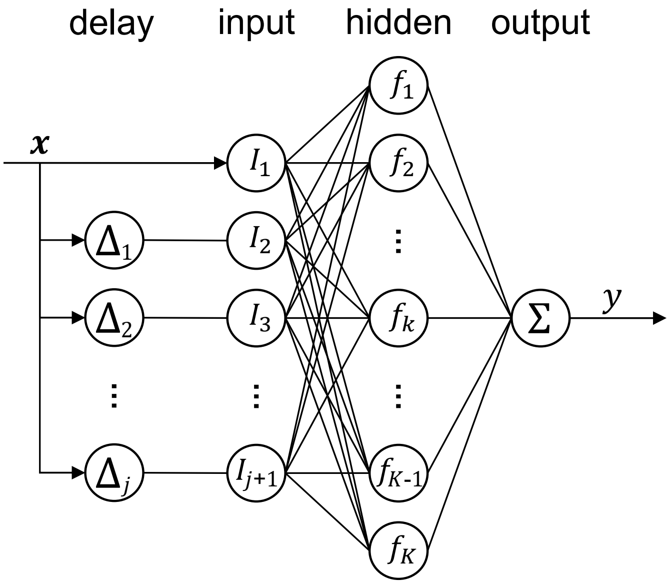


Figure 2.22: RNN architecture

Between the input and output layers, the RNN functions as a feedforward network. The output neuron has a feedback connection to one or more time delays, which then link back to the input layer. The input also includes the exogenous input which adds new information to the network.

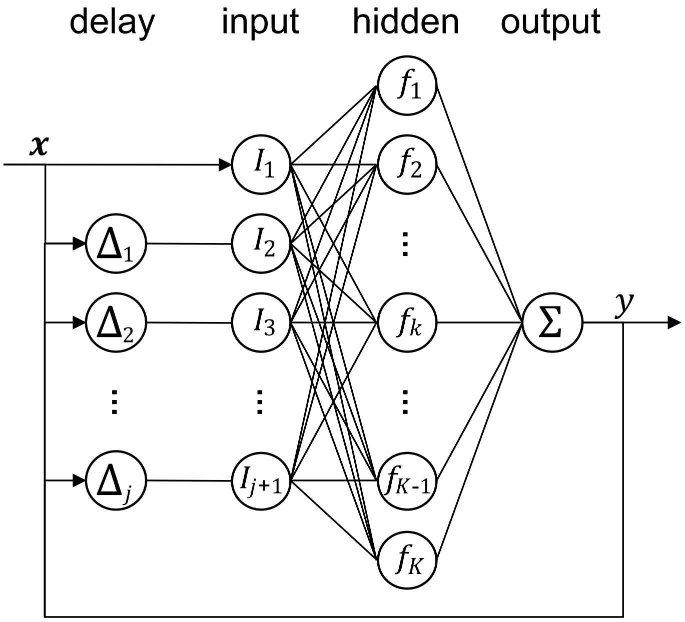


Figure 2.23: NARX architecture

Ignatyev and Khrabrov [157, 158] used neural networks to model the characteristics of an aircraft in forced oscillation at large angles of attack. This oscillation moved the vehicle through regimes of flow separation and attachment. They compared an RNN and an FFNN, finding the RNN to be favorable for its ability to capture the unsteady behavior of flow separation and reattachment.

Wang et al. [159] created RNNs to model the lift and pressure distributions of an airfoil. CFD was used to create time-accurate simulations of the airfoil pitching and plunging in the transonic regime where it was subject to shock motion. They compared two RNN architectures, NARX and Long Short-Term Memory (LSTM) [160], for predicting scalar and distributed quantities trained using a set of time-series samples. They found that the selection of the number and interval of the delays was critical to predicting scalar quantities. The results for scalar quantities were found to achieve a higher smoothness and accuracy than those of the distributed quantities.

2.4.4 Observations

The surrogate methods in the above section have a demonstrated history of use in modeling problems similar to entry vehicles. The machine learning-based methods in particular have been demonstrated for nonlinear dynamic effects such as shock motion and flow separation.

Considering that the potential use of these surrogate models would be as a substitute for the state-of-the-art surrogates, their input/output characteristics are examined. It can be assumed that their output can be made to be compatible with the requirements from 6DOF flight simulation framework, especially since many of these models are demonstrated for this purpose. The inputs of these surrogates are time-series data from dynamic, time-accurate models, which is consistent with some, but not all, disciplinary models reviewed in sections 2.2 and 2.3.

2.5 Summary of Observations

The observations made throughout the literature review presented above can be used to make a number of higher-level findings in response to the guiding questions introduced at the beginning of this chapter.

State-of-the-art Methodology

The literature review conducted in sections 2.1 to 2.4 answer the first guiding question:

Guiding Question 1: What is the state-of-the-art methodology for creating the necessary flight dynamics, aerodynamic, and control system models for entry controller tuning?

The static aerodynamics are modeled using steady-state CFD, with validation from wind tunnel tests and corrections based on flight data. Dynamic behavior is modeled using ballistic range tests, with the dynamic coefficients extracted by parameter estimation. The control system interaction is modeled using static, steady-state CFD as a correction between the actated and non-actuated states. The data from the aerodynamic and control system models are consolidated into aerodynamic databases for the nominal values and the uncertainty. These databases are then used as the disciplinary surrogate models for the flight simulation.

Suitability of the State-of-the-Art Methodology to Model Unsteady, Coupled Behavior

With an understanding of the state-of-the-art methodology, an assessment can now be made about its capability as directed by the second guiding question:

Guiding Question 2: What are the limitations of this methodology with respect to the physics of actively-controlled entry vehicle flight?

Entry vehicles experience highly unsteady flight during atmospheric entry. This was previously noted in Observation 1.1 and expanded upon in Sections 2.2.1 and 2.3.1.

The vehicle oscillates at a frequency related to the freestream conditions and its own mass properties. Large-scale fluid structures are formed in the wake of the vehicle, communicating across the region of subsonic flow. These effects can operate over multiple time constants. When the vehicle is free to move, the aerodynamics and flight dynamics are fundamentally coupled.

For a vehicle with an active control system whose effects propagate outside of (or change) the OML, the control system is fundamentally coupled with the aerodynamics. The degree of coupling is different depending on the control system, with RCS being the most coupled and control surfaces being less coupled. Separating these models out is an artifact of the way these models are currently constructed. In reality, they are governed by the exact same physics. RCS plumes can excite the subsonic region of the wake, which is already a primary contributor to aerodynamic unsteadiness. The plumes can also interact with the high-velocity flow around the vehicle, affecting separation and attachment. In turn, the plume behaves differently in separated or attached flow. Furthermore, the actuation of control systems depends on the guidance algorithm and control law, which in turn depends on the vehicle motion calculated by the flight dynamics model.

Observations 2.5 and 2.6 indicate the limitations of the state-of-the-art methodology to capture higher-order coefficients, as well as shorter time-constant behavior from the wake. This could theoretically be solved by using more input dimensions in the database and accepting the associated increase in cost to sample the higher-dimensional space. However, this will still not be able to capture time-dependent effects.

Based on these observations, a formalized definition for unsteady, coupled behavior is used throughout this dissertation[2]:

Definition 2: Unsteady, coupled behavior: The inherent, time-dependent flight dynamics, control system, and aerodynamic behaviors of an actively-controlled entry vehicle, and

[2] Note that the definition of coupling describes interaction on the disciplinary level, not within it (e.g., pitch/roll cross-coupling).

the interactions between them, that are not accounted for in the state-of-the-art aerodynamic and control system surrogate models.

Alternate Forms of Surrogate Model

The final two findings respond to the third guiding question:

Guiding Question 3: What alternative methods exist that could ameliorate these limitations?

If the state-of-the-art aerodynamic databases are unsuitable for modeling the unsteady, coupled behavior of an entry vehicle with active control, it is possible to use a different form of surrogate model. Section 2.4.3 surveys a subset of the model forms which have progeny in aerospace applications. This includes a number of models that are used in parameter estimation to fit data from ballistic range experiments, a process that is already used in modeling entry vehicle aerodynamics. However, within the scope of entry vehicles, and even for the same OML, there exists some variation in the specific format of model used. The capability of these parametric models to capture the unsteady behavior and transients from RCS interaction is unknown. Since the model assumes an explicit functional form, it would be limited by the designer's understanding of the system behavior and the inclusion of applicable terms in the model.

As an alternative, nonparametric models could be used, such as neural networks. Since they do not assume a functional form, they could be better suited to modeling the unsteady, coupled behavior.

However, a key point about the use of surrogate models is that they still require a source of training data. The surrogate model choice may also be dependent on the qualities of the fundamental disciplinary models. Thus, the underlying models must still be capable of modeling the desired behavior.

Applicability of the Underlying Aerodynamic and Control Interaction Models

While the state-of-the-art surrogate models have a deficiency in modeling the unsteady, coupled behavior, this is not necessarily the case for the underlying models used to populate the surrogates.

Static wind tunnel tests accurately model the unsteady behavior of the wake, albeit with added interference from the sting balance. Ballistic range tests accurately model the wake for a vehicle in free-flight. As noted in Observation 2.1, it is the measurement process that limits the application of these models to capture the unsteady, coupled behavior. Observation 2.2 shows that coupled CFD-RBD models ameliorate this issue, simulating the vehicle in time-accurate free flight, with the ability to sample data with any desired precision. Even though these models have not yet been used directly in a Mars entry vehicle aerodynamic surrogate, they have been successfully validated against experimental data. The underlying aerodynamic models are sufficient to capture the coupled behavior of aerodynamics and flight dynamics.

For RCS interaction, wind tunnel experiments similarly model the unsteady behavior with fixed vehicle attitude. However, the CFD models used in the state-of-the-art methodology are still limited to steady-state solutions. CFD flow solvers are capable of modeling this behavior; the largest hurdle is the lack of validation to date for entry vehicles.

Figure 2.24 summarizes the literature review conducted for the modeling and simulation, and the key observations that will contribute to the problem formulation in Chapter 4.

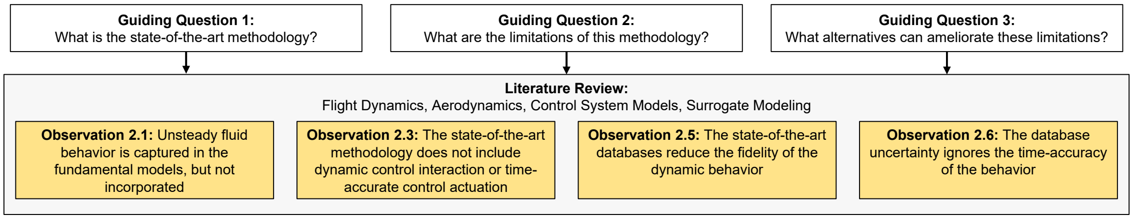

Guiding Question 1: What is the state-of-the-art methodology?
Guiding Question 2: What are the limitations of this methodology?
Guiding Question 3: What alternatives can ameliorate these limitations?
Literature Review: Flight Dynamics, Aerodynamics, Control System Models, Surrogate Modeling
Observation 2.1: Unsteady fluid behavior is captured in the fundamental models, but not incorporated
Observation 2.3: The state-of-the-art methodology does not include dynamic control interaction or time-accurate control actuation
Observation 2.5: The state-of-the-art databases reduce the fidelity of the dynamic behavior
Observation 2.6: The database uncertainty ignores the time-accuracy of the behavior

Figure 2.24: Summary of modeling and simulation literature review

CHAPTER 3
CONTROLLER TUNING

While Chapter 2 provides a foundation for addressing the first part of the overarching research objective, Chapter 3 addresses the second: the utilization of a “direct” accounting for coupled, unsteady behavior in the process of entry controller tuning. First the current methodology must be understood:

Guiding Question 4: What is the state-of-the-art methodology for entry controller tuning for a Mars entry vehicle?

The following chapter presents a literature review of the process of entry controller tuning for atmospheric entry vehicles. Figure 3.1 identifies the high-level activities and products used by designers for entry controller tuning.

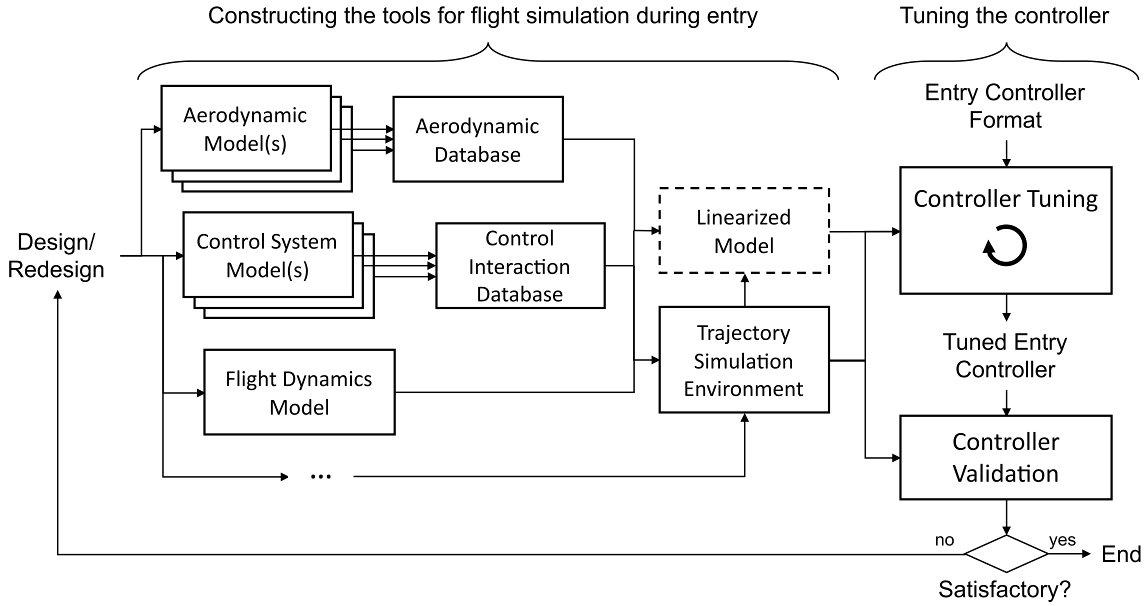


Figure 3.1: Baseline Entry Controller Tuning Process

A subset of the vehicle design process is shown, starting from an initial design or a significant redesign. If disciplinary models do not yet exist, or have been invalidated by design changes, these models must be created or updated. For aerodynamics and control system disciplines, one or more models may be consolidated into a surrogate model, as found in Section 2.4. The aerodynamic, control system, flight dynamics, and additional disciplinary models are used to construct the trajectory simulation environment. This serves as a starting point for identifying the remaining processes shown. An investigation is made into the controller and control law in order to determine the extent of the input variables to the process of controller tuning.

Given this knowledge about the baseline methodology, the impact of potential changes to the modeling methodology imposed on the tuning process must be identified:

Guiding Question 5: What limitations would be imposed on the entry controller tuning process by accounting for coupled, unsteady behavior in the models?

3.1 Entry Controller Architecture

Before discussing the controller tuning process, the controller introduced in section 1.1.1 must be understood in more detail.

3.1.1 Attitude Controller

The guidance algorithm for MSL and Mars 2020 serve as the best examples for Mars entry vehicles (the Viking landers, the only previous Mars entry vehicles with active control, used a simpler guidance algorithm [161]). Brugarolas et al. [27, 162] describe the construction of the entry-phase attitude controller for MSL. The controller operates within the guidance algorithm as shown in Figure 3.2.

During the entry phase, the controller needs to stabilize the vehicle attitude. The actual angle of attack needs to be stabilized at the predicted trim angle of attack, within a certain deadband. Similarly, the actual sideslip angle must be stabilized about the predicted trim sideslip angle, which is close to zero. The α and β deadbands are functions of the Mach

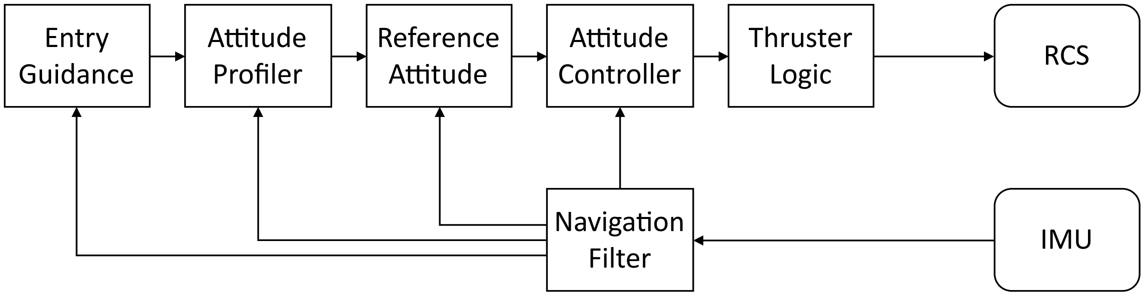


Figure 3.2: MSL Entry Phase Guidance Functional Diagram [27]

number. The controller also responds to bank angle commands generated by the guidance algorithm. During the bank reversal maneuvers, bank angle could change rapidly, but outside of these maneuvers the bank angle would be held constant. 80% of RCS propellant was reserved for bank reversals, with the remaining 20% used for stabilizing the attitude [27].

The entry controller operates in the control frame C, illustrated in Figure 3.3. the x- and y-axes are aligned with the z- and y-axes of the body reference frame F, respectively. The z-axis is opposite to the x-axis of F and offset by the trim angle of attack, making the control frame non-orthogonal. Using the control frame separates the lateral and longitudinal motion, so the entry controller can be constructed as three separate channels for α, β and bank angle.

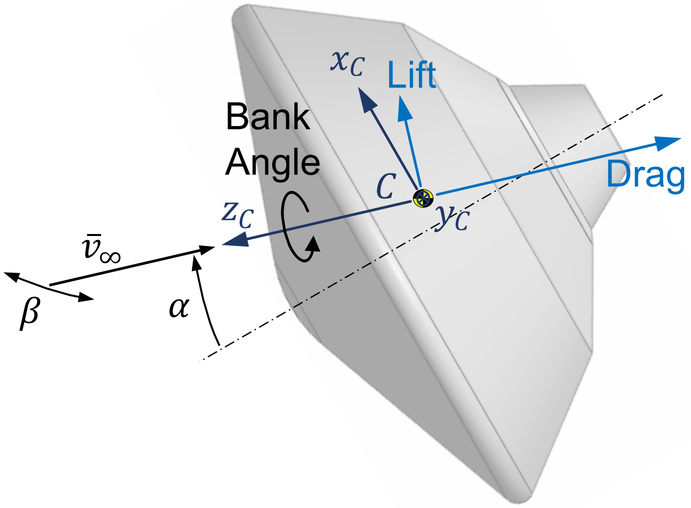


Figure 3.3: Control frame [27]

The structure of the controller for each individual channel is shown in Figure 3.4 [27]. The controller includes both a feedback and feedforward path. The feedforward path is primarily used for the bank reversals, when the commanded angles are changing rapidly. The feedback path stabilizes the plant, using a proportional-derivative (PD) controller. Earlier development of MSL considered the use of a phase plane approach or linear-quadratic regulator [163]. In practice, the feedback primarily operates as a rate controller since the attitude deadbands are set wide [162].

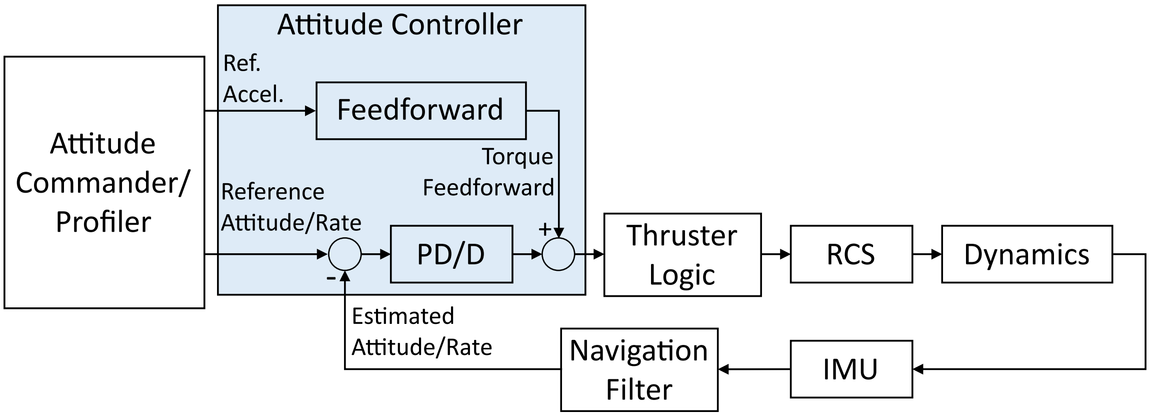


Figure 3.4: MSL single-axis controller functional diagram [27]

3.1.2 Control Law

The control law is the mathematical relationship between the measured vehicle state, the desired state, and the control system inputs [164].

The MSL feedback controller used a phase plane formulation with attitude and rate deadbands as shown in Figure 3.5 [162]. In the green region, the attitude error is considered to be inconsequential, so the controller acts purely as a derivative controller. In the diagonal blue-shaded region, the attitude error is increased, so the controller acts as a PD controller.

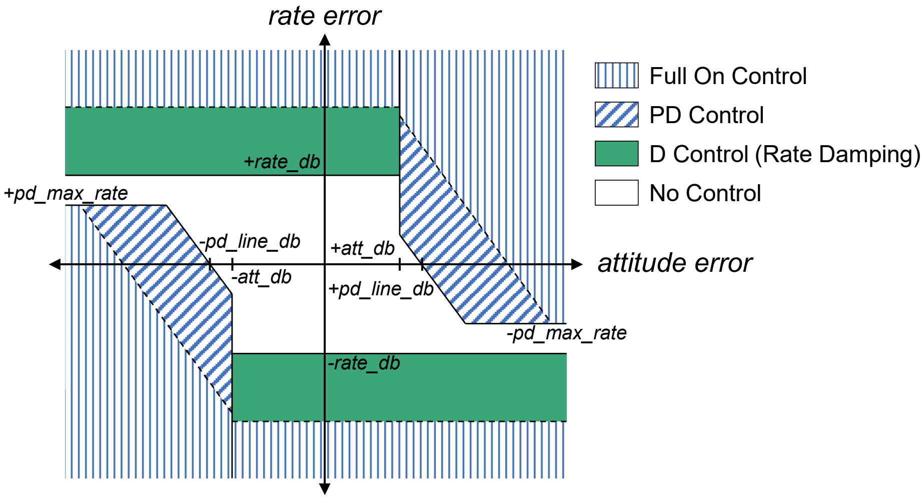


Figure 3.5: MSL Feedback Path Controller [162]

The controller has the form of Equation 3.1:

$$\bar{u} = -\boldsymbol{K}\bar{x} \tag{3.1}$$

where the control vector $\bar{u}$ contains the commanded torque for each axis, $\boldsymbol{K}$ is a matrix of feedback proportional gains, and $\bar{x}$ is the state vector with the channel value and rate. For example, the angle of attack error α_{err} is calculated as $\alpha_{cmd} - \alpha$. In the state vector, this error is modified according to the deadbands:

$$\bar{x} = \begin{bmatrix} \begin{cases} \alpha_{err} - \alpha_{db} & \alpha_{err} \geq \alpha_{edb} \\ 0 & \alpha_{db} > \alpha_{err} > -\alpha_{db} \\ \alpha_{err} + \alpha_{db} & \alpha_{err} \leq \alpha_{edb} \end{cases} \\ \begin{cases} \dot{\alpha}_{err} - \dot{\alpha}_{db} & \dot{\alpha}_{err} \geq \dot{\alpha}_{db} \\ 0 & \dot{\alpha}_{db} > \dot{\alpha}_{err} > -\dot{\alpha}_{db} \\ \dot{\alpha}_{err} + \dot{\alpha}_{db} & \dot{\alpha}_{err} \leq \dot{\alpha}_{db} \end{cases} \end{bmatrix} \tag{3.2}$$

Further calculation is performed downstream to convert this torque into the pulse width and modulation frequency required for each jet. When the torque reaches saturation (represented by the light-shaded region), the jets will fire continuously, making the controller equivalent to a bang-bang controller [162].

The constants in this control law include the control gains and the values of the deadbands. Since the dynamic behavior of the vehicle changes with the flight conditions, these values are scheduled based on flight conditions like Mach and dynamic pressure, as well as phase and timing during entry [162]. Gain scheduling is a common and well-established method of control [165]. To date, all Mars entry vehicles with active control have used gain scheduling. The Viking landers used gain scheduling for roll control and pitch and yaw damping [161]. MSL used a gain scheduled attitude controller for full 3-axis control [26], and Mars 2020 uses an improved version of the same system [10]. Therefore, this is considered to be the "state of the art" for controller development. Gain scheduling may be preferred over more advanced techniques as its development process is considered to be lower risk than the alternatives [166].

Controller architecture selection is somewhat dependent on the control system. Atkins [167] develops a Linear Quadratic (LQ) controller with integral action for an entry vehicle equipped with two-axis IMMA control. The control system operates by imparting a force on the IMMAs, whose motion is modeled as a spring-mass-damper system. Etkin was able to linearize the system about the equilibrium point. The control law is shown in Equation 3.3:

$$\bar{u} = -\boldsymbol{K}(\bar{x} + \boldsymbol{R}\bar{r}_{cmd}) - \boldsymbol{K_{int}}\bar{x}_{int} \tag{3.3}$$

The control vector $\bar{u}$ consists of the forces applied to each IMMA. The controller tracks commanded angle of attack and sideslip:

$$\bar{r}_{cmd} = [\alpha_{cmd} \quad \beta_{cmd}]^T \tag{3.4}$$

The state vector $\bar{x}$ consists of the angular velocity vector, orientation vector, slip angles, and the velocity and acceleration of each of the IMMA masses. $\boldsymbol{R}$ and $\boldsymbol{K_{int}}$ are matrices of feedforward control gains and feedback integral gains, respectively. Atkins also used gain scheduling, based on atmospheric density.

More recent nonlinear control laws have been proposed for entry vehicles [168]. Van Soest et al. [169] created a constrained model predictive controller for an entry vehicle with aerodynamic control surfaces. They found that, in combination with feedback linearization, the predictive controller performed better than a PID control, and that optimal control was easier to develop. However, they noted that the process assumed a highly accurate aerodynamic model.

Adaptive controllers have a mechanism that allows the controller to learn the vehicle behavior in real time and update the control law [170]. This class of controller is useful for problems with a high degree of uncertainty. Restrepo and Valasek [171] compared two adaptive control schemes for a Mars entry vehicle through numerical simulation subject to vehicle and environmental uncertainties. They compared a model reference adaptive control commonly found in literature to a structured adaptive model inversion controller. Both controllers were found to perform well for the problem, but differed in the difficulty of implementation. They also investigated the effect of different learning rates. Steinberg [170] notes that “proving stability and convergence of the total system is very challenging.”

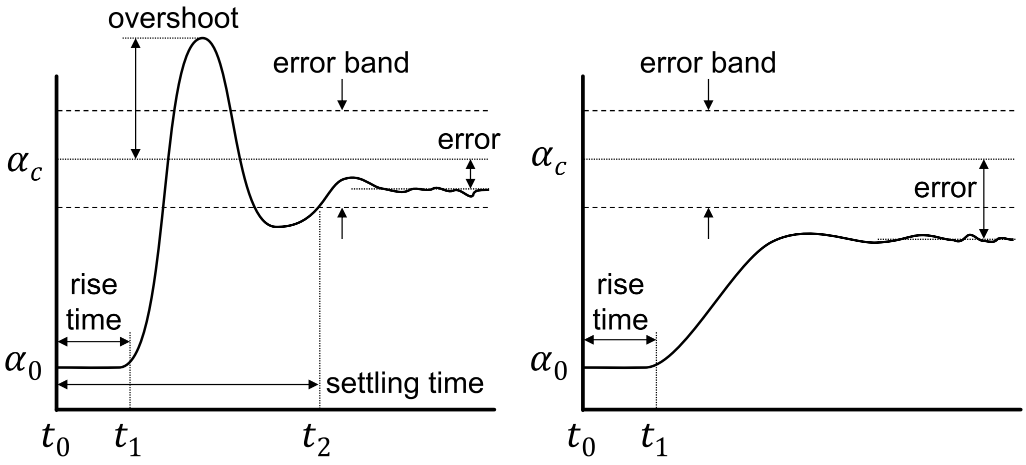
3.1.3 Observations

Designers develop an entry controller format, which has some parameters that need to be tuned. This can include parameters like gains and deadbands as in the MSL controller, or learning rates for an adaptive controller. Some process is necessary to find satisfactory values for these parameters.

The architecture of the controller itself is more dependent on the control system than the control law. The baseline control law is compatible with multiple types of control systems, including RCS, control surfaces, and IMMA.

3.2 Controller Performance

Metrics must be established for determining the performance of the controller and control law relative to a set of requirements. Schoenwetter [172] documents the process for measuring settling time and other characteristics for a generic value under control. These metrics are visualized for the angle of attack in Figure 3.6, starting at some initial value α_0 or perturbation, with a controller attempting to meet a commanded value.



(a) In control

(b) Out of control

Figure 3.6: Control Metrics

Figure 3.6a shows the response coming under control, whereas Figure 3.6b shows the response failing to come under control. For a constant commanded value and acceptable deadband, the controller seeks to bring the system under control from its initial condition. Rise time is the duration between receiving the command and the system starting to respond. Settling time is the duration required for the system to enter and remain within the error band. Relative or absolute overshoot is the maximum error of the system with opposite sign to the starting value.

Additional controller performance metrics are identified through the vehicle design literature. For an entry controller, the error measured over a short duration is called the tracking error. This value is considered to be instantaneous compared to the long duration of the entry phase.

The MSL controller was also tuned with the desire to minimize fuel usage [162]. Restrepo and Valasek [171] identified control effort, defined as $\bar{u}^T\bar{u}$, as a metric for controller comparison. They also evaluated the integral of the control vector magnitude over time as it is proportional to propellant usage. McNamara et al. [173] used phase and gain margin requirements.

The controller performance must be evaluated over multiple initial conditions, resulting in distributions of the metrics defined above. As Marrison and Stengel [174] write, "[a] probabilistic design metric is equally applicable to systems that are linear or nonlinear, time-varying or time-invariant, continuous or discrete." A set of control gains might be evaluated at different initial attitude and angular velocity conditions, as well as uncertainty, with the overall performance measured according to the probability that the controller meets the performance requirements. The MSL controller in particular experienced sensitivity to the uncertainty bounds. Schoenenberger et al. note that "the controller [had] no way of identifying and correcting for a persistent roll torque explicitly" [40]. The controller had a tendency to fail in a bank reversal near the peak dynamic pressure when the persistent roll moment was too high. Thus, an overly-conservative estimate of roll moment uncertainty could have artificially decreased the controller performance.

Controller requirements act as an interface between designers during development. For the purposes of guidance development and testing, the entry controller must deliver a certain minimum of performance, measured according to some testing parameters. Thus, designers are primarily concerned with the "worst-case" performance of a controller, subject to uncertainties in mass properties, initial conditions, control mechanism performance, etc.

Since the controller must simply satisfy the requirements, as opposed to the necessity of being the true optimum, at some point continuing the controller tuning activity would serve no engineering purpose. If this work is to evaluate controller performance, a threshold for significance must be established with respect to real-world applications. Literature that specifically cites an entry controller performance requirement is uncommon. Thus, understanding significance relies on abstraction of the UQ process. Queen et al. [175] present the dispersion analysis for the Mars Polar Lander, which flew on an unguided ballistic trajectory. The dispersion analysis included uncertainty due to lateral C.G. offset (from the nominal exactly on the axis of symmetry). Considering that the offset in C.G. would result in a trim angle of attack, the dispersion can be considered as a tracking error relative to the commanded $\alpha = 0°$ Given the aerodynamic properties of the vehicle, the dispersion due to tracking error can be calculated:

$$\alpha_e = \frac{C_A \Delta z_{cg}}{C_{M_\alpha} d} = \frac{1.628 \cdot 0.0025}{-0.0011 \cdot 2.4} = -1.5731 \tag{3.5}$$

Assuming that the maximum downrange landing error resulted from the maximum α_{trim} (C.G. offset), the sensitivity of downrange landing error magnitude to tracking error is $5.9158\mathrm{km}/°$. If the requirement for a precision landing within $100\mathrm{m}$ was applied, a tracking error of only $0.02°$ would result in excessive downrange landing error.

A similar analysis can be applied for a vehicle which uses a lifting trajectory. However, since all flight vehicles that have used lifting trajectories have flow bank-to-steer trajectories, the results for angle of attack cannot be assumed to compound. Instead, the trajectory simulations performed by Atkins [167] for an IMMA-equipped vehicle are used. This vehicle modulated its lateral velocity by displacing the C.G., not by bank-to-steer. Therefore, the same assumption used for the ballistic trajectory case can be applied.

The results for both trajectory types are listed in Table 3.1. As expected, the threshold of significance for a guided, lifting trajectory is much larger than for an unguided, ballistic trajectory. The lifting trajectory is much less sensitive to tracking error, since the guidance algorithm can correct for persistent errors during entry.

Table 3.1: Threshold of significant instantaneous tracking error for precision Mars landing

Ballistic trajectory	Lifting trajectory
0.02°	0.17°

Observations

Controller performance metrics that measure the properties of the controlled variable are valid regardless of the control system architecture. While this may not be the case for other variables, e.g., the consumables that are of interest for RCS, analogues such as control effort can be used in their place.

Designers are concerned with the robustness of the controller to varying initial conditions and to uncertainty in the disciplinary models. Therefore, a probabilistic approach is taken to measuring the controller performance.

The performance metrics reviewed above can be determined by evaluating the time-series vehicle and controller state data. These products can be produced through flight simulation. The simulation duration necessary will be on the same order as any temporal performance requirements. It is unnecessary to determine that a controller has failed beyond a certain threshold — as much may be learned about the problem from a controller that failed a settling time requirement by 100% as from one that failed by 1,000%.

3.3 Controller Tuning

Controller or control law tuning is the process of finding the values of the parameters in the control law, such as gains and deadbands, at one or more design conditions.

3.3.1 Linearized Control Problems

For systems that are linear or can be linearized about the area of interest, analytical methods exist for finding optimal control gains [176]. McNamara et al. [173] document the process of gain scheduling for the Orion launch abort vehicle controller. Considering the lack of sources for the gain scheduling process for historic Mars entry vehicles, this serves as an example of a process used within NASA, for a vehicle that presents similar problems for controller development, at a similar time to the development of MSL. The vehicle uses jets for attitude control, and also encounters aerodynamic/jet interaction. It must perform satisfactorily over a wide Mach number and altitude envelope, and it is unstable for most of the flight regime.

The tuning process started with the construction of a linearized model [173]. The vehicle is modeled as a rigid body, and the aerodynamic stability derivatives are found from a database by using central differencing of small perturbations. This linear model, shown in Equation 3.6, was used to tune the gains.

$$\dot{\bar{x}} = \boldsymbol{A}\bar{x} + \boldsymbol{B}\bar{u} \tag{3.6}$$

Multiple methodologies were considered for tuning, but the linear quadratic regulator (LQR) was selected due to its simplicity in implementation considering the complexity of the aerodynamic behavior. LQR was also investigated during the initial design of the MSL entry controller [27]. Atkins [167] also used LQR to determine IMMA controller gains.

The objective of LQR is to determine the control law as defined in Equation 3.1 that minimizes the cost function shown in Equation 3.7 [177]:

$$\boldsymbol{J} = \int_0^\infty (\bar{x}^T \boldsymbol{Q}\bar{x} + \bar{u}^T \boldsymbol{R}\bar{u})\,dt \tag{3.7}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the state and control input weights, respectively. $\boldsymbol{J}$ has a global minimum when $\boldsymbol{Q}$ is positive semi-definite and $\boldsymbol{R}$ is positive definite. The optimal control gains are found using the equation:

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} \tag{3.8}$$

where $\boldsymbol{P}$ is the solution to the algebraic Riccati equation [177]:

$$\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = 0 \tag{3.9}$$

McNamara et al. [173] use Newton's method to iterate on the weights until the controller performance requirements were met. They defaulted some of the weights by setting the ratio of the weights for some variables, such as α and β, to be larger compared to other state variables. The gains were scheduled as a function of time (used as a surrogate for changing mass properties), Mach number, and altitude, with LQR applied at each of the selected design conditions. Design of the schedule was concurrent with the tuning process. Since the gains are interpolated at the flight conditions, care was taken to avoid discontinuities in the gain designs. Tuning is initiated for all of the design conditions at the same time, with further iteration at each design stopping when it is found to meet the performance requirements.

Validation is performed for each of the designs using Monte Carlo simulation [173]. A first assessment was made using simulations of the linearized model, with a final verification performed using a 6DOF trajectory simulation. The Monte Carlo simulations included dispersions for mass properties, aerodynamic coefficients, atmospheric parameters, and initial state about the nominal trajectory.

3.3.2 Empirical Approach

Entry vehicles, especially those flying lifting trajectories, experience large ranges of α_T and can exhibit highly nonlinear behavior, as identified in Sections 2.2.1 and 2.3.1. The degree of nonlinearity may also be exacerbated by new decelerator technologies. If linearization is not feasible, gains must be determined empirically.

The tuning of an entry controller is a form of optimization problem. For an entry controller that is designed to meet commanded angle of attack and sideslip α_c and β_c, the task of minimizing the settling time is written in standard form in Equation 3.10:

$$
\begin{aligned}
\min_{t_{set}} \quad & t_{set} = f(\bar{x}_0, \boldsymbol{K}, \alpha_c, \beta_c, \bar{U}) \\
\text{s.t.} \quad & x_{i,0,min} \leq x_{i,0} \leq x_{i,0,max} \\
& k_{j,min} \leq k_j \leq k_{j,max} \\
& -\alpha_{trim} \leq \alpha_c \leq \alpha_{trim} \\
& -\beta_{trim} \leq \beta_c \leq \beta_{trim} \\
& u_{m,min} \leq u_m \leq u_{m,max}
\end{aligned}
\tag{3.10}
$$

where $x_{i,0}$ is the value of the initial condition for $i = 1...n_x$ non-trivial initial conditions, k_j is the gain for $j = 1...n_k$ non-trivial gains, α_{trim} and β_{trim} are the maximum trim angle of attack and sideslip, and u_m is the value of the uncertainty for $m = 1..n_u$ uncertainty parameters. The function f represents the evaluation of the controller through flight simulation and the post-processing of the results to find the controller performance metrics. In the worst-case scenario, this function must be treated as a black box. This function is highly dimensional, with $n_x + n_k + n_u + 2$ inputs. For the above example, the total number of gains in each set is given by Equation 3.11

$$n = n_u(n_x + 2n_r) \tag{3.11}$$

where n_u is the number of actuator state variables, n_x is the number of state variables, and n_r is the number of commanded variables. Some of these gains can be defaulted to zero based on assumptions of symmetry and low cross-channel interaction, but this can still leave tens of gains that must be determined. The process of determining which gains are non-trivial may also require evaluation in the flight simulation environment.

This function may be difficult to optimize because of the highly nonlinear nature. McNamara et al. [173] found that even “the LQR space is highly nonlinear and first derivative information is unknown.” Furthermore, the settling time is undefined if the vehicle does not come under control. Depending on the choice of disciplinary models, the function could also be non-deterministic. All of these attributes contribute to the complexity of the gain optimization problem.

In reality, it is not necessary to find the optimal gains — it is sufficient to find a set of gains that satisfy the performance requirements. Therefore, a method can be used to simulate a large number of designs, evaluate the performance metrics, then filter the results based on their conformity to the requirements [178]. At its most basic, this is equivalent to a random search optimization algorithm. Each set of control gains requires experimentation to determine its performance metrics. A similar process to that used for uncertainty quantification for guidance system validation is applied. 6DOF trajectory simulations are run with control in the loop, and the metrics are evaluated from the resulting trajectory. There are two main points of difference with the process of guidance system validation. First, instead of simulations over the entire duration of EDL, simulations only need be executed for a short duration, comparable to the time constants of the dynamic behavior, to allow the vehicle to be brought under control. Second, the stochastic parameters include not only the uncertainties, but also the gain design and initial conditions.

The controller can also be tuned using a more directed optimization methodology. Marrison and Stengel [174] used a genetic algorithm (GA) and Monte Carlo simulations to design a robust controller for a hypersonic vehicle. The genetic algorithm searched the controller design space, and at each design a Monte Carlo simulation was run with 28 uncertainty parameters for aerodynamics, atmosphere, etc. The designs were evaluated based on 39 probabilistic performance metrics for stability, settling time, rise time, overshoot, load factor, control input magnitude, etc. The GA executed several generations to find a controller that met the requirements, evaluating over 20,000 designs. The algorithm was also repeated eight times, yielding a range of minima, but produced a controller that performed favorably compared to their baseline, achieving a 99.9% probability of meeting their stability requirement.

Miyazawa and Motoda [179] studied the feasibility of stochastic parameter tuning of a reentry vehicle controller. Optimization was performed with the mean tracking technique and with Monte Carlo simulation of the uncertainty parameters. Motoda et al. [180] also investigated the use of simulated annealing for optimization, comparing this to the results achieved using a downhill-simplex method.

Observations

In the general case where the degree of behavioral nonlinearity is unknown and the properties of the disciplinary models and flight simulation are unknown, the controller tuning process can always be considered to be an optimization problem, seeking to optimize the controller performance metrics to a required threshold.

Observation 3.1: Controller tuning is an optimization activity.

The tuning process is dependent on the complexity and behavior of the system. For systems that are linearizable, analytical methods can be used. In some cases these methods require access to the disciplinary models to produce a linearized model.

Observation 3.2: The controller tuning process can utilize a linearized model of the flight simulation environment.

At the theoretical extremity, the vehicle behavior may be so complex as to require empirical methods for controller tuning. In these cases, the trajectory simulation environment might be used directly. Regardless of the controller tuning methodology used, the controller must be validated to ensure that the performance is satisfactory in the highest available fidelity of flight simulation, and to ensure that the performance is robust to the uncertainty parameters.

Observation 3.3: Controller validation is performed using a large number of 6DOF flight simulations.

3.4 Summary of Observations

State-of-the-art Controller Tuning Methodology

Based on the literature review for entry vehicles and closely related projects presented above, Guiding Question 4 can be addressed with a reasonable degree of certainty:

Guiding Question 4: What is the state-of-the-art methodology for entry controller tuning for a Mars entry vehicle?

The activities and products involved in the controller tuning process have been identified, filling in the flowchart in Figure 3.1. The process is considered here from the point where a design decision has been made as to the format of the controller and control law. With the format of the parameters known, the process of tuning can start. It requires the availability of at least the flight simulation, but could possibly use the disciplinary models directly or create a linearized model. The process produces a set or design of parameters. This design is then validated using the flight simulation, where the controller performance metrics are evaluated. If the metrics satisfy the requirements, then the controller is successfully tuned.

Continued Use of Gain Scheduling

The use of gain-scheduled controllers is likely to continue for future Mars entry vehicles. Gain scheduling has been used for the entry controllers for each of the previous Mars entry vehicles with active control, as well as for the Orion capsule. This heritage provides an ease of implementation and large knowledge base that make its continued use attractive for designers. Given that EDL is already one of the highest-risk phases of flight This conservatism has been documented in the design of MSL, the last new Mars entry vehicle OML that was developed. Designers elected to use a velocity trigger over a range trigger for parachute deployment, accepting increased landing uncertainty “due to some perceived risk considerations” with the range trigger [50]. Controllers that are fully developed and characterized *a priori* are (at least perceived to be) lower risk compared to adaptive or fuzzy control schemes.

Use of Models with Multiple Levels of Fidelity

The controller tuning methods examined in section 3.3 generally utilize two models. One is the flight simulation environment described in Chapter 2. The second is a linearized model constructed from the aerodynamic and control system models. The linearized model can be considered as a reduced-order model of the flight simulation environment, resulting in a lower computational expense to tune a controller. However, this is achieved at the cost of lower fidelity to the physics of the problem. This is why the controller must still be validated in the higher-fidelity flight simulation environment. Since a more limited number of controller designs are validated, the higher computational expense is not as much of a concern.

Use of Many Trajectory Simulations

With this understanding of the baseline process, the final guiding question can be addressed:

Guiding Question 5: What limitations would be imposed on the entry controller tuning process by accounting for coupled, unsteady behavior in the models?

The process of controller tuning relies on performing trajectory simulations, where the number of simulations required depends on the tuning methodology used. For a system that is easily linearized, although determination of the gains may be possible without trajectory simulation, the controller still needs to be validated. Due to the need to validate the controller under realistic flight conditions, these simulations are performed with the E2E trajectory simulation environment (or its constituent models). As a result, thousands or tens of thousands of simulations must be run.

When considering the validation of the controller (as opposed to the guidance algorithm as a whole), the simulation duration only needs to be on the order of the vehicle modes. Thus, the simulations may only need to be executed for seconds or tens of seconds. In summary, the controller tuning process relies on a large number of short-duration flight simulations.

The controller tuning process is therefore sensitive to the execution time of the flight simulation. If the computational cost of each run were to increase, it would have a multiplicative effect on the time required to tune and validate a controller.

Figure 2.24 summarizes the literature review conducted for the entry controller and controller tuning process, and the key observations that will contribute to the problem formulation in Chapter 4.

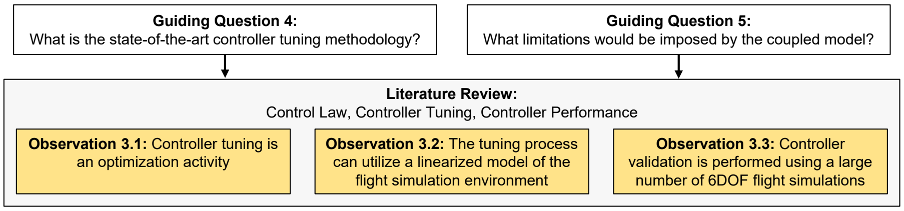

Guiding Question 4: What is the state-of-the-art controller tuning methodology?
Guiding Question 5: What limitations would be imposed by the coupled model?
Literature Review: Control Law, Controller Tuning, Controller Performance
Observation 3.1: Controller tuning is an optimization activity
Observation 3.2: The tuning process can utilize a linearized model of the flight simulation environment
Observation 3.3: Controller validation is performed using a large number of 6DOF flight simulations

Figure 3.7: Summary of controller tuning literature review

CHAPTER 4
PROBLEM FORMULATION

The following chapter uses the overarching research objective and the observations made during the literature review to define major research questions and create hypotheses. For each research question, experiments are proposed to validate the hypotheses.

4.1 Research Objective

Chapter 1 identified the need to reduce aerodynamic and control system uncertainty during entry to support Mars precision landing. This uncertainty includes a significant contribution due to model fidelity choices; the unsteady behavior of entry vehicles, especially with active control, is accounted for indirectly through uncertainty quantification. While this methodology has proved mostly successful for the common heritage of Mars entry vehicles, it could be compromised by the added uncertainty from new decelerator technologies and stricter requirements for future missions. The motivation to reduce this uncertainty and leverage the reduction for increased controller performance leads to the overarching research objective, restated below:

Research Objective

Develop an entry controller tuning methodology that directly accounts for coupled, unsteady entry vehicle aerodynamic and control system behavior.

This overarching objective can be achieved by successfully meeting the following sub-objectives:

1. Develop a method for modeling the coupled, unsteady behavior of entry vehicle aero-

dynamics, control system, flight dynamics, and guidance and control systems.

2. Develop a method for incorporating this model into the process of entry controller tuning.

3. Quantify the change in entry controller performance between the proposed methodology and a baseline representing the state of the art.

The scope of the proposed work is subject to several assumptions to ensure feasibility of the analysis tasks:

1. The first assumption is that the methodology created and evaluated in one particular flight regime is similarly feasible across the whole range of continuum aerodynamics. As documented in Section 1.1.1, control authority is negligible within the rarified and transitional regimes due to low dynamic pressure.The entry phase will still end in the low supersonic regime, even with new decelerator or propulsion technologies.The literature review of aerodynamic models in Section 2.2 shows that the state-of-the-art aerodynamics and control system modeling methodology (the use of CFD and physical experiments) is consistent across this whole span. Therefore the proposed work will be scoped to the low supersonic regime. This range can be selected based on ease of model construction, capturing specific vehicle behavior, or matching the availability of existing experimental data.

2. The second assumption is the investigation using a single, fixed configuration for the vehicle and control system. The unsteady, coupled behavior of this system is assumed to be categorically representative of the behavior exhibited by all of the control systems identified in Section 1.1.1. As noted in Section 2.2.1, an entry vehicle inherently excites unsteady aerodynamic behavior, so the primary rationale for control system selection is the inclusion of interaction effects with either aerodynamics or flight dynamics. In addition, the flexibility of the vehicle is assumed to

be negligible, so that it can be modeled as a rigid body. This eliminates the need to model aeroelastic effects, reducing the problem complexity.

3. The third assumption concerns the consideration of computational cost in the evaluation of the proposed methodology. It is recognized that computational costs for coupled simulation exceeds the cost of the state-of-the-art methodology by orders of magnitude. However, computational power is constantly increasing. Moore's law provides a simple empirical relationship that the number of transistors in an integrated circuit will double every one to two years [181]. Therefore it can be expected that the physical cost will eventually become feasible.

   Continued advancements have been made in parallelization; planned exascale supercomputing systems will use GPU acceleration for increased throughput and efficiency [182, 183, 184]. CFD software is already being adapted to leverage this next generation of systems [185], and the capacity to run significant numbers of longer-duration time-accurate CFD simulations is being anticipated by NASA stakeholders [107]. Since the methodology is proposed for future use, an evaluation based on the cost of current computer systems is inappropriate. Therefore, comparison to the state of the art will be primarily based on performance. Where cost does come into focus, the evaluation will consider relative computational cost between alternatives.

4. The fourth assumption is that the computational models will act as a source of truth. This allows for the measurement of small absolute differences without measurement uncertainty.

5. The fifth assumption is that the value for instantaneous tracking error is suitable as a threshold of significance for the angular error results. During actual flight, the guidance algorithm has the ability to correct for persistent errors This assumption is made in addition to the assumptions involved in the caluculation of that value in ... In the absence of more detailed simulation that can provide an understanding into

the relationship between the instantaneous tracking error and the landing, the value provides the most concrete estimate of that relationship available.

4.2 Research Questions and Proposed Methodology

Investigation of the overarching research objective is directed by the major research questions posed in the following sections. The proposed methodology is presented in tandem with the research questions.

4.2.1 Modeling Unsteady, Coupled Behavior

Since the Research Objective seeks to modify the baseline entry controller tuning methodology, the first step to consider is the construction of the tools that are necessary for tuning, namely the flight simulation framework and its constituent disciplinary models. An examination of the physics made in Sections 2.1, 2.2, and 2.3 showed that the unsteady, coupled behavior of an entry vehicle are the result of the aerodynamics, control systems, and flight dynamics disciplines. Therefore, the accounting (or lack thereof) for this behavior is accomplished in these disciplinary models. This motivates the search for a solution within these disciplines as directed by Research Question 1:

Research Question 1

What is the appropriate methodology to model aerodynamics, control mechanisms, and flight dynamics for an entry vehicle with unsteady, coupled behavior?

Chapter 2 presented a literature review into the construction of the aerodynamics, control system, and flight dynamics models, and their incorporation into a flight simulation environment, from which several observations were made.

It was found that the baseline methodology cannot directly account for unsteady, coupled behavior because of development choices made in constructing the fundamental disciplinary models and the surrogate models. While the static aerodynamic models are capable of capturing time-accurate behavior, the CFD models use steady-state solutions, and the static wind tunnel models average out unsteady effects from the wake and control system interaction in the measurement process. The static control interaction models have the same limitation, but they also fail to measure the time-varying behavior of control system actuation. Coupling between the aerodynamics and flight dynamics disciplines is modeled, but control interaction is not modeled with dynamic flight. The surrogate models introduce a further reduction in the fidelity of the dynamic behavior.

In examining the gap between the models used in the state-of-the-art methodology and the actual physics of the problem, it becomes apparent that two fundamental changes must be made. Capturing unsteady behavior in each discipline requires time-accurate models, and capturing the interaction between the disciplines requires coupling between these models. Furthermore, it is apparent that the proposed methodology must use numerical simulation due to the infeasibility of modeling the desired behavior with ground-based physical testing. While ballistic range simulations provide an accurate model for free-flight behavior, they have no provision for modeling RCS control, the most common control system used by entry vehicles. Subscale or full-scale flight tests are prohibitively expensive, notwithstanding the fact that they can only be performed in Earth atmosphere.

However, it was also observed that among the disciplinary models, exist a subset of models that are capable of capturing the desired behavior. CFD-RBD models, described in Section 2.2.4, are capable of modeling the coupled time-accurate aerodynamics and 6DOF flight dynamics. These "CFD-in-the-loop flight simulations" are already in use for similar classes of vehicle, and are also being validated for entry vehicles. Furthermore, CFD can be used to model the time-accurate interaction between aerodynamics and control systems. Control surfaces can be modeled with overset meshes or mesh deformation, and propulsive control can be modeled with inlet boundary conditions.

This enables the capability of driving the control system actuation based on a guidance system or controller. Incorporation of the control system into the coupled simulation has been demonstrated by the "virtual fly-out" of projectiles. The problems faced by projectiles are similar to those of an entry vehicle: unsteady behavior, and control/aerodynamic interaction. This lends credence to the idea of using the same approach for entry vehicles. This idea forms the basis of Hypothesis 1:

Hypothesis 1: If aerodynamics, flight dynamics, guidance and control mechanisms are captured in a coupled, time-accurate model, then this model will simulate unsteady, coupled entry vehicle behavior that is not captured in the baseline model.

This hypothesis can be decomposed into two sub-hypotheses considering the measurement of the response from interactions between aerodynamics and flight dynamics, and combined aerodynamics/flight dynamics/control mechanism interaction:

Hypothesis 1.1: If an entry vehicle is simulated in free flight in the coupled and baseline models, then the difference in response due to unsteady, coupled behavior will be quantified.

Hypothesis 1.2: If an entry vehicle is simulated in with open-loop control actuation in the coupled and baseline models, then the difference in response due to unsteady, coupled behavior with control system interaction will be quantified.

However, entry vehicles and projectiles may have very different OMLs and fly at different flight regimes, so a definitive answer cannot be found through the literature search; the efficacy of the proposed method must be determined through experimentation.

Experiments 1.1 and 1.2 will compare the performance between flight simulations of an entry vehicle using two flight simulation models: a "baseline" representing the state of the art, and a "coupled" CFD-RBD-FCS model. The experiment is summarized below, with a detailed description in Chapter 7.

Experiments 1.1 and 1.2: Coupled Simulation Execution

As a prerequisite to Experiments 1.1 and 1.2, the baseline and coupled models must be constructed. The coupled model will be constructed from a CFD flow solver and a trajectory propagator, linked together in a CFD-RBD-FCS framework. The baseline model will be constructed as a representation of the state-of-the-art process for aerodynamic and control system modeling, including design of experiments and database generation. The baseline model is then executed using the trajectory propagator linked to these databases. Crucially, both of these models will use the same flow solver and trajectory propagator as the source of truth for the physics of the problem.

A series of four simulations will be run using the baseline and coupled models, listed in Table 4.1. Experiment 1.1 consists of runs 1 and 2, and Experiment 1.2 consists of runs 3 and 4. Each of the runs will be initialized at the same freestream and initial conditions, and executed for a duration of two times the natural oscillation period of the vehicle. Two of the runs will have no control input, to establish the unsteady aerodynamic behavior and its interaction with flight dynamics. The other two runs will have open-loop control. Simulations with commanded open-loop control will characterize control interaction with aerodynamics and flight dynamics. Trajectory results from the coupled simulation can also be used to verify that control performance metrics can be calculated.

Table 4.1: Experiment 1 simulations

Run	Model	Control	Command
1	Baseline	None	None
2	Coupled	None	None
3	Baseline	Open-loop	z-axis square wave
4	Coupled	Open-loop	z-axis square wave

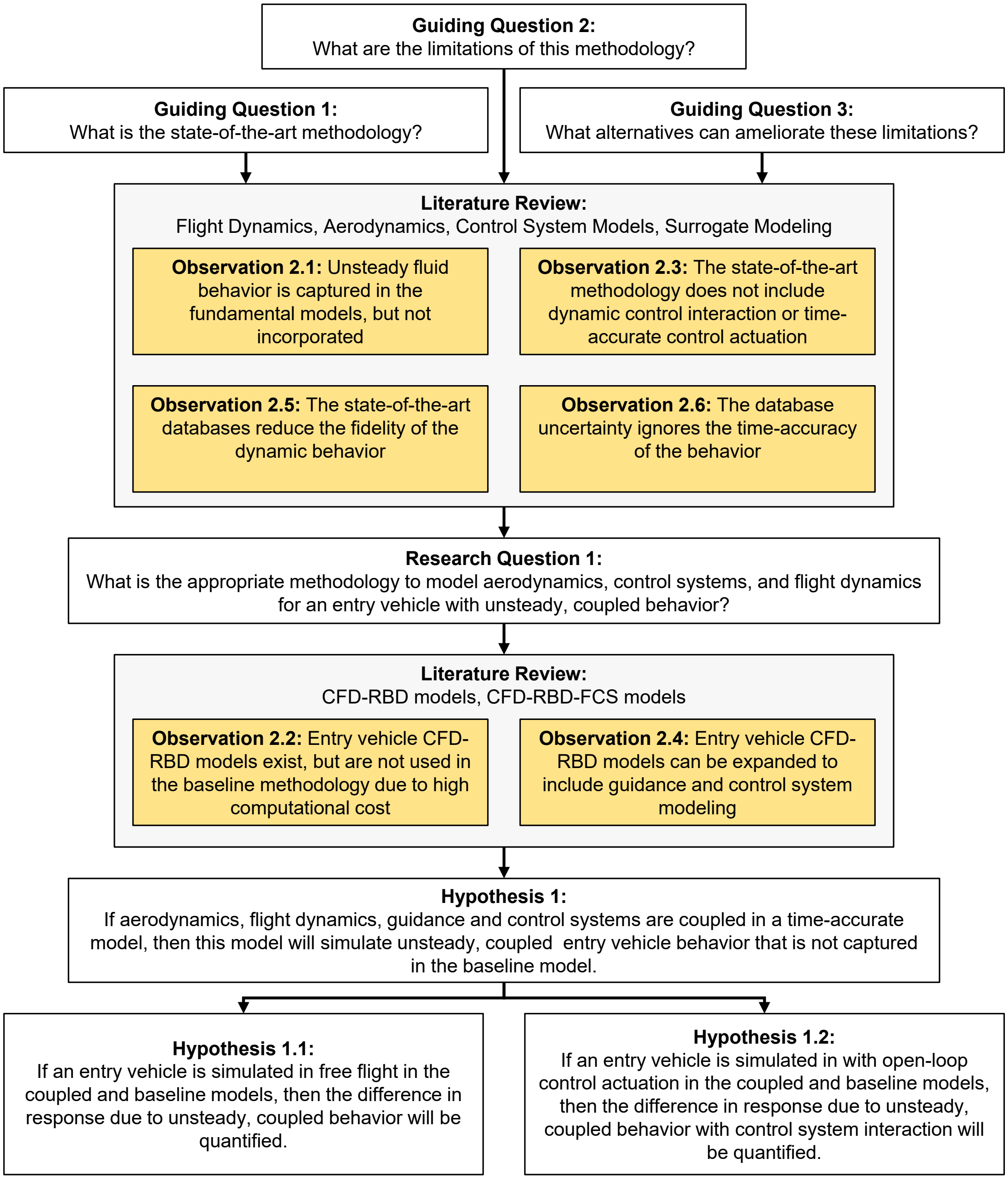
The differences in the resulting trajectories can be examined directly. In addition, each trajectory will be used to execute a forced motion simulation using the other model: the trajectory results from the baseline simulation will be used to drive the motion of a time-accurate CFD simulation, and the coupled model trajectories will be used to interpolate parameters from the baseline databases. This allows for the comparison of instantaneous aerodynamic forces and moments at the same vehicle state and time.

The success criteria for the experiment are that significant differences are observed in the trajectory components and aerodynamic parameters between the results from the coupled and baseline models. Significance in the trajectory results is defined with respect to the magnitude of controller performance metrics, such as overshoot and deadband, as determined from the literature review. Significance in the aerodynamic coefficients is defined with respect to the magnitude of aerodynamic and control interaction database uncertainties described in Section 2.4.1.

Figure 4.1 summarizes the argumentation used to develop Research Question 1 and Hypothesis 1.

4.2.2 Controller Tuning

A coupled trajectory simulation environment is proposed to replace the baseline environment, where the disciplinary models would be directly incorporated into the framework and coupled together for time-accurate simulation. Assuming that Hypothesis 1 is retained, the coupled model would provide higher-fidelity results than the baseline model. Any advantage of the high-fidelity results from the coupled model must be realized by leveraging the model for entry controller tuning. Figure 4.2 illustrates the proposed inclusion of the coupled model in the tuning process (compared to the baseline shown in Figure 3.1).



Guiding Question 2:
What are the limitations of this methodology?

Guiding Question 1:
What is the state-of-the-art methodology?

Guiding Question 3:
What alternatives can ameliorate these limitations?

Literature Review:
Flight Dynamics, Aerodynamics, Control System Models, Surrogate Modeling

Observation 2.1: Unsteady fluid behavior is captured in the fundamental models, but not incorporated

Observation 2.3: The state-of-the-art methodology does not include dynamic control interaction or time-accurate control actuation

Observation 2.5: The state-of-the-art databases reduce the fidelity of the dynamic behavior

Observation 2.6: The database uncertainty ignores the time-accuracy of the behavior

Research Question 1:
What is the appropriate methodology to model aerodynamics, control systems, and flight dynamics for an entry vehicle with unsteady, coupled behavior?

Literature Review:
CFD-RBD models, CFD-RBD-FCS models

Observation 2.2: Entry vehicle CFD-RBD models exist, but are not used in the baseline methodology due to high computational cost

Observation 2.4: Entry vehicle CFD-RBD models can be expanded to include guidance and control system modeling

Hypothesis 1:
If aerodynamics, flight dynamics, guidance and control systems are coupled in a time-accurate model, then this model will simulate unsteady, coupled entry vehicle behavior that is not captured in the baseline model.

Hypothesis 1.1:
If an entry vehicle is simulated in free flight in the coupled and baseline models, then the difference in response due to unsteady, coupled behavior will be quantified.

Hypothesis 1.2:
If an entry vehicle is simulated in with open-loop control actuation in the coupled and baseline models, then the difference in response due to unsteady, coupled behavior with control system interaction will be quantified.

Figure 4.1: Summary of problem formulation for Research Question 1

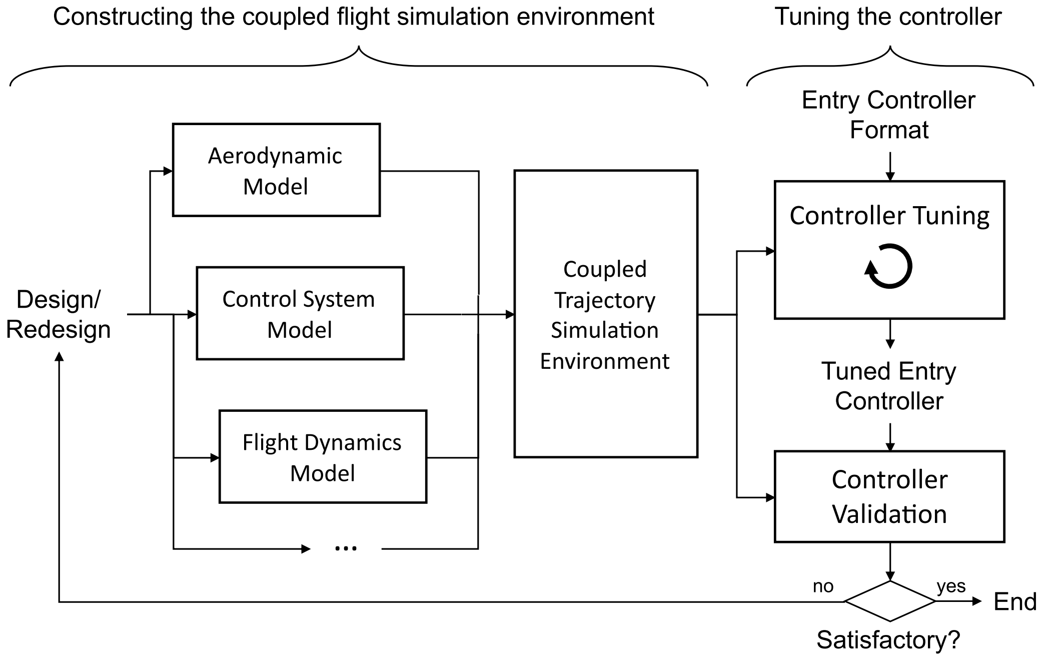


Figure 4.2: Proposed "coupled" entry controller tuning process

The guiding questions posed in Chapter 3 were formed in anticipation of a potential change in the tools used in the tuning process, and the subsequent literature review led to several findings that are applicable here.

From a compatibility standpoint, coupled model has the same inputs and outputs as the baseline model, meaning that it would be theoretically possible to substitute it into the controller tuning process. However, Observation 3.2 noted that some tuning methods make direct use of the disciplinary models for linearization, which is not possible for the coupled model.

The problem of cost is also detrimental. It was found that a large number of trajectory simulations are necessary to tune and/or validate a controller. This is feasible in the baseline methodology because the aerodynamic and control system surrogate models are inexpensive. However, Observation 2.2 noted that the execution time for CFD-RBD simulations is many orders of magnitude higher, and adding control system coupling into the model can only increase the execution time. Directly substituting the coupled simulation into the controller tuning methodology would be impractical, even when anticipating advancements in computing power. Therefore, a significant change in controller tuning methodology is required. This technical gap motivates Research Question 2:

Research Question 2

Given its increased cost, how can a coupled model viably be used to tune a controller for an entry vehicle exhibiting unsteady, coupled behavior?

The problem posed here bears similarity to the baseline controller tuning process. It was observed in Chapter 3 that the baseline flight simulation environment and the linearized model, although not named as such, function as lower- and higher-fidelity models for the controller tuning process.

In the literature, the desire to achieve high fidelity and low cost from a single model for which those two attributes are mutually exclusive has led to the use of multifidelity data fusion methods [186]. "Data fusion" methods combine multiple sources of data to perform a single operation, and "multifidelity" methods are data fusion methods that specifically use sources with different levels of fidelity. While multifidelity methods can use any number of sources, the simplest architecture would use two models. These models are generally referred to as "high-fidelity" and "low-fidelity." In the discussion of multifidelity models the descriptors "high" and "low" are relative terms, and may not reflect the actual fidelity of these models in comparison to all other models.

By combining models of different levels of fidelity, multifidelity methods can achieve both high accuracy and low cost where those attributes are most relevant. The cheaper, low-fidelity model provides rapid exploration of the space, and the expensive high-fidelity model provides an accurate response at specific regions of interest. The high-fidelity model is almost always the same model that designers desire to use, capturing the physics of the problem as completely as possible. The low-fidelity model is typically a simplification of the high-fidelity model. It might use simpler physics or geometry; reduce the dimensionality or linearize the problem; or run with less refinement or partial convergence [187]. The low-fidelity model could have completely independent progeny, or it could be a surrogate of the high-fidelity model [143].

These observations suggest the possibility of formalizing the use of multifidelity methods for controller tuning. This would allow the use of the coupled model because it would be supplemented by a lower-fidelity, lower-cost model. Therefore, Hypothesis 2 expresses that mutltifidelity methods are a potential solution to the problem posed by Research Question 2:

Hypothesis 2: If the controller tuning process uses models at multiple levels of fidelity, then the coupled simulation can viably be used to tune a controller.

As noted in Observation 3.1, controller tuning is fundamentally an optimization activity. Multifidelity optimization is a commonly-used method in aerospace applications, for multiple disciplines [188], with uncertainty [189], for robust design [190], etc. The field is also well established outside of aerospace research [187, 186]. The following sections formulate a methodology that incorporates multifidelity methods into the controller optimization process.

Multifidelity Controller Tuning Algorithm

The architecture for a generic multifidelity optimization algorithm is shown in Figure 4.3 [191]. At any particular iteration of the optimization, the algorithm chooses which model to run, how many times to run it, and what (sets of) inputs to use. The algorithm then executes the high- or low-fidelity model (or both), which generate a high- or low-fidelity response. These responses may then be combined by a data fusion method before being used in the optimizer. The algorithm will check to see if the stopping criteria are met. If the criteria are not met, then the optimizer will run to determine the guesses for the next loop. It may use the combined response, or it may use the relationship between the high-

and low-fidelity responses to calculate the next guess.

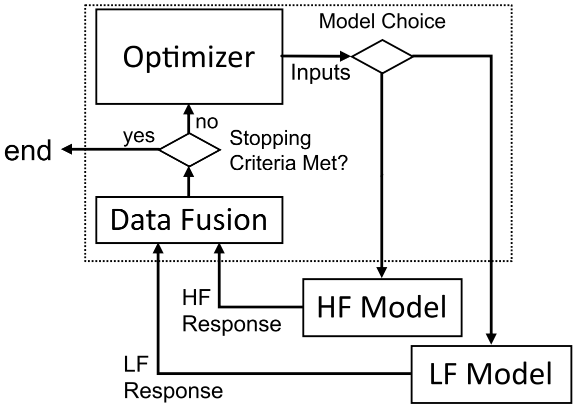


Figure 4.3: Multifidelity optimization algorithm architecture

The coupled simulation framework will serve as the high-fidelity model. A black-box representation of this model is shown in Figure 4.4. The model has inputs for initial state $\bar{x}_0$, controller gains $\boldsymbol{K}$, and commanded attitude angles α_c and β_c. In the paradigm of optimization, the primary outputs of this model are the post-processed controller performance metrics. Uniquely though, this model also outputs the time-series data for the trajectory and aerodynamic coefficients.

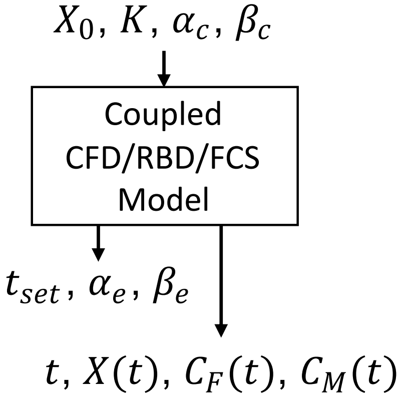

X_0, K, α_c, β_c
Coupled CFD/RBD/FCS Model
t_{set}, α_e, β_e
t, $X(t)$, $C_F(t)$, $C_M(t)$

Figure 4.4: High-fidelity model architecture

A “low-fidelity” model is needed to compliment the high-fidelity coupled model. The strategy for creating this model depends on two decisions. First, the hierarchical nature of the coupled simulation means that the low-fidelity model could be a wholesale replacement, or it could replace the disciplinary models. It has been observed that the flight dynamics and guidance models are reasonably inexpensive; there is little impetus to create lower-fidelity models for these disciplines. Therefore, the most effective way to reduce the computational cost of the low-fidelity model is to use lower-fidelity aerodynamic and control system models.

Second, the low-fidelity model could either be a surrogate or a reduced-order model. The use of an aerodynamic surrogate has three compelling factors: the aerodynamic response is likely to be less complex than the function for the controller performance metrics, making it easier to create a surrogate; the field of aerodynamic surrogate models is widely-explored, as was observed in Section 2.4, presenting a ready suite of alternatives; and uniquely for this problem, a surrogate could be made for the secondary output of the coupled model: the time-series aerodynamic and trajectory data. Whereas the output from the baseline aerodynamic database yields no new data, each execution of the coupled model creates new aerodynamic data — data that would otherwise be unused in the optimization process.

A flowchart of this architecture is shown in Figure 4.5. The low-fidelity model would operate like the coupled simulation, where the RBD and guidance models update the vehicle and control system state. It would then run the surrogate model, which would output the net aerodynamic coefficients.

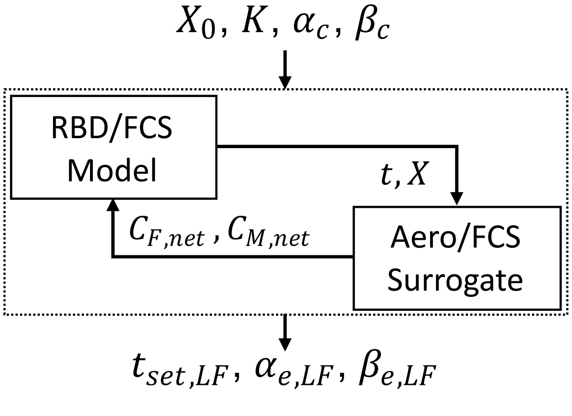

$X_0, K, \alpha_c, \beta_c$
RBD/FCS Model
t, X
$C_{F,net}, C_{M,net}$
Aero/FCS Surrogate
$t_{set,LF}, \alpha_{e,LF}, \beta_{e,LF}$

Figure 4.5: Low-fidelity model architecture

The proposed multifidelity controller tuning process is shown in Figure 4.6. The optimization algorithm, high-fidelity model, and low-fidelity model operate as described above. However, when the high-fidelity model is called, it also makes the time-series aerodynamic and trajectory data available to the low-fidelity surrogate. The low-fidelity surrogate can be retrained whenever new data is available.

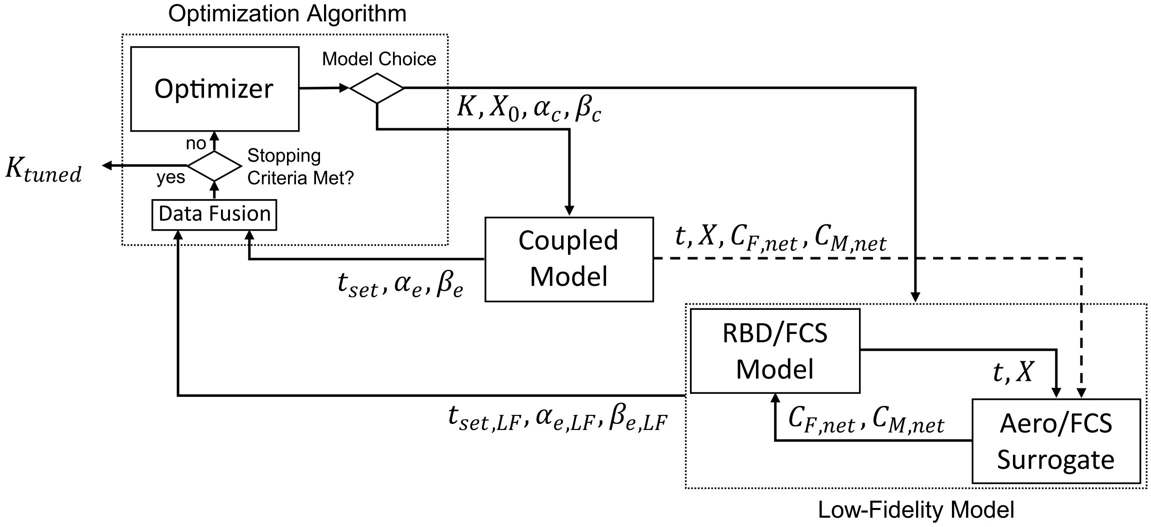


Figure 4.6: Multifidelity controller tuning process

The high-fidelity model and the RBD/FCS models used for the low-fidelity model have already been stipulated. To complete the process, a low-fidelity surrogate model and optimization algorithm must be selected. This motivates Research Questions 2.1 and 2.2.

Surrogate Modeling

Research Question 2.1 directs the investigation to fill the technical gap of the low-fidelity surrogate model. The requirements for the aerodynamic surrogate are driven by the qualities of the data from the coupled model from which it must be fitted. This data will contain the time-series aerodynamic coefficients, vehicle trajectory, and control system state. It can be considred as effectively continuous data with much higher local density compared to a space-filling pattern that might normally be used in a DOE to populate an aerodynamic database. As a result, the regions of extrapolation will also be larger for an individual run.

The differences between these types of data are illustrated in Figure 4.7, which compares the free-flight simulation of an entry vehicle to an aerodynamic database.

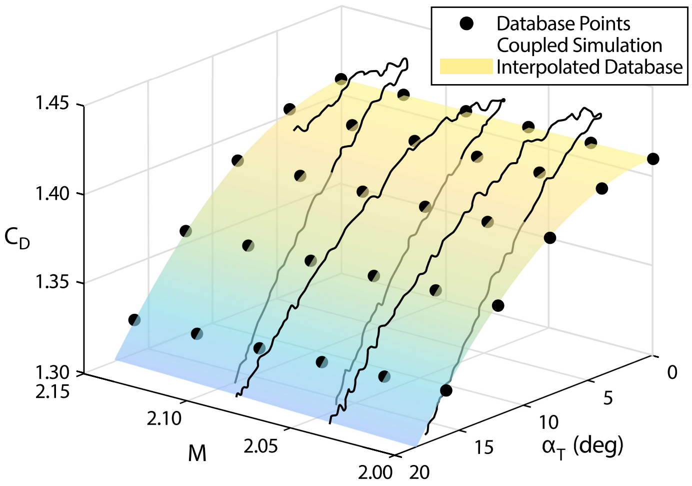


Figure 4.7: Comparison of aerodynamic data from a coupled simulation vs. a database

The surrogate is also subject to requirements based on its usage in the controller tuning process. The surrogate must be capable of accepting new data and being retrained during the process. There is also a strong desire for the surrogate to capture the time-dependent aerodynamics and control system interaction forces and moments. The error with respect to the high-fidelity results must converge as the training set size increases (i.e., as controller tuning proceeds).

Research Question 2.1

How can an aerodynamic surrogate model be constructed using time-series data from the coupled model as it is generated during the tuning process?

The literature review of surrogate models in Chapter 2 identified a number of models available as alternatives to the baseline aerodynamic database. Parameter estimation is considered due to its use in extracting data from dynamic experiments. However, the use of an assumed functional form is a limiting factor, especially for modeling the unsteady behavior of the control system. In contrast, neural networks do not assume a functional form. Two broad categories of ANN are considered: the feed-forward and recurrent networks.

As noted in Section 2.4.4, all of these models are trained with time-series data, matching the output of the coupled model. An RNN includes feedback connections that allow the time history of the response to be used as an input. Based on its universal approximation properties for dynamical systems, it is hypothesized that RNNs are the most suitable surrogate.

Hypothesis 2.1: If a neural-network-based surrogate model is fitted with time-series trajectory and aerodynamic data from the coupled simulation, then it will have improved error convergence to the coupled model over a baseline database.

However, the literature review for RNNs does not include entry vehicle applications, and recurrent networks are known to be sensitive to feedback error during training. Therefore, these models must be evaluated experimentally. Experiment 2 is designed to evaluate candidate surrogate model architectures based on their capability to model the unsteady, coupled behavior, as well as to develop the best procedures for training the surrogate.

Experiment 2: Surrogate Model Development

Experiment 2 simulates the training of the low-fidelity surrogate as it would be used in the multifidelity optimization methodology.

A set of eight coupled simulation runs are executed to provide a data set. These runs will have random gain designs and initial conditions. This training set is used to evaluate the different surrogate architectures.

For each surrogate architecture, a leave-k-out cross-validation is performed, splitting the runs into training and validation sets. Each surrogate design is trained multiple times, using an increasing fraction of the training set. The model fit error (MFE) and the model response error (MRE) to the aerodynamic data can be evaluated as a function of training set size. This serves as an evaluation of the surrogate on its own. Each of the surrogates will also be evaluated in the low-fidelity model by replicating the simulations used to build the data set. This provides an evaluation of the predicted controller performance as a function of training set size.

Model suitability will be evaluated based on the convergence toward the validation data. The model that achieves the lowest error with respect to the high-fidelity simulation with the smallest training set (i.e., number of high-fidelity runs required) will be judged to be the best.

Multifidelity Method

Research Question 2.2 pertains to the technical gap of the optimization process. With the high- and low-fidelity models identified, some multi-fidelity methodology is necessary to manage the data, in the form of control performance metrics, produced by each. Multiple architectures exist for multi-fidelity optimization, operating differently on the data produced by the models [191]. The key distinction is whether the methodology should adapt or fuse the metrics directly via a surrogate, or whether it should rely solely on the training connection between the high- and low-fidelity models.

Research Question 2.2

How can the multifidelity optimization method utilize the high- and low-fidelity controller metrics to reduce the computational cost of controller tuning?

The selection of this major design choice can be made by evaluating two methods representative of each architecture. A “default” architecture would use the results from the low- and high-fidelity models independently, in sequence, to find a satisfactory set of gains. This is akin to a filtering-based multi-fidelity method, a progression of the gain scheduling technique used in the state of the art. Convergence would rely on robustness of the gain designs, and the updated low-fidelity model after each optimization iteration.

As an alternative, a fusion-based method such as Cokriging would use information about the relationship between the high- and low-fidelity models. The covariance between the performance metrics outputs are calculated and used to develop the surrogate. The information can also be used to predict the variance of the response at off-design points, to determine the best design point to run to improve the model. Taking advantage of this additional direction is hypothesized to allow the method to achieve higher efficiency:

Hypothesis 2.2: If a data-fusion-based multifidelity optimization method is used to tune an entry controller, then it will have less computational cost than using a quasi-hierarchical filtering-based method for an equal performance.

Experiment 3 will perform controller tuning using two multi-fidelity optimization techniques, comparing their rate of convergence to a satisfactory set of control gains.

Experiment 3: Comparison of Multi-Fidelity Methods

Each of the multi-fidelity methods are executed to find a satisfactory set of control gains. Each method will be executed using the same number of calls to the high-fidelity coupled simulation for each gain design. Each run in the coupled simulation will be at the same duration, even if the vehicle achieves control before the end of the run. The low-fidelity surrogate will be re-fitted after each coupled simulation. The number of calls to the low-fidelity model will be allowed to vary. The bounds for each control gain will be determined by Monte Carlo results generated in Experiment 1. The control gains of the database-tuned controller will be used as the initial guesses for both methods.

The convergence toward a satisfactory set of control gains will be compared between both methods in terms of number of calls to the high-fidelity model. This is effectively equivalent to the total computational cost. The superior method is the one that achieves a satisfactory set of gains with the lowest number of runs.

Experiment 3 will also test Hypothesis 2; if either method yields a tuned controller, then Hypothesis 2 is retained.

Figure 4.8 summarizes the argumentation used to develop Research Question 2, its sub-questions, and hypotheses.

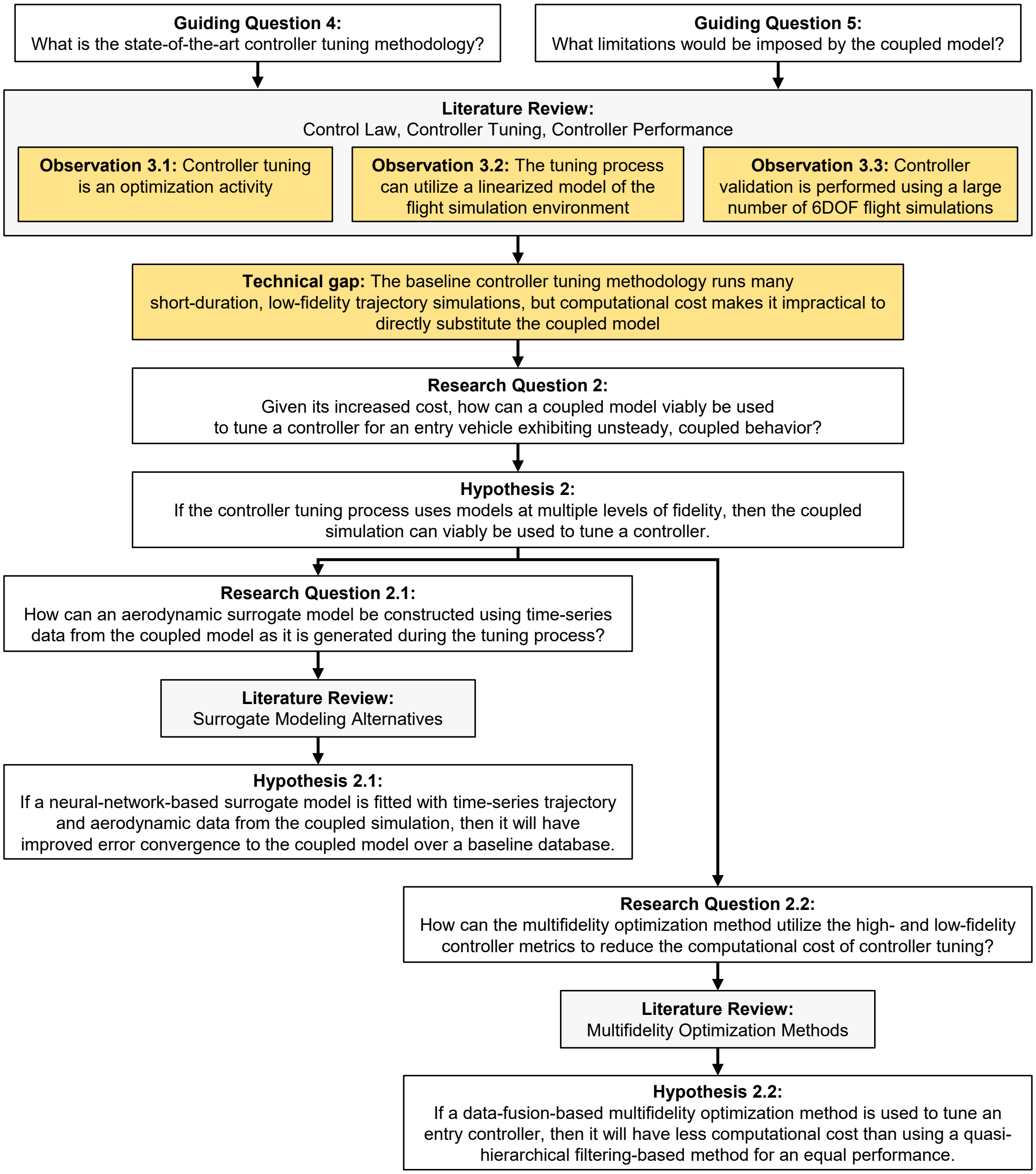

Guiding Question 4: What is the state-of-the-art controller tuning methodology?
Guiding Question 5: What limitations would be imposed by the coupled model?
Literature Review: Control Law, Controller Tuning, Controller Performance
Observation 3.1: Controller tuning is an optimization activity
Observation 3.2: The tuning process can utilize a linearized model of the flight simulation environment
Observation 3.3: Controller validation is performed using a large number of 6DOF flight simulations
Technical gap: The baseline controller tuning methodology runs many short-duration, low-fidelity trajectory simulations, but computational cost makes it impractical to directly substitute the coupled model
Research Question 2: Given its increased cost, how can a coupled model viably be used to tune a controller for an entry vehicle exhibiting unsteady, coupled behavior?
Hypothesis 2: If the controller tuning process uses models at multiple levels of fidelity, then the coupled simulation can viably be used to tune a controller.
Research Question 2.1: How can an aerodynamic surrogate model be constructed using time-series data from the coupled model as it is generated during the tuning process?
Literature Review: Surrogate Modeling Alternatives
Hypothesis 2.1: If a neural-network-based surrogate model is fitted with time-series trajectory and aerodynamic data from the coupled simulation, then it will have improved error convergence to the coupled model over a baseline database.
Research Question 2.2: How can the multifidelity optimization method utilize the high- and low-fidelity controller metrics to reduce the computational cost of controller tuning?
Literature Review: Multifidelity Optimization Methods
Hypothesis 2.2: If a data-fusion-based multifidelity optimization method is used to tune an entry controller, then it will have less computational cost than using a quasi-hierarchical filtering-based method for an equal performance.

Figure 4.8: Summary of problem formulation for Research Question 2

4.3 Demonstration of the Proposed Methodology

If Hypothesis 1 is retained, the coupled model proposed in Section 4.2.1 will be capable of modeling the unsteady, coupled behavior of an entry vehicle with an active control system, predicting significant differences in the behavior of the vehicle compared to the baseline flight simulation model. This satisfies the first sub-objective of the overarching research objective. From the surrogate model and multifidelity model alternatives proposed in Section 4.2.2, the experimentation will identify which methods can be used to leverage the coupled model in the controller tuning process. As a result, the proposed methodology will be complete, satisfying the second sub-objective. To completely satisfy the overarching research objective, the performance of the proposed methodology must be evaluated against the baseline methodology:

Overarching Hypothesis: If the entry controller is tuned using a coupled, unsteady aerodynamic and control system model, the performance of the entry controller will be improved compared to a controller tuned using the state-of-the-art methodology.

Experiment 4 acts as a demonstration for the Overarching Hypothesis.

Experiment 4: Comparison of Entry Controller Performance

Entry controllers tuned using the baseline and proposed methodologies are evaluated in both the coupled and baseline models by running the simulations listed in Table 4.2. The goals for controller gain optimization are:

- Trim the vehicle at $\alpha_t rim = -2.5°$.
- Minimize the root-mean-square (RMS) error between the α response and α_{cmd}.
- Minimize the worst-case results from a multitude of initial orientation and angular velocity conditions.

Each simulation is initialized at the same sets of freestream and initial conditions (that are distinct from those used in Experiment 1). The simulations are executed for a duration of three times the natural oscillation frequency of the vehicle, with closed-loop control to meet a commanded attitude. Post-processing of the trajectory results is used to find the controller performance metrics.

Table 4.2: Experiment 4 simulations

Run	Controller	Model
1	Baseline	Baseline
2	Baseline	Coupled
3	Coupled	Baseline
4	Coupled	Coupled

Comparing the results from runs 1 and 3 will demonstrate that the baseline controller is the best controller as evaluated by the baseline model, and that the performance improvement of the coupled controller is not just due to it being universally better. Comparing runs 1 and 2 will demonstrate that the performance of the baseline controller decreases when it is simulated in the coupled model. Finally, comparing runs 2 and 4 will demonstrate that the performance increase of the coupled controller relative to the baseline controller. Meeting these success criteria will demonstrate that the proposed controller tuning methodology recoups the performance losses identified by using the coupled model.

Figure 4.9 presents a summary of the problem formulation, including the research questions, hypotheses, and experiments.

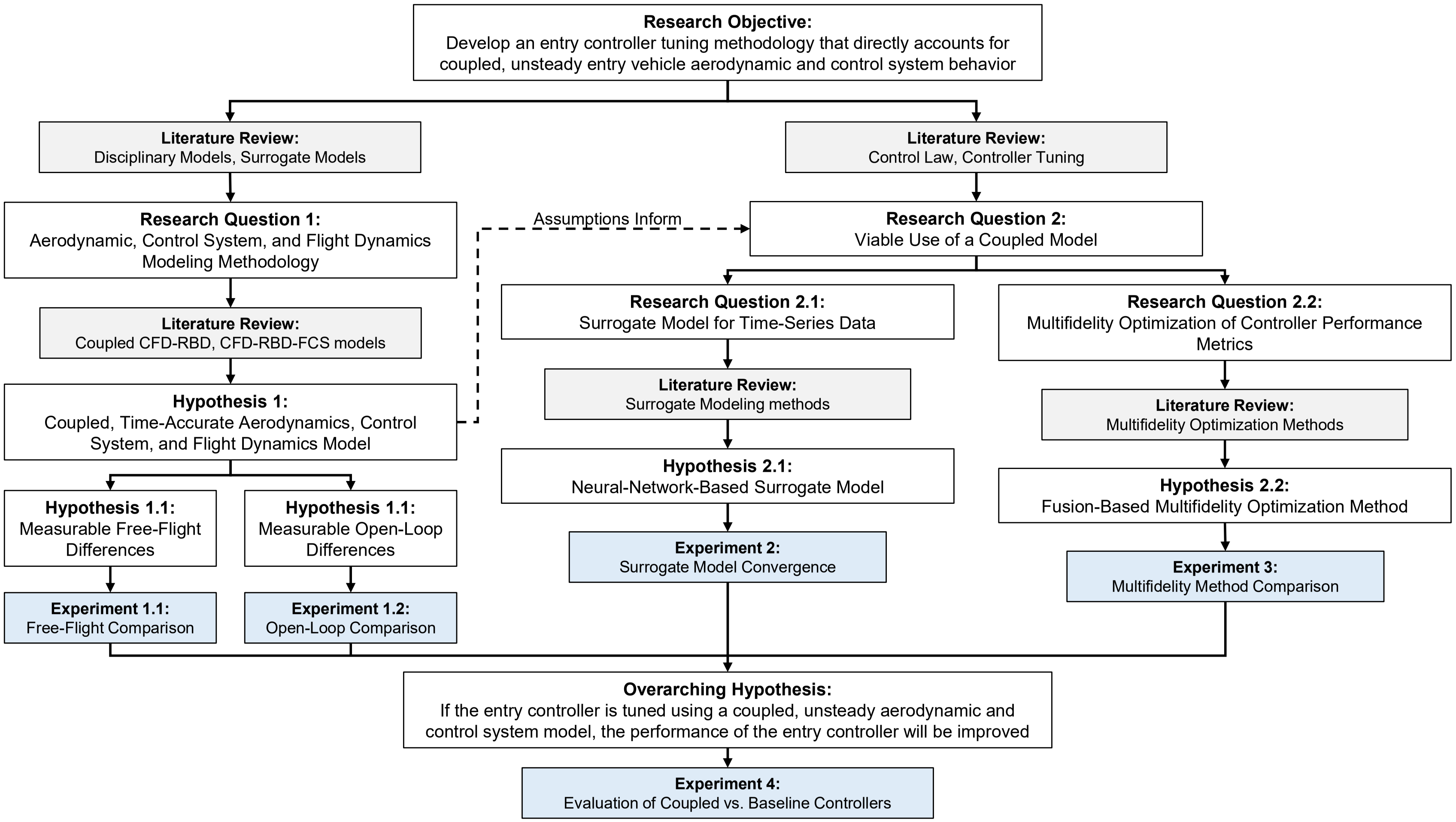

Research Objective:
Develop an entry controller tuning methodology that directly accounts for coupled, unsteady entry vehicle aerodynamic and control system behavior
Literature Review:
Disciplinary Models, Surrogate Models
Literature Review:
Control Law, Controller Tuning
Research Question 1:
Aerodynamic, Control System, and Flight Dynamics Modeling Methodology
Assumptions Inform
Research Question 2:
Viable Use of a Coupled Model
Literature Review:
Coupled CFD-RBD, CFD-RBD-FCS models
Research Question 2.1:
Surrogate Model for Time-Series Data
Research Question 2.2:
Multifidelity Optimization of Controller Performance Metrics
Hypothesis 1:
Coupled, Time-Accurate Aerodynamics, Control System, and Flight Dynamics Model
Literature Review:
Surrogate Modeling methods
Literature Review:
Multifidelity Optimization Methods
Hypothesis 2.1:
Neural-Network-Based Surrogate Model
Hypothesis 2.2:
Fusion-Based Multifidelity Optimization Method
Hypothesis 1.1:
Measurable Free-Flight Differences
Hypothesis 1.1:
Measurable Open-Loop Differences
Experiment 2:
Surrogate Model Convergence
Experiment 3:
Multifidelity Method Comparison
Experiment 1.1:
Free-Flight Comparison
Experiment 1.2:
Open-Loop Comparison
Overarching Hypothesis:
If the entry controller is tuned using a coupled, unsteady aerodynamic and control system model, the performance of the entry controller will be improved
Experiment 4:
Evaluation of Coupled vs. Baseline Controllers

Figure 4.9: Summary of problem formulation

CHAPTER 5
COUPLED AND BASELINE MULTIDISCIPLINARY MODELS

This chapter details the construction of both the coupled and baseline multidisciplinary models to be used for flight simulation.

5.1 Coupled Model

A coupled CFD/RBD/FCS model is required for each of the experiments outlined in Chapter 4. The two primary components of the model are the CFD flow solver and the trajectory propagator. The description of the coupled model in Section 4.2.1 is decomposed into a set of requirements for the flow solver:

1. The flow solver must be able to run time-accurate simulations of an entry vehicle with the proper solver settings to model the flow features around the vehicle; namely, the forebody shock structures, flow separation, and subsonic recirculation. Based on the literature in Section 2.2.4, the proper model would use the Detached Eddy Simulation (DES) modification of the Reynolds-Averaged Navier-Stokes (RANS) equations.

2. The flow solver must be able to account for rigid grid motion in the equations of flow. In essence, the flow solver must use the arbitrary Lagrangian-Eulerian (ALE) formulation of the equations to allow grid nodes to be moved in an arbitrary manner.

3. The flow solver must be able to model the effect of the control mechanism on the OML (e.g., including propulsion inlet boundary conditions for RCS simulation).

4. The flow solver must be capable of starting a simulation with arbitrary initial vehicle state conditions. At a minimum, this would include arbitrary orientation.

The baseline model will also require the use of CFD to populate the aerodynamic surrogate. The maximum utility can be gained if the flow solver selected for the coupled model is also used for the baseline model:

5. The flow solver must be capable of independent use for the calculation of static aerodynamic coefficients for an aerodynamic database.

Similarly, a set of requirements is composed for the trajectory propagator:

1. The trajectory propagator must be able to model 6DOF rigid body motion. At a minimum, this must be with respect to a single inertial reference frame as specified in Section 2.1.

2. The trajectory propagator must be able to model an entry controller with the functionality as identified from the literature in Section 3.1.

3. The trajectory propagator must be able to model the physics of the control mechanism that do not interact with the OML of the vehicle (e.g., the throttle lag and response curve of a thruster for RCS simulation).

No existing CFD-RBD capability was identified from the literature that met all of these requirements. The most significant technical gaps were the capability to model arbitrary control mechanism actuation based on a GN&C model with the complexity expected for entry vehicles. However, the search for software tools can be expanded if independent flow solver and trajectory propagator tools are considered. Whereas existing CFD-RBD capabilities (e.g., Kestrel or US3D FF-CFD) have already accomplished the tasks of communication between the processes, the technical approach must also handle these tasks. Thus, a derived set of requirements must be constructed. For the flow solver these are:

1. The flow solver must be capable of setting the grid motion based on the 6DOF rotation and translation components from an external source at each time step.

2. The flow solver must be capable of setting the control mechanism state from an external source at each time step.

3. The flow solver must allow for the solver to be paused and restarted at the frequency of one time step, in order to allow the simulation to wait for the execution of the RBD model and the calculation of the new vehicle motion state.

4. The flow solver must be capable of communicating the net aerodynamic force and moment loads to the trajectory propagator at each time step.

And for the trajectory propagator:

1. The trajectory propagator must be capable of communicating the vehicle motion state in the reference frame required by the flow solver at each time step.

2. The trajectory propagator must be capable of communicating the control mechanism state to the flow solver at each time step.

3. The trajectory propagator must allow for the propagation to be paused and restarted at the frequency of one time step, to allow the simulation to wait for the flow solver execution.

4. The trajectory propagator must be capable of reading the net aerodynamic loads from the flow solver at each time step.

5. The trajectory propagator must allow for modification or alteration of the numerical integration scheme to allow for monotonic increase of the simulation time within the flow solver.

These sets of requirements were used for the selection of software tools.

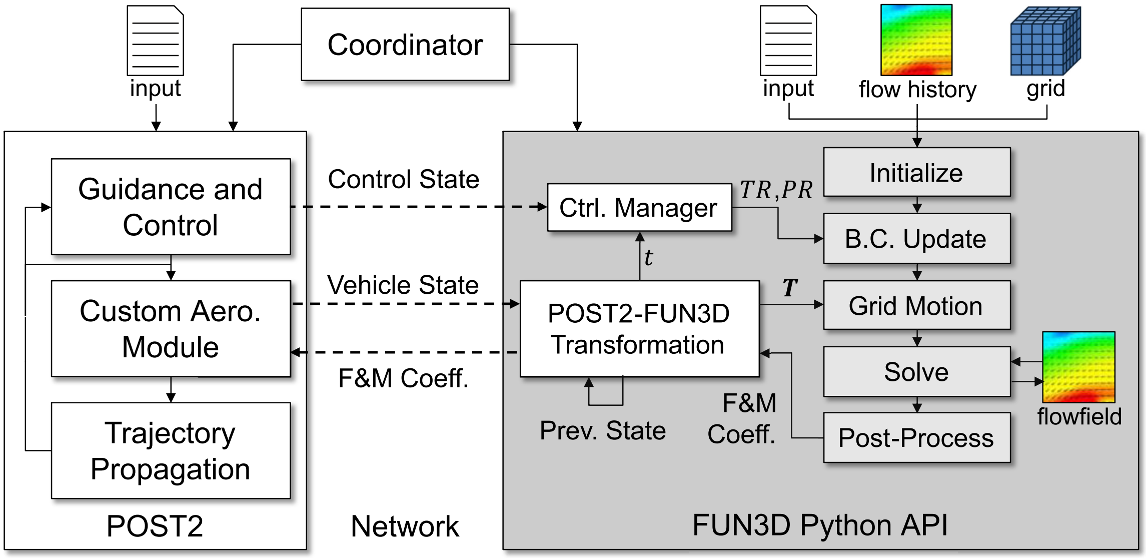


Figure 5.1: Coupled model architecture

5.1.1 Model Architecture

A high-level overview of model operation is shown in Figure 5.1.

The model uses independent CFD and trajectory propagator processes to perform the CFD-RBD simulation. A Python3-based wrapper handles communication via network sockets and performs data transformation.

The coupling is solved using a modified, staggered nonlinear block-Gauss-Seidel algorithm [192]. Both codes operate using the same, constant physical timestep; at each time step, FUN3D receives vehicle position and orientation from POST2, updates the mesh transformation, and computes the flow state. The state update calculated by POST2 in the previous step is held constant for the duration of the CFD iteration at this step (i.e., there is no fluid-structure interaction iteration). POST2 receives the aerodynamic coefficients, calculates the forces and moments, and uses numerical integration to update the vehicle state.

5.1.2 Flow Solver

Fully Unstructured Navier-Stokes 3D (FUN3D) is a production flow analysis and design tool developed by NASA Langley Research Center [193]. From its inception in the late 1980s, FUN3D has been developed into a suite of tools for flow analysis, mesh adaptation, and design optimization [87, 194]. The flow solver is capable of solving the RANS or Euler equations for steady-state or time-accurate flow. FUN3D uses a node-based finite-volume discretization capable of solving unstructured or mixed-element meshes [193]. It can model perfect gas or a generic gas with multiple chemical species; in compressible or incompressible flows; at thermochemical equilibrium or non-equilibrium. FUN3D uses a second-order-accurate spatial discretization, with options for first- to fourth-order temporal discretization (with temporal error controllers). The software includes a variety of turbulence models, including modification of RANS for DES, and flux splitting schemes [193].

FUN3D was selected as the flow solver based on its capability to model the problem of interest in the hypersonic and supersonic flight regime[1]. FUN3D has a long history of usage in government and industry projects, including for analysis of the Phoenix [12] and MSL entry vehicles [11], as well as for the ADEPT SR-1 flight test geometry [195]. FUN3D can model boundary conditions for internal flow and propulsion simulation, including inlets and nozzles. The flow solver can also be compiled with overset mesh capability to model control surface actuation [196, 193].

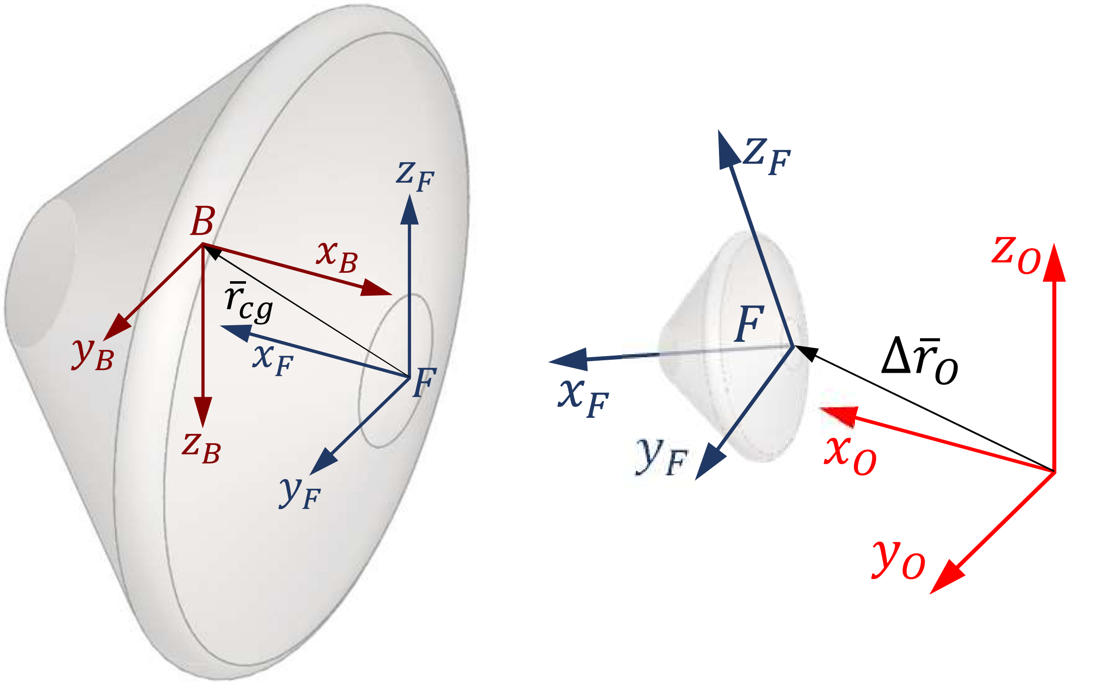
FUN3D uses the ALE formulation of the governing equations defined in Section 2.2.3, allowing for both rigid mesh movement and mesh deformation [86]. Under rigid mesh movement, the points in the grid are transformed with a common rotation and translation. For mesh deformation, unique transformations can be applied at every grid point. Rigid motion and deformation can be combined in the same simulation. This grid motion capability allows for the results of the RBD model to be communicated to the flow solver.

[1] This work specifically uses FUN3D version 13.7.

FUN3D also allows for fine control of the execution down to the individual time step using a Python-based application programming interface (API) [193]. Thus FUN3D meets the requirements for the flow solver specified above.

Reference Frames

FUN3D tracks calculations in two reference frames, defined in Figure 5.2. The first frame is the body reference frame F, shown in Figure 5.2a in relation to the body frame B. The frame is aligned to the mesh coordinates with the x-axis pointing aft, the y-axis pointing right, and the z-axis pointing up. The origin is at an arbitrary point, conventionally the nose of the vehicle. The second frame is the freestream frame O, shown in Figure 5.2b. The O frame can be conceptualized as an inertial frame that moves along the freestream velocity vector.



(a) Body Reference frame

(b) Freestream frame

Figure 5.2: FUN3D Reference Frames

Nondimensionalization

Calculations in FUN3D are performed with non-dimensional quantities. Any parameters input into FUN3D through file I/O or the Python API must be non-dimensionalized by length and speed. For a length measurement x, the non-dimensionalized quantity x_{nd} is found by Equation 5.1:

$$x_{nd} = x/(L_{ref}/L_{grid}) \tag{5.1}$$

where L_{ref} is the reference length of the grid in dimensional units, and L_{grid} is the same length expressed in grid units. For reference area S_{ref}, the non-dimensionalization is:

$$S_{nd,ref} = S_{ref}/(L_{ref}/L_{grid})^2 \tag{5.2}$$

Time is non-dimensionalized by:

$$t_{nd} = ta_{ref}/(L_{ref}/L_{grid}) \tag{5.3}$$

where a_{ref} is the reference speed of sound in dimensional units. Thus, a speed s is non-dimensionalized by Equation 5.4:

$$s_{nd} = s/a_{ref} \tag{5.4}$$

The density ρ, temperature T, and pressure P at a point in the flow can be non-dimensionalized as:

$$\rho_{nd} = \rho/\rho_{ref} \quad T_{nd} = T/T_{ref} \quad P_{nd} = P/(P_{ref}a_{ref}^2) \tag{5.5}$$

where ρ_{ref}, T_{ref}, and p_{ref} are the freestream reference quantities in dimensional units. The Reynolds number per grid unit Re_{nd} is found by Equation 5.6:

$$\mathrm{Re}_{nd} = \mathrm{Re}/L_{ref} = \frac{\rho_{ref}s_{ref}L_{ref}}{\mu_{ref}L_{grid}} \tag{5.6}$$

where s_{ref} is the reference speed and μ_{ref} is the dynamic viscosity, both in dimensional units.

Rigid Body Motion

General rigid body motion can be specified in FUN3D as a mapping between its position and orientation at initialization and at time t. For a reference point at the initial position $\bar{r}_F$ in the body reference frame, a 4×4 transformation matrix performs translation and rotation to the new position $\bar{r}_O$ in the observer frame. The transformation matrix is expressed in Equation 5.7:

$$\begin{bmatrix} \bar{x}_O \\ 1 \end{bmatrix} = \begin{bmatrix} & \boldsymbol{R}_{OF} & & \Delta\bar{x}_O \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_F \\ 1 \end{bmatrix} \tag{5.7}$$

where the 3×3 submatrix $\boldsymbol{R}_{OF}$ in the upper left is an orthonormal rotation about the origin, and the upper-right column vector $\Delta\bar{x}_O$ defines a translation. The bottom row of each matrix remains constant.

Specifying rigid body motion also requires updating the position of the body's center of gravity $\bar{r}_{cg,O}$, which is also used as the moment center for integrating the aerodynamic coefficients. Given a schedule of transformation matrices and times, FUN3D uses linear interpolation to find the position at the desired solution time. With the change in position calculated at each time step, the velocity and acceleration of the body with respect to O is implicitly defined.

Aerodynamic Coefficients

The axial, normal, and side force coefficients C_A, C_N, and C_Y are calculated by integrating the normal and shear stress along the x-, y-, and z-axes of the body reference frame, respectively. Equation 5.8 shows the integration of C_A as an example.

$$C_A = \frac{\int_S [P_{nd} n_x + \tau_x] dS}{S_{nd,ref} Q_{nd,ref}} \tag{5.8}$$

where τ_x is the shear stress in the x-axis and $Q_{nd,ref}$ is the non-dimensionalized reference dynamic pressure. The moment coefficients C_{LL}, C_M, and C_{LN} about the x-, y-, and z-axes of the body reference frame, respectively, are calculated by integrating the normal and shear stress multiplied by the moment arm about the moment center.

$$C_M = \frac{\int_S [P_{nd}(r_x n_z - r_z n_x) + r_z \tau_x - r_x \tau_z] dS}{S_{nd,ref} L_{grid} Q_{nd,ref}} \tag{5.9}$$

Note that the coefficients are calculated with respect to the freestream dynamic pressure, which may not be equal to the dynamic pressure on a body with relative velocity to the O frame.

5.1.3 Trajectory Propagator and Control Mechanism Model

Although FUN3D can be compiled with a 6DOF RBD trajectory propagator [67], this RBD code has limited capabilities in the selection of gravitational models, numerical integration scheme, or initial conditions. It also does not allow for guidance and control simulation. An alternative implementation uses an independent process to perform the RBD simulation and drive the motion in the CFD flow solver. This allows the flow solver to be linked with a state-of-the-art trajectory propagation software, with greatly expanded capability; it can be coupled with other disciplines such as propulsion or guidance, navigation, and control (GN&C).

The Program to Optimize Simulated Trajectories II (POST2) is a trajectory propagation and optimization program developed at NASA Langley Research Center [197]. POST2 models the trajectory of one or more vehicles in flight about one or more attracting bodies, inside or outside of an atmosphere. The propagator can solve problems in 3DOF by integrating the translational equations of motion, or in 6DOF by integrating the rotational equations of motion as well. The planetary model is generalized, with arbitrary rotation, gravitational (oblateness), and atmospheric parameters. In addition, POST2 is capable of modeling complex vehicle mass properties, such as a composite vehicle with multiple, internal, time-varying masses.

POST2 is a state of the art trajectory propagation software, used for trajectory analysis of orbital launch vehicles [198], and Earth and Mars entry vehicles [199, 50, 200]. POST2 is designed with great flexibility in enabling custom code modules. Simulations are run with POST2 acting as the integrator for disciplinary models, such as aerodynamics, GN&C, and propulsion that are specifically built for the vehicle under study [19]. This is the key reason for which POST2 is used as the trajectory propagator in this work[2]. The trajectory propagator models direct effects from the control system (including interaction between the control system and flight dynamics), while the interaction between the control system and aerodynamics are modeled in the flow solver. Thus POST2 satisfies the requirements for the trajectory propagator specified above.

Reference Frames

The reference frames used in POST2 are shown in Figure 5.3. For most CFD-RBD models, the inertial frame I is assumed to be the local horizontal frame in a "Flat Earth" model with a uniform gravitational field. However for a spheroidal planetary model with nonzero rotation, I represents the Earth-centered inertial (ECI) frame. The ECI frame is oriented with the x-axis pointing through the equator and zero longitude at the epoch, the y-axis pointing through the equator at a perpendicular direction, and the z-axis pointing through the planet's axis of rotation. As the planet rotates with angular velocity ω_e it carries with it the Earth-centered rotating frame E, whose x- and y-axes remain pointing through 0° and 90° longitude, respectively.

The local vertical frame V has its origin on the planet's surface directly below the vehicle. Its x- and y-axes lie in the horizontal plane with the x-axis pointing North. The z-axis lies along the local vertical pointing inward towards the planet (note that for an oblate

[2] This work specifically used POST2 version 2.430D.

spheroid this may not point directly at the planet's center). The orientation of V with respect to E is given in terms of the longitude λ and geodetic latitude Φ angles. POST2 is configured to read the aerodynamic coefficients in the body reference frame F.

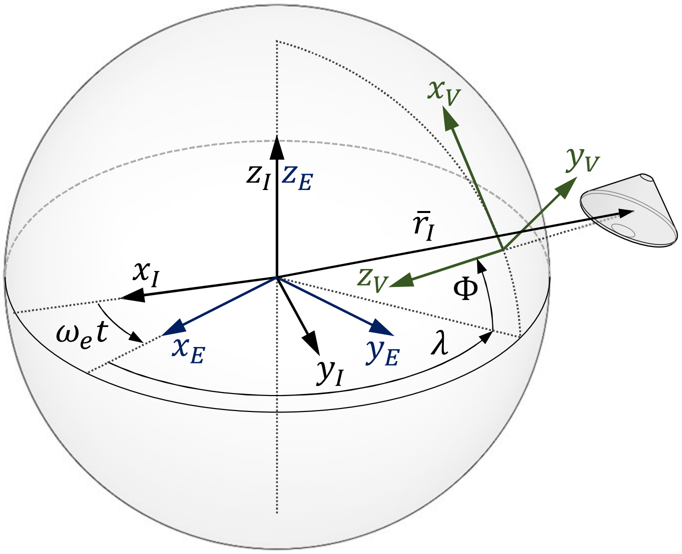


Figure 5.3: POST2 reference frames

Numerical Integration

Time integration is performed using the Runge-Kutta 4th-order (RK4) method, with aerodynamic force and moment coefficients held constant during integration across the RK4 process each time step. Using RK4, or many other numerical integration schemes, as intended would require resetting the vehicle to its state and time at the beginning of the step, whereas the flow solver expects the flow history to be monotonic. Performing the full numerical integration scheme requires expensive manipulation of multiple flow history versions. Using constant aerodynamic forces and moments allows FUN3D to keep a single flow field in memory and preserve the monotonic nature of the solution. While using constant force and moment coefficients reduces the order of accuracy of the integration scheme, the time step is orders of magnitude smaller than typical RBD time steps, mitigating the error propagation [65]. This is a common approach for CFD-RBD simulation [67, 108, 126].

Guidance and Control Simulation

The guidance and control system model is built as a custom code module for POST2. With the assumption of perfect navigation, the state variables can be read directly from the trajectory propagator. The calculations for updating the guidance algorithm can be run at the same time step as the simulation, or at a reduced frequency. A custom variable is created to store the control system state vector, which is visible to other components in POST2. The architecture of the guidance algorithm and controller are documented in the next chapter.

5.1.4 Data Transformation

Transforming the vehicle state between POST2 and FUN3D requires calculations in each of the reference frames defined in the sections above. The rotation, position, and angular and linear velocity are calculated at each time step to update the transformation in FUN3D. Similarly, the coefficients are transformed back into the POST2 reference frames at the end of each time step.

Intermediary Frame

Translation and orientation are defined in FUN3D with respect to an observer frame O, which moves at a constant velocity relative to the atmosphere. However, the orientation of the O frame is arbitrary. It is therefore convenient to define an intermediary frame P as shown in Figure 5.4. The origin of this frame is coincident with the initial local vertical frame V_0, with the x-axis pointing along the vehicle's initial velocity vector v_{0,V_0} and the y-axis along the local horizontal. The orientation of P with respect to V_0 is defined by the azimuth angle Ψ and flight path angle γ. The O frame is defined as a 180° rotation about the y-axis from the P frame. Using the intermediary frame decouples specification of freestream conditions in FUN3D from the initial orientation within POST2, making it possible to change the azimuth and flight path of the vehicle without having to recreate the initial flow field.

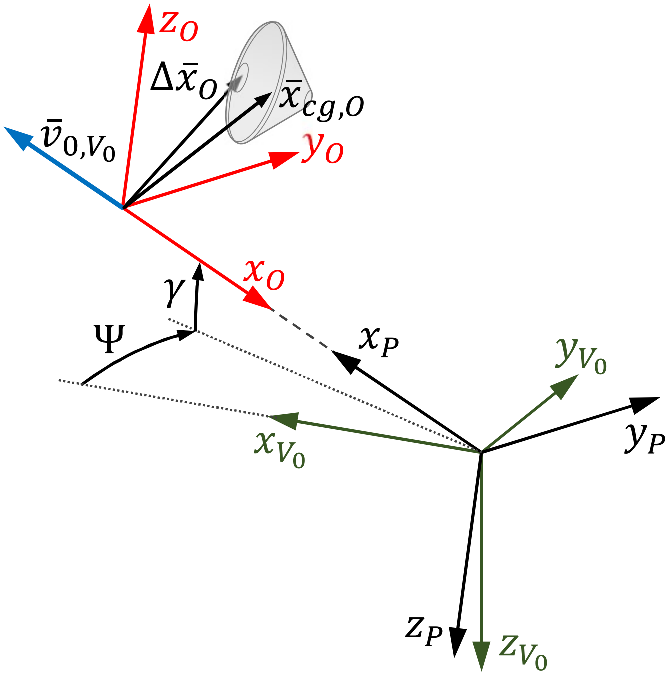


Figure 5.4: Vertical, Intermediary, and Freestream frames

Angular Components

FUN3D requires the orientation of the vehicle to be input as a rotation matrix between the O and F frames. This matrix is calculated using Equation 5.10:

$$\boldsymbol{R}_{OF} = \boldsymbol{R}_{OP}\boldsymbol{R}_{PB}\boldsymbol{R}_{BF}, \qquad \boldsymbol{R}_{OP} = \boldsymbol{R}_{BF} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{5.10}$$

where $\boldsymbol{R}_{PB}$ is the transformation between the Intermediary and Body frames. $\boldsymbol{R}_{PB}$ is calculated using Equation 5.11:

$$\boldsymbol{R}_{PB} = \boldsymbol{R}_{PV_0}\boldsymbol{R}_{V_0E}\boldsymbol{R}_{EV}\boldsymbol{R}_{VB} \tag{5.11}$$

$\boldsymbol{R}_{PV_0}$ is the rotation matrix between the intermediate and initial vertical frames:

$$\boldsymbol{R}_{PV_0} = \begin{bmatrix} \cos\Psi\cos\gamma & \cos\gamma\sin\Psi & -\sin\gamma \\ -\sin\Psi & \cos\Psi & 0 \\ \cos\Psi\sin\gamma & \sin\Psi\sin\gamma & \cos\gamma \end{bmatrix} \tag{5.12}$$

where γ and Ψ are available from POST2. $\boldsymbol{R}_{V_0E}$ and $\boldsymbol{R}_{EV}$ are the 3-2 rotation matrices calculated using the initial and current geodetic latitude & longitude, respectively.

$$\boldsymbol{R}_{V_0E} = \begin{bmatrix} -\cos\lambda_0\sin\Phi_0 & -\sin\Phi_0\sin\lambda_0 & \cos\Phi_0 \\ -\sin\lambda_0 & \cos\lambda_0 & 0 \\ -\cos\Phi_0\cos\lambda_0 & -\cos\Phi_0\sin\lambda_0 & -\sin\Phi_0 \end{bmatrix} \tag{5.13}$$

$$\boldsymbol{R}_{VE} = \begin{bmatrix} -\cos\lambda\sin\Phi & -\sin\Phi\sin\lambda & \cos\Phi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\Phi\cos\lambda & -\cos\Phi\sin\lambda & -\sin\Phi \end{bmatrix} \tag{5.14}$$

$\boldsymbol{R}_{VB}$ is the 3-2-1 rotation matrix calculated from the relative Euler angles ϕ, θ, and ψ, with the same form as Equation 2.5. The Euler angles, latitude, and longitude are available directly within POST2.

The angular velocity in the O frame is calculated using Equation 5.15:

$$\bar{\omega}_{rel,O} = \boldsymbol{R}_{OP}\boldsymbol{R}_{PB}\bar{\omega}_{rel,B} \tag{5.15}$$

where $\bar{\omega}_{rel,B}$ is the angular velocity of the vehicle relative to the V frame, available directly from POST2. This is calculated using Equation 5.16:

$$\bar{\omega}_{rel,B} = \bar{\omega}_B - \boldsymbol{R}_{BI}[0 \quad 0 \quad \omega_{earth}]^T - \boldsymbol{R}_{BV}\bar{\omega}_{ned} \tag{5.16}$$

where $\bar{\omega}_{ned}$ is the angular velocity of the vertical frame relative to the E frame:

$$\bar{\omega}_{ned} = [\dot{\lambda}\cos\Phi \quad -\dot{\Phi} \quad -\dot{\lambda}\sin\Phi]^T \tag{5.17}$$

Linear Components

FUN3D requires a center of gravity position and a grid translation vector, both in the O frame. POST2 provides the position vector in the I frame. The position in the ECR frame, $\bar{x}_E$, is calculated using Equation 5.18:

$$\bar{x}_E = \boldsymbol{R}_{EI}\bar{x}_I \tag{5.18}$$

where $\boldsymbol{R}_{EI}$ is the transformation between the ECI and ECEF frames. This rotation matrix is calculated as:

$$\boldsymbol{R}_{EI} = \begin{bmatrix} \cos\omega_{earth}t & \sin\omega_{earth}t & 0 \\ -\sin\omega_{earth}t & \cos\omega_{earth}t & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.19}$$

where ω_{earth} is the magnitude of the planetary rotation rate and t is the time since epoch. The displacement is calculated relative to the initial position:

$$\Delta\bar{x}_E = \bar{x}_E - \bar{x}_{0,E} \tag{5.20}$$

Transforming the displacement into the observer frame must account for that frame's velocity. The position in the V_0 frame is calculated using Equation 5.21:

$$\Delta\bar{x}_{V_0} = \boldsymbol{R}_{V_0E}\Delta\bar{x}_E - \bar{v}_{0,V_0}t \tag{5.21}$$

where $\bar{v}_{0,V_0}$ is calculated as:

$$\bar{v}_{0,V_0} = s_0[\cos\gamma\cos\Psi \quad \cos\gamma\sin\Psi \quad -\sin\gamma]^T \tag{5.22}$$

where s_0 is the initial freestream speed. The position in the O frame is calculated using Equation 5.23:

$$\Delta\bar{x}_O = \boldsymbol{R}_{OP}\boldsymbol{R}_{PV_0}\Delta\bar{x}_{V_0} + \bar{x}_{cg,0,O} \tag{5.23}$$

where $\bar{x}_{cg,0,O}$ is the initial position of the center of gravity in the O frame. The grid translation vector is calculated as:

$$\Delta\bar{x}_{g,O} = \Delta\bar{x}_O - \boldsymbol{R}_{OF}\bar{x}_{cg,F} \tag{5.24}$$

where $\bar{x}_{cg,F}$ is the center of gravity in the body reference frame.

The velocity vector is calculated using Equation 5.25:

$$\bar{v}_O = \boldsymbol{R}_{OE}(\boldsymbol{R}_{EI}\bar{v}_I - \bar{x}_E \times [0 \quad 0 \quad \omega_{earth}]^T), \qquad \boldsymbol{R}_{OE} = \boldsymbol{R}_{OP}\boldsymbol{R}_{PB}\boldsymbol{R}_{BE} \tag{5.25}$$

Aerodynamic Coefficients

FUN3D provides the axial, side, and normal force coefficients C_A, C_Y, and C_N in the Body frame that POST2 expects, so no transformation is needed for the force coefficients. The moment coefficients are calculated in the Body Reference frame, so these are transformed into the Body frame using Equation 5.26:

$$\begin{bmatrix} C_{LL} \\ C_M \\ C_{LN} \end{bmatrix}_B = \boldsymbol{R}_{BF} \begin{bmatrix} C_{LL} \\ C_M \\ C_{LN} \end{bmatrix}_F \tag{5.26}$$

These coefficients are non-dimensionalized according to the initial freestream conditions in FUN3D. This non-dimensionalization remains constant throughout the run, whereas POST2 expects coefficients scaled to the actual dynamic pressure. Each coefficient must be corrected to the actual freestream conditions in POST2 using Equation 5.27:

$$C_{POST2} = \frac{Q_{ref}}{Q} C_{FUN3D} \tag{5.27}$$

where Q_{ref} and Q are the reference and current dynamic pressures.

5.1.5 Communication

Communication between FUN3D and POST2 is handled via network sockets. The calculations for data transformation are split between the custom POST2 module and the Python-based wrapper.

Custom POST2 Aerodynamic Module

The interface between POST2 and the Python-based wrapper is written in the POST2 user-defined aerodynamic code module. Each time POST2 requires new values for the aerodynamic coefficients, it calls this custom module. Depending on the choice of numerical integration scheme and the inclusion of disciplinary code modules, POST2 may require updating the coefficients more than once per time step. Therefore, this module includes the logic to call FUN3D only once at the beginning of each time step, passing stored values for the coefficients in subsequent calls.

At the beginning of the first time step, the module records the initial state variables and performs the calculations described in Section 5.1.4 that remain constant throughout the run. The module then calls for the initialization of FUN3D. On subsequent time steps, the module performs the reference frame transformations described above and passes the dimensional vehicle state in the observer frame. The state variables for the control system are also passed to the wrapper at this time. The module calls FUN3D and waits for it to complete execution of one time step, then reads the aerodynamic coefficients. The coefficients are calibrated to the actual dynamic pressure and returned for use within POST2.

Python-Based FUN3D Wrapper

The Python-based wrapper uses the FUN3D Python API to run FUN3D as a server. Each time the POST2 aerodynamic module calls the FUN3D wrapper, it creates a client that communicates over network sockets to the server. Network communication is also used to gracefully shut down FUN3D at the end of the simulation. FUN3D is developed primarily for operation on Linux, with parallelization using Message Passing Interface (MPI). Thus, the wrapper must also manage communication from POST2 running on one processor to all of the processors running FUN3D. The communication with POST2 is only run on one processor.

On the first call, the wrapper initializes FUN3D, loading the input file, mesh, and flow history. The wrapper then returns the coefficients as calculated from the startup flow field.

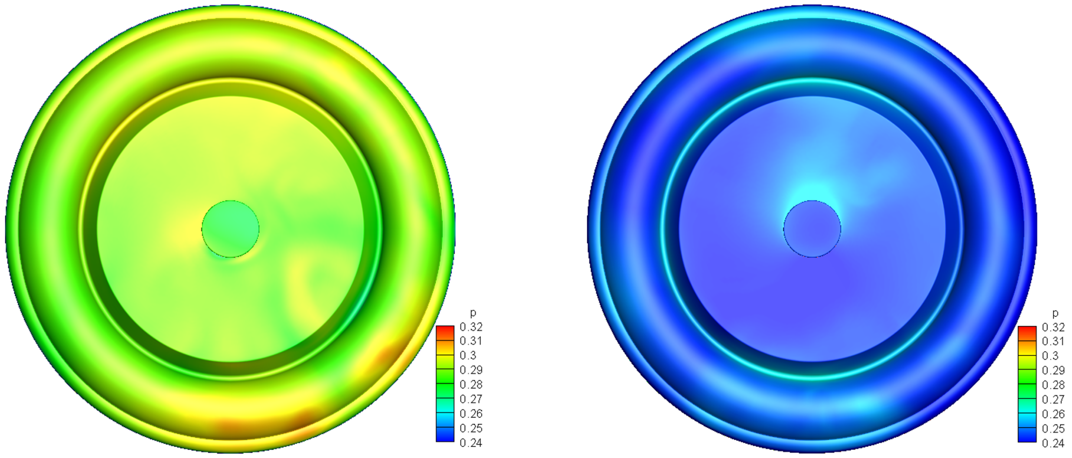
On subsequent time steps, the wrapper reads the current vehicle and control state and performs the non-dimensionalization into grid units. These calculations are performed on a single processor, and the results are broadcast to all of the other processors. The wrapper updates the transformation matrix, sets the C.G. location in the inertial frame, and sets the linear and angular velocities in the body reference frame. FUN3D is executed for one time step, and post-processing is called to find the force and moment coefficients. Equation 5.27 is used to correct the coefficients to the current dynamic pressure as expected in POST2. The coefficients are transformed into the body frame and communicated back to POST2.

5.1.6 Initialization

Since the coupled simulation uses the aerodynamic force and moment coefficients to update the 6DOF equations of motion, these coefficients must be converged at the start of the coupled simulation. Otherwise, numerical instability could cause non-physical forces and moments to be applied to the vehicle, affecting the trajectory or causing simulation failure.

In addition to meeting the threshold for convergence, the flow field must be accurate to the desired initial conditions. However, while initializing the trajectory propagator with the vehicle and control system state is sufficient, this does not ensure that the flow field matches. Initializing from a steady CFD solution would result in a flow history that does not capture the unsteady behavior that presumably leads to the desired initial conditions. For example, consider the flow field around a vehicle at constant speed and slip angles. The flow field around the vehicle would be different if it was at a nonzero angular velocity, or if it were accelerating. Achieving the correct angular velocity conditions is especially important given its effect on the stability characteristics of EDL vehicles [125]. Instead, a forced motion initialization is performed to start the free-flight simulation with the vehicle state and flow field at the desired initial conditions. Figure 5.5 illustrates the result of using a forced-motion initialization. Both figures show the pressure distribution for a vehicle at the same initial velocity and orientation. The vehicle in Figure 5.5b was initialized with a forced motion trajectory, whereas the vehicle in Figure 5.5a was initialized with a time-accurate simulation at fixed conditions. The differences in the net pressure and pressure distribution will result in differences in both forces and moments on startup.

Starting from the initial conditions, a trajectory is back-propagated using the equations of motion. The trajectory assumes no external forces. The trajectory is back-propagated for a sufficient number of steps to achieve convergence of the residuals in the time-accurate simulation. This trajectory is converted into a set of transformation matrices as described in Equation 5.7. The first position and orientation serve as a set of "pre-initial conditions." A steady solution is executed in FUN3D at these new conditions, which is run for a sufficient number of iterations to achieve convergence. The steady flow field is used to initialize the forced motion run, which moves the vehicle along the back-propagated trajectory, using the same flow solver settings and time step anticipated for the coupled simulation. The



(a) Fixed

(b) Moving

Figure 5.5: SIAD backshell pressure distribution with moving vs. fixed time-accurate initialization

flow field at the end of the forced motion simulation is converged, with the flow history and vehicle state matching the desired initial conditions. This flow field is used to initialize the coupled simulation. An example of initialization is shown in Figure 5.6, with the convergence of the residuals during the steady and time-accurate initialization phases demonstrated in Figure 5.7.

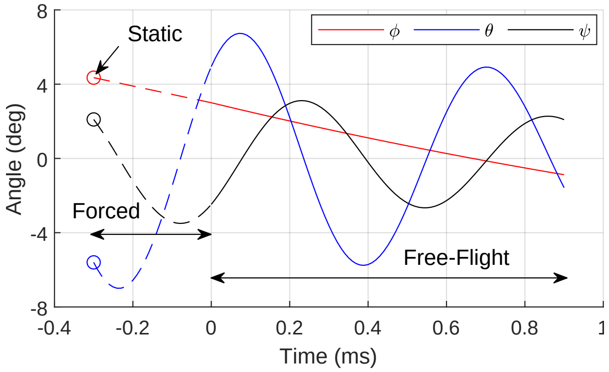
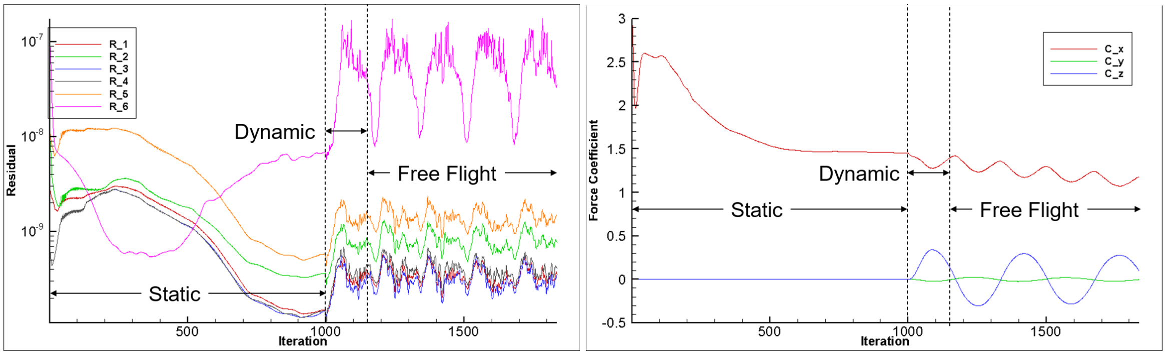


Figure 5.6: Trajectory initialization example

5.1.7 Verification and Validation

Verification of the coupled model was performed by comparing its execution to an existing CFD-RBD model, a 6DOF library that can be compiled with FUN3D [133, 67]. This cross-



(a) Residuals (b) Force Coefficients

Figure 5.7: Initialization results

code verification was performed by running a simulation with the same vehicle, initial conditions, flow solver settings, and initial flow history. The only differences were the trajectory propagator and the method of communication between the disciplinary codes.

Verification Vehicle

The Army-Navy Finner (ANF) projectile was selected as the vehicle for the verification activity due to its common usage for aerodynamic research in both physical [201, 202] and computational [203, 204] experiments. The dimensions of the vehicle are shown in Figure 5.8, and the mass properties are listed in Table 5.1.

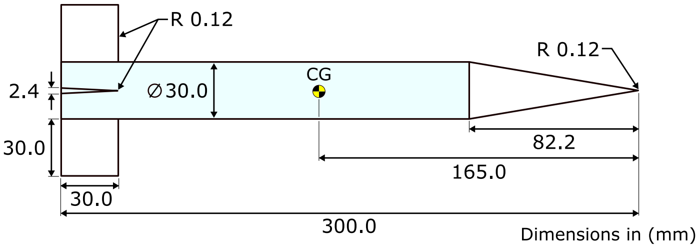


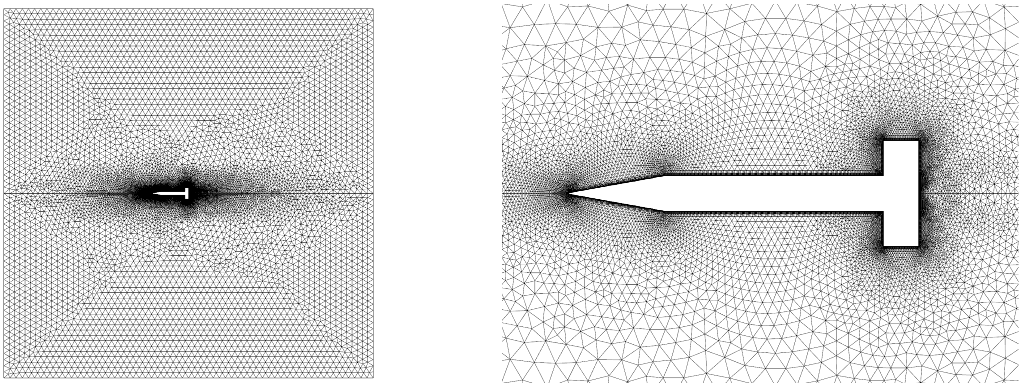
Figure 5.8: Army-Navy finner projectile dimensions.

Table 5.1: ANF mass properties.

m, kg	I_{xx}, kg·m^2	I_{yy} and I_{zz}, kg·m^2
1.588	$1.92526{\cdot}10^{-4}$	$9.87035{\cdot}10^{-3}$

Grid Generation

An unstructured, mixed-element, viscous grid was constructed for the ANF in Pointwise® V18.0R2 [205] with double-precision format and combined anisotropic cells. Rectangular farfield boundaries were spaced at a distance of 5 times the characteristic length in all directions. A hexahedral boundary layer was generated with an initial wall spacing of $4 \times 10^{-7} m$ in order to ensure a y^+ less than unity at the wall for freestream conditions up to Mach 3.0. The surface and volume were generated for a quarter of the vehicle, then mirrored into a full volume to ensure symmetry. The full grid had just over 7 million volume nodes. Cross-sections of the ANF grid are shown in Figure 5.9.

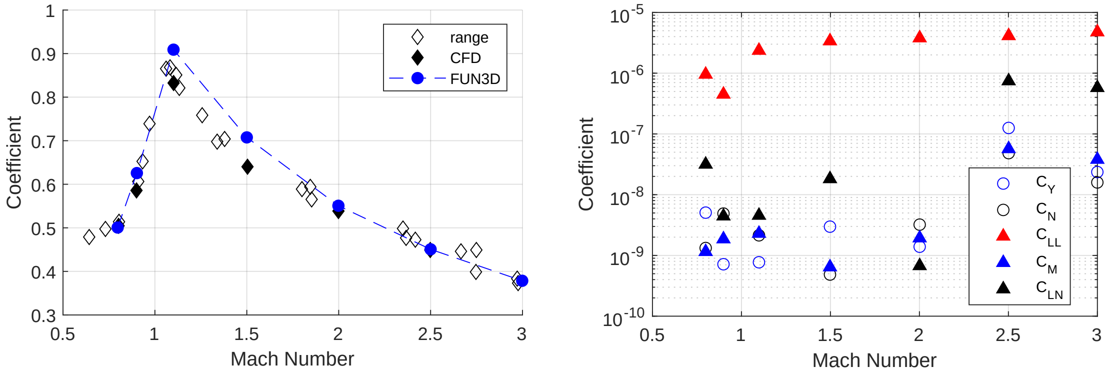


(a) Full grid

(b) Grid detail

Figure 5.9: ANF computational grid

Validation of the drag coefficient was performed against a set of ballistic range tests conducted by Dupuis and Hathaway [202, 206], as well as CFD experiments performed by Sahu and Heavey [203]. The comparison of axial results is shown in Figure 5.10a. The results for the axial coefficient are a close match to both the physical and computational experiments, especially in the supersonic regime. An examination of the other force and moment coefficients are made for the same conditions, in order to verify that their residuals are sufficiently small. The magnitude of the residuals for the off-axis coefficients are shown in Figure 5.10b. The roll moment coefficient is resolved to less than 10^{-5}, and all other components are resolved to less than 10^{-6}. This precision was judged to be sufficient for the use of this grid in 6DOF simulations.



(a) Axial force coefficient.

(b) Magnitude of off-axis residuals.

Figure 5.10: ANF Grid Validation Results, $\alpha = 0°$[207]

Verification Problem

The ANF projectile was simulated in a ballistic drop at Mach 2.0 at standard sea-level conditions, with zero initial slip angles and angular rates.

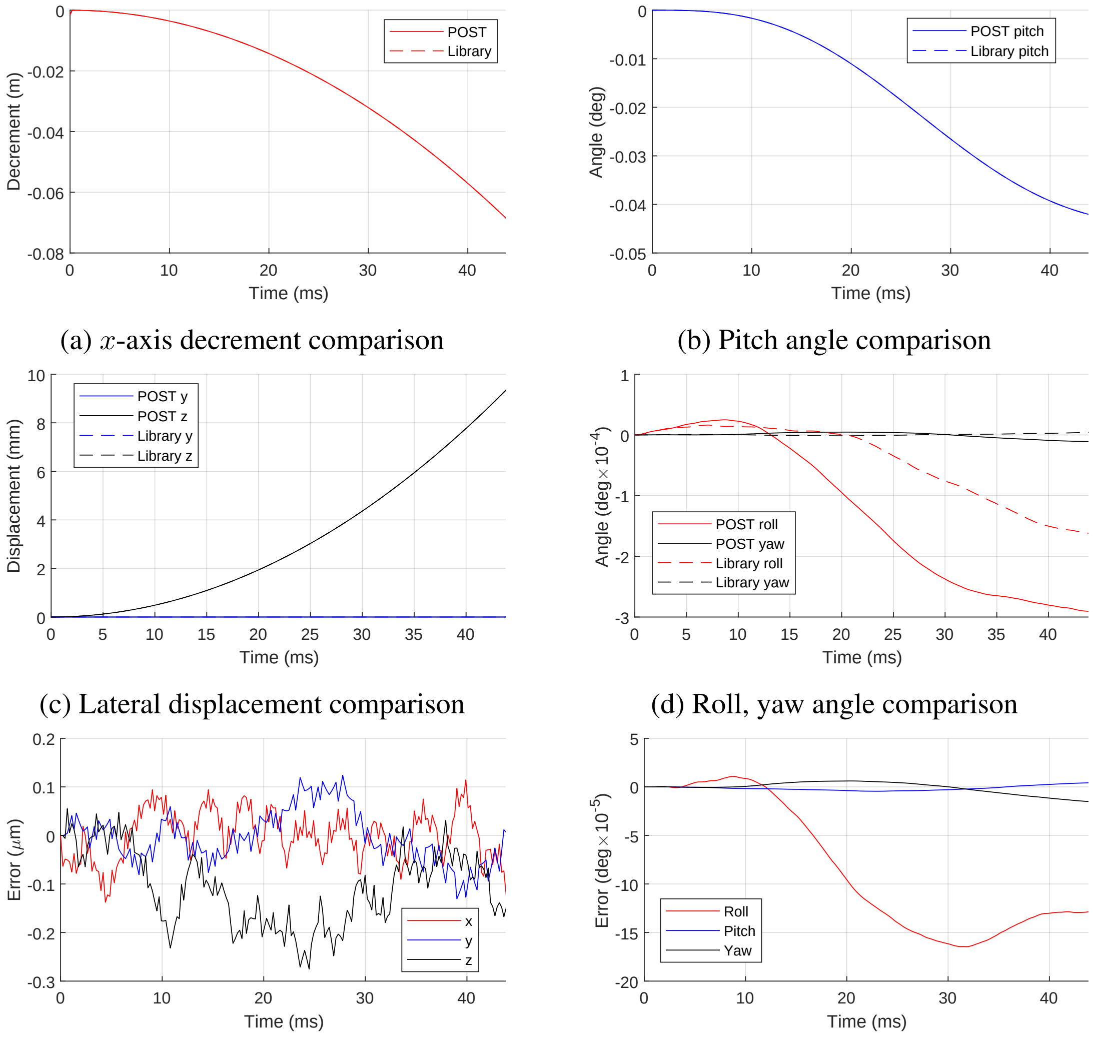
The coupled model was set up in order to approximate the FUN3D 6DOF Library as closely as possible. The 6DOF Library uses a “flat Earth” model with a uniform gravitational field and inertial vertical frame. The flat Earth model is approximated in POST2 by simulating the vehicle around a non-rotating planet with a radius of 10^9 m and a gravitational constant chosen to keep the sea-level gravitational acceleration equal to the standard acceleration of gravity. The 6DOF Library also calculates forces and moments using the flow solver reference quantities for density and speed, which are constant during the simulation. Therefore, constant sea-level atmospheric conditions were enforced within POST2. The 6DOF Library uses the same modification to the time integration scheme as the FUN3D-POST2 framework, so no modification was necessary in that regard.

The flow solver used a k-omega turbulence model [208] with a low-diffusion flux-splitting scheme. The boundary conditions consisted of a viscous, no-slip condition on the projectile surface and a Riemann invariant farfield at the outside surfaces of the grid. Both simulations were performed on the Aerospace Systems Design Laboratory computing cluster within the Partnership for an Advanced Computing Environment at the Georgia Institute of Technology [209]. The simulations were run with $\Delta t = 0.44079$ms for 200 steps, or about 44ms of simulation time.

Verification Results

A comparison of the coupled model ('POST') and FUN3D 6DOF Library ('Library') results for the ballistic drop are shown below in Figure 5.11. The results shown in Figures 5.11a and 5.11b describe a ballistic trajectory with pitch oscillation, which is the expected behavior. Figures 5.11c and 5.11d show that nonzero but small lateral motion develops due to computational precision. The error in each of the displacement components, shown in Figure 5.11e, is small compared to the overall motion of the vehicle, and each is of about the same order of mangitude. The orientation error plotted in Figure 5.11f shows that the error in the roll angle is the largest, which is expected from the behavior identified during grid validation.

Figure 5.12 plots the magnitude of the displacement and orientation errors by simulation step. Both simulations are started from the same initial conditions, so the error is initially zero. After the first step, the error in all components is on the order of 10^{-16}. The displacement error magnitude rises more sharply than the orientation, reaching between 10^{-8} and 10^{-7} for the duration of the simulation. The orientation error magnitude increases more slowly but reaches between 10^{-7} and 10^{-4}. Based on this rate of error growth over time, the cross-code verification was judged to be successful.



(a) x-axis decrement comparison

(b) Pitch angle comparison

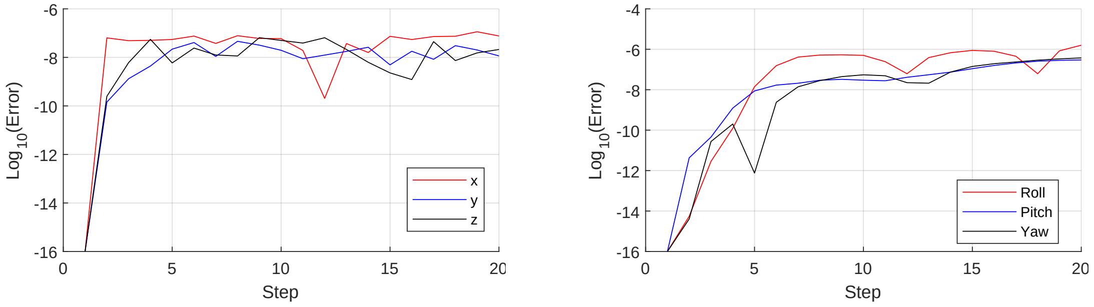
(c) Lateral displacement comparison

(d) Roll, yaw angle comparison

(e) Displacement error

(f) Orientation error

Figure 5.11: ANF cross-code verification results



(a) Displacment

(b) Orientation

Figure 5.12: ANF cross-code verification error growth

5.2 Uncoupled Model

Uncoupled models are required for multiple purposes for the experiments outlined in Chapter 4. A “baseline” model, with construction representing the state of the art, is used as a point of comparison to the coupled model. The low-fidelity model, described in Section 4.2.2 that is used as part of the multi-fidelity controller tuning methodology, is also be an uncoupled model. The only difference between the two as specified in Chapter 4 is the need to accommodate different aerodynamic models. Because the overarching form of these models is the same, they can be considered to be a single uncoupled model with multiple options for aerodynamic model.

POST2 will again be used as the trajectory propagator and control mechanism model, as it has already been established to meet the requirements for the problem of interest. In a similar manner to the custom module constructed to link POST2 with FUN3D, custom aerodynamic modules can also be constructed to perform the calculations required by any of the surrogate models considered in Section 2.4.

CHAPTER 6
TEST CONFIGURATION

This chapter details the construction of the test configuration, in the form of a representative vehicle and problem, that will be applied to the experiments.

6.1 Aeroshell

The work will model an entry vehicle equipped with a Supersonic Inflatable Aerodynamic Decelerator (SIAD). SIAD is designed to be a lightweight decelerator for Mars entry vehicles [210]. Similar to HIAD, the SIAD is an inflatable torus that extends around a rigid nose, though the SIAD has a greater proportion of the vehicle surface covered with a conventional heatshield. SIAD-R was one configuration of the Low Density Supersonic Decelerator (LDSD) Project, developed in consideration for large robotic missions [210, 211]. Figure 6.1 details the SIAD-R in its stowed and deployed configuration.

The choice to use a vehicle with new decelerator technology anticipates the desire to use these technologies on future Mars missions. With a nearly flat afterbody and complex geometry around the shoulder, torus, and burble fence, SIAD exhibits a large unsteady wake. However, it has more benign separation and attachment behavior than a traditional cone body, since the entire afterbody is almost always in separated flow. The geometry was found to be unstable at small α_T in the supersonic regime [97]. Therefore, this vehicle presents suitably unsteady behavior for evaluation in the coupled simulation.

Two flight tests of SIAD were performed on the Supersonic Flight Dynamics Test (SFDT) vehicle in 2015 [101]. These flights tested SIAD deployment in Earth's atmosphere at speeds over Mach 3 and an altitude of 52km. In support of these flight tests, a series of ballistic range tests were performed on SIAD models at the Hypervelocity Free-Flight Aerodynamics Facility (HFFAF) at the NASA Ames Research Facility [97]. The

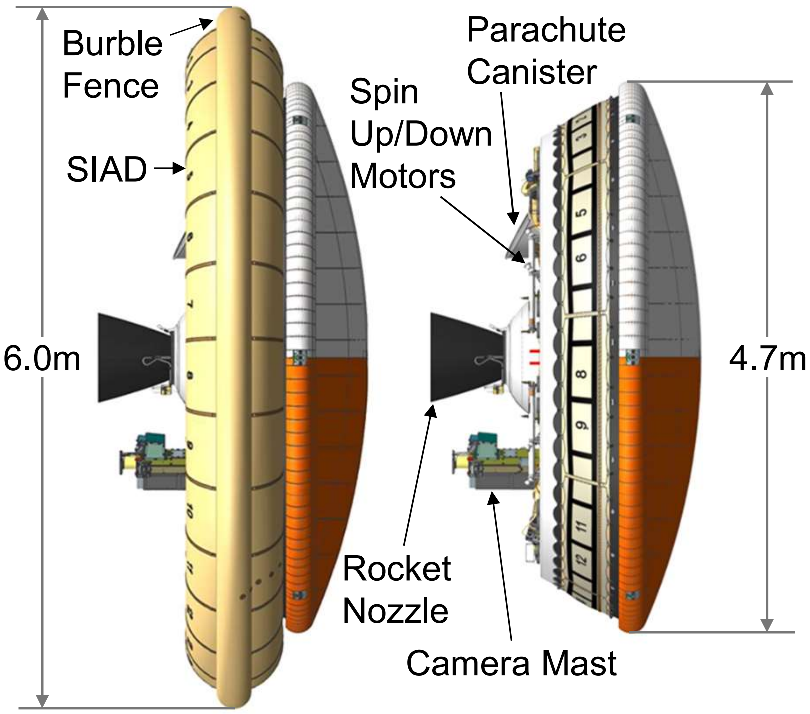
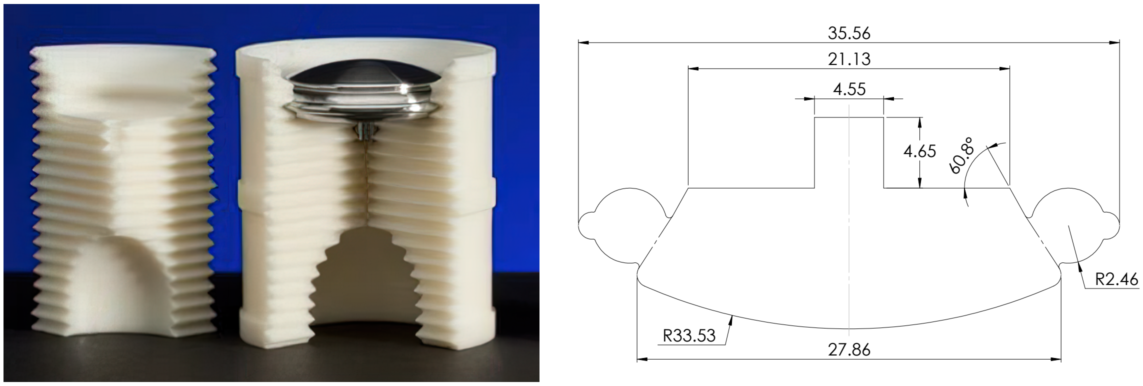


Figure 6.1: Stowed and Deployed SIAD Configuration [97]

deployed-configuration ballistic range model geometry used in this work is shown in Figure 6.2. The selection of SIAD therefore allows this work to leverage the computational & physical experimental data for establishing feasibility and validation.



(a) Model in sabot

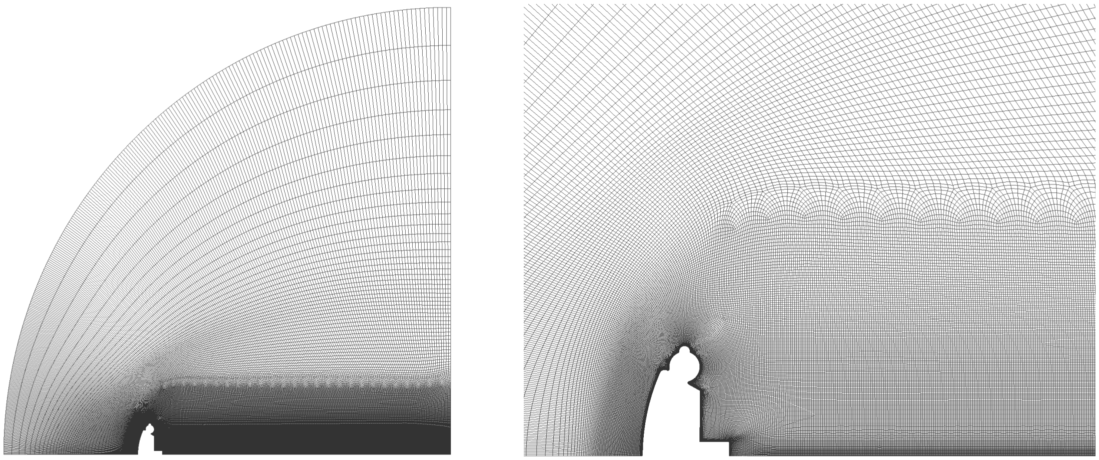
(b) Cross-section (dimensions in mm)

Figure 6.2: SIAD ballistic range model (adapted from Wilder et al. [97])

6.1.1 Computational Grid

The deployed-configuration grid used in [110] has been provided by Joe Brock. This grid, shown in Figure 6.3, was generated for the cell-centered US3D flow solver, and will be adapted for use in the node-centered FUN3D solver. The hemispherical volume captures the forebody out to 2 vehicle diameters, and the wake out to 4.5 diameters. The grid spacing is 0.1 μm at the wall and 0.4mm in the wake to ensure a value of y^+ less than unity in those regions. The mesh was created as a quarter-symmetry, then mirrored about the xy- and xz-planes to form the full grid. The full mesh is composed of about 22 million hexahedral elements.

It is important to note that this grid was developed for use with mesh deformation in the US3D-based CFD-RBD model. Therefore when undergoing rigid mesh movement, the high density region of the grid aft of the vehicle can rotate out of the subsonic wake, decreasing the resolution of any flow features. The satisfactory performance of this grid with rigid motion must be determined by validation.



(a) Quarter-Symmetry Plane

(b) Wake Detail

Figure 6.3: SIAD Ballistic Range Model Mesh

6.1.2 Vehicle Properties

The vehicle geometry and mass properties are scaled for execution cost feasibility. Table 6.1 compares the geometry and mass properties for MSL, SDFT, and the ballistic range model which is used in this work. The table also lists the vehicle properties as modified for this work. The vehicle mass is artificially doubled. This doubles the ballistic coefficient, so that the vehicle is more resistant to acceleration in the simulation, giving a longer time at the desired conditions. The inertia products are kept the same so as to maintain the oscillation frequency, thus preventing the required simulation duration from increasing.

Table 6.1: Comparison of entry vehicle characteristics

	MSL [54, 42]	SFDT [101]	Model [97]	Modified model
Diameter (m)	4.5	6.0	0.0356	0.0356
Entry Mass (kg)	2,919	1461	0.0462	0.0919
Ref. Area (m^2)	15.90	28.27	9.931×10^{-4}	9.931×10^{-4}
C_D at Mach 2	1.347	1.45	1.45	1.45
$x_{cg,F}/d$	0.214	-	0.197	0.197
$z_{cg,F}/d$	0.0134	0.0	0.0	0.0
B.C.	154.0	35.6	32.1	64.2
I_{xx} (kg m^2)	3055	3149	5.568×10^{-6}	5.568×10^{-6}
I_{yy} (kg m^2)	3952	1784	3.023×10^{-6}	3.023×10^{-6}
I_{zz} (kg m^2)	4836	1784	3.023×10^{-6}	3.023×10^{-6}

6.1.3 Aerodynamic Model

To perform the proposed work, it is critical to determine the correct settings for both the CFD flow solver and the trajectory propagator. Setting the physical time step is chosen to resolve the unsteady flow around the vehicle. The number of subiterations and CFL settings are chosen such that the aerodynamic flow is resolved at the subiteration level for each time step with proper numerical convergence to accurately determine the unsteady forces and moments.

The FUN3D flow solver settings resulting from the validation activities are summarized here. The simulation is run with compressible, calorically-perfect gas. Fully turbulent flow is used with a Spalart-Allmaras-based DES model [212]. A stencil-based Van Albada flux limiter [213] is used with a dissipative low-diffusion flux splitting scheme. For time-accurate simulations, an optimized second-order backward differencing scheme is used for temporal time integration [214]. The time-accurate simulations use a physical time step of 2.354×10^{-5}, which corresponds to a non-dimensional time step of 0.4773. This time step was evaluated in a time step resolution study and determined to result in converged trajectory responses [215]. Each time step used 75 subiterations with constant meanflow and turbulence CFL numbers of 60.0 and 45.0, respectively. POST2 and FUN3D will use the same physical time step, so no interpolation of the transformation matrix will be performed by FUN3D.

6.2 Vehicle Model Validation

Validation was performed for the coupled model of the SIAD ballistic range test vehicle, including the vehicle mesh, mass properties, flow solver & trajectory propagator settings. Replications were made of a subset of ballistic range experiments performed at NASA Ames Research Center's Hypervelocity Free-Flight Aerodynamics Facility (HFFAF) [97]. Test shots were run from Mach 2 to 4, with the model impacting a sheet of paper to induce α_T oscillations up to $28°$. Replications of a subset of these ballistic range tests for the deployed model were also conducted by Brock et al. in a coupled free-flight simulation to validate the predictive capabilities of their US3D-based environment [110]. The conditions for the test shots and the CFD replications are shown in Figure 6.4.

Determination of the flow solver settings was validated by performing a replication of one of the US3D experiments. Further validation was performed by replicating three ballistic range test shots and comparing the resulting trajectories.

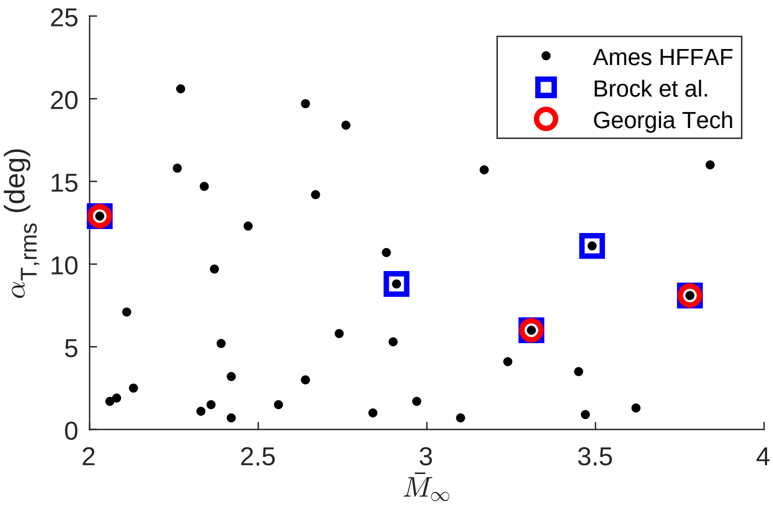


Figure 6.4: SIAD experiment conditions [215]

6.2.1 Replication of a US3D-based CFD-RBD Simulation

The initialization method used by Brock et al. is a development of the method used by Stern et al. [41] expanded to allow nonzero initial pitch and yaw rates. The freestream conditions and initial orientation were chosen to match the first experimental observation. With the grid held at the initial orientation, time-accurate flow was run for long enough to eliminate transients. As Stern et al. [41] point out, this results in an unrealistic flow field, but the unphysical initial conditions are assumed to dissipate quickly in time-accurate simulation. Brock et al. [110] obtained the initial Euler angle rates $\dot{\theta}$ and $\dot{\psi}$ by fitting cosine functions, with the form of Equation 6.1, to the experimental pitch and yaw data, then taking their derivative at the first experimental observation.

$$f = a_1 \cdot \cos\left(a_2 + a_3 t\right) \tag{6.1}$$

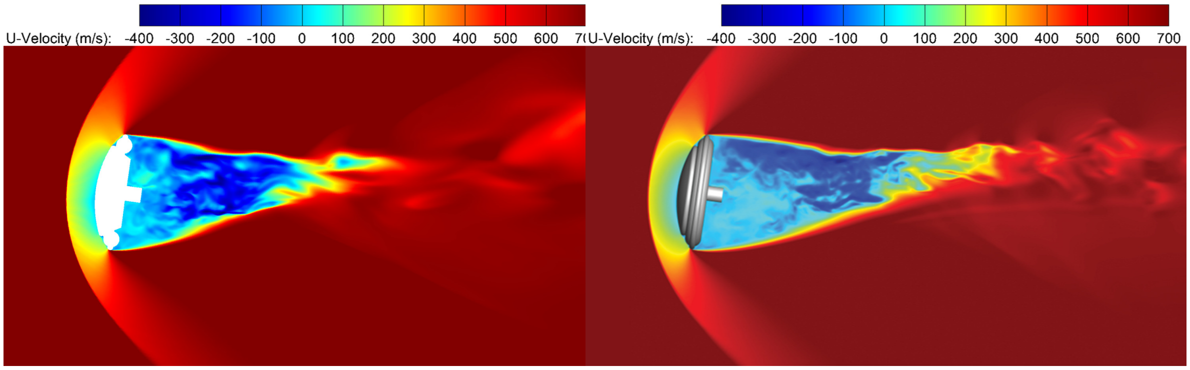
The initial roll angle ϕ and roll rate p were assumed to be zero. Therefore, the initial angular velocity q and r components are approximately equal to the Euler angle rates $\dot{\theta}$ and $\dot{\psi}$, respectively. The angular velocity was applied instantaneously at the start of RBD simulation. v and w, the linear velocities in the y- and z-axes, were assumed to be negligible.

The initial conditions are listed in Table 6.2.

Table 6.2: Shot 2623 Initial Conditions [110]

Shot	u, m/s	v, m/s	w, m/s	ϕ, °	θ, °	ψ, °	p, °/s	q, °/s	r, °/s
2623	728.88	0	0	0	-7.68	0.16	0	6790	-741

A comparison of the streamwise velocity flow fields at initialization is shown in Figure 6.5. The wake calculated in FUN3D is slightly shorter than the US3D results from Brock et al., but the nearfield behavior is similar. In addition to the effects of the remaining discrepancies between the simulations described above, differences will also arise due to the stochastic nature of the flow in time-accurate simulation. Even with these differences, the settings of the FUN3D simulation yielded a satisfactory replication of the initial flow field.



(a) FUN3D

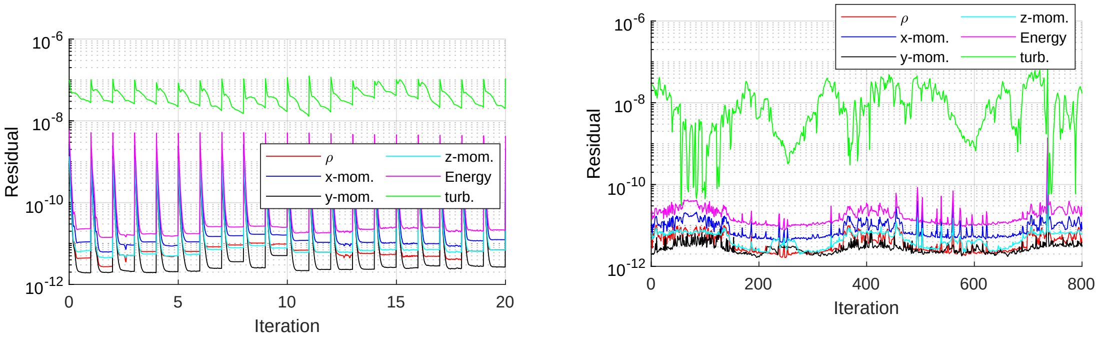
(b) US3D [110]

Figure 6.5: Streamwise velocity flow field for initialization of Shot 2623

The validation simulations were executed on hardware provided by the Partnership for an Advanced Computing Environment at the Georgia Institute of Technology [209] and the K-cluster at NASA Langley Research Center[1]. The meanflow and turbulence residuals are plotted in Figure 6.6 by time step. Figure 6.6a plots a detailed view of the residual convergence over the first 20 time steps. The meanflow residuals drop by about two orders of

[1] Runs were executed on the K3 and K4 subclusters. The K3 subcluster has compute nodes with dual-socket 8-core 2.60 GHz Intel E5-2670 Sandybridge nodes with 32GB of memory per node. The K4 subcluster has compute nodes with dual-socket 20-core 2.40 GHz Intel Gold 6184 Skylake processors with from 96GB to 384GB of memory per node.

magnitude across the time step, reaching converged values over the second half of the step. All of these residuals converge below 10^{-10}, which is sufficiently small for this problem. The turbulence residual converges by an average of about a half of an order of magnitude, but the number of subiterations is not sufficient for it to approach a steady state. However, the value of the residual is below 10^{-7}, which is satisfactory for this problem. Figure 6.6b plots the residuals at the final subiteration for the first 800 steps (approximately half of the total run). This shows that each of the residuals is consistent across a wide range of time steps, where the vehicle motion state moves significantly.



(a) Detailed view of residual convergence

(b) Residuals at final subiteration of each time step

Figure 6.6: Residuals for replication of Shot 2623 computational simulation

The validation of the simulation is based on comparison of the resulting trajectory to both sets of experiments. Figure 6.7 overlays the trajectory components for the FUN3D-POST2 simulation, the Brock et al. simulation, and the ballistic range experiment. Figure 6.7a plots the x-axis displacement decrement x_{dec}, calculated using Equation 6.2:

$$x_{dec} = x - u_0 t \tag{6.2}$$

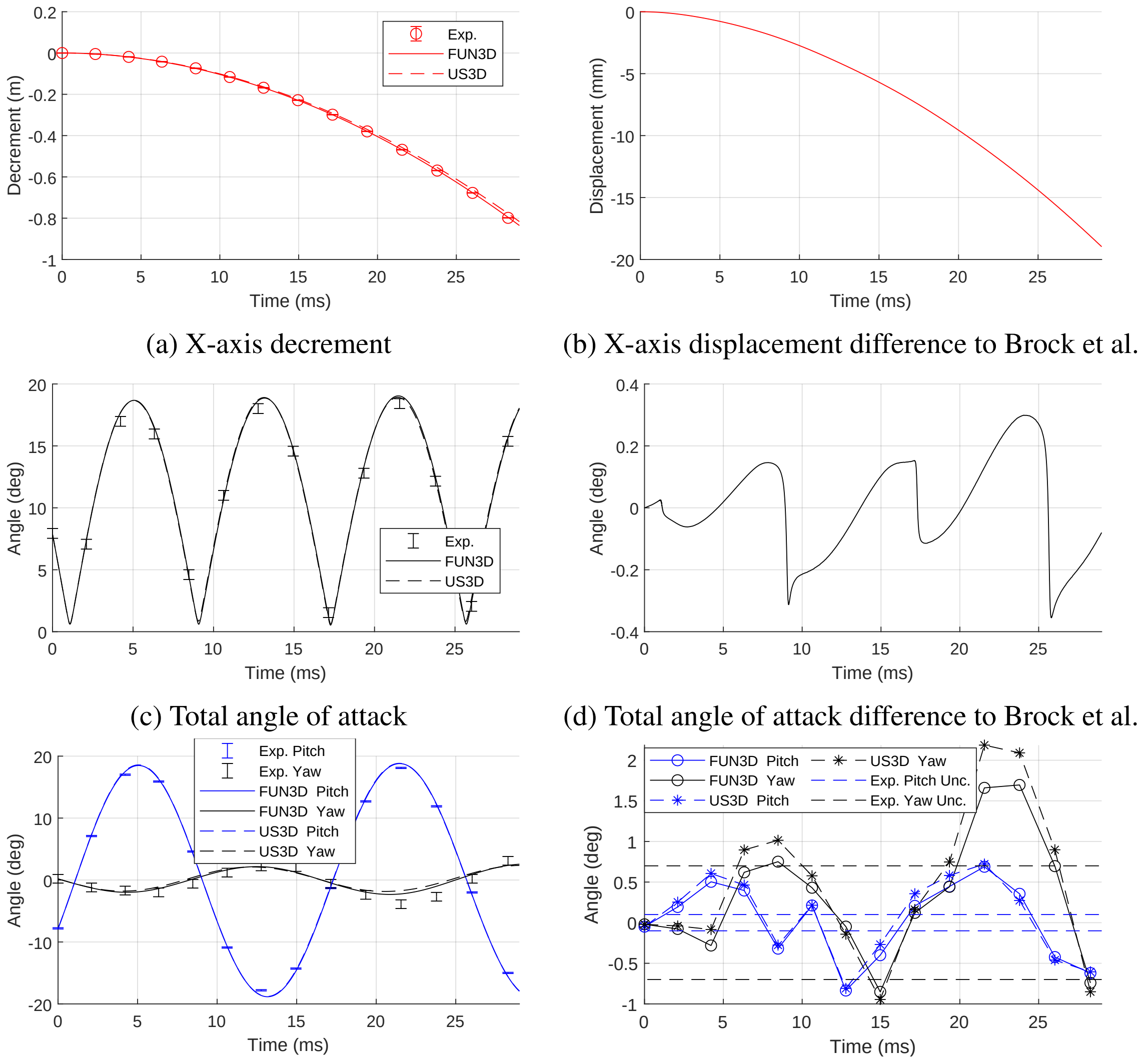
where x is the displacement from the position of the first experimental measurement, u_0 is the x-axis velocity of the vehicle at the first experimental measurement, and t is the time measured from the first experimental observation. This is equivalent to the displacement relative to an inertial observer moving at the initial velocity. Both the FUN3D-POST2 and the Brock et al. simulations match the behavior of the ballistic range experiment, with a maximum displacement error of less than 0.1% of the total displacement. The difference between the simulations is plotted in Figure 6.7b. The difference grows smoothly and gradually over time, which is expected due to the propagation of slight differences in axial force.

Total angle of attack α_T is plotted for both simulations and the experimental data in Figure 6.7c. The difference in α_T remains less than 0.4° throughout the simulation, and less than 0.2° at its maximum amplitude. Evaluating the results of the differences in axial decrement and α_T demonstrates that the two simulations predict very similar behavior in both linear and angular motion. Both simulations also have similar error with respect to the experimental data. Even with the known differences in flow solver and RBD calculation settings, the differences between the angular components of the simulated trajectories are on the same order of magnitude as the experimental uncertainty. These results demonstrate that the FUN3D-POST2 framework and the selected settings provide a satisfactory free-flight simulation of the SIAD model.

However, an examination of the results for the pitch and yaw angles reveals additional information. As shown in Figure 6.7f, both simulations deviate from the experimental results, with the maximum error increasing over time. The failure to predict the amplitude growth in the yaw oscillation is particularly noticeable.

Both simulations show a discrepancy between the accuracy of α_T and the Euler angles. While α_T is close to the experimental measurements, when this is decomposed into pitch and yaw, the error is increased. Amplitude growth is over- and under-predicted for pitch and yaw oscillation, respectively. For an axisymmetric vehicle, the behavior in pitch and yaw oscillation (i.e. damping) should be similar if the roll rate is negligible, which is the assumption for both CFD simulations. However, the exchange of oscillation amplitude in the pitch and yaw present in the experimental data suggests that the roll rate is nonzero. This type of behavior can be observed in ballistic range tests of other axisymmetric vehicles

[98]. While these deficiencies themselves prompt further investigation, this similarity in behavior upholds the validation between the computational simulations.



(a) X-axis decrement

(b) X-axis displacement difference to Brock et al.

(c) Total angle of attack

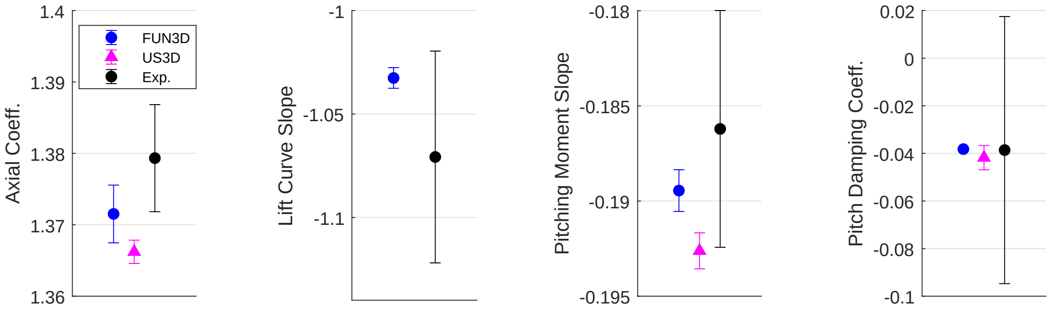
(d) Total angle of attack difference to Brock et al.

(e) Euler angles

(f) Euler angle error

Figure 6.7: Replication of Shot 2623 computational simulation

In addition to comparisons of the trajectory results, the extracted aerodynamic behavior was also compared. System identification was used to determine time-average aerodynamic coefficients from the trajectory results for the FUN3D simulation, the US3D simulation by Brock et al. [110], and the ballistic range data [97]. This establishes a like-to-like comparison using only the measurements from this single shot. The time-average coefficients are plotted in Figure 6.8. Figure 6.8a shows that both FUN3D and US3D slightly underpredict the axial force coefficient. The lift curve slope is plotted in Figure 6.8b (except for the US3D results, for which the y and z trajectory data was not available). For the pitching moment coefficient plotted in Figure 6.8c, both CFD simulations slightly overpredict the magnitude. Results for the pitch damping coefficient plotted in Figure 6.8d show that the CFD results are close to each other and close to the mean value extracted from the experimental data. However, the large uncertainty resulting from the fit over a single experimental run precludes a more detailed comparison. Overall, the average coefficients demonstrate good agreement between the CFD models.



(a) Axial force (b) Lift curve slope (c) Pitch moment slope (d) Pitch damping

Figure 6.8: Comparison of average aerodynamic coefficients

6.2.2 Recreation of Ballistic Range Shots

The three ballistic range shots indicated above were recreated by running free-flight simulations of the vehicle using the initial conditions and freestream properties calculated and measured from the experimental results. The freestream conditions and vehicle properties are listed in Table 6.3. The exact mass properties of each shot are modeled in POST2, but the computational mesh represents the nominal dimensions of the model. However, the largest difference in diameter between any of the experiments and the nominal dimensions is less than 0.25%. Therefore, the assumption is made to use the nominal dimensions (equivalent to the surface dimensions used in the grid) for the simulations. The location of the center of mass and center of moment from the nose are represented in POST2 and FUN3D by multiplying the x_{cg}/d by the nominal diameter of 35.56mm.

The initial conditions are listed in Table 6.4. These were determined by fitting the measured trajectory results and back-propagating to the start of the range. Since the roll angle of the vehicle was not measured, the roll rate was reconstructed from the other trajectory components. For a detailed description of this process, see Reference [215].

Table 6.3: Freestream conditions and model properties for selected shots [97]

Shot	Mach	$\alpha_{T,max}$, °	Pres., kPa	Temp., K	d, mm	Mass, g	x_{cg}/d
2623	2.12	18.4	22.58	292.8	35.51	45.93	0.161
2643	3.46	8.8	20.67	294.3	35.58	45.85	0.159
2638	3.93	11.7	19.87	294.2	35.52	45.93	0.160

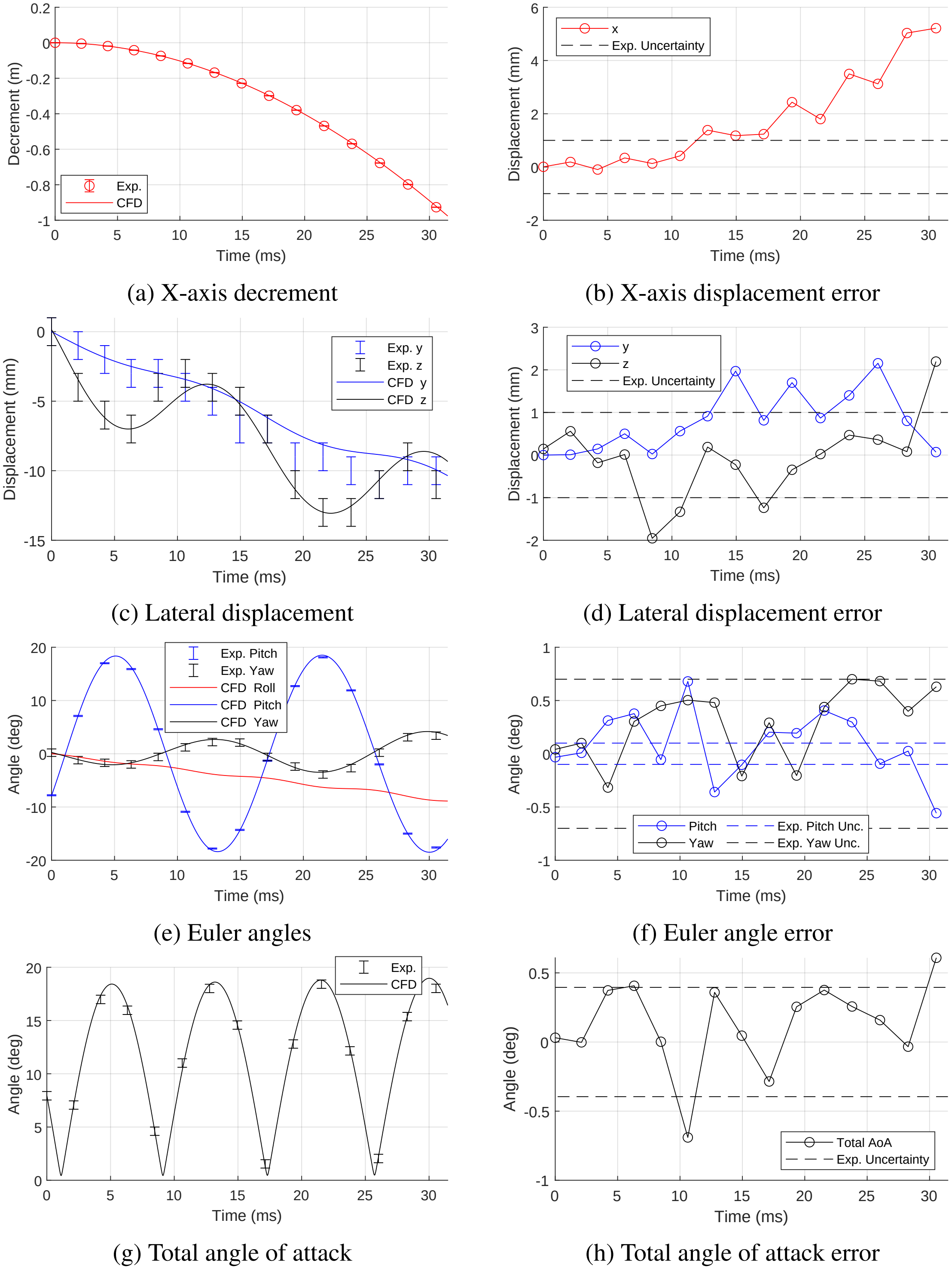
Table 6.4: Initial conditions for selected shots [215]

Shot	u, m/s	v, m/s	w, m/s	ϕ, °	θ, °	ψ, °	p, °/s	q, °/s	r, °/s
2623	728.95	-0.48	-1.62	0	-7.85	0.18	-290.	6640	-601
2638	1351.0	-0.42	-0.45	0	-4.60	0.24	-91.3	8150	-153
2643	1188.9	-0.75	-0.11	0	-1.64	0.74	-1890	6100	-1030

The trajectory component results for Shot 2623 are plotted in Figure 6.9 along with the measured experimental data. Figure 6.9a plots the x-axis decrement, showing the vehicle decelerating over time. Figure 6.9b shows that the x-axis displacement has increasing deviation with the experiment over time, consistent with the behavior shown in Figure 6.7b. However, the maximum error in displacement at the end of the run is less than 0.025% the length of the range. The lateral position components are plotted in Figure 6.9c. Note that the oscillatory behavior of the displacement is distinctly captured, even for such small displacement magnitudes relative to the downrange motion. Figure 6.9d shows that the lateral components remain on the same order of magnitude as the experimental uncertainty bounds. The pitch and yaw angles are plotted in Figure 6.9e, along with the roll angle showing the nonzero initial roll rate. Compared to the results in Figure 6.7f, the error plotted in Figure 6.9f is much smaller, with the yaw angle entirely within the experimental uncertainty bounds. Even considering the total angle of attack in Figure 6.9h, the orientation error remains on the same order of magnitude as the experimental uncertainty bounds. Overall, the results for Shot 2623 show excellent agreement to the experimental data.

The quality of these results is largely repeated for the other two shots, Shot 2638 and Shot 2643 (plotted in Figures 6.10 and 6.11, respectively). Both of these shots had larger errors in the linear and angular components, but with each of the components remaining within the same order of magnitude as the experimental uncertainty bounds. Shot 2638 had the largest orientation error, as shown in Figure 6.10h, of slightly greater than $1°$.

Based on the overall close agreement between the recreations and the experimental data, the coupled model of the SIAD vehicle was judged to be successfully validated. Furthermore the results validate the use of the SIAD grid with rigid mesh rotation for total angles of attack less than $18°$.



(a) X-axis decrement

(b) X-axis displacement error

(c) Lateral displacement

(d) Lateral displacement error

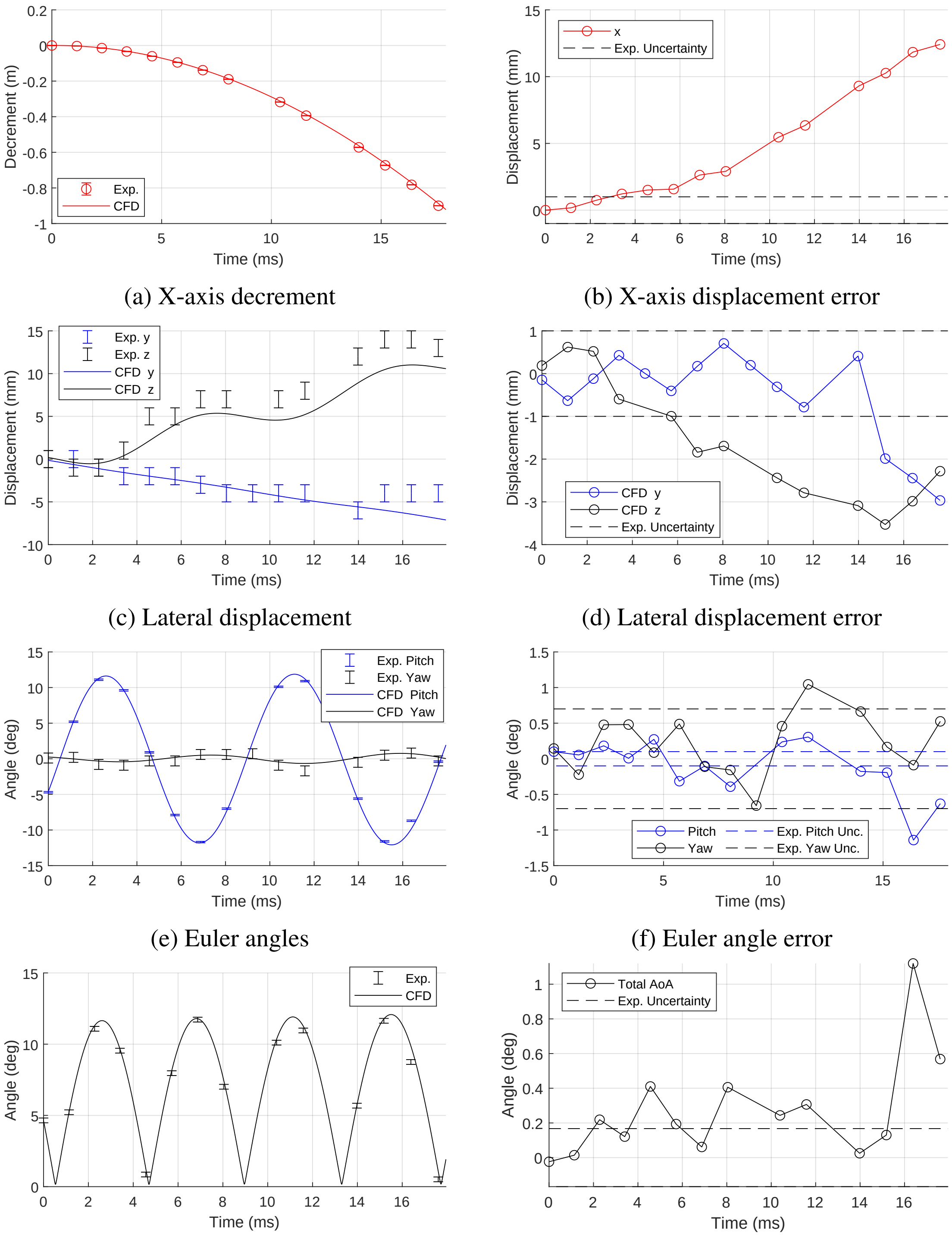
(e) Euler angles

(f) Euler angle error

(g) Total angle of attack

(h) Total angle of attack error

Figure 6.9: Recreation of Shot 2623



(a) X-axis decrement

(b) X-axis displacement error

(c) Lateral displacement

(d) Lateral displacement error

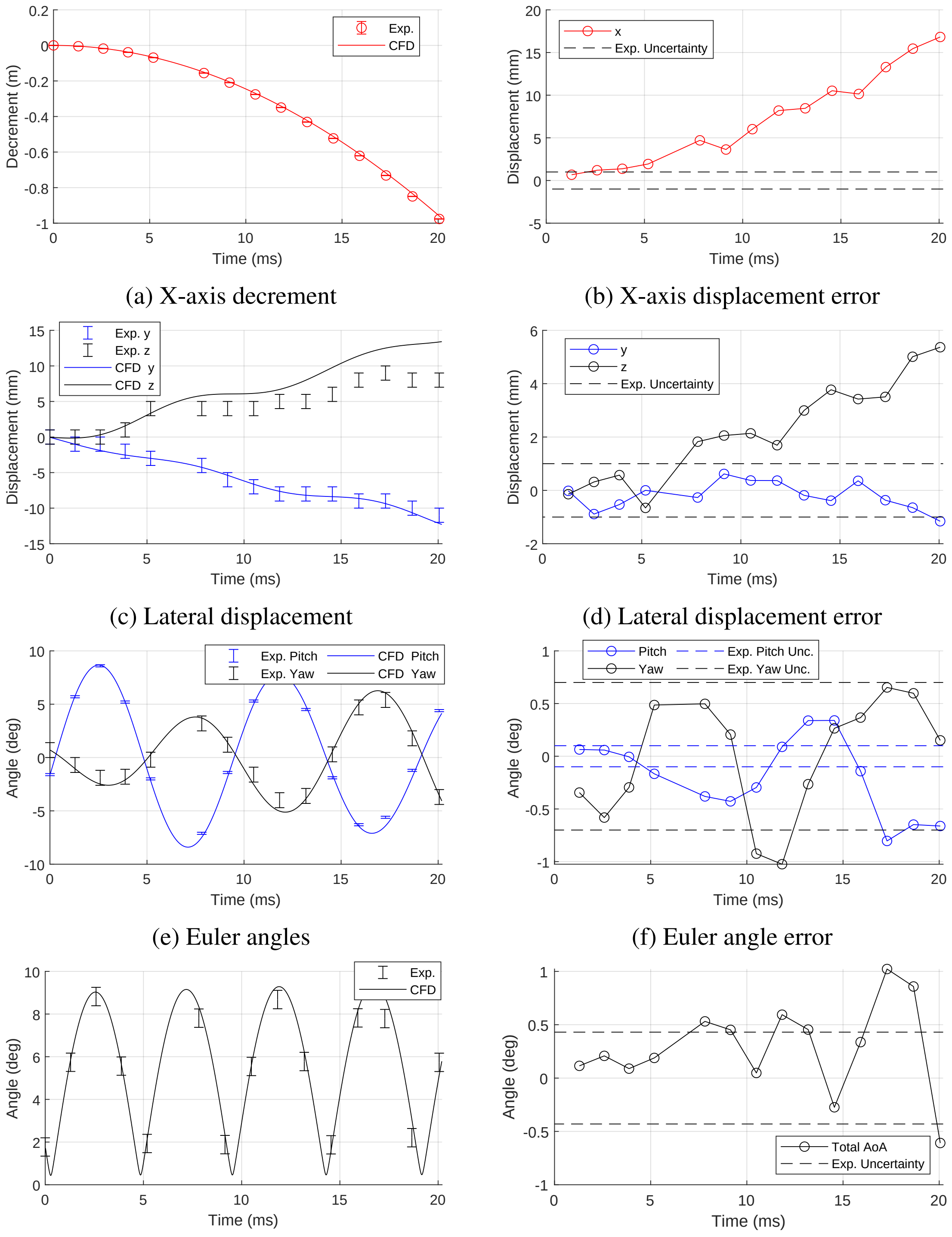
(e) Euler angles

(f) Euler angle error

(g) Total angle of attack

(h) Total angle of attack error

Figure 6.10: Recreation of Shot 2638



(a) X-axis decrement

(b) X-axis displacement error

(c) Lateral displacement

(d) Lateral displacement error

(e) Euler angles

(f) Euler angle error

(g) Total angle of attack

(h) Total angle of attack error

Figure 6.11: Recreation of Shot 2643

6.3 Control Mechanism

The control mechanism selection is motivated by the ease of implementation while still meeting the requirements to use a system that exhibits unsteady interaction with aerodynamics and flight dynamics.

Internal Moving Mass Actuators (IMMA) are an alternative control system studied for entry vehicles. An IMMA moves masses within the vehicle to change the composite mass properties or momentum. The mass can be translating along a linear or circular track, or rotating about an axis (e.g., a spacecraft reaction control wheel). A translating IMMA changes the composite center of gravity of the vehicle relative to the center of pressure. This allows the vehicle to directly control the magnitude of the lift vector. By using two masses oriented along the lateral axes of the vehicle as shown in Figure 6.12, the system can have independent angle of attack and sideslip control. This would decouple control over downrange and crossrange distance, eliminating the need for lifting trajectories to fly through bank reversals. For larger vehicles like HIAD and ADEPT, IMMA control can have favorable feasibility compared to RCS, considering the mass penalties for thrusters and fuel [216].

Since all of the control apparatus is contained within the OML, there is no interaction with the aerodynamic flow. However, the moving masses do interact with the flight dynamics. The acceleration of the masses must be accounted for by additional terms in the equations of motion, and the inertia derivative will be nonzero. While there is still time-dependent interaction within the system, for a comparable performance an IMMA will have less complex interaction than an RCS.

IMMAs were proposed for roll control on RVs as early as 1967 [217]. Petsopoulos and Regan developed a linear moving-mass roll control system for a fixed-trim, non-spinning RV [218]. Their results showed that control was possible but that the system produced dynamic and aerodynamic cross-coupling. Wang et al. analyzed the nonlinear dynamics of

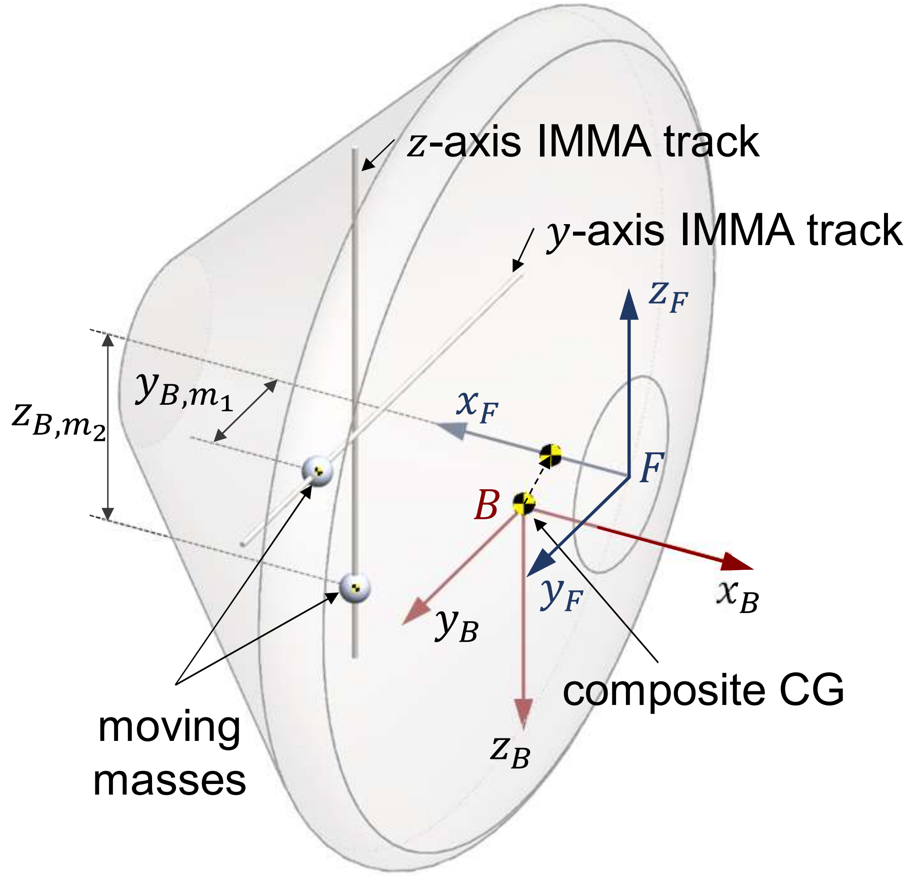


Figure 6.12: Two-axis IMMA configuration

such a system, showing that the stability of the system was sensitive to the location of the moving mass [219]. In addition to roll control, they were able to develop a guidance law to control the deceleration of the RV [220].

Robinett III et al. demonstrated an autopilot for an alternate control scheme, using a translational IMMA control system with an axisymmetric, slow-spinning RV [221]. Menon et al. developed a three-axis translational IMMA system for control of a kinetic warhead inside and outside the atmosphere [222].

IMMA control systems have also been studied for entry vehicles. Balaram described a system for a Mars entry vehicle using two masses sharing a single, circular track [223]. Mukherjee and Balaram developed a nonlinear controller using dynamic inversion [224].

Davis et al. analyzed guidance algorithm performance for HIAD and rigid aeroshell entry vehicles using CG offset control, compared to bank angle control [216]. This study considered generalized movement of point masses, without specifying an actuation method. Atkins and Queen developed a two-axis IMMA for the Mars Phoenix entry vehicle, modifying the Apollo Earth return terminal guidance algorithm to achieve precision guidance [15]. The masses were actuated as damped, driven, harmonic oscillators with given damping and spring constants. They chose masses of about 7.7% the vehicle mass, achieving an α_T control authority greater than $10°$. Compare this mass fraction to the 6% allocated on MSL for entry mass balance devices to achieve the desired α_T (excluding mounting and jettisoning hardware, and excluding the RCS hardware & fuel) [40]. Atkins has further developed IMMA control systems for HIAD with both translation and rotation configurations [167].

In another parallel with the field of entry vehicles, IMMA control systems have also been studied for projectiles. Murphy studied the influence of moving internal components on the motion of spinning projectiles [225]. Rogers and Costello have extensively studied spinning projectiles with a single moving mass. They demonstrated control for configurations with the mass moving in the lateral [226] and longitudinal [227] axes, as well as rotation at the end of a cantilever [228]. The actuation was modeled as a set of electromagnetic actuators moving a permanent magnet. Control was achieved by dynamic effect, not by changing center of pressure.

The difference between the activities for tuning a two-axis IMMA system compared to a one-axis system amounts to a difference in scale. Therefore, the vehicle will be equipped with a single translational IMMA for controlling angle of attack. Compared to RCS, this system requires no modification of the grid geometry and density to include nozzles and properly resolve the plume.

The vehicle mass is split into 97.5% static mass, and 2.5% moving mass. This means the moving mass is 2.3g, or about 2.7% the static mass of the vehicle. The mass travels on a 32mm track aligned along the z-axis of the body frame. The IMMA track is constrained to within the OML of the stowed configuration of the vehicle, as shown in Figure 6.13. This mechanism allows for a trim angle of attack up to ±4°. The mass properties of the spring and damper are assumed to be constant, and therefore are condensed into the body mass properties.

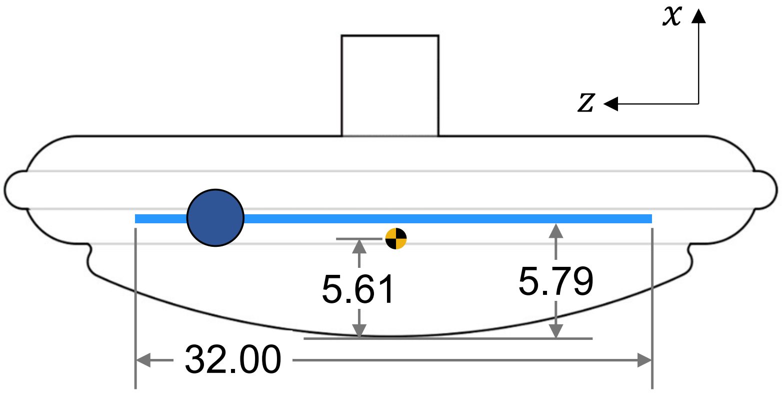


Figure 6.13: IMMA track dimensions on SIAD model (mm)

The actuation is modeled as a damped harmonic oscillator with an input driving force. The state variables for the mass is:

$$\mathbf{H} = (z_a, \dot{z}_a, \ddot{z}_a)^T \tag{6.3}$$

where z_a, $\dot{z}_a$, $\ddot{z}_a$ are the position, velocity, and acceleration of the z-axis mass. The equation of motion is:

$$\ddot{z}_a = (1/m_a)(F_z - k_a z_a - c_a \dot{z}_a) \tag{6.4}$$

where F_z is the driving force, k_a is the spring coefficient, and c_a is the damping coefficient. The behavior of the mass is constrained at the end of the track such that:

$$-z_{a,max} \leq z_a \leq z_{a,max} \tag{6.5}$$

$$\begin{cases} \dot{z}_a \leq 0, & z_a = z_{a,max} \\ \dot{z}_a \geq 0, & z_a = -z_{a,max} \end{cases} \tag{6.6}$$

$$\begin{cases} \ddot{z}_a \leq 0, & z_a = z_{a,max} \\ \ddot{z}_a \geq 0, & z_a = -z_{a,max} \end{cases} \tag{6.7}$$

6.4 Guidance Algorithm and Entry Controller

The guidance algorithm commands a constant, chosen angle of attack α_{cmd}. The algorithm is relatively simple, since the short duration of the simulations would not be able to demonstrate complex behavior like predictor-corrector algorithms or even reference trajectory. The instantaneous operation of these algorithms can be represented by commanding constant slip angles, assuming that the rate of change of the desired slip angles would be small over the simulation duration.

$$u_z = k_{z,0}\alpha_{cmd} + k_{z,1}\alpha_{cmd}^2 + k_{z,2}\alpha_e + k_{z,3}\dot{\alpha} + k_{z,4}\alpha_{\mathcal{I}} + k_{z,5}\dot{z} + k_{z,6}\ddot{z} \tag{6.8}$$

The $k_{z,0}$ and $k_{z,1}$ terms manage the trim behavior of the vehicle. The $k_{z,2}$, $k_{z,3}$, and $k_{z,4}$ terms form a PID controller to control the error relative to α_{cmd}. Finally, the $k_{z,5}$ and $k_{z,6}$ terms allow for responsiveness to the IMMA state. These two terms were included based on the example of Atkins [167].

6.5 Flight Conditions

The freestream conditions selected are equivalent to the conditions of ballistic range test Shot 2623 [97]. This allows the flow solver settings to be largely unchanged from the validation. The initial Mach number chosen as 2.1, with a simulation duration allowing the vehicle to decelerate to about Mach 1.95. This small range of Mach number was chosen to limit the computational cost and duration of simulations. While the small range represents a limitation to any results, this is a regime in which the ballistic range tests demonstrate dynamic instability, a key characteristic of interest for this problem.

Table 6.5: Freestream conditions and model properties for selected shots [97]

Mach	$\alpha_{T,max}$, $^\circ$	Pres., kPa	Temp., K	d, mm	Mass, g	x_{cg}/d
2.12	18.4	22.58	292.8	35.51	45.93	0.161

CHAPTER 7
EVALUATION OF UNSTEADY, COUPLED BEHAVIOR (RESEARCH QUESTION 1)

This chapter documents the investigation of Research Question 1 and its associated Hypothesis 1, both of which are restated below:

> **Research Question 1**
>
> What is the appropriate methodology to model aerodynamics, control mechanisms, and flight dynamics for an entry vehicle with unsteady, coupled behavior?

Hypothesis 1: If aerodynamics, flight dynamics, guidance and control mechanisms are captured in a coupled, time-accurate model, then this model will simulate unsteady, coupled entry vehicle behavior that is not captured in the baseline model.

As the capability of the coupled model to accurately capture bluff-body aerodynamics was established in the literature review and validated in Chapter 6, the investigation must determine the significance of the differences in behavior prediction for models that do and do not capture the unsteady, coupled behavior identified in Section 2.2.1. To this end, Hypothesis 1 is decomposed into two sub-hypotheses which are each tested through experimentation. The objective of Experiment 1.1 is to test Hypothesis 1.1, which is restated below:

Hypothesis 1.1: If an entry vehicle is simulated in free flight in the coupled and baseline models, then the difference in response due to unsteady, coupled behavior will be quantified.

Once the behavioral differences that arise in free flight are quantified, further investigation is performed in Experiment 1.2 to test Hypothesis 1.2:

Hypothesis 1.2: If an entry vehicle is simulated in with open-loop control actuation in the coupled and baseline models, then the difference in response due to unsteady, coupled behavior with control system interaction will be quantified.

Section 7.1 documents the generation of the coupled free-flight simulations that will be used for these and future experiments. Section 7.2 describes the construction of a baseline aerodynamic database, a necessary prerequisite to both Experiments 1.1 and 1.2. Sections 7.3 and 7.4 document the procedures, execution, findings, and conclusions of Experiments 1.1 and 1.2, respectively. Based on these results, Section 7.5 draws conclusions that can be applied to Hypothesis 1 as a whole.

7.1 Generation of Coupled Free-Flight Simulations

Free-flight simulations are required to provide trajectory and aerodynamic data for the investigation of aerodynamic surrogate model alternatives, and to serve as a comparison for the baseline flight model. Assuming roughly sinusoidal motion, a Latin Hypercube design was used to sample the space of pitch amplitude and phase, which were then used to generate initial conditions for pitch and pitch rate. A series of eight initial condition cases were created, shown in Table 7.1. These conditions are designed, when used to execute free-flight simulations, to provide persistent excitation [229]. There is only a single mode of oscillation for bluff body entry vehicles, so achieving persistent excitation devolves into adequate sampling of the flight condition space.

Forced-motion startup simulations were executed to develop consistent vehicle motion states and flowfields in accordance with the procedure described in Section 5.1.6. Free-flight simulations at each case were run in the coupled model for a duration of about 0.05

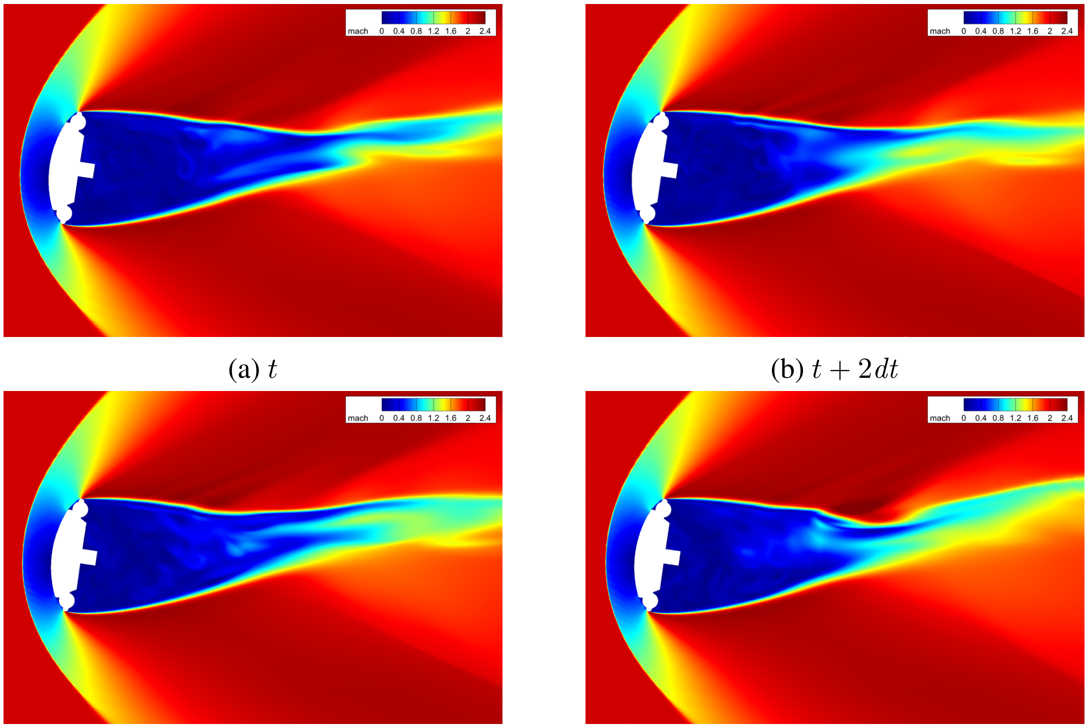
Table 7.1: Coupled simulation initial conditions

Case	Pitch (°)	Pitch Rate ($^\circ/s$)	Pitch Amplitude (°)
1	0.90	997	2.5
2	3.83	3322	9.2
3	8.98	2918	11.6
4	3.57	-2094	6.4
5	0.21	-468	1.4
6	3.23	-877	4.0
7	5.02	235	5.3
8	7.68	-582	7.7

to 0.06 seconds[1]. Each simulation was computed with 400 processors running for 144 hours on the NASA Langley Research Center K-cluster.

The Case 2 free-flight simulation is used as an example to demonstrate the results of the coupled model. The simulation was run for 2,167 time steps, equivalent to about 50 ms. A sequence of four x-z-pane snapshots of the flowfield Mach number from this run are plotted in Figure 7.1. Here the vehicle is at about $\alpha = 9^\circ$ and a freestream Mach number of 2.09. The freestream velocity is oriented directly from left to right. Each subfigure is a snapshot taken two time steps apart. The figure clearly shows the behavior of the flowfield around the vehicle: the bow shock, separated wake, and expansion shocks are all visible. Of particular note is the variation of the wake between each shot, even with negligible motion of the vehicle.

[1] Except for the run at case 8, which had a duration of about 0.03 seconds



(a) t

(b) $t + 2dt$

(c) $t + 4dt$

(d) $t + 6dt$

Figure 7.1: Case 2 free-flight simulation flowfield Mach number sequence starting at $t = 3.625ms$

Figure 7.2 plots the Mach number, showing the vehicle decelerating at a relatively constant rate. The trajectory component history is plotted with respect to the Freestream reference frame in Figure 7.3. Figure 7.3a plots the x-axis decrement, which is calculated using the formula:

$$x_{dec} = r_x - v_{x,0}t \tag{7.1}$$

The results show the vehicle accelerating away from the Freestream frame, equivalent to deceleration for an atmosphere-fixed observer. Figure 7.3b plots the lateral displacement of the vehicle. The z-axis displacement increases over time as a result of gravitational force, with an added oscillatory component resulting from the aerodynamic normal force. The y-axis displacement remains much smaller, and both y and z are many orders of magnitude smaller than the x-axis displacement.

The pitch Euler angle, plotted in Figure 7.3c, shows the large-scale sinusoid-like oscillatory behavior of the vehicle in the angular dimensions. The roll and yaw Euler angles are plotted in Figure 7.3d. The yaw also shows oscillatory behavior, with amplitude increasing over time. However, the maximum amplitude of about $0.45°$ at the end of the run is much smaller than the pitch oscillation, allowing it to be ignored when considering motion in pitch. Although the vehicle starts with zero initial roll rate, a slight roll rate develops over the course of the simulation. However, since the maximum roll angle is less than $0.2°$, this can also be ignored. Since the magnitude of the lateral velocity is small, the values of α and β plotted in Figure 7.4 are essentially equivalent to the pitch and yaw Euler angles, respectively.

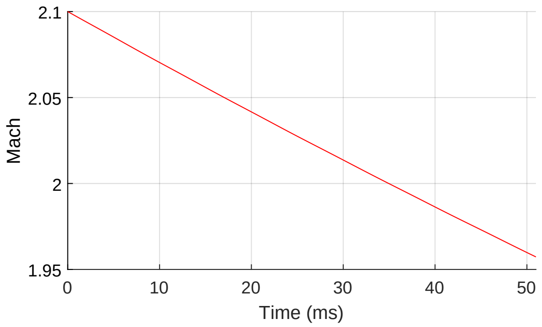
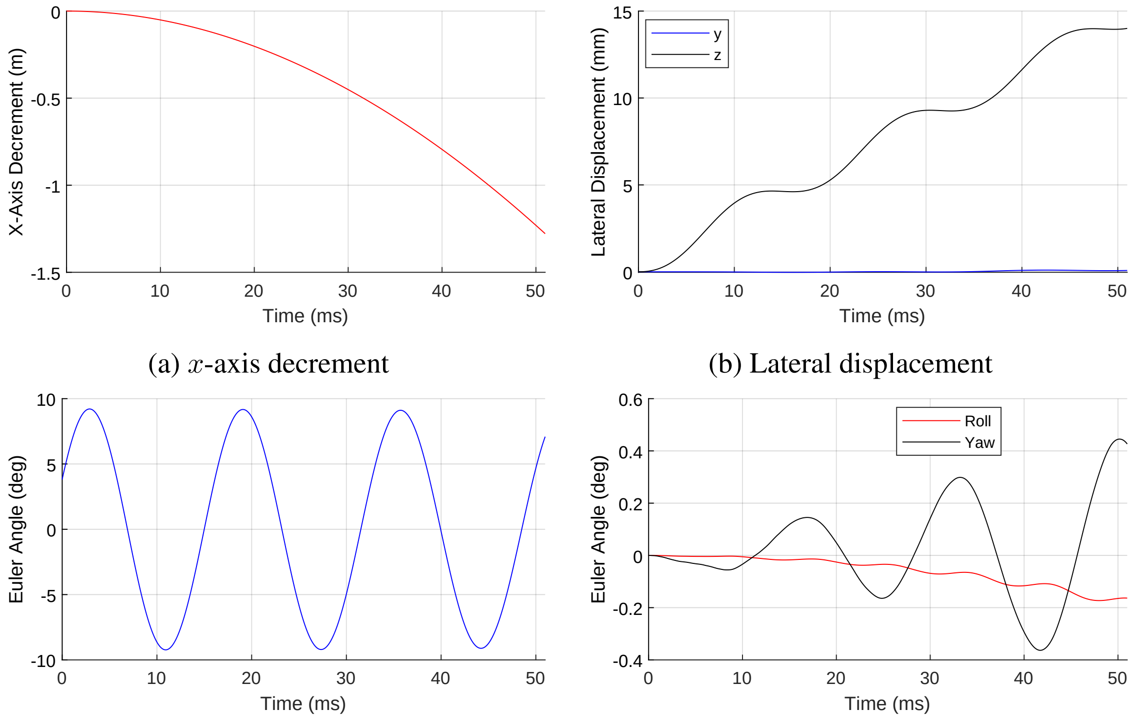


Figure 7.2: Case 2 free-flight simulation Mach number



(a) x-axis decrement (b) Lateral displacement

(c) Pitch angle (d) Roll and yaw angles

Figure 7.3: Case 2 free-flight simulation trajectory

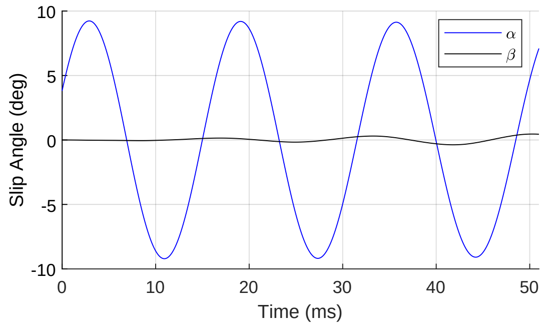


Figure 7.4: Case 2 free-flight simulation aerodynamic slip angles

Figure 7.5 plots the net aerodynamic coefficients[2]. The axial force coefficient plotted in Figure 7.5a shows the unsteady response resulting from oscillation, along with higher-frequency behavior resulting from recirculation in the wake. It is important to note that the variation due to oscillation and the higher-frequency noise are both small relative to the time-average magnitude of the coefficient. This shows similar behavior to the pres-

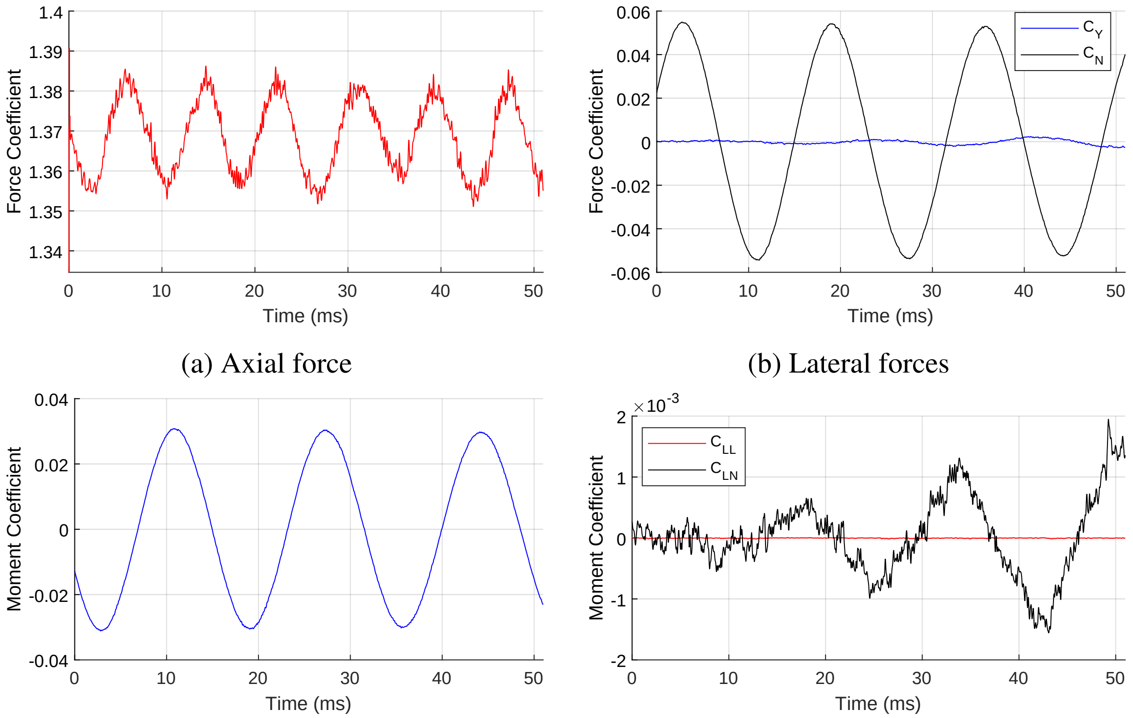
[2] All aerodynamic coefficient results are calibrated to the instantaneous dynamic pressure.

sure results from the MSL ballistic range test in Figure 2.7. The side and normal force coefficients plotted in Figure 7.5b show oscillatory behavior. The amplitude of the normal force is much larger than the side force coefficient, commensurate with the difference in magnitude between the pitch and yaw oscillation. Magnitudes for both lateral force coefficients are much smaller than the axial coefficient. While the high-frequency noise resulting from the wake is still present, it has a much smaller effect relative to the amplitude when compared with the axial coefficient.

Figure 7.5c plots the pitch moment coefficient. Once again, the behavior is mainly oscillatory, with the high-frequency noise still present, but small relative to the amplitude. The yaw and roll moment coefficients are plotted in Figure 7.5d. Notably, the oscillatory behavior of C_{LN} appears to have a triangular pattern as opposed to the sinusoid-like behavior of all the other components. C_{LN} also has a more pronounced high-frequency effect due to the small magnitude of the component. The magnitude of C_{LL} is much smaller than C_{LN}, and both are much smaller than C_M.

The examination of the trajectory and aerodynamic components demonstrates that the model results are behaving qualitatively as expected for a bluff-body entry vehicle that is free to pitch, heave, and decelerate.

Figure 7.6 shows the coverage of these runs across the input dimensions for the database. Given that the initial conditions are constrained to a single plane and the vehicle is axisymmetric, the resulting motion is largely constrained to the same (vertical) plane. The range of Mach numbers is covered in a generally uniform manner. Figure 7.6a demonstrates that the overlapping results de-correlate the Mach and α inputs. The density of points in the α_T dimension decreases roughly linearly with increasing angle. However, this allows for a generally even spacing of data in the dimensions of α_T and the nondimensional pitch rate $\hat{q}$ as shown in Figure 7.6b. While the α and $\hat{q}$ inputs remain correlated, the close spacing of runs will help to mitigate any problems due to interpolation. These free-flight simulations result in trajectory and aerodynamic data at about 17,000 total points.



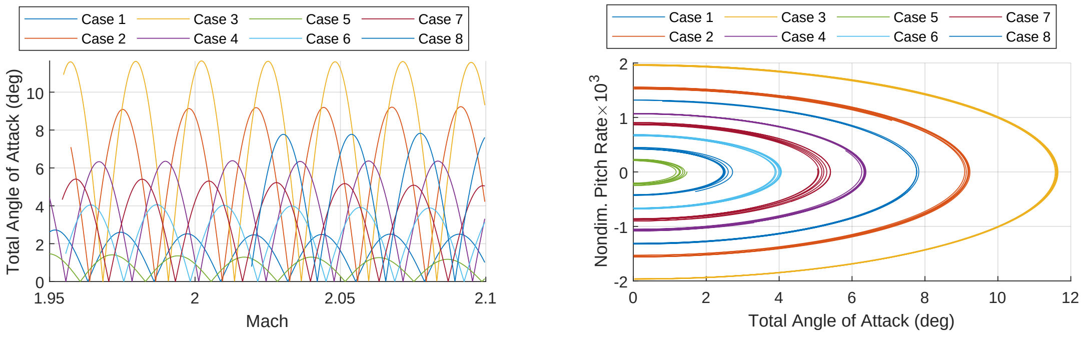
(a) Axial force

(b) Lateral forces

(c) Pitch moment

(d) Roll and yaw moments

Figure 7.5: Case 2 free-flight simulation aerodynamic coefficient history



(a) α_T vs. Mach number

(b) Pitch rate vs. α_T

Figure 7.6: Span of coupled free-flight simulations across database dimensions

7.2 Construction of the Baseline Aerodynamic Database

Experiments 1.1 and 1.2, as well as following experiments, rely on comparisons to a baseline model constructed to represent the state of the art for entry vehicle aerodynamic models. This consists of an aerodynamic database that provides static and dynamic coefficients as a function of Mach and the slip angles. The data used to populate the database will be generated using the FUN3D flow solver [133]. The flow solver is run with the same settings as for the coupled model in Section 7.1. This allows the database to share its source of truth with the coupled model.

7.2.1 Static Coefficients

The static aerodynamic coefficients will be found using time-accurate fixed-attitude FUN3D solutions, acting as an analogue to static wind tunnel tests. A design of experiments (DOE) spanning the range of Mach and angle of attack is created as shown in Figure 7.7. Since the vehicle is axisymmetric, the results are also assumed to hold for total angle of attack. The DOE uses a full-factorial design, consistent with the state of the art for Mars entry vehicle aerodynamic databases as demonstrated by MSL [54]. The points are evenly distributed across the Mach 1.95-2.10 range specified in Chapter 6. The distribution of points in the α_T dimension is weighted toward the smaller values, within which the vehicle will be able to achieve a trim angle of attack. The density of points is chosen to both a) adequately describe the coefficients in the range of conditions, and b) provide enough points to selectively change the density as required in Experiment 2. Table 7.2 compares the average density of points along each input dimension in the baseline static database to the databases used for MSL and Phoenix, the two most recent unique development efforts for Mars entry vehicles. The density of points in the baseline database is much higher in both dimensions, satisfying the first requirement. The total number of points, 44, allows the database to be constructed in increments of 4 points (Chapter 8 will demonstrate that the computational expense of 4 static points is comparable to one coupled simulation). This satisfies the second requirement.

Table 7.2: Average density of static aerodynamic database in the vicinity of Mach 2

Model	M	α_T (1/deg)
Phoenix [56]	1.000	0.200
MSL [54]	1.087	0.208
Baseline	20.00	0.718

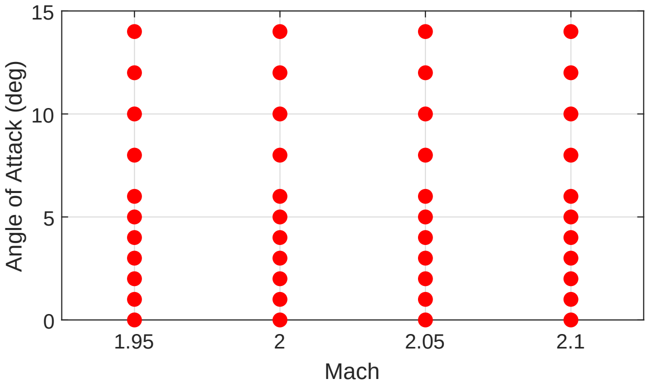


Figure 7.7: Static solution design of experiments

An example of the process is documented below for a single database point. First, a steady solution is generated with 3000 iterations to resolve the gross features of the flow field. The residuals for this run are plotted in Figure 7.8. The meanflow components (density; momentum in the x, y, and z directions; and energy) drop by about an order of magnitude to below 10^{-9}, which is satisfactory for the steady solution. The turbulence residual decreases initially, but then increases after about iteration 500 until it approaches a steady state with a net decrease less than an order of magnitude. This behavior is due to the use of the DES-based turbulence model in a steady simulation. Since the magnitude of the resdual remains below 10^{-7}, this is acceptable for a steady solution.

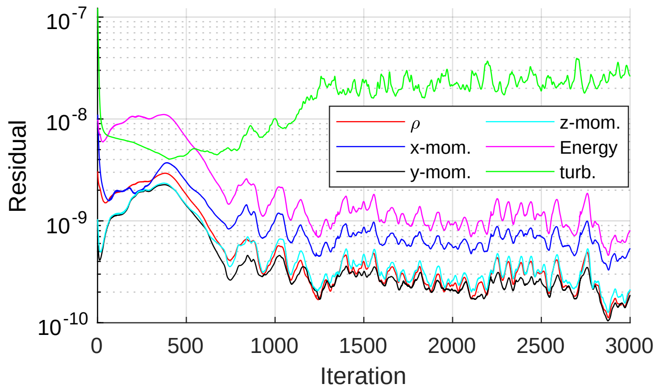
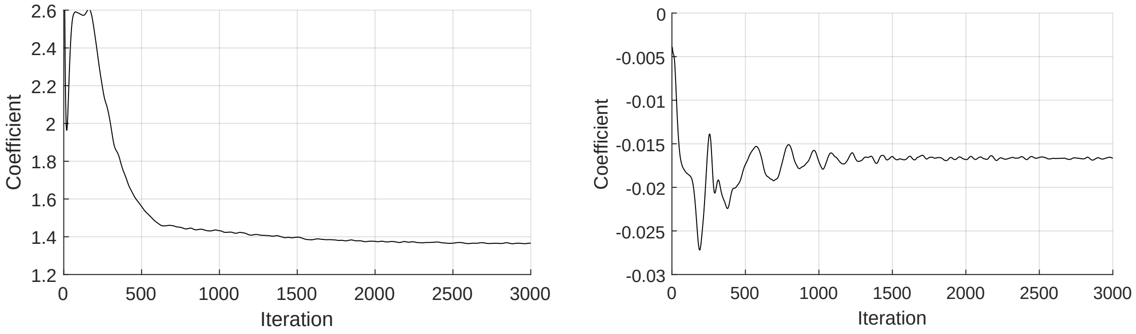


Figure 7.8: Steady-state solution residuals

Figures 7.9a and 7.9b plot the static axial force and pitch moment coefficients, respectively. The number of iterations is shown to be large enough for both coefficients to converge.



(a) Axial force coefficient

(b) Pitch moment coefficient

Figure 7.9: Steady-state coefficient history for M=2.05, α=5° database point

The converged steady flow field is then used to start a time-accurate simulation which is run for 500 time steps, or a duration of 0.12s. This duration is about 70% of the vehicle's damped oscillation period at these conditions. Therefore, the time-accurate solution will be long enough to capture the lowest-frequency steady-state behavior expected in the coupled model. This approach is consistent with the methodology of Stern et al. [128] for the generation of static coefficients.

The residuals are plotted in Figure 7.10 for the first 20 time steps. The initial time steps show variations in the convergence, especially for the turbulence residual. These are both likely the result of transients due to the startup from a steady flow field solution. After iteration 15, and for the rest of the simulation, the residuals behave in a similar manner to that shown in Figure 6.6 for the validation simulations.

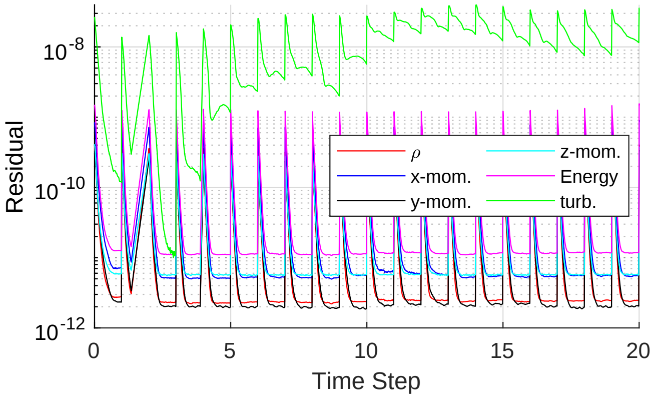
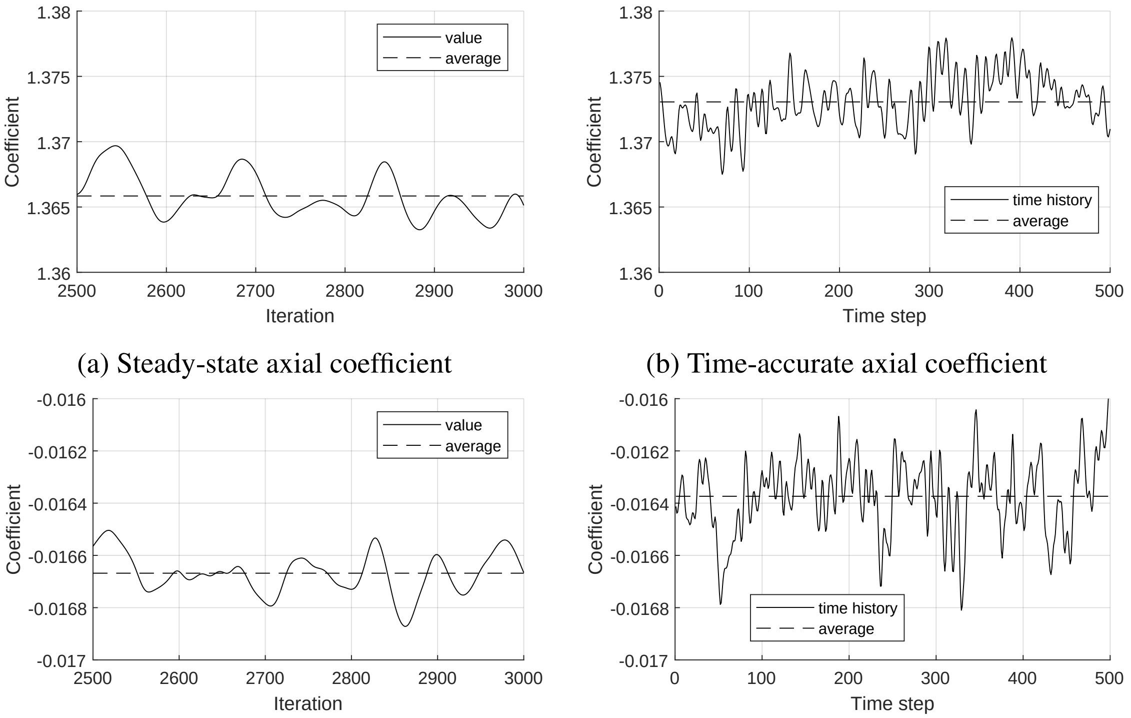


Figure 7.10: Static solution residuals

The examination of Figures 7.8 and 7.10 demonstrate that the flow solver settings are valid for the steady and time-accurate simulations.

Figures 7.11b and 7.11d plot the static axial force and pitch moment coefficient history and their averages. The results show consistent behavior centered around the average, indicating that the effect of the startup transients on the coefficients is negligible.

The time-accurate results can also be compared to the final 500 iterations of the steady solution as shown in Figures 7.11a and 7.11c. The average values for both C_A and C_M are different between the steady and time-accurate solutions. This amounts to about 0.5% for the axial force and about 1.8% for the pitch moment. In addition, the variance of both of the time-accurate results is also increased. These results demonstrate that even considering only the change in fidelity between steady and time-accurate models results in a noticeable change to the aerodynamic behavior.



(a) Steady-state axial coefficient

(b) Time-accurate axial coefficient

(c) Steady-state moment coefficient

(d) Time-accurate moment coefficient

Figure 7.11: Static coefficient history for M=2.05, α=5° database point

7.2.2 Dynamic Coefficients

The baseline database must also include the effects of damping. The roll damping coefficient is ignored since no runs are planned with a significant initial roll rate, hence the only term included will be the pitch damping coefficient C_{M_q}.

As identified in Section 2.4.1, the state-of-the-art models use parameter identification to determine a fit for C_{M_q} from ballistic range tests. These fits are then evaluated at the desired inputs to produce values for the database. Therefore, the free-flight simulations generated in Section 7.1 will be used as an analogue to the ballistic range tests.

The model form for the pitch moment coefficient is based on examples from entry vehicle literature [97, 96]. Given the general uncertainty about the form of the damping coefficient in this flight regime, where little full-scale flight data exists for verification, the forms identified from the literature are not used directly. Instead, promising model terms were included in Equation 7.2:

$$\begin{aligned}\tilde{C}_M &= (c_0 + c_1\Delta M)\sin\alpha + (c_2 + c_3\Delta M)\sin^3\alpha \\ &+ (c_4 + c_5\Delta M)\,\hat{q} + (c_6 + c_7\Delta M)\,|\sin\alpha|\,\hat{q}\end{aligned} \tag{7.2}$$

where ΔM is the difference between Mach number and a chosen reference Mach number M_{ref}, and $\hat{q}$ is the nondimensionalized pitch rate. With eight coefficients, the model has a complexity between those described in References [96] and [97].

The process of fitting the model takes advantage of the fact that the coupled simulation provides direct measurements of the aerodynamic loads. This is done so as not to unfairly burden the baseline database with error propagation that would result from the data reduction and parameter identification solely from the trajectory components. However as a result, the lack of this error represents a deviation from the state-of-the-art which must be considered when evaluating database performance. Substituting data points $1..n$ into Equation 7.2 results in the overconstrained system of equations in Equation 7.3:

$$\begin{bmatrix} s_{\alpha_1} & s_{\alpha_1}\Delta M_1 & s_{\alpha_1}^3 & s_{\alpha_1}^3\Delta M_1 & \hat{q}_1 & \hat{q}_1\Delta M_1 & |s_{\alpha_1}|\hat{q}_1 & |s_{\alpha_1}|\hat{q}_1\Delta M_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{\alpha_n} & s_{\alpha_n}\Delta M_n & s_{\alpha_n}^3 & s_{\alpha_n}^3\Delta M_n & \hat{q}_n & \hat{q}_n\Delta M_n & |s_{\alpha_n}|\hat{q}_n & |s_{\alpha_n}|\hat{q}_n\Delta M_n \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_7 \end{bmatrix} = \begin{bmatrix} C_{M_1} \\ \vdots \\ C_{M_n} \end{bmatrix} \tag{7.3}$$

where $s_{\alpha_i} = \sin\alpha_i$. The value of the coefficients are found using linear least squares (which is equivalent to maximum likelihood estimation in this case). The data are split into snippets of no greater than 200 continuous time steps, and 15% of these snippets are randomly selected to be withheld from the fitting process to serve as validation points[3].

The goodness of fit parameters for the pitch moment coefficient are plotted in Fig-

[3] This procedure is used since it allows for re-stratification while preserving the time-series nature of the data, which will be necessary for some surrogate model alternatives evaluated through Experiment 2.

ure 7.12. Figure 7.12a plots the actual value of the C_M response against the value predicted by the fit. The plot shows a tight line of data lying along the diagonal line with a slope of 1, which is an indication of a good fit.

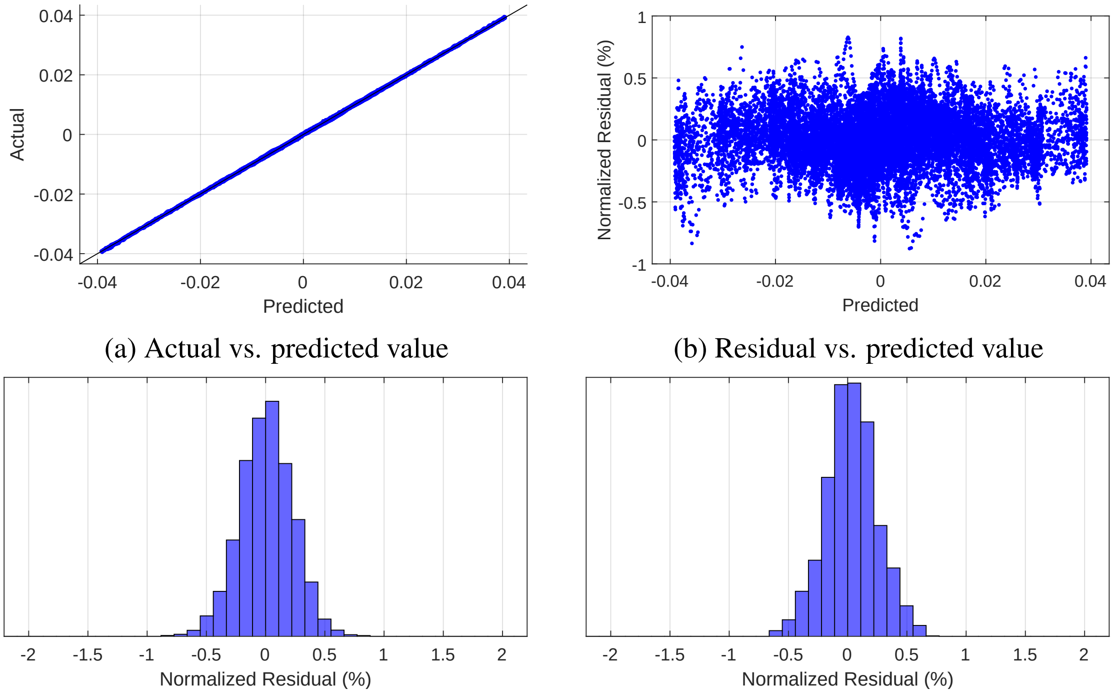
Figure 7.12b plots the residual between the actual and predicted values, normalized by the total range of the response. While there is some clumping in the response, this is a result of highly dense time history of the data. Overall, the residual shows no distinguishable patterns.

The model fit error (MFE) represents the residual for the points used to train the model (i.e., it is a measure of how well the training data fits the surrogate model). The model representation error (MRE) represents the residual for the validation data (i.e., it is a measure of how well the surrogate model matches the actual response). These are plotted in Figures 7.12c and 7.12d, respectively, showing roughly normal distributions centered around zero. Overall, the moment coefficient fit has a three standard deviation uncertainty of $\pm 0.66\%$, which is of a similar magnitude to the $\pm 1.4\%$ average error for the coefficient found by system identification of the ballistic range tests on the same vehicle [97]. Based on the similarity in the error magnitude compared to the ballistic range tests, the moment coefficient fit is judged to be a satisfactory representation.

The damping moment coefficient is found by using the last four terms of Equation 7.2. Rather than using the equation to calculate C_{M_q} directly, the fit is sampled at the same design points used to construct the static coefficient database. Interpolation is used to find the coefficient at the desired conditions in the same manner as the static coefficients. This reflects the way dynamic coefficients are incorporated into the state-of-the-art database models in the literature.

7.2.3 Compilation of Results

The average coefficient results for the time-accurate simulations are shown in Figure 7.13. The axial coefficient results plotted in Figure 7.13a demonstrates the variation in C_A vs. α



(a) Actual vs. predicted value

(b) Residual vs. predicted value

(c) Model fit error

(d) Model representation error

Figure 7.12: Baseline pitch moment coefficient surrogate goodness of fit metrics

as a function of Mach number. The maximum predicted axial coefficient is not at $\alpha = 0°$; rather, it slightly increases until about $\alpha = 4°$ then starts decreasing.

Instead of directly using CN and CM, the database will store normal force and pitching moment data as the coefficient slopes, $C_{N_\alpha} = C_N/\alpha$ and $C_{M_\alpha} = C_M/\alpha$ respectively.This will allow for extrapolation from a small subset of points to be performed during the simulated training in Experiment 2 without an unacceptable breakdown of the model. The lift-curve and pitch moment slopes are plotted in Figures 7.13b and 7.13c, respectively. Note that the error bars decrease with increasing alpha as a result of the inverse performed on the measured data. The model also assumes that the time-average static pitch moment and normal force are zero at $\alpha = 0°$; hence, these values are not reported. Both C_{N_α} and C_{M_α} vary measurably with Mach and α.

The pitch moment damping coefficient is plotted in Figure 7.13d. The coefficient has a large variation in both Mach and α. It is beneficial to examine these values for pitch damping in relation to the pitch damping for neutral stability (i.e., no amplitude growth).

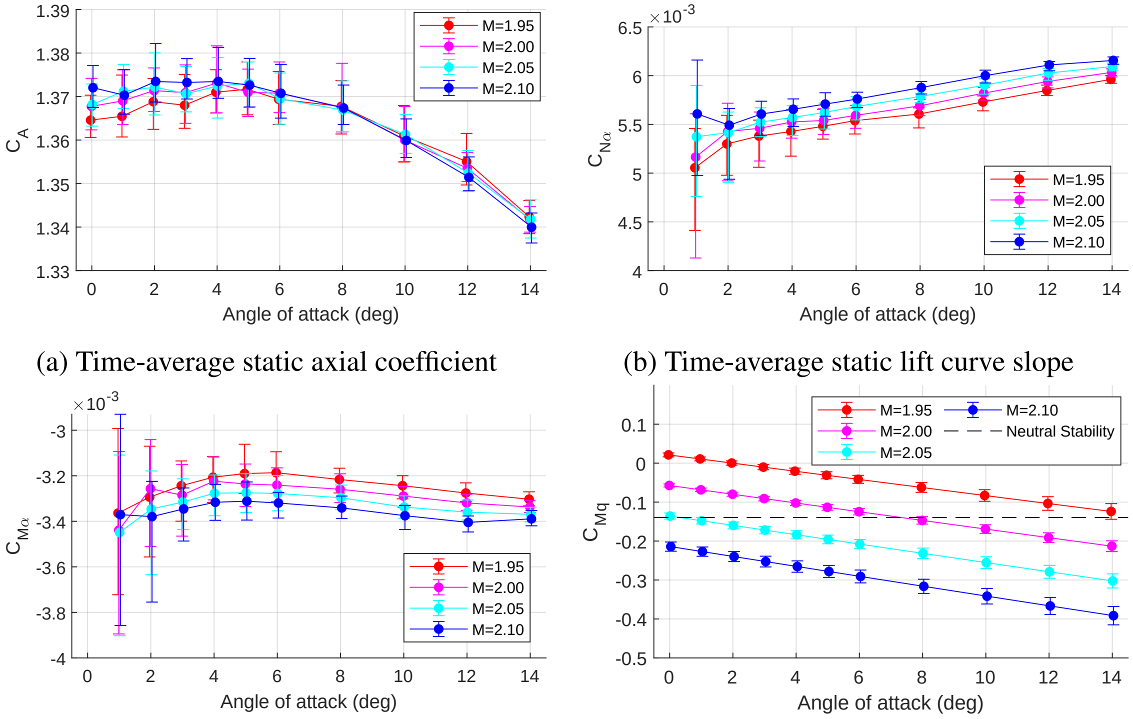
The predicted neutral stability is calculated for the vehicle assuming 3DOF motion and linear aerodynamic coefficients. Schenenberger and Queen [230] present the equations of motion for this case, along with an analytical solution:

$$\alpha = At^{\mu} \cos\left(\nu \log t + \delta\right) \tag{7.4}$$

where the coefficient μ controls the oscillation growth over time:

$$\mu = \frac{mL_{ref}^2 (C_{M_q} + C_{M_{\dot{\alpha}}})}{4 I_{yy} C_A} + 1 \tag{7.5}$$

Setting this coefficient to zero yields the pitch damping coefficient required for neutral stability. Using an average value for the axial coefficient, the pitch damping coefficient for neutral stability is about -0.14. The baseline model therefore predicts that the vehicle is dynamically unstable as both Mach and α_T decrease.



(a) Time-average static axial coefficient

(b) Time-average static lift curve slope

(c) Time-average static pitch moment slope

(d) Pitch moment damping coefficient

Figure 7.13: Database coefficient results

Since the vehicle is symmetric, the models for $\tilde{C}_Y$, $\tilde{C}_{LN}$, $\tilde{C}_{Y_r}$, and $\tilde{C}_{LN_r}$ are assumed to take the same form as and use the same magnitude of coefficients as $\tilde{C}_N$, $\tilde{C}_M$, $\tilde{C}_{N_q}$, and $\tilde{C}_{M_q}$, respectively. The completed aerodynamic database can now be used for the baseline flight simulation model.

7.3 Comparison of Free-Flight Behavior in Baseline and Coupled Models (Experiment 1.1)

7.3.1 Experimental Procedure

Experiment 1.1 will perform free-flight simulations of the vehicle using both the coupled model documented in Section 5.1 and the baseline model documented in Section 5.2 (whose aerodynamic surrogate is described in Section 7.2). This is achieved by using the baseline model to replicate the Case 7 free-flight simulation shown in Section 7.2.2. This case has an initial pitch amplitude that is close to the median of all the cases, at about $5.3°$. Case 7 was also selected due to the fact that it will allow for maximum IMMA actuation in Experiment 1.2 without causing the pitch amplitude to exceed the bounds of the aerodynamic database. The simulation will be run with the same initial conditions, freestream conditions, mass properties, and trajectory propagator settings. The model will be run for the same simulated duration.

Comparisons will be made based on instantaneous results (i.e., the aerodynamic model) and the propagated results over time (i.e., the trajectory). The trajectory responses from both simulations will be directly compared. The aerodynamic responses, however, will not; since the aerodynamic parameters are a function of the trajectory, the results from two different trajectories would also include error propagation. Instead, the trajectory from the coupled simulation is input into the aerodynamic database to provide the response of the baseline model. The aerodynamic results from the coupled simulation are then compared to the baseline response for the exact same trajectory conditions. This eliminates the effect of error propagation on the trajectory that would result from directly comparing the aerodynamic results from the two simulations[4].

Findings will be made based on observations of the time histories of the trajectory and aerodynamic components. In addition, overall metrics will be calculated to quantify the performance of the model.

The root-mean-square (RMS) value of the error is used as an overall measure of the error between the predicted and actual results. It provides a measure of the total deviation from the response. Equation 7.6 shows the formulation with normalization based on the magnitude of the response (as is used for the axial coefficient above):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(\frac{C_i - \tilde{C}_i}{C_i}\right)^2} \tag{7.6}$$

Equation 7.7 shows the formulation where the error is normalized by the range of the response (as is used for the normal force and pitch moment coefficients):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(\frac{C_i - \tilde{C}_i}{C_{max} - C_{min}}\right)^2} \tag{7.7}$$

The RMS error allows for a comparison that is insensitive to variations in the number of data points in each simulation. The change in the formula between Equations 7.6 and 7.7 only precludes direct comparison of different components (e.g. comparison of the axial to the moment coefficients). The RMSE will be evaluated for both the aerodynamic and trajectory components.

The mean error is used as a second metric for the aerodynamic results as a measure of the bias. This metric is considered due to the largely periodic behavior of the vehicle in free flight. The deficiencies of a model with small bias might be more likely to have those errors canceled out through propagation. Conversely, a net bias may be more detrimental

[4] The opposite comparison, i.e., running a time-accurate forced motion CFD simulation along the baseline trajectory, was not performed because the CFD model would not be simulated over a substantially different portion of the design space, nor would any different physical mechanisms be excited.

to the trajectory results than a model with smaller bias yet a larger RMSE.

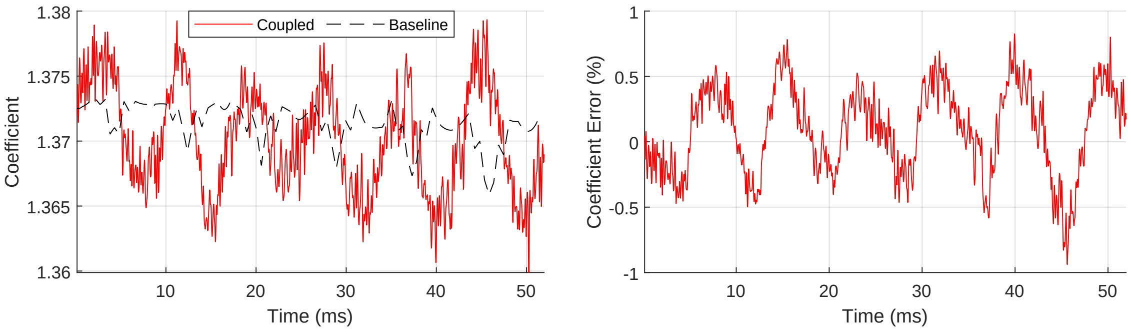
The trajectory components considered are the x-axis decrement, z-axis displacement, and pitch Euler angle; the other three components are ignored since they are inconsequential to the longitudinal behavior. For the same reasoning only C_A, C_N, and C_M are considered for the aerodynamic components.

7.3.2 Results

Aerodynamic Response

The coupled axial coefficient results are plotted with the baseline response in Figure 7.14. Figure 7.14b shows that the magnitude of the error is small. However Figure 7.14a shows qualitative differences in the behavior between the coupled and baseline models. First, the coupled model shows high frequency noise, the result of the wake on the afterbody of the vehicle. This behavior is not predicted in the baseline aerodynamic model.

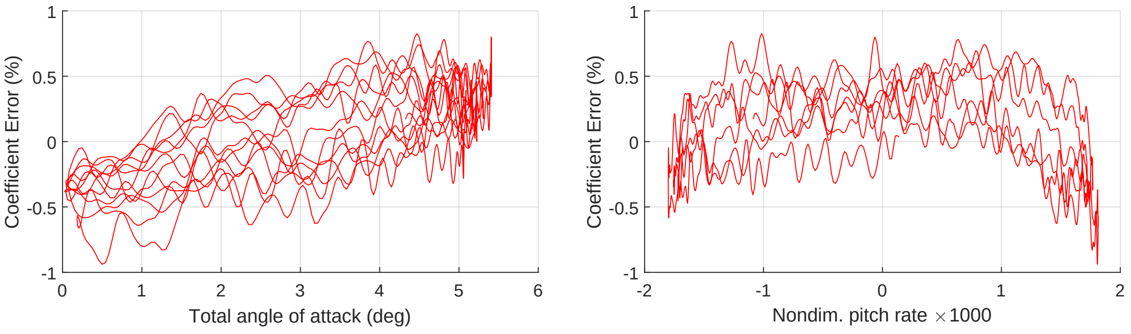
However, there is another noticeable trend in the results: the lower-frequency response of the baseline model sometimes moves in the opposite direction to the coupled model (most noticeable around 45ms, which corresponds to an α_T close to zero). Plotting this error against the total angle of attack and pitch rate in Figure 7.15 shows a noticeable pattern in each dimension. In addition, the axial coefficients from the coupled simulation are being predicted for a relatively constant but nonzero acceleration - effectively some dM/dt - whereas the baseline database assumes a $dM/dt = 0$.



(a) Coefficient

(b) Coefficient error

Figure 7.14: Free-flight axial force comparison



(a) As a function of α_T

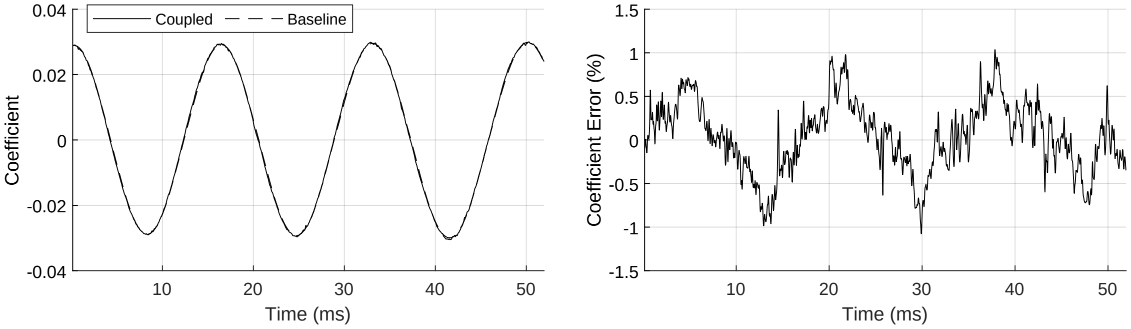
(b) As a function of $\hat{q}$

Figure 7.15: Free-flight axial coefficient error

For a single simulation of free-flight oscillation, it is not possible to de-correlate the α_T and $\hat{q}$ inputs, but the omission of any dynamic component to C_A, namely the effect of pitch rate on the axial force coefficient C_{A_q}, is likely a significant cause. However even for durations of small pitch rate (corresponding to $\alpha_T \approx 5^\circ$ for this run), where conditions in the runs used to produce the database should most closely match the conditions of the dynamic simulation, there is a noticeable offset in the error. One possible explanation is that the time spent at any particular amplitude is not long enough for the flow to develop to the extent that it did in the static simulations.

The normal force coefficient results are plotted in Figure 7.16a, with the error plotted in Figure 7.16b. As with C_A, the coupled model captures high-frequency variation in the moment coefficient as a result of the continuous integration of the unsteady flow field.

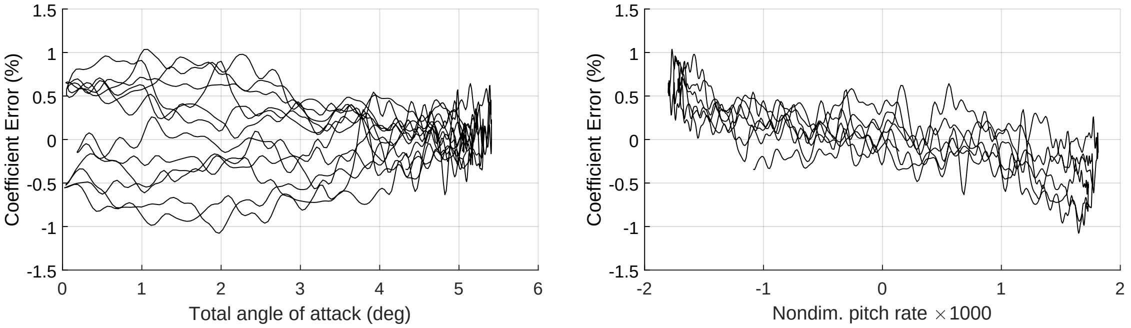
The error again shows a clear oscillatory pattern. The error is plotted against α_T and $\hat{q}$ in Figure 7.17, demonstrating a noticeable trend. As with the C_A error, the variation with $\hat{q}$ is likely due to the omission of dynamic components of the coefficient.



(a) Normal coefficient

(b) Normal coefficient error

Figure 7.16: Free-flight normal force comparison

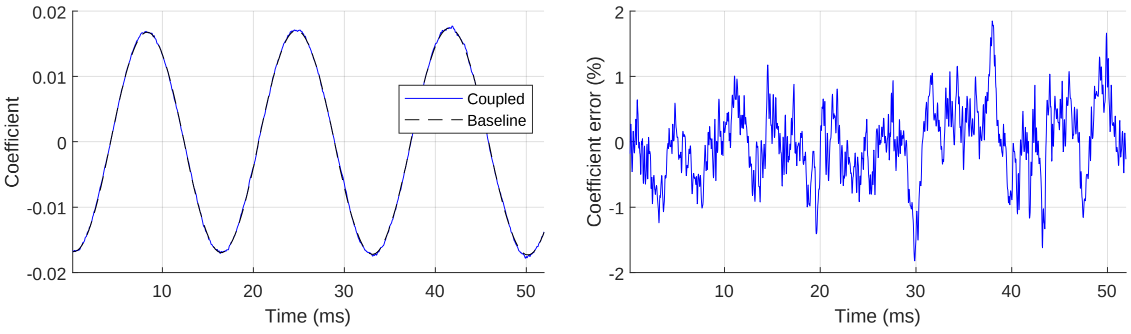


(a) As a function of α_T

(b) As a function of $\hat{q}$

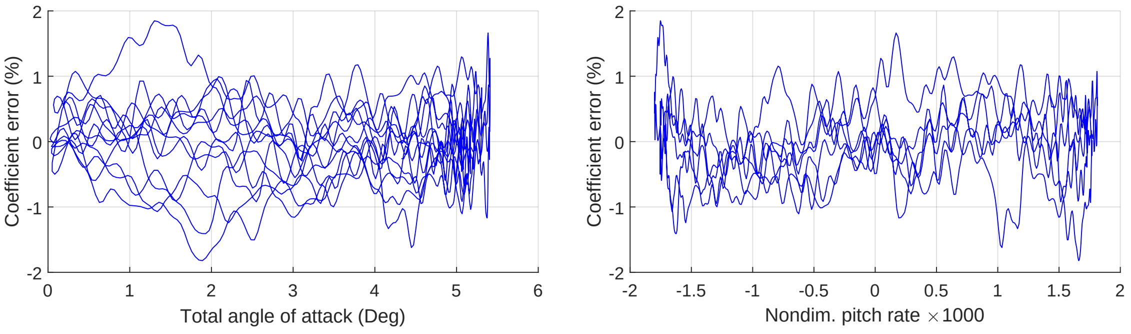
Figure 7.17: Free-flight normal coefficient error

The moment coefficient results are plotted in Figure 7.18. Figure 7.18b does not show a strong oscillatory pattern with time, and there are no distinguishable patterns with α_T or $\hat{q}$ as shown in Figure 7.19.



(a) Moment coefficient (b) Moment coefficient error

Figure 7.18: Free-flight aerodynamic moment comparison



(a) As a function of α_T (b) As a function of $\hat{q}$

Figure 7.19: Free-flight pitch moment error

The overall metrics for each coefficient are listed in Table 7.3.

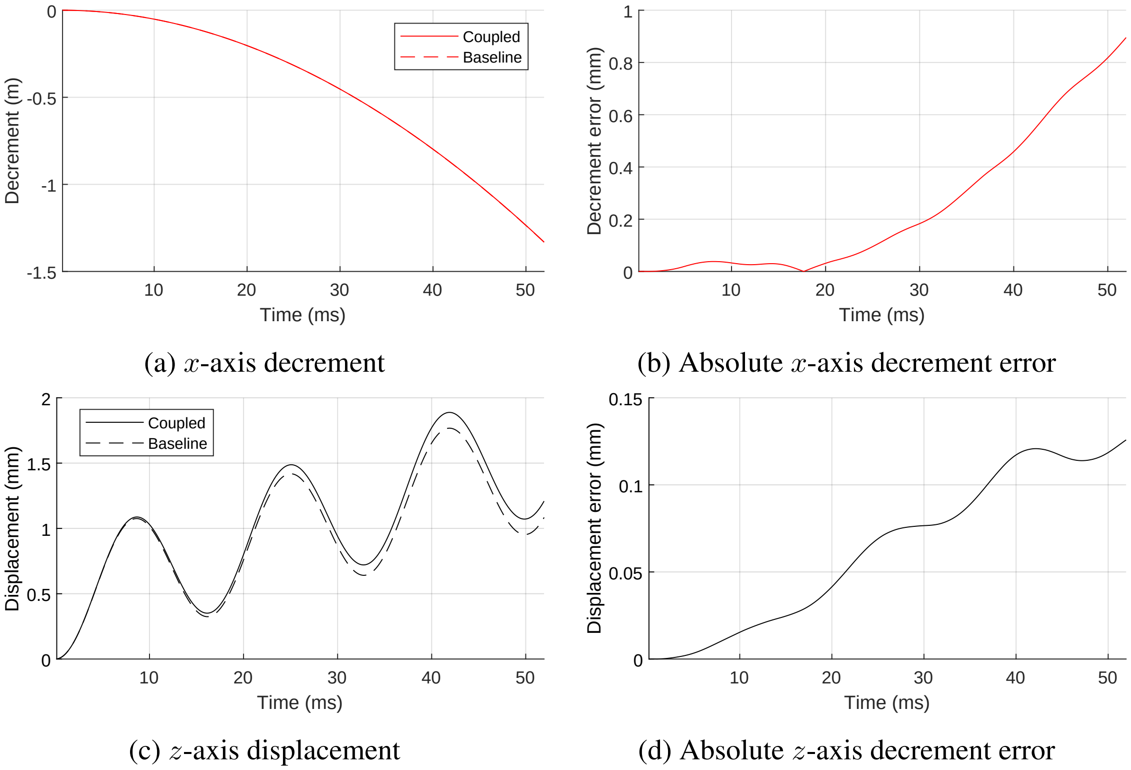
Table 7.3: Free-flight aerodynamic prediction metrics

Coefficient	Mean error (%)	RMS Error (%)
C_A	0.09	0.35
C_N	0.08	2.14
C_M	-0.02	0.56

Trajectory Response

While the aerodynamic components serve as a measure of the instantaneous error, a comparison of the error propagated over time can also be made by considering the trajectory components. The x-axis decrement results are shown in Figure 7.20a. The results for the coupled and baseline models are indistinguishable at full scale. Figure 7.20b shows that the absolute error remains below 1mm over the duration of the run. As expected, the magnitude of the error generally increases with time due to error propagation. The maximum error is less than 0.003% of the total x-axis displacement of the vehicle.

The z-axis displacement is plotted in Figure 7.20c, with the error plotted in Figure 7.20d. As with the x-axis, the z-axis displacement deviates over time due to error propagation.



(a) x-axis decrement

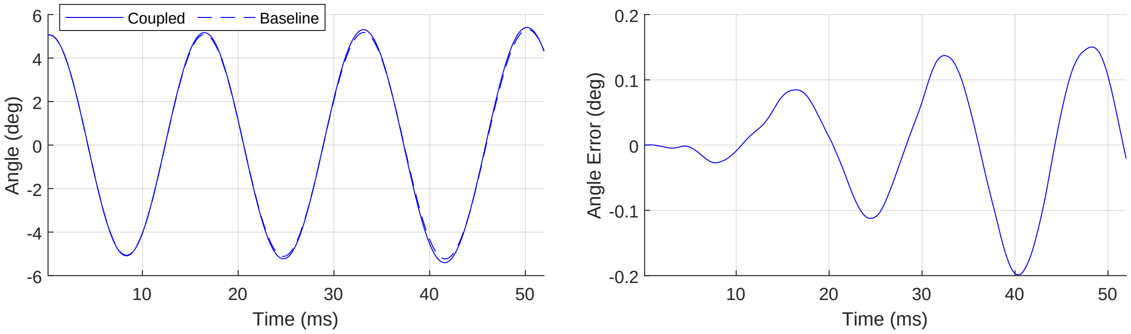
(b) Absolute x-axis decrement error

(c) z-axis displacement

(d) Absolute z-axis decrement error

Figure 7.20: Free-flight translation comparison

The pitch angle results are plotted in Figure 7.21, showing good agreement between the baseline and coupled models. Figure 7.21b shows that the error amplitude of the baseline model increases over time. The plot of total angle of attack in Figure 7.22 more clearly shows that the baseline model under-predicts the amplitude growth over time.



(a) Pitch

(b) Pitch error

Figure 7.21: Free-flight orientation comparison

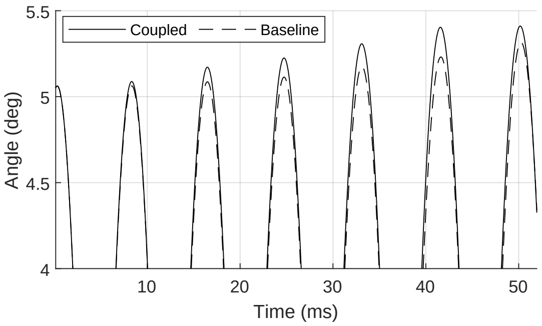


Figure 7.22: Total angle of attack comparison

The overall RMS values of the trajectory component errors are listed in Table 7.4. The magnitudes of the errors are larger than for the aerodynamic components as a result of error propagation over time.

Table 7.4: Free-flight trajectory prediction metrics

Component	RMS Error (%)
x_{dec}	2.75
r_z	4.05
θ	0.80

7.3.3 Findings

The comparison of free-flight simulations on the coupled and baseline models demonstrates the difference in aerodynamic behavior. As expected from the literature review, the baseline model does not capture the high-frequency noise due to the unsteady flow of the wake.

Distinguishable patterns in the axial force coefficient are observed as a result of the differences between the coupled and baseline model. This is expressed in Finding 7.1:

Finding 7.1: The error in C_A and C_N predictions for the free-flight simulation both show a measurable trend with $\hat{q}$, reflecting the omission of dynamic components of the coefficients from the state-of-the-art entry vehicle database.

However, Table 7.3 shows that the mean and RMS errors of C_A and C_N are small, within the magnitude of uncertainty bounds for entry vehicle aerodynamic databases from literature [97, 40].Therefore, this finding helps to inform the search for a replacement aerodynamic surrogate as stipulated by Research Question 2.1, but it does not impact Hypothesis 1.1.

The most significant discrepancy in the trajectory results is the under-prediction of the pitch amplitude growth by the baseline model. Even though no discernible pattern was found in the pitch moment error, this nevertheless resulted in a change in the damping behavior when the error was propagated over time:

Finding 7.2: The baseline model predicts more stable (less) pitch oscillation growth than the coupled model for the free-flight simulation.

Aside from the behavioral findings specifically called out above, the experiment was able to measure the error between the baseline and coupled simulation responses:

Finding 7.3: The results of Experiment 1.1 quantify the differences in aerodynamic and trajectory component responses of the baseline and coupled models for free-flight simulations.

The angular errors were all less than 1%, and the largest linear error, the z-axis displacement, was less than 5%. With a maximum difference of less than $0.15°$, the angular error is not significant compared to the value of 0.17° for precision landing with active guidance:

Finding 7.4: The RMS pitch error for the baseline free-flight simulation is not significant compared to the threshold established for instantaneous tracking error.

7.3.4 Conclusions

On the basis of Finding 7.3, Hypothesis 1.1 is accepted. The difference in free-flight response between the baseline and coupled models is measurable, but not significant for the problem of interest. The comparison between these two runs establishes a threshold for the trajectory and aerodynamic differences between the models for a free-flight simulation of the SIAD vehicle (i.e., in the absence of control actuation).

7.4 Comparison of Open-Loop Control Behavior in Baseline and Coupled Models (Experiment 1.2)

7.4.1 Experimental Procedure

Experiment 1.2 performs further investigation into the differences between the coupled and baseline models. Two more runs are executed using the case 7 initial conditions, now with identical open-loop control actuation. After $t = 1$ms, the z-axis IMMA is commanded to deflect to -16mm, its maximum travel, and remain in this position for the rest of the simulation. The commanded and actual deflection of the IMMA are plotted in Figure 7.23. The IMMA driven by the simulated spring-damper system reached this position at $t =$ 3.2ms. This actuation is representative of a command that might be made during active guidance, to bring the vehicle to some specific α_{trim}.

This deflection corresponds to a static trim angle of attack of about $4.3°$. As a result, the vehicle is anticipated to have a significantly different pitch response, making direct comparison to the free-flight simulations more difficult. However, there is no way to design the experiment such that the vehicle flies through the same region of the input variables while simultaneously exercising the control actuation to a significant degree.

Aside from the actuation of the control mechanism, the experimental procedure is identical to Experiment 1.1. However in addition to the comparison of results between the baseline and coupled simulations (shown in Section 7.4.2), Experiment 1.2 will also compare the change in error relative to the two free-flight simulations performed in Experiment 1.1 (shown in Section 7.4.3).

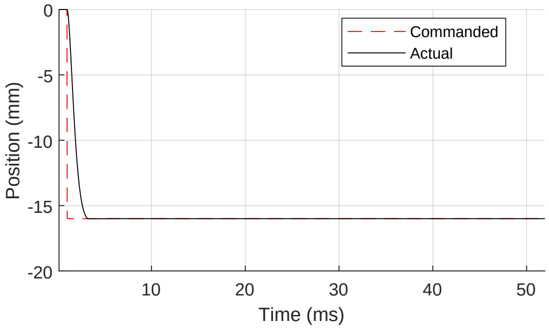
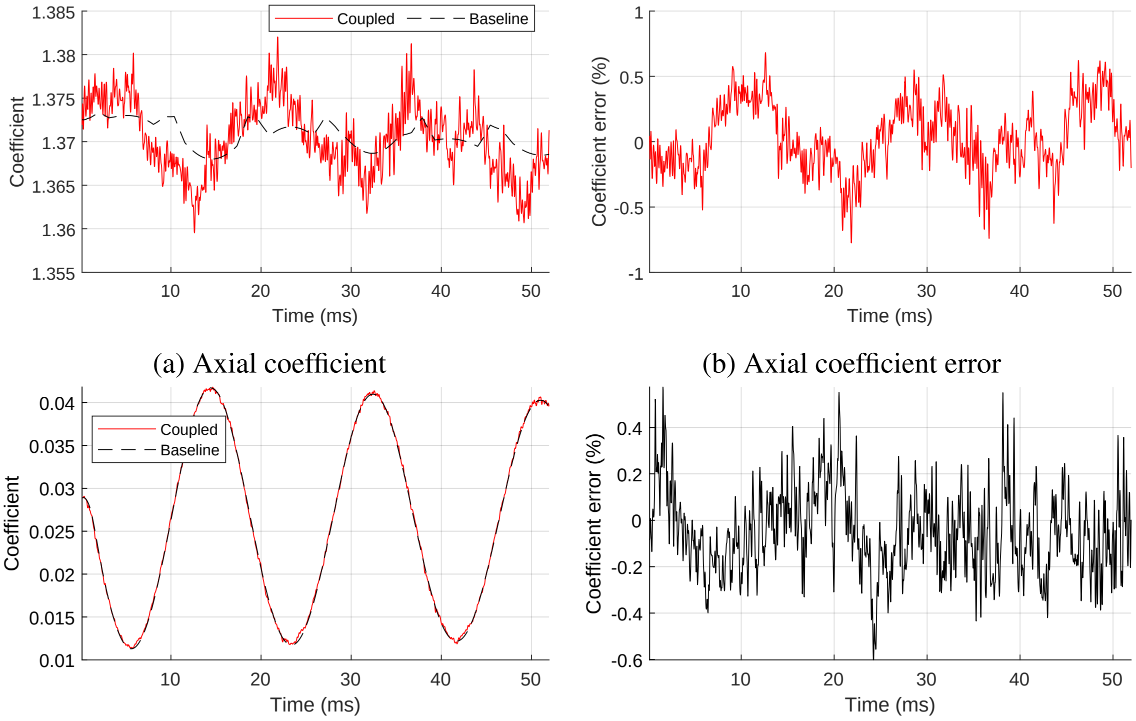


Figure 7.23: z-axis IMMA position

7.4.2 Results

The axial and normal force coefficients plotted in Figure 7.24 show the same qualitative behavior observed in the free-flight simulations, as do the moment coefficients plotted in Figure 7.25a.



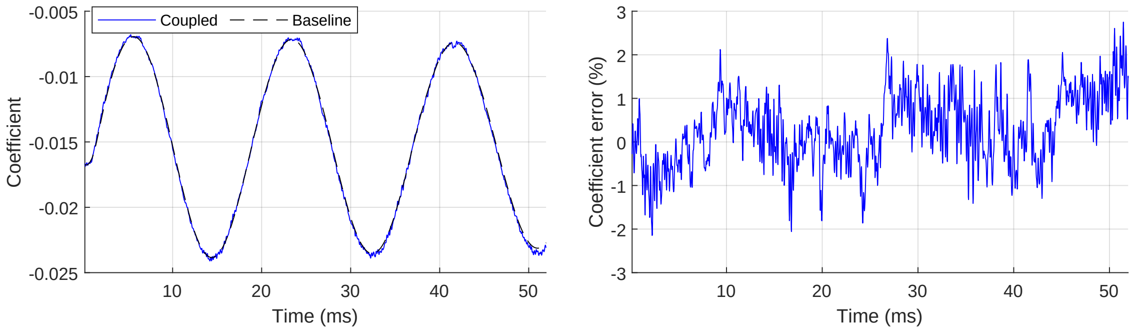
(a) Axial coefficient

(b) Axial coefficient error

(c) Normal coefficient

(d) Normal coefficient error

Figure 7.24: Open-loop aerodynamic force comparison



(a) Moment coefficient

(b) Moment coefficient error

Figure 7.25: Open-loop aerodynamic moment comparison

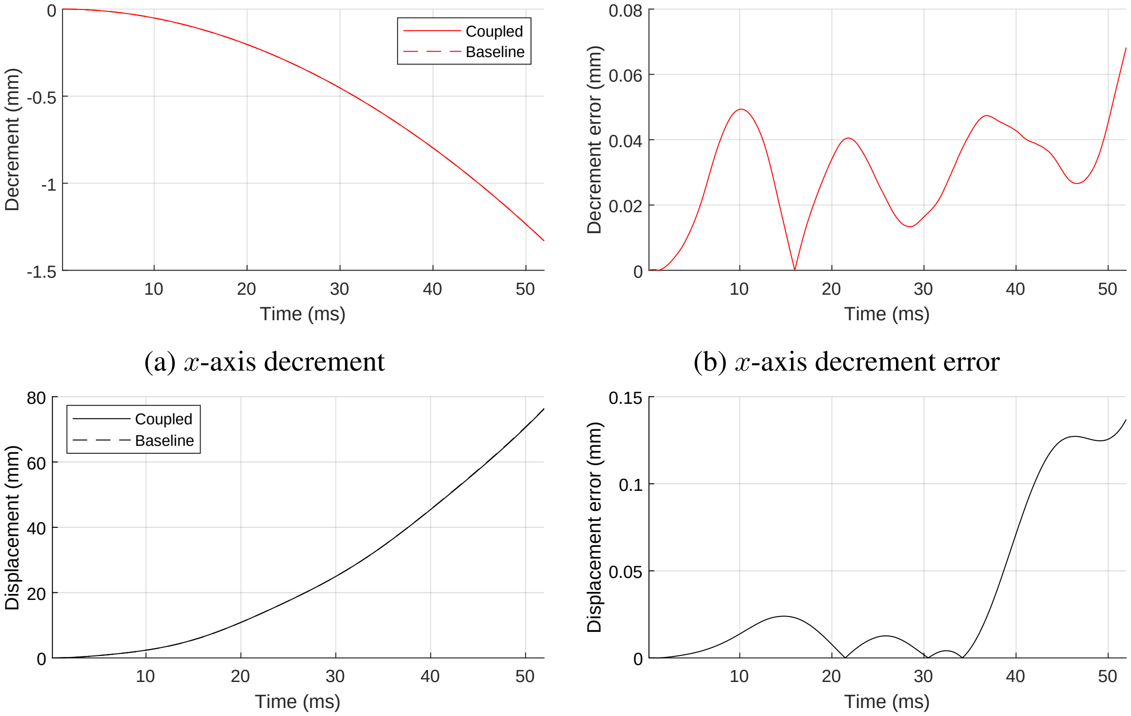
The overall metrics for the open-loop aerodynamic prediction are listed in Table 7.5:

Table 7.5: Open-loop aerodynamic prediction metrics

Coefficient	Mean error (%)	RMS Error (%)
C_A	0.03	0.26
C_N	-0.01	1.87
C_M	0.31	0.86

Figure 7.26 demonstrates that, as with the free-flight simulations, error in the x-axis decrement and z-axis displacement is insignificant.

The pitch response of both simulations is shown in Figure 7.27. As with the free-flight simulations, the largest difference is in the amplitude growth. In this instance the baseline model predicts an amplitude growth, whereas the coupled model shows a slight decrease in amplitude. The oscillation frequency also starts to deviate between the simulations.



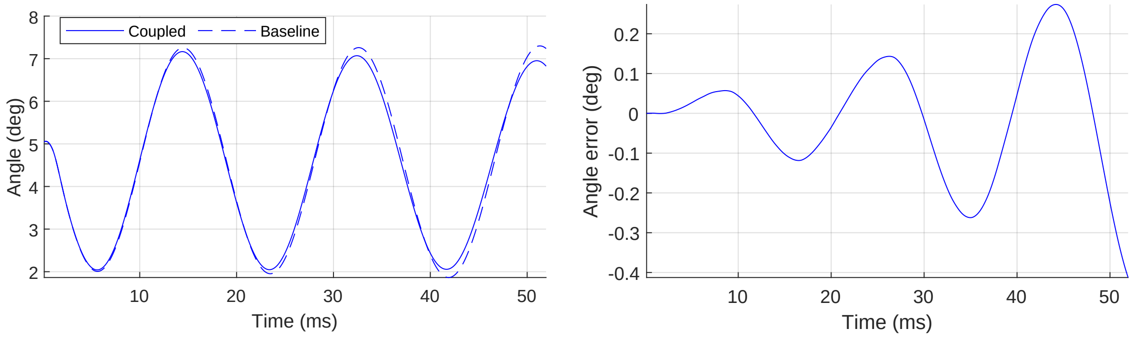
(a) x-axis decrement

(b) x-axis decrement error

(c) z-axis displacement

(d) z-axis displacement error

Figure 7.26: Open-loop translation comparison



(a) Pitch (b) Pitch error

Figure 7.27: Open-loop orientation comparison

The overall metrics for the trajectory components are listed in Table 7.6:

Table 7.6: Open-loop trajectory prediction metrics

Component	RMS Error (%)
x_{dec}	0.00
r_z	0.08
θ	3.47

7.4.3 Evaluation of the Impact of Control Actuation on Error

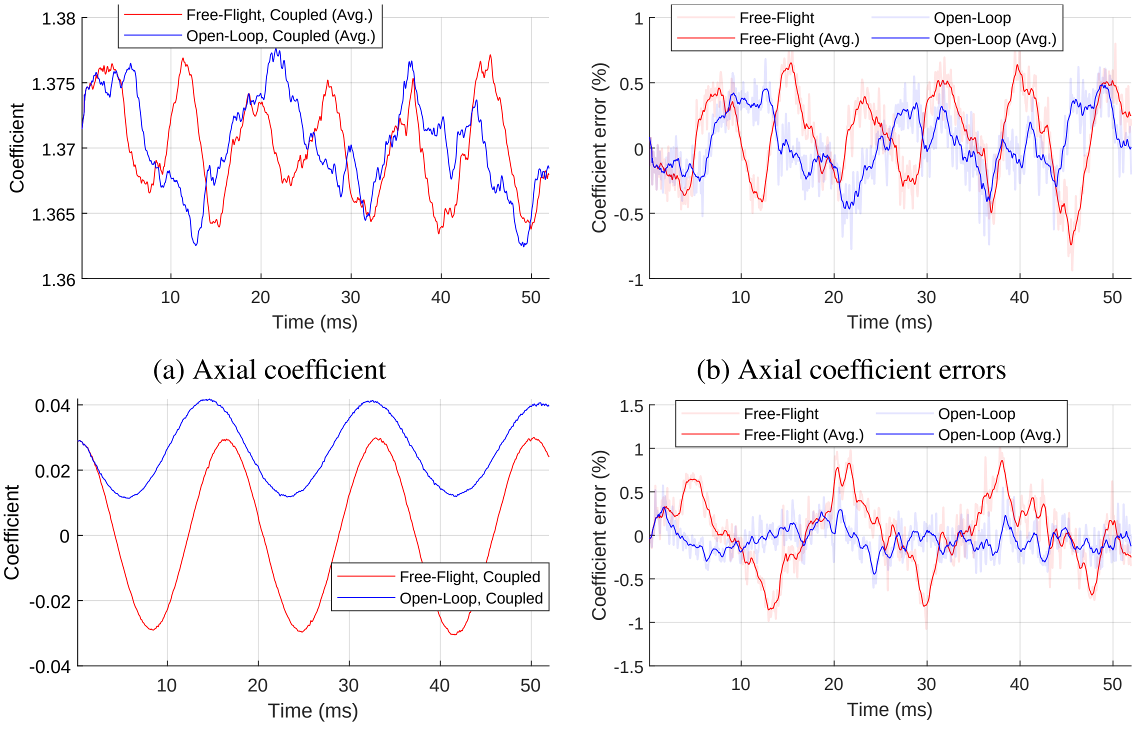
Evaluation of the open-loop simulations on their own yields little additional information beyond the results from the free-flight simulations. Hwever by comparing the simulations in the baseline and coupled model for both free-flight and open-loop control, an investigation can be made into the effect of control actuation on the propagated and instantaneous error.

Aerodynamic Response

Figure 7.28a plots the coupled model responses for the free-flight and open-loop simulations. A centered, 31-step moving average is applied to the errors in the figure to improve readability. The figure demonstrates that the true behavior in these two responses is similar.

Note that the apparent difference in frequency between the oscillatory behavior is due to the different behavior of the α_T oscillation (the open-loop response does not cross $\alpha_T = 0$, and is therefore not reflected). Figure 7.28 plots a comparison of the error between the baseline aerodynamic response for the free-flight and open-loop simulations against their respective coupled responses. The open-loop C_A error has smaller peaks than the free-flight error, but in each the error is generally centered about zero.

The normal coefficients are plotted in Figure 7.28c, with the error plotted in Figure 7.28d. Figure 7.28c shows how the responses are closely overlaid before the IMMA causes the C.G. to shift. Notably, the free-flight simulation has a much higher relative error, even considering the large oscillation amplitude of the response.



(a) Axial coefficient

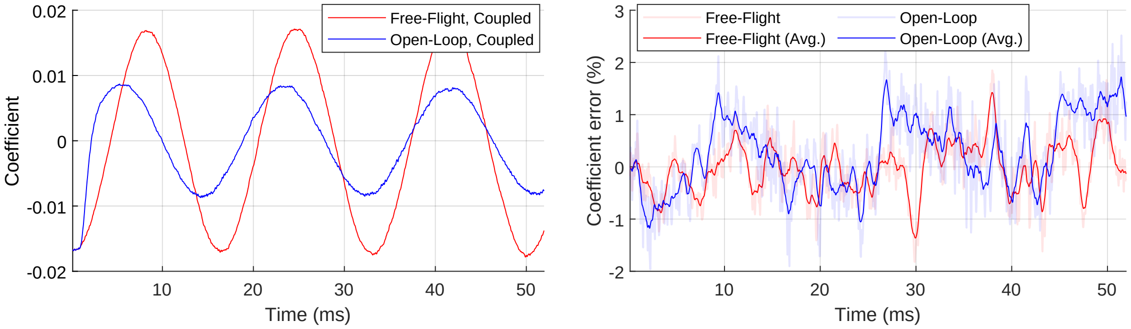
(b) Axial coefficient errors

(c) Normal coefficient

(d) Normal coefficient errors

Figure 7.28: Comparison of predicted to actual force coefficient

Figure 7.29a plots the pitch moment coefficient about the C.G. of the vehicle, $C_{M,cg}$, and Figure 7.29b plots the error for each simulation. Again, the responses closely match before the IMMA actuation. The open-loop simulation shows consistently higher peaks in error magnitude, as well as a bias that trends away from zero with increasing time.



(a) Pitch moment coefficient (b) Pitch moment coefficient errors

Figure 7.29: Comparison of predicted to actual pitch moment coefficient

The values for the overall error metrics are compared in Tables 7.7 and 7.8. Table 7.7 shows that the mean error for both C_A and C_N are decreased for the open-loop simulation. $C_{M,cg}$ is the only component that increases, but the increase is larger than any other change. All of the mean errors fall under 0.5%. Table 7.8 shows similar behavior for the change in RMS error: the error for the open-loop C_A and C_N prediction is decreased, while the $C_{M,cg}$ error increases.

Table 7.7: Comparison of coefficient mean error

Simulation	C_A Error (%)	C_N Error (%)	$C_{M,cg}$ Error (%)
Free-Flight	0.09	0.08	-0.02
Open-Loop	0.03	-0.01	0.31

Table 7.8: Comparison of coefficient RMS error

Simulation	C_A Error (%)	C_N Error (%)	$C_{M,cg}$ Error (%)
Free-Flight	0.35	2.14	0.56
Open-Loop	0.26	1.87	0.86

Comparison of Baseline and Coupled Models at Similar Flight Conditions

While the aerodynamic comparison was performed in such a way that it eliminates the effect of trajectory error propagation from each baseline and coupled simulation, the free-flight and open-loop runs have entirely different trajectories. Therefore an attempt is made to compare the aerodynamic behavior at as close to the same flight conditions as possible. The maximum difference in instantaneous Mach number between the runs is less than 0.0001, so the vehicle state will be considered based on pitch rate and angle of attack.

Figure 7.30 plots the angle of attack and pitch rate for the free-flight and open-loop simulations run in the coupled model. A region is indicated where the these two input variables are substantially similar. The region is bounded by $4.5° \leq \alpha \leq 5.0°$ and $3.37 \times 10^{-3} \leq \hat{q} \leq 4.36 \times 10^{-3}$, which are a small fraction of the total range spanned by the simulations. The free-flight and open-loop coupled simulations have a similar number of points within this region: 161 and 141 points, respectively[5].

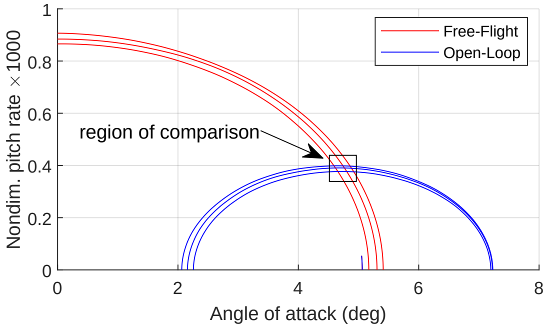
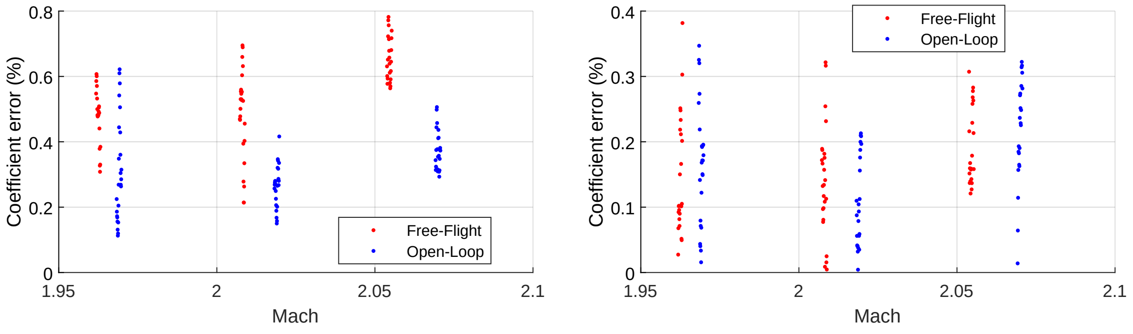


Figure 7.30: Trajectory conditions

Figure 7.31a plots the absolute C_A error as a function of Mach for time steps within the region of comparison. The clustering of the data shows that the Mach number does not change significantly within each individual snippet. The free-flight simulation has a higher average error than the open-loop simulation, a deviation that increases with Mach number. Since the vehicles pass through this region at different times, the Mach number is slightly offset, but assuming that the behavior was the result of change in Mach number would mean that the error oscillates with Mach — an unlikely rationale. Figure 7.31b shows that the average absolute C_N error is roughly equal between the two simulations, with the difference being much smaller than the variance of each cluster.

[5] The similar region with negative $\hat{q}$ values contains only half the number of open-loop data points as free-flight points and is therefore not suitable for comparison.



(a) Absolute C_A error

(b) Absolute C_N error

Figure 7.31: Force coefficient error comparison at similar flight conditions

Figure 7.32 shows the absolute $C_{M,cg}$ error as a function of Mach for time steps within the region of comparison. Since the vehicles pass through this region at different times, the Mach number is slightly offset, but for each snippet, the open-loop data trends higher than the free-flight data.

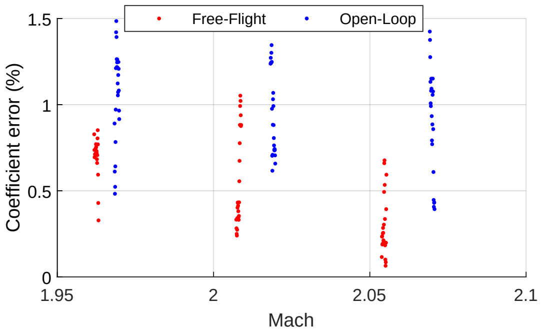


Figure 7.32: Absolute $C_{M,cg}$ error comparison at similar flight conditions

Figures 7.9 and 7.10 list the overall error metrics evaluated for this region. While the RMS C_M error has a similar magnitude to the average over the whole run, the mean C_M error is increased for both the free-flight and open-loop simulations. For both moment coefficient metrics, the open-loop simulation has about twice the error as the free-flight simulation.

Table 7.9: Comparison of coefficient mean error for region of similar flight conditions

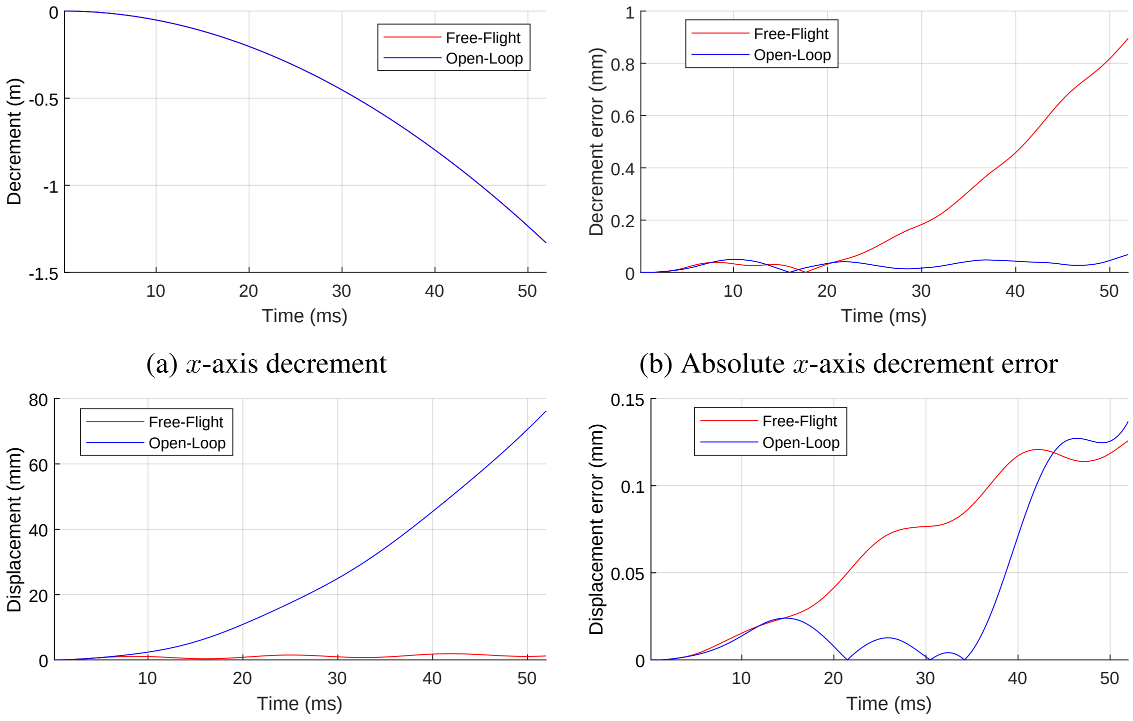
Simulation	C_A Error (%)	C_N Error (%)	$C_{M,cg}$ Error (%)
Free-Flight	0.54	-0.12	0.52
Open-Loop	0.32	-0.14	0.97

Table 7.10: Comparison of coefficient RMS error for region of similar flight conditions

Simulation	C_A Error (%)	C_N Error (%)	$C_{M,cg}$ Error (%)
Free-Flight	0.55	0.18	0.58
Open-Loop	0.34	0.19	1.01

Trajectory Response

Figure 7.33b plots the x-axis decrement error between the free-flight simulations in the coupled and baseline models, and compares this to the same error for the open-loop simulations. The final error for the open-loop simulations is much smaller than for the free-flight simulations, however both errors are still negligible compared to the total x-axis displacement. Figure 7.33c plots the error in the z-axis displacement, which has a similar magnitude between the two sets of simulations even though the magnitude of the open-loop response is much larger than the free-flight simulation. The error in the free-flight simulation generally grows at a linear rate, whereas the open-loop simulation has a substantial increase in error starting around 35ms.



(a) x-axis decrement

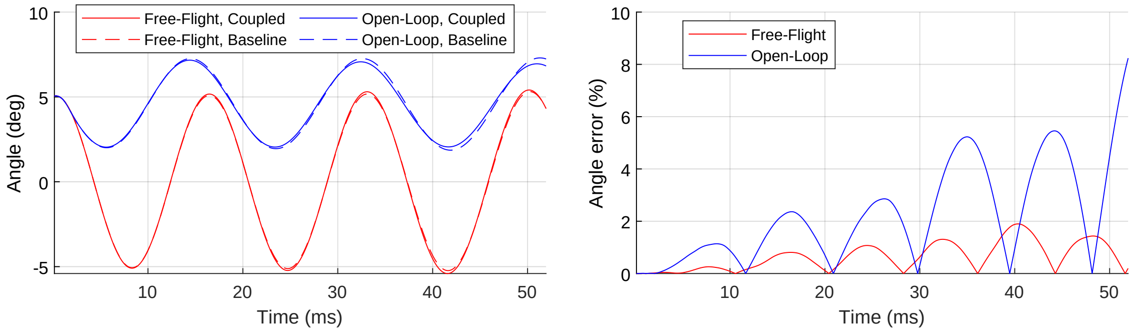
(b) Absolute x-axis decrement error

(c) z-axis displacement

(d) Absolute z-axis displacement error

Figure 7.33: Comparison of translation error between free-flight and open-loop simulations

The pitch response is examined in terms of the relative error in both frequency and amplitude of oscillation. The pitch response for all four simulations is plotted in Figure 7.34a, showing the change in frequency of the baseline simulations to their respective coupled simulations. The frequency, measured from peak to peak, of the baseline free-flight simulation deviates by less than 0.25%, whereas the baseline open-loop simulation is about 2% off after two full periods. Figure 7.34b plots the pitch error of each baseline simulation as a fraction of the average amplitude of their respective coupled simulations. The relative error of the open-loop simulation is significantly higher than for the free-flight simulation. While the maximum error magnitude of the free-flight simulation is less than 2%, the open-loop simulation error grows to over 8%, demonstrating that the open-loop simulation has a much lower success in predicting the pitch damping.



(a) Pitch angle

(b) Absolute pitch angle error

Figure 7.34: Comparison of pitch angle error between free-flight and open-loop simulations

The overall metrics for the trajectory component errors are compared in Table 7.11. The open-loop simulation has substantially less error in the x and z axes. The change is larger than observed for C_A and C_N shown in Tables 7.7 and 7.8, due to the compounding effect of error propagation. However, the average pitch angle error is substantially increased.

Table 7.11: Comparison of trajectory RMS error

Simulation	x_{dec} Error (%)	r_z Error (%)	θ Error (%)
Free-Flight	2.75	4.05	0.80
Open-Loop	0.00	0.08	3.47

7.4.4 Findings

Substitution of the baseline aerodynamic model for the coupled CFD-RBD model results in errors in the trajectory and aerodynamics for a simulation with open-loop IMMA control actuation, as was found for the baseline free-flight simulation:

Finding 7.5: The results of Experiment 1.2 quantify the differences in aerodynamic and trajectory component responses of the baseline and coupled models for free-flight simulations.

The remaining findings of Experiment 1.2 focus on the comparison of results between Experiments 1.1 and 1.2.

For the axial and normal force coefficients, the open-loop simulation actually yielded smaller errors. A natural possibility is that since the open-loop simulation passes through a different range of α_T it encounters different magnitudes of error. Section 7.4.3 only provides a direct comparison at similar flight conditions for a limited region of the data, but this showed similar differences in the error.

The moment coefficient, in contrast, had increased error:

Finding 7.6: The aerodynamic moment response predicted by the baseline model for the open-loop simulation had increased error to the coupled model compared to the free-flight simulation.

The results for the whole simulation were again corroborated by the results for the region of similar flight conditions. Both the mean and RMS error were increased. Nevertheless, the error is still about 1%, which cannot be judged to be significant on its own.

For the trajectory response, the translational components had decreased error, but the pitch response had increased error. It was determined that the pitch error arises from the pitch damping behavior:

Finding 7.7: The baseline model had increased error in pitch oscillation amplitude growth/decay for the open-loop simulation compared to the free-flight simulation.

For the free-flight simulation, the difference in amplitude growth rate was measurable but not significant; it cannot be stated that for a longer-duration simulation the models would not arrive at the same stable frequency (assuming the deviation in Mach number is negligible). In contrast for the open-loop simulation, the baseline model predicted amplitude growth, whereas the coupled model predicted amplitude decay.

With a maximum difference of about $0.4°$ after only three oscillation periods, the angular error is significant compared to the value of 0.17°for precision landing with active guidance:

Finding 7.8: The RMS pitch error for the baseline open-loop simulation is significant compared to the threshold established for instantaneous tracking error.

7.4.5 Conclusions

On the basis of Finding 7.5, Hypothesis 1.2 is accepted. Furthermore, the difference in the pitch results was found to be significant, demonstrating that the angular results are the most important for these short-duration simulations, because this has the greatest effect on the outer guidance loop.

Finding 7.8 indicates that for a given stringency in the controller requirements, the increase in error of the baseline model with the inclusion of the guidance and control system could provide an engineering justification for the use of the coupled model. This demonstrates the suitability of the problem of interest for the purpose of investigating the remaining research questions.

7.5 Synthesis of Conclusions from Experiments 1.1 and 1.2

Research Question 1 (What is the appropriate methodology to model aerodynamics, control mechanisms, and flight dynamics for an entry vehicle with unsteady, coupled behavior?) can be answered for the problem of interest using the findings from Experiments 1.1 and 1.2.

The flight model with coupled aerodynamics and flight dynamics predicts unsteady behavior similar to that measured in physical experiments for bluff bodies as identified in the literature review, demonstrating the model's suitability for the problem of interest. This also supports the assumption that the coupled model can act as a source of truth for the purposes of this work.

The baseline database has measurable deficiencies when compared to the coupled model.

Using an aerodynamic database with the twin advantages of 1) a substantially higher density of points than is commonly found in the state of the art, and 2) directly using the results of the coupled simulation for characterizing C_{M_q}, the baseline flight simulation model still yielded results that were measurably different from the coupled model. As expected, the baseline model is incapable of predicting the high-frequency fluctuations in force and moment coefficients that are captured in the coupled simulation. However, the magnitude of their effects on the aerodynamics is small, as is their effect on the trajectory. The difference in the lower-frequency behavior does result in measurable differences in the free-flight predictions of the baseline and coupled models. As hypothesized, these changes are also exacerbated when simulating control actuation.

Based on these conclusions Hypothesis 1 is accepted. The coupled model is suitable for modeling the problem of interest, and it achieves results that are significantly different in some cases from the results of the baseline model.

The limitations of these experiments are rooted in the limitations on the problem of interest. The unsteady aerodynamic behavior on its own (without control actuation) did not result in significant differences to the baseline results. This is likely a result of the benign separation and attachment behavior of the vehicle as discussed in Section 6.1. There is also the possibility that significant differences in the free-flight runs could be achieved with a wider range of freestream conditions. Greater differences might also have been achieved by simulating open-loop runs at other initial condition cases with different pitch oscillation amplitude. It is also possible that simulating a vehicle with RCS would exhibit larger differences for the open-loop simulation. However while the real-world significance of the differences between the models is small, the important point is that even for a control mechanism as benign as an IMMA, the differences are measurable.

Figure 7.35 presents a summary of the investigation performed in order to answer Research Question 1.

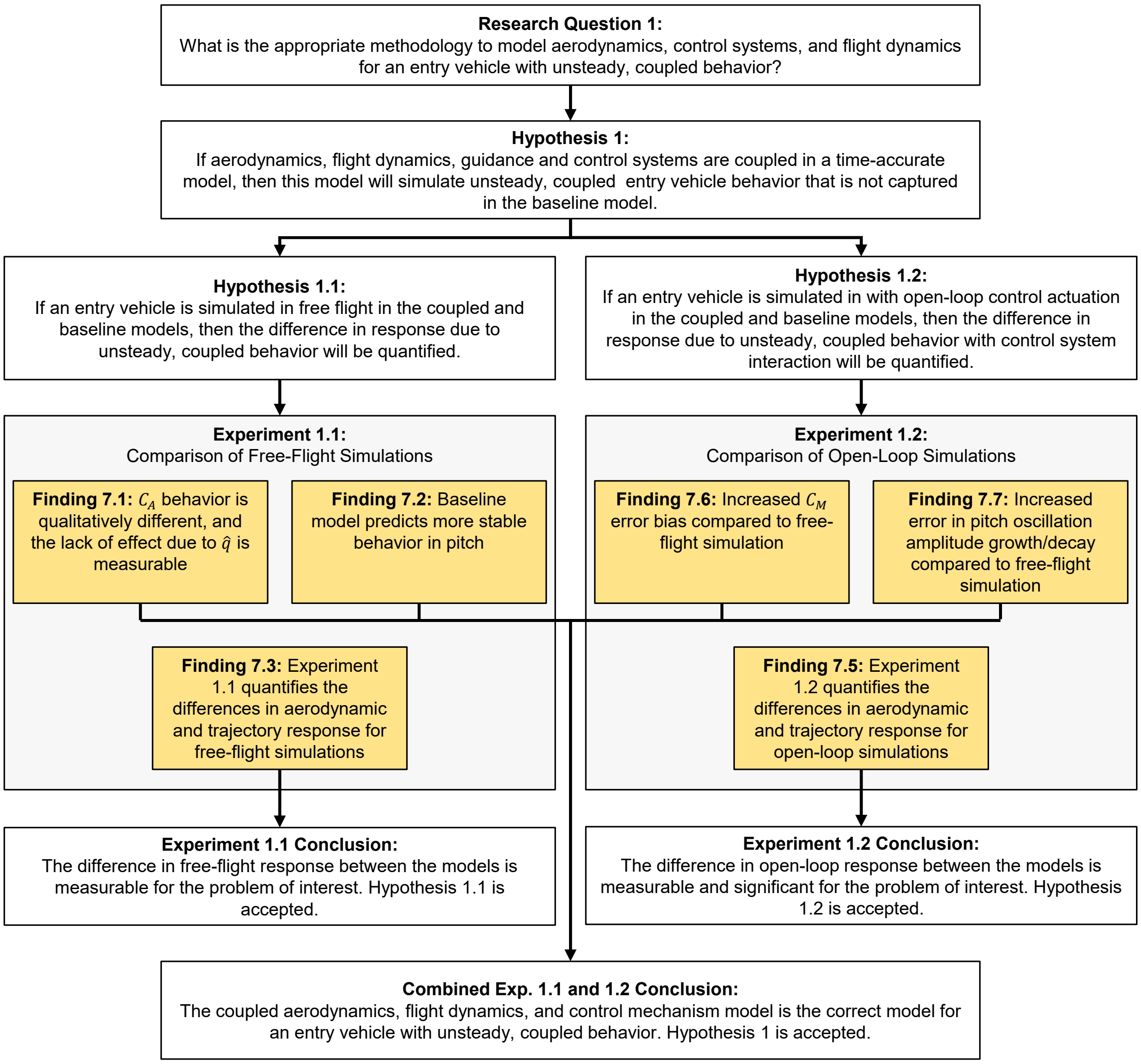

Research Question 1: What is the appropriate methodology to model aerodynamics, control systems, and flight dynamics for an entry vehicle with unsteady, coupled behavior?
Hypothesis 1: If aerodynamics, flight dynamics, guidance and control systems are coupled in a time-accurate model, then this model will simulate unsteady, coupled entry vehicle behavior that is not captured in the baseline model.
Hypothesis 1.1: If an entry vehicle is simulated in free flight in the coupled and baseline models, then the difference in response due to unsteady, coupled behavior will be quantified.
Hypothesis 1.2: If an entry vehicle is simulated in with open-loop control actuation in the coupled and baseline models, then the difference in response due to unsteady, coupled behavior with control system interaction will be quantified.
Experiment 1.1: Comparison of Free-Flight Simulations
Finding 7.1: C_A behavior is qualitatively different, and the lack of effect due to $\hat{q}$ is measurable
Finding 7.2: Baseline model predicts more stable behavior in pitch
Finding 7.3: Experiment 1.1 quantifies the differences in aerodynamic and trajectory response for free-flight simulations
Experiment 1.2: Comparison of Open-Loop Simulations
Finding 7.6: Increased C_M error bias compared to free-flight simulation
Finding 7.7: Increased error in pitch oscillation amplitude growth/decay compared to free-flight simulation
Finding 7.5: Experiment 1.2 quantifies the differences in aerodynamic and trajectory response for open-loop simulations
Experiment 1.1 Conclusion: The difference in free-flight response between the models is measurable for the problem of interest. Hypothesis 1.1 is accepted.
Experiment 1.2 Conclusion: The difference in open-loop response between the models is measurable and significant for the problem of interest. Hypothesis 1.2 is accepted.
Combined Exp. 1.1 and 1.2 Conclusion: The coupled aerodynamics, flight dynamics, and control mechanism model is the correct model for an entry vehicle with unsteady, coupled behavior. Hypothesis 1 is accepted.

Figure 7.35: Summary of investigation into Research Question 1

CHAPTER 8
EVALUATION OF AERODYNAMIC SURROGATE MODEL ALTERNATIVES (RESEARCH QUESTION 2.1)

This chapter documents the investigation of Research Question 2.1 and its associated Hypothesis 2.1, both of which are restated below:

Research Question 2.1

How can an aerodynamic surrogate model be constructed using time-series data from the coupled model as it is generated during the tuning process?

Hypothesis 2.1: If a neural-network-based surrogate model is fitted with time-series trajectory and aerodynamic data from the coupled simulation, then it will have improved error convergence to the coupled model over a baseline database.

Experiment 1 demonstrated the deficiencies of the baseline aerodynamic database for modeling an entry vehicle with unsteady, coupled behavior. Therefore, a new surrogate model must be identified for use in the multi-fidelity controller tuning methodology. An aerodynamic surrogate model must be identified that satisfies the requirements developed through this Research Question (i.e., the characteristics of the model required for use in the multi-fidelity controller tuning process). These requirements are that the surrogate model must:

1. Accept as training data the dense, continuous time-series trajectory and aerodynamic response generated by the coupled model.
2. Incorporate and be re-trained/re-fitted with additional time-series data sets that are

generated as the multi-fidelity controller tuning process iterates.

3. Converge in its aerodynamic and trajectory response prediction to those of the coupled model as the training set size increases, in order to satisfy the assumptions made about the quasi-adaptive nature of the controller tuning process.

The success criteria of the first two requirements are binary, and can be evaluated through literature review. Therefore, Experiment 2 will serve as verification for these requirements. The objective of Experiment 2 is to identify a surrogate model that satisfies the last requirement, and assuming that multiple alternatives satisfy the requirement, to determine which surrogate alternative has the best convergence.

The experimental procedure is to simulate the training scheme that the surrogate model will undergo when used as part of the multi-fidelity controller tuning process. The data generated for the experiment also allows for investigation into the relationship between the convergence of instantaneous coefficient prediction and the convergence of the trajectory prediction. Section 8.1 enumerates the surrogate model alternatives considered and details their construction: a quasi-linear parametric model, a feed-forward neural network (FFNN), and a time-dependent neural network (TDNN). Section 8.2 documents the experiment, its results, and its findings. Section 8.3 revisits the analysis performed in Experiment 1 using the best surrogate model alternative identified through Experiment 2. Finally conclusions are made in Section 8.4.

8.1 Model Alternatives

Construction of the other model alternatives considered is described in the subsections below. In general, each model is designed to produce coefficients in the Missile frame, with POST2 automatically converting the results into the Body frame. This is valid since the vehicle is axisymmetric.

8.1.1 Database

The database model to be used as the basis for comparison has the form described in Section 7.2: the static coefficients C_A, C_N, and C_M are found using time-accurate fixed-attitude CFD solutions, and the dynamic coefficients are found by fitting the aerodynamic moment history from free-flight simulations.

8.1.2 Parametric Model

A quasi-linear parametric model is constructed for the coefficients. This type of model was selected as an alternative due to its common use in aerodynamics for other vehicles and for other uses, as identified in Section 2.4.3. The model form for each of the coefficients is based on examples from entry vehicle literature [97, 96]:

$$\tilde{C}_A = (c_0 + c_1\Delta M)\cos\alpha_T + (c_3 + c_4\Delta M)\ \sin^2\alpha_T + (c_6 + c_7\Delta M)\ \hat{\dot{\alpha}}_T \tag{8.1}$$

$$\tilde{C}_N = (c_0 + c_1\Delta M)\sin\alpha_T + (c_2 + c_3\Delta M)\sin^3\alpha_T + (c_4 + c_5\Delta M)\ \hat{\dot{\alpha}}_T \tag{8.2}$$

$$\begin{aligned}\tilde{C}_M = (c_0 + c_1\Delta M)\sin\alpha_T + (c_2 + c_3\Delta M)\sin^3\alpha_T \\ + (c_4 + c_5\Delta M)\ \hat{\dot{\alpha}}_T + (c_6 + c_7\Delta M)\ |\sin\alpha_T|\ \hat{\dot{\alpha}}_T\end{aligned} \tag{8.3}$$

where ΔM is the difference between Mach number and a chosen reference Mach number M_{ref}, and $\hat{q}$ is the nondimensionalized pitch rate. By including $\hat{\dot{\alpha}}_T$ as an input variable, each of these three equations effectively includes both the static and dynamic components of the aerodynamic loads. Note that Equation 8.3 is identical to Equation 7.2, the moment coefficient fit used for the baseline database. Based on the findings in Section 7.3 that the omission of pitching terms in the baseline axial coefficient resulted in measurable errors to the coupled model, $\hat{\dot{\alpha}}$ terms were included in Equation 8.1.

Nonlinearity is introduced by allowing the constants to effectively vary with Mach number. Given the small range of Mach numbers, a linear variation across Mach was judged to be sufficient for each of the equations.

8.1.3 Feed-Forward Neural Network

The next model alternative considered will use single-layer feed-forward neural networks (FFNN) to predict the net aerodynamic coefficients. The model will use the architecture shown in Figure 8.1, with one hidden layer. Each hidden layer node will use a perceptron neuron with a sigmoid activation function, a well-understood node design [151]. The output layer node uses a linear activation function. Individual networks are trained for each coefficient C_A, C_N, and C_M. Each network uses the same inputs: M, α_T, and $\hat{\dot{\alpha}}_T$.

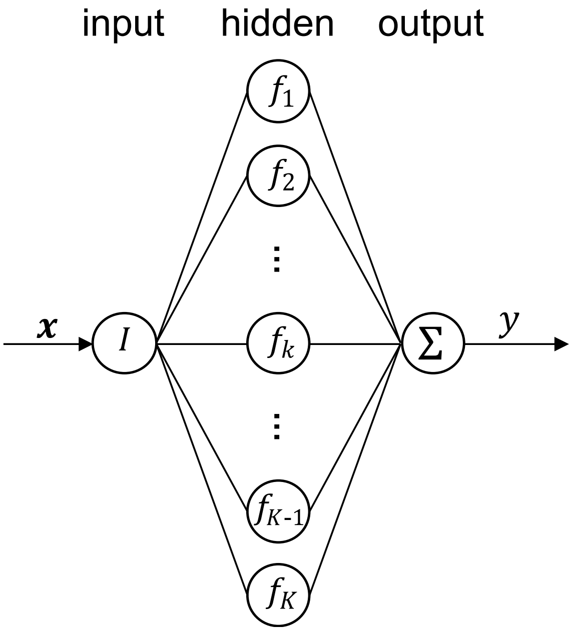


Figure 8.1: FFNN architecture

A Levenberg Marquardt training algorithm [231] was used to minimize the mean squared error to the response. The data was randomly split into training, validation, and testing sets of 70%, 15%, and 15%, respectively. The MATLAB® Deep Learning Toolbox™ was used to train each of the networks [232].

For each coefficient, an exploration of the network design was performed in order to identify the size of network that yielded the smallest MRE when trained with the whole set of coupled free-flight simulations. The number of hidden layer nodes was the only hyper-parameter varied for the FFNNs. Networks of varying sizes were repeatedly trained using all the data sets in order to identify when the model would start suffering from over-fitting. The exploration is shown for the C_N surrogate in Figure 8.2. The error bars represent the maximum and minimum error from the replications at each size. does not start to suffer from overfitting until reaching about 22 nodes. The size of each network was found by minimizing the average RMS error for the validation data, resulting in the designs listed in Table 8.1. The total number of unknowns for each net includes the weights and biases for each node that must be fitted. For C_N and C_M, this gives a ratio of about 163:1 between the total number of training points and the number of unknowns, far above the rule-of-thumb ratio of 10:1.

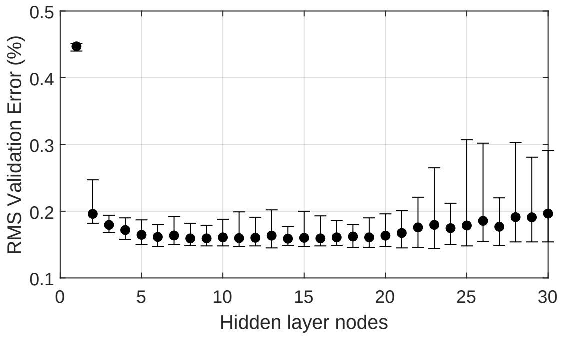


Figure 8.2: C_N FFNN size study

8.1.4 Time-Delay Neural Network

The third surrogate model alternative considered is a time-delay neural network (TDNN). The architecture will use one hidden layer as shown in Figure 8.3. As with the FFNN, the

Table 8.1: FFNN hyperparameters

Coefficient	Hidden layer nodes	Total unknowns
C_A	8	41
C_N	16	73
C_M	16	73

hidden layer nodes of the TDNN will use perceptron neurons with the sigmoid activation function, and the output layer node will use a linear activation function. Each of the $1..j$ delay nodes is given a delay of $j \cdot dt$. Again, individual networks are trained for each coefficient C_A, C_N, and C_M, each with M, α_T, and $\hat{\dot{\alpha}}_T$ as inputs.

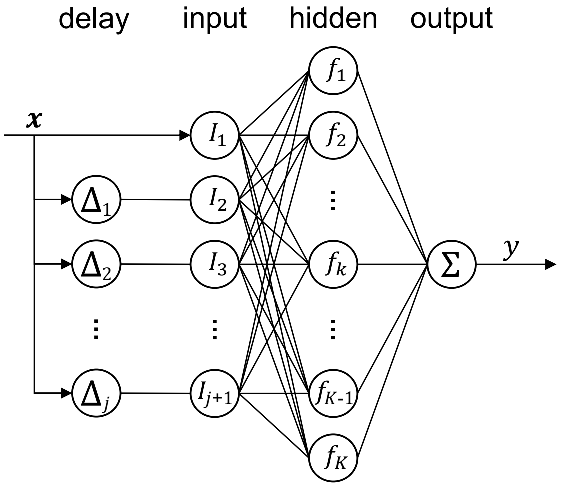


Figure 8.3: RNN architecture

Training of the TDNN was performed with the same tools, process, and settings as described for the FFNN. A design exploration of the number of hidden layer nodes and the number of delay nodes was made to determine the optimal network design. The hyperparameters identified are listed in Table 8.2.

Table 8.2: Time-delay neural network design

Response	Delay nodes	Hidden nodes	Total unknowns
C_A	6	6	139
C_N	6	12	277
C_M	6	12	277

Although the size of the hidden layer is decreased for each coefficient compared to the FFNN design, the total size of the networks increases due to the connectivity from the delay nodes. However, the C_N and C_M networks still yield a training data to unknown ratio of about 43:1.

The delays can recover the relationship dM/dt, which, assuming negligible changes in atmospheric properties and small oscillation amplitudes, is proportional to the acceleration dv/dt. Since the acceleration of the vehicle is largely driven by the axial force; in effect, this creates a feedback loop within the C_A net. The same mechanism exists with the C_M net and the $\hat{\dot{\alpha}}_T$ input.

Execution of the TDNN requires initialization with the time history of the inputs in order to populate the delay nodes on the first function call. This is achieved by using the vehicle motion state history from the forced-motion startup runs described in Section 7.2.2 for α and $\hat{\dot{\alpha}}_T$. However, since the Mach number is held constant during the forced-motion startup, the Mach number was instead back-propagated from the first free-flight time step, assuming a constant deceleration as measured from finite differencing. Figure 8.4 demonstrates the difference between this back-propagation and the actual trajectory flown in the coupled model.

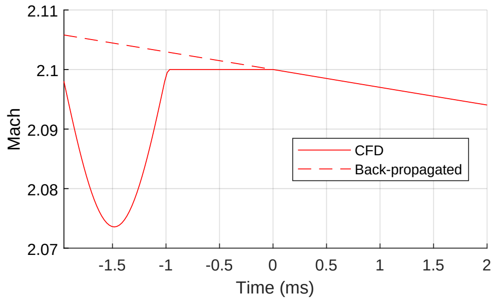


Figure 8.4: Back-propagation of Mach number for TDNN initialization

8.1.5 Other Model Alternatives

In addition to the surrogate model alternatives discussed in the sections above, exploratory investigation was performed on a number of other alternatives. These activities are briefly discussed here.

In addition to the FFNN and TDNN architectures, a Nonlinear AutoRegressive model with eXogenous variables (NARX) architecture was also considered. This adds connectivity for the recurrence of the output as shown in Figure 8.5. This architecture was considered based on its use for aerodynamic modeling as identified in the literature review in Section 2.4.3. The explicit feedback loop can exacerbate the problems discussed for the TDNN. In practice, test models constructed using this architecture were too unstable, quickly deviating into regions of extrapolation resulting in non-physical behavior. Exploration was made into reducing the number of delay nodes, and independently changing the number of input and response delay nodes, with little effect unless the number was reduced to zero, effectively defaulting the model to a TDNN or FFNN. A complete set of results was not able to be obtained for this model architecture. A more intelligent utilization of this model might combine it with another model for the lower-frequency response, with the TDNN providing bounded values for the higher-frequency response.

FFNN and TDNN models were also constructed for the coefficients directly in the Body frame, using M, α, and $\hat{q}$ as inputs. Since the vehicle is symmetric, the models for $\tilde{C}_Y$ and $\tilde{C}_{LN}$ could be assumed to take the same form as and use the same magnitude of coefficients

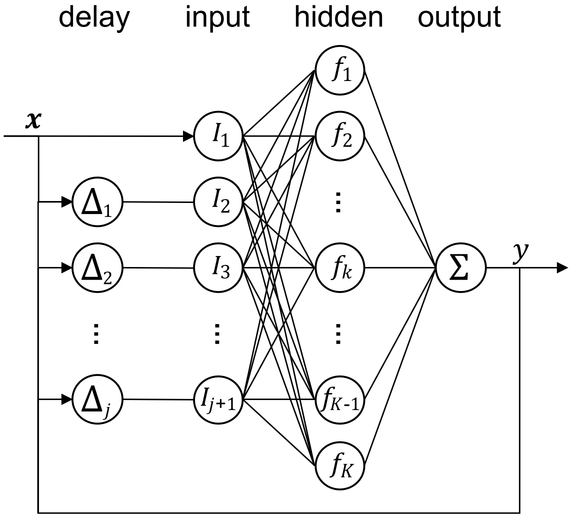


Figure 8.5: NARX architecture

as $\tilde{C}_N$ and $\tilde{C}_M$, respectively, assuming small α and β.

While the quasi-linear model enforces symmetry about $\alpha = 0$, the body-frame FFNN and TDNN models do not. This means that negative values of α will uniquely affect the training. The distribution of design points in the input dimensions, shown in Figure 7.6 for α_T, is actually much more sparse for α, forcing the neural network models to rely more heavily on extrapolation. The results for both of these models were similar but inferior to the FFNN and TDNN models that will be shown in Sections 8.2.2 and 8.2.3. Therefore, these models were not considered.

8.2 Simulated Training of Aerodynamic Surrogate Models (Experiment 2)

8.2.1 Experimental Procedure

The procedure for Experiment 2 is to simulate the training of each surrogate model in turn through the use of a common set (and order of inclusion) of training data. A series of eight free-flight simulations are run in the coupled model to serve as the training data. These are the same runs used to find the baseline database C_{M_q} in Section 7.2.2. The absence of closed- or open-loop control mechanism actuation in the training data is acceptable due to the design of the problem of interest. Since the IMMA does not interact with the fluid flow, its actuation does not result in any new flow physics being captured.

Given the small number of training sets, a leave-one-out cross-validation is performed for each of the surrogate architectures [233]. A randomized order of the training sets is chosen and maintained for all the surrogate models.

For each of the $i = 1..8$ data sets:

1. Run i is set aside to serve as validation data[1], while the remaining seven runs are used for the sequential training of the surrogate model. The order of these runs is the same as for the outer loop (except for the removal of validation run i).
2. For $j = 1..7$ iterations:
    (a) Runs $1..j$ of the sequential training set are used to train the aerodynamic surrogate. Each of the models is trained directly using the aerodynamic data from the coupled simulations, as opposed to the parameter identification method of fitting simulated trajectories directly to the trajectory measurements.

    (b) Metrics based on the aerodynamic coefficient error are calculated. The MFE of the surrogate is calculated for the same runs used for its training, and the coefficient MRE is calculated for the validation run i.

    (c) The surrogate model is substituted into the low-fidelity model and used to simulate all eight coupled runs.

    (d) Error metrics based on the trajectory components are calculated. The MFE is calculated for the simulations of the sequential training runs, and the MRE is calculated for the simulation of validation run i.

As a result of the above procedure, the surrogate models are trained with an average of 2,100 time steps added for each simulated iteration of the controller tuning methodology.

[1] Note that this validation data is not the same as the validation and testing sets used for the neural network training - this validation set is entirely excluded from the training process.

After the runs are completed the k-fold metrics are calculated using the formula below, where the k-fold metric for the jth iteration is:

$$f_j = \frac{1}{8} \sum_{i=1}^{8} f_i \tag{8.4}$$

Comparison of the convergence of the surrogate model alternatives is performed by examining several metrics constructed from the MRE to the validation data. These are the same metrics used for Experiment 1.1, documented in Section 7.3.1.

Equivalent Training Method for the Baseline Aerodynamic Database

The performance of the baseline aerodynamic database with increasing training cost can be measured using a slightly modified procedure. The sequential training (the inner loop) provides $4j$ static coefficients to populate the database[2]. The j coupled simulation runs are only used to identify the pitch damping coefficient, as shown in Section 7.2.2. For example, the database on the first iteration would contain data at four randomly-selected points, with the damping coefficient calculated using parameter identification on a single time-series run. On the second iteration, the database would contain a total of eight points, with the damping coefficient calculated from two runs, and so forth. As evaluated in Table 8.3, even for the first iteration, the density of points in the baseline database is comparable to the state of the art for entry vehicles.

[2] As with the database constructed in Section 7.2, the normal and moment coefficients information is stored as C_{N_α} and C_{M_α} so the model does not fail to extrapolate the restoring nature of the pitch moment for the initial iterations

Table 8.3: Average density of static aerodynamic database in the vicinity of Mach 2

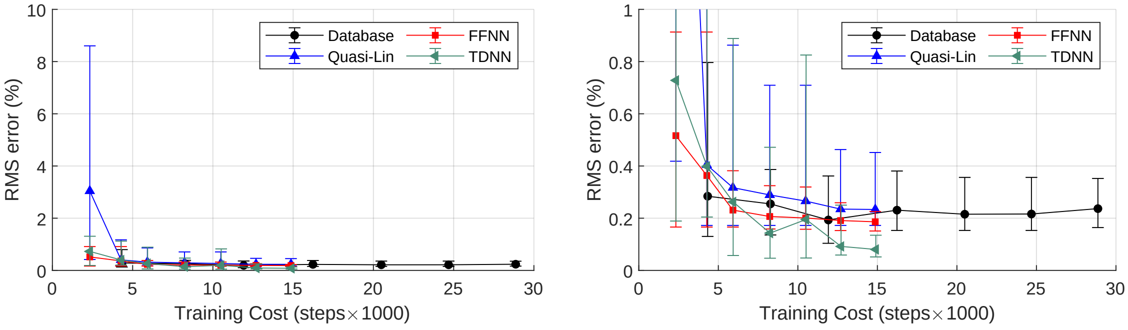
Vehicle	M	α_T (1/deg)
Phoenix [56]	1.000	0.200
MSL [54]	1.087	0.208
Baseline (Exp. 1)	20.00	0.718
Baseline (min. for Exp. 2)	6.667	0.167

Since the static database points were generated with the same CFD flow solver at the same settings, the the cost per time step is substantially similar when run on the same computing hardware. This allows for a comparison between the cost of the baseline database and the surrogate model alternatives[3].

8.2.2 Comparison of Aerodynamic Prediction Convergence

Axial Force Coefficient

Figure 8.6a plots the convergence of the normalized RMS error to the validation C_A data, with a detailed view of the same data plotted in Figure 8.6b. The points represent the k-fold metric, with the error bars representing the maximum and minimum values from the cross-validation.



(a)

(b) Detailed view

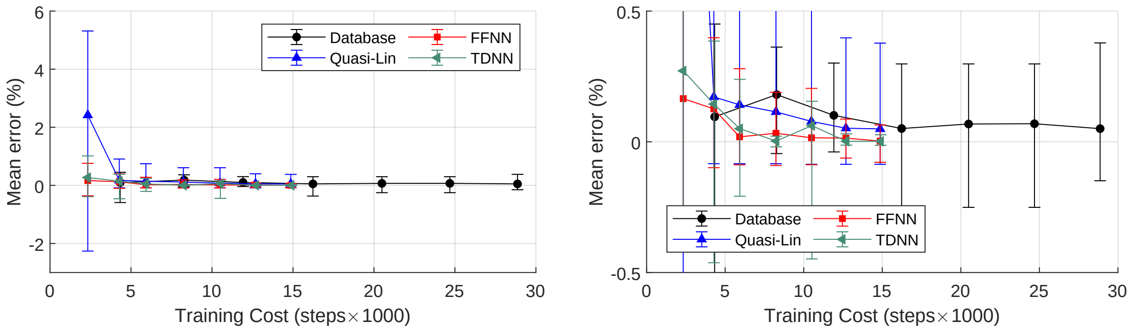
Figure 8.6: Convergence of axial force coefficient RMS error

[3] A direct comparison of computing time is precluded since runs were executed on different hardware, different numbers of processors, and different memory allocations.

The database has relatively little change in performance with increasing training set size, with the k-fold error reaching a minimum at iteration 3. However, the database does provide the best result for the first iteration (with minimum training cost). Note that each iteration of the database requires around two times the training data for the same iteration of the other surrogate alternatives. The quasi-linear model has the worst initial performance, but converges to about the same error as the database. Both the database and the quasi-linear model do not account for dynamic components of C_A, which could be a contributing factor to their similar final performance. Both neural net models show good convergence, with results that exceed the other two models. Since these models have no specified functional form, they are capable of capture the dynamic contribution to C_A. The TDNN achieves the best performance overall, with about 50% of the k-fold error of the FFNN. The TDNN has an increase in error on the 5th iteration, possibly as a result of the stochastic nature of the training process. The uncertainty of the FFNN and the TDNN on the final iteration are also both smaller than the uncertainty for the database and the quasi-linear model.

Figure 8.7a plots the convergence of the mean C_A error, with a detailed view of the same data plotted in Figure 8.7b[4]. The figure shows that the baseline database has a measurable bias that persists across each iteration with only a slight decrease. The quasi-linear model has the worst initial performance, but converges to about the same level as the database. Both the FFNN and TDNN show excellent convergence toward a mean of zero, and with much smaller uncertainty than the baseline or quasi-linear models.

[4] The values of the points and error bar limits plotted for all of the mean error plots in this section are multiplied by the sign of the mean value, so that the mean value is always plotted in the positive y-axis. This simplifies comparison of the magnitude and k-fold variation from iteration to iteration.



(a)

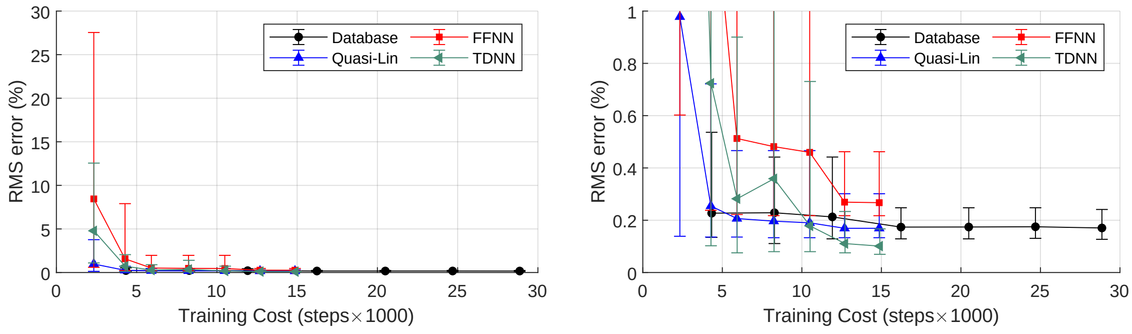
(b) Detailed view

Figure 8.7: Convergence of axial force coefficient mean error

In all cases the models have converging RMS and mean error with increasing training set size. On the final iteration, the neural network models provide the lowest RMS error with very little bias in the mean error. They also have substantially smaller k-fold variation than the quasi-linear and database models. The TDNN provides the best overall result.

Normal Force Coefficient

Figure 8.8a plots the convergence of the RMS C_N error, with a detailed view of the same data plotted in Figure 8.8b. The database provides the best initial result, with only a slight decrease with increasing training set size. The quasi-linear model converges to about the same performance as the database. Both the FFNN and the TDNN have substantial error on the first iteration, but both rapidly converge. The FFNN has a worse final performance than the database, but the TDNN yields the best result for the final iteration. As with C_A RMS results, the TDNN also has the smallest uncertainty bounds, but in this case the FFNN has increased uncertainty.

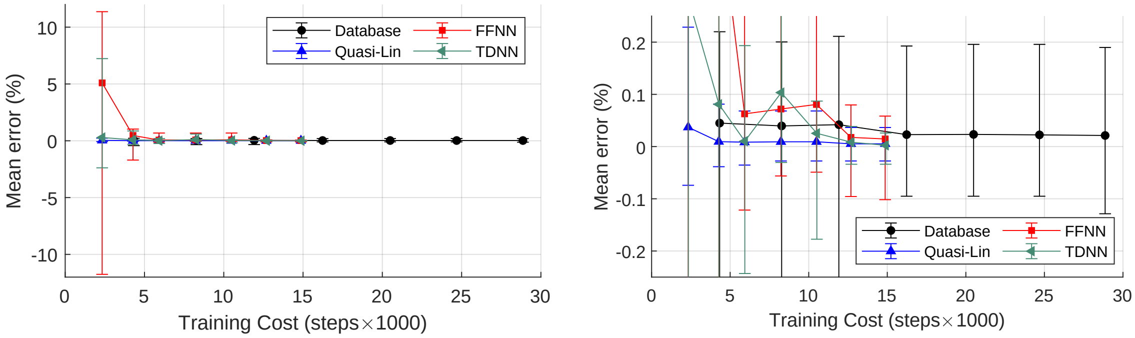


(a)

(b) Detailed view

Figure 8.8: Convergence of normal force coefficient RMS error

Figure 8.9a plots the convergence of the mean C_N error, with a detailed view of the same data plotted in Figure 8.9b. The baseline database once again shows a bias that starts relatively small, but decreases only slightly with increasing training set size. It has the worst performance after its 3rd iteration, and the k-fold variation also remains larger than that of the other models. The neural net models have the largest initial bias, but both quickly converge to below the database model. The quasi-linear model has the lowest initial bias, and outperforms all other models except for the TDNN on the final iteration.



(a)

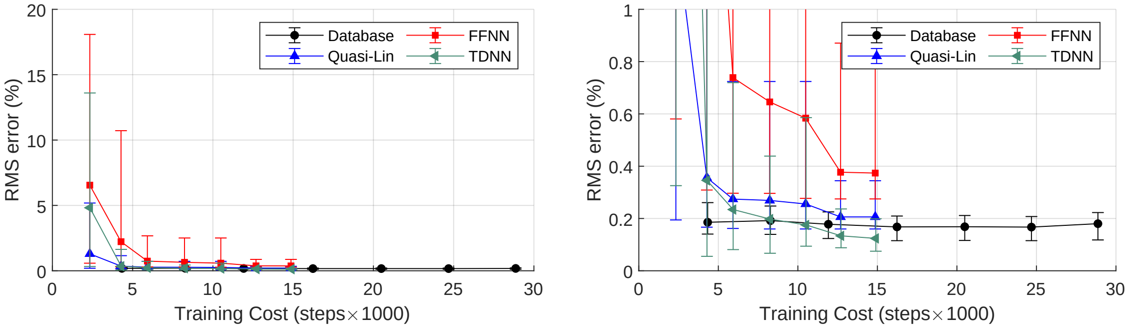
(b) Detailed view

Figure 8.9: Convergence of normal force coefficient mean error

Each C_N surrogate model converges in its error with increasing training set size. The TDNN again provides the best overall result. Unlike with C_A the two neural net models do not have a similar level of performance; the FFNN provides worse performance than the quasi-linear model.

Pitch Moment Coefficient

Figure 8.10a plots the convergence of the RMS C_M error, with a detailed view of the same data plotted in Figure 8.10b. Again the database provides the best initial result, but does not significantly improve with added training data. In fact, the mean result on the final iteration is only slightly improved from the initial iteration, with the minimum at iteration 6. Both the FFNN and the quasi-linear model converge, but do not achieve results as good as the database. The variability of the FFNN is also larger than for the other models. Again, the TDNN achieves the best final k-fold performance with the smallest variability.

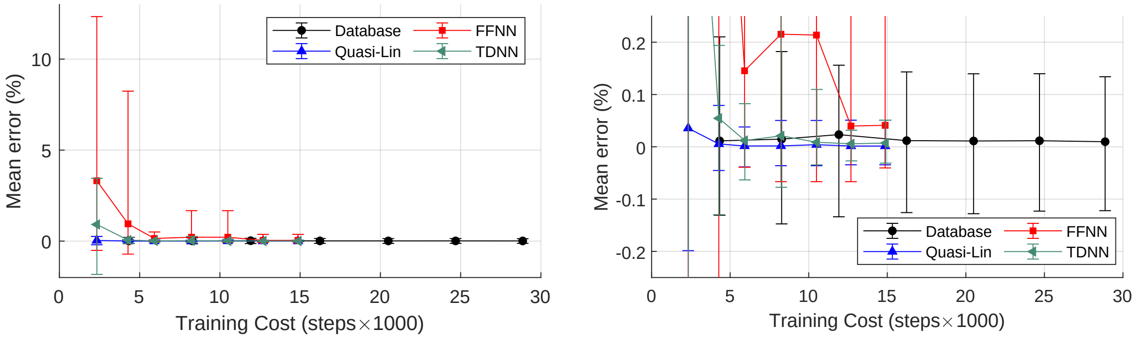


(a)

(b) Detailed view

Figure 8.10: Convergence of pitch moment coefficient RMS error

Figure 8.11a plots the convergence of the mean C_M error, with a detailed view of the same data plotted in Figure 8.11b. For this metric, the database slightly diverges. The FFNN has the worst performance across all iterations, with relatively substantial k-fold variation. The TDNN and quasi-linear model have very small bias, with the quasi-linear model providing the best performance across all iterations.



(a)

(b) Detailed view

Figure 8.11: Convergence of pitch moment coefficient mean error

Once again, the FFNN performs measurably worse than the TDNN for this coefficient. The TDNN performs the best for the RMS error, and performs relatively equally to the quasi-linear model in terms of the mean error.

Findings

For all of the models, the RMS and mean errors in the coefficients trend toward zero with increasing iteration, so they can be considered to converge:

Finding 8.1: All four of the evaluated aerodynamic surrogate models have converging coefficient error with respect to the coupled simulation response as training set size increases.

The results from the final iteration of the training are presented for all four models in Table 8.4. Overall, each of the models appears satisfactory from an engineering perspective. The k-fold mean and RMS errors are all less than 1% on the final iteration. All of the surrogate models achieve a maximum final error of less than 0.5% for C_A and less than 1% for C_M.

Table 8.4: Iteration 7 k-fold aerodynamic coefficient error metrics

	Axial Force		Normal Force		Pitch Moment	
Model	RMS (%)	Mean (%)	RMS (%)	Mean (%)	RMS (%)	Mean (%)
Baseline	0.237	0.050	0.170	0.021	0.180	0.010
Quasi-linear	0.233	0.049	0.169	0.005	0.206	0.001
ANN	0.186	0.003	0.267	0.015	0.374	0.041
TDNN	0.080	0.002	0.101	0.002	0.124	0.007

The database and quasi-linear model generally perform better than the neural net models in the first few iterations (i.e., with a more limited training budget). The smaller the training set, the more likely it is that the model is relying on extrapolation to predict the validation response. Ae expected, the alternatives with prescribed functional form result in less performance degradation in extrapolation. However, after a certain number of iterations, the neural network models begin to outperform the other two. In addition, neither of the neural net models, nor to some extent the quasi-linear model, appear to have reached the asymptotic behavior which would indicate that they are fully converged.

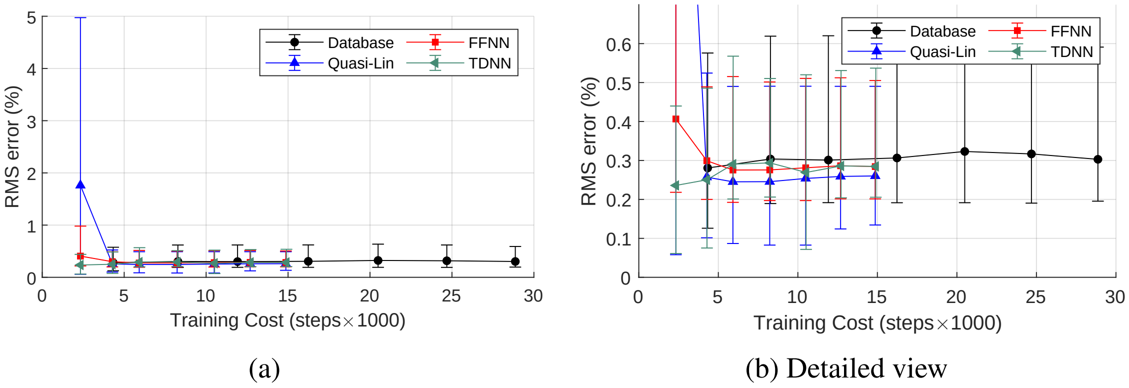
Finding 8.2: With a sufficient training set size, the neural-network-based aerodynamic surrogate models generally have better convergence than the models with an assumed functional form.

The TDNN has the best result in all metrics except for the mean pitch moment error.

8.2.3 Comparison of the Trajectory Prediction Convergence

Evaluating the performance of the aerodynamic surrogates when implemented into the flight simulation is important because it represents the models' intended use in the multifidelity controller tuning methodology.

Figure 8.12a plots the convergence of the x-axis displacement, with a detailed view of the same data plotted in Figure 8.12b. Simulations with the database and the TDNN actually result in a slight decrease in performance with increasing training set size. After an initial convergence, both the quasi-linear and FFNN models also have increasing error with increasing training set size. The TDNN model actually yields the best overall result on the first iteration, however after the second iteration, the quasi-linear model yields the best result for a given training set size. All of the models also have a similar magnitude of variation after the initial few iterations.

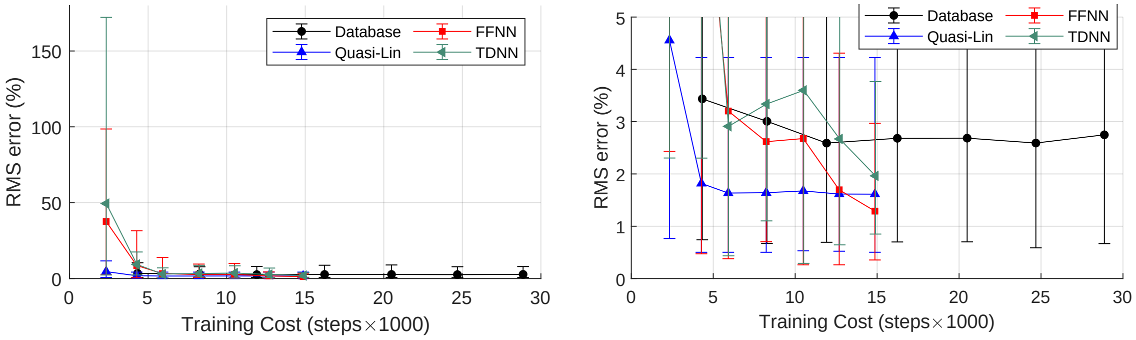


(a)

(b) Detailed view

Figure 8.12: Convergence of the x-axis displacement RMS error

Figure 8.13a plots the convergence of the z-axis displacement, with a detailed view of the same data plotted in Figure 8.13b. The quasi-linear model provides the best result for all steps except for the final iteration. While both the FFNN and TDNN have very high error on the initial iteration, these both quickly converge. The FFNN yields the best result on the final iteration, and the FFNN and TDNN still appear to be converging at a significant rate.

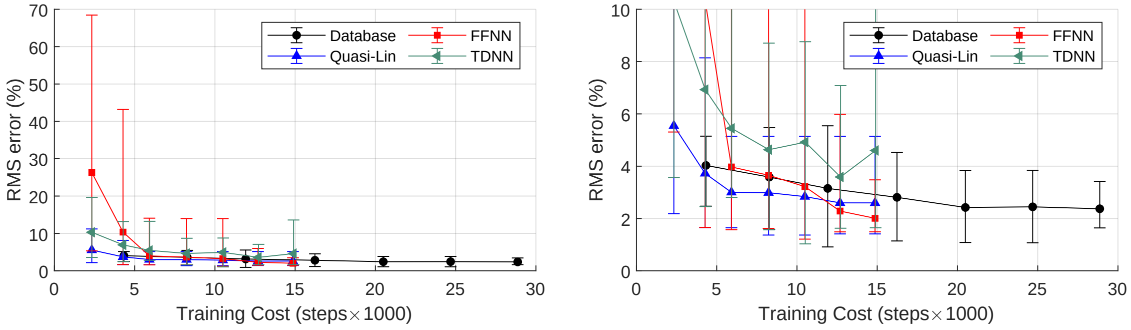


(a)

(b) Detailed view

Figure 8.13: Convergence of the z-axis displacement RMS error

Figure 8.14a plots the convergence of the pitch angle, with a detailed view of the same data plotted in Figure 8.14b. Each of the models converges with increasing training set size. The database converges at a consistent rate until the 5th iteration, where it appears to converge around 2.4% error. The quasi-linear and FFNN models converge at a better rate than the database for a given training set size, with the FFNN yielding the best overall result at around 2%. The TDNN performs worst of all the models after the 2nd iteration.



(a)

(b) Detailed view

Figure 8.14: Convergence of the pitch angle RMS error

Findings

All models have decreasing RMS error for the z-axis displacement and pitch angle responses. The x-axis decrement is the only component for which all of the models do not converge; the baseline and TDNN models show a slight increase in error. However even the worst result on the final iteration has an error of less than 0.4%, which is satisfactory for the problem of interest.

Finding 8.3: All four evaluated models have satisfactory or converging x-axis decrement error with respect to the coupled simulation responses. All four models have converging z-axis displacement and pitch angle error with respect to the coupled simulation responses.

The final trajectory component error metrics are shown for all four models in Table 8.5

Table 8.5: Iteration 7 k-fold trajectory component RMS error

Model	x_{dec} (%)	r_z (%)	θ (%)
Baseline	0.303	2.74	2.37
Quasi-linear	0.260	1.61	2.60
ANN	0.284	1.29	2.01
TDNN	0.285	1.96	4.60

In real-world terms, the differences between the models is only slight, as was found for the evaluation of the aerodynamic components.

In the z-axis displacement and pitch results, the trend observed for the aerodynamic components — that the database and quasi-linear models outperform the neural network models for small training set sizes — continues, albeit with the TDNN as an outlier (the trend in its convergence is similar, but it does not perform better than the quasi-linear model).

Finding 8.4: With sufficient training set size, the FFNN-based aerodynamic surrogate model general has better trajectory error convergence than the models with an assumed functional form.

The best performing models are generally the quasi-linear and FFNN models. For both

z and pitch, the FFNN performs best on the final iteration.

8.2.4 Investigating the Relationship between Instantaneous and Propagated Prediction Error

In Section 8.2.2 it was found that the TDNN generally yielded the best results for the aerodynamic components. However in Section 8.2.3 this model was not found to yield the best results for the trajectory components. Especially notable is the behavior in angular motion: the TDNN had the best RMS error and nearly equal mean error in C_M to the quasi-linear model, but these models performed differently in pitch prediction, where the TDNN was the worst model on the final iteration. Conversely, the FFNN, which had the worst performance in C_M, had the best performance in pitch prediction on the final iteration. While not strictly relevant to the testing of Hypothesis 2.1, a comparison of the results is performed in order to elucidate some relationship between the instantaneous and propagated prediction error.

The RMS, mean, and standard deviation of error in C_A, C_N, and C_M measured from each individual (not k-folded) model design, are considered as the predictors (9 in total) for the response of RMS error in pitch. The JMP® statistical sofware program was used to execute a bootstrap forest model that calculates the contribution of each predictor to the response [234]. The results are plotted for each model and for all models in Figure 8.15.

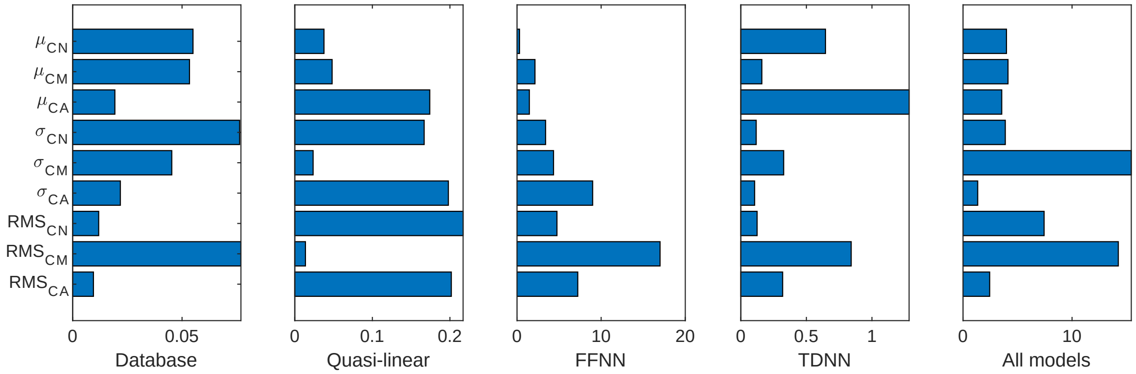


Figure 8.15: Percent contribution of predictors to RMS error in pitch response

The contributions for all models shows that after the standard deviation and RMS values of the C_M error, the largest contributor is the C_N RMS error. It is known that for a blunt body, the pitch damping response changes considerably when it is modeled with and without freedom of motion in the vertical direction [230]. The results could be an expression of this phenomenon.

8.2.5 Surrogate Model Selection for Use in the Multifidelity Methodology

No surrogate alternative provides an optimal performance across every metric evaluated. However, the results in pitch represent the most important characteristic of the response, since it directly effects guidance & control throughout EDL. The FFNN provides the best performance in pitch and z-axis displacement prediction along with an acceptable performance in x-axis displacement. Therefore, in the absence of other considerations, the FFNN would be the best surrogate (out of the alternatives evaluated) to use in the low-fidelity model for the multifidelity controller tuning methodology.

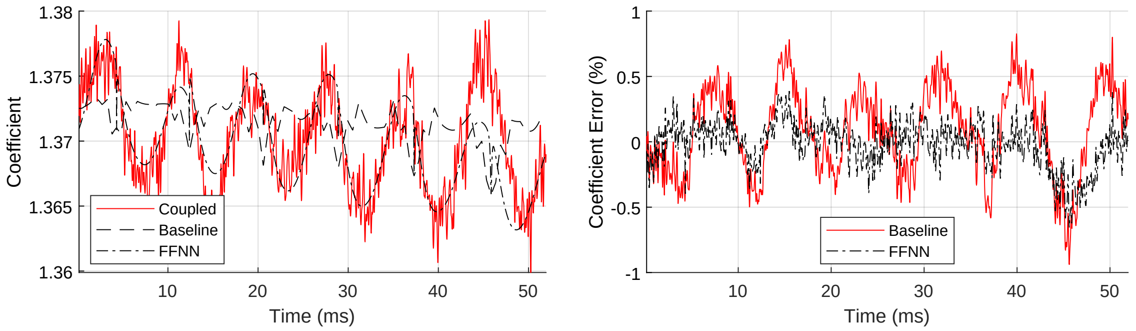
8.3 Comparison of FFNN and Database Surrogates

As a final verification of the suitability of the FFNN, the procedures from Experiments 1.1 and 1.2 are repeated, substituting the FFNN in place of the database.

8.3.1 Free-Flight

Aerodynamic Coefficients

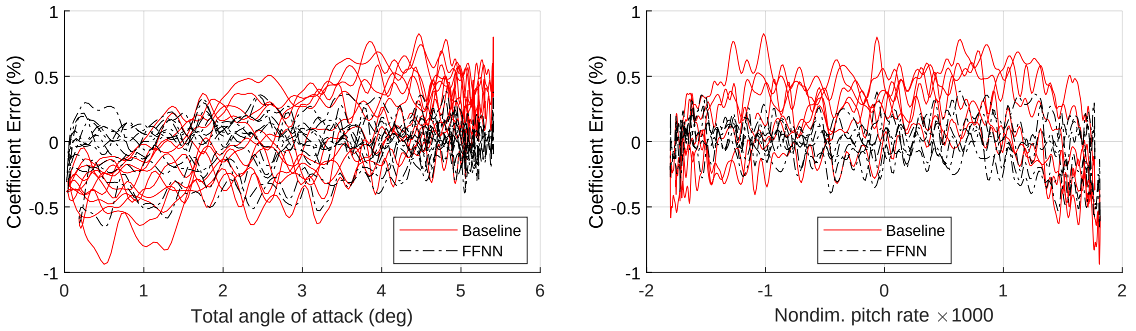
The axial force coefficient comparison plotted in Figure 8.16 shows that the FFNN model is much better than the baseline at predicting the oscillatory behavior of the coefficient, resulting in smaller peak error amplitudes. Examining the error as a function of α_T and $\hat{q}$ in Figure 8.17 shows that the noticeable trend present in the baseline model results is largely eliminated. However, the FFNN still does not predict the high-frequency behavior due to the unsteady wake.



(a) Coefficient

(b) Coefficient error

Figure 8.16: Free-flight axial force comparison

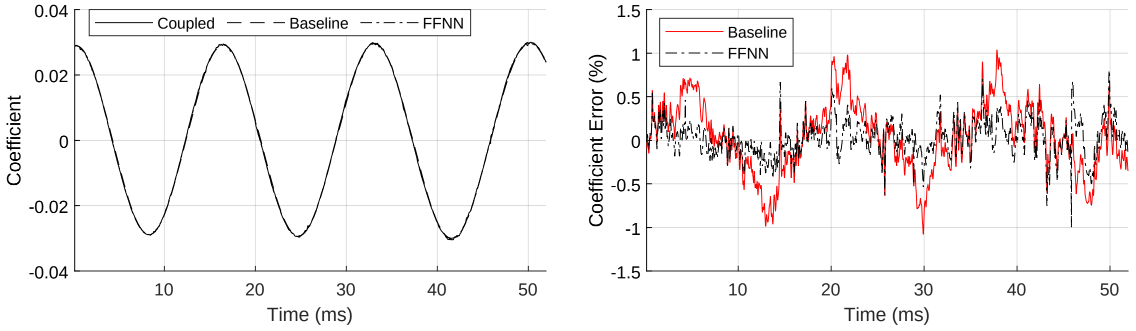


(a) As a function of α_T

(b) As a function of $\hat{q}$

Figure 8.17: Free-flight axial coefficient error

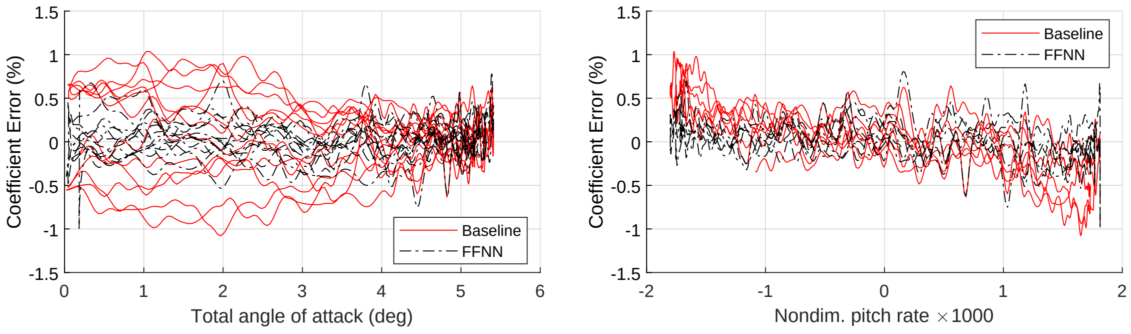
Figures 8.18b and 8.19 show that the FFNN normal force prediction is similarly improved over the baseline model.



(a) Normal coefficient

(b) Normal coefficient error

Figure 8.18: Free-flight normal force comparison

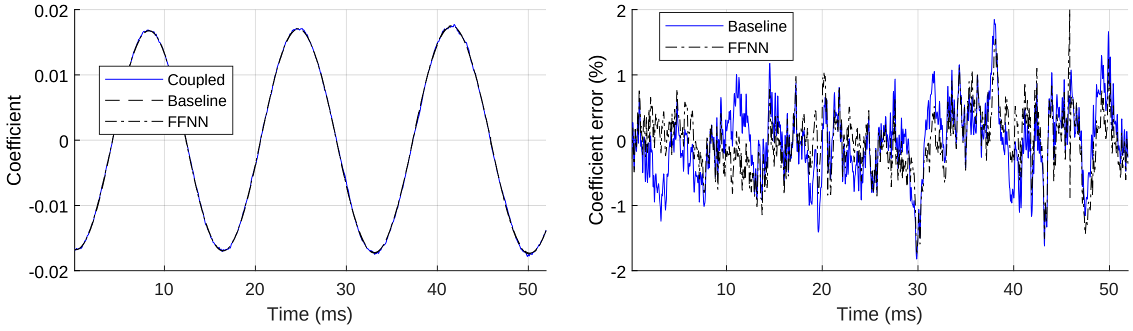


(a) As a function of α_T

(b) As a function of $\hat{q}$

Figure 8.19: Free-flight normal coefficient error

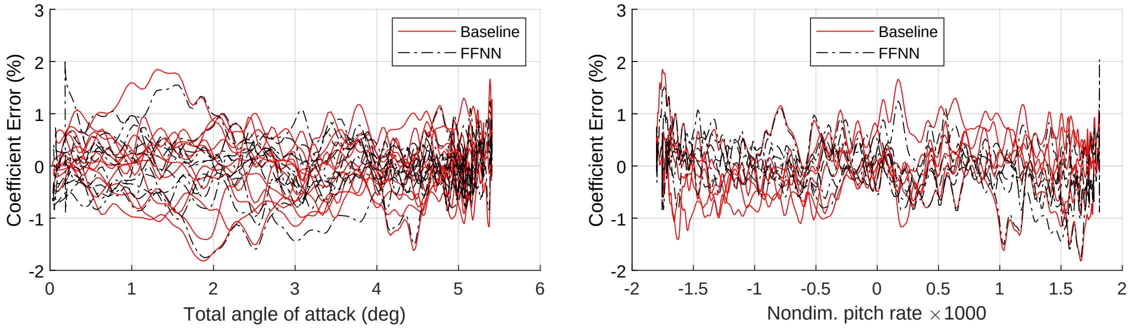
The pitch moment error plotted in Figure 8.20b shows that the FFNN still has similar performance to the baseline model. Figure 8.21a reveals that the behavior of the error as a function of α_T remains substantially the same, although the FFNN yields some improvement with respect to $\hat{q}$ as shown in Figure 8.21b.



(a) Moment coefficient

(b) Moment coefficient error

Figure 8.20: Free-flight aerodynamic moment comparison



(a) As a function of α_T

(b) As a function of $\hat{q}$

Figure 8.21: Free-flight pitch moment error

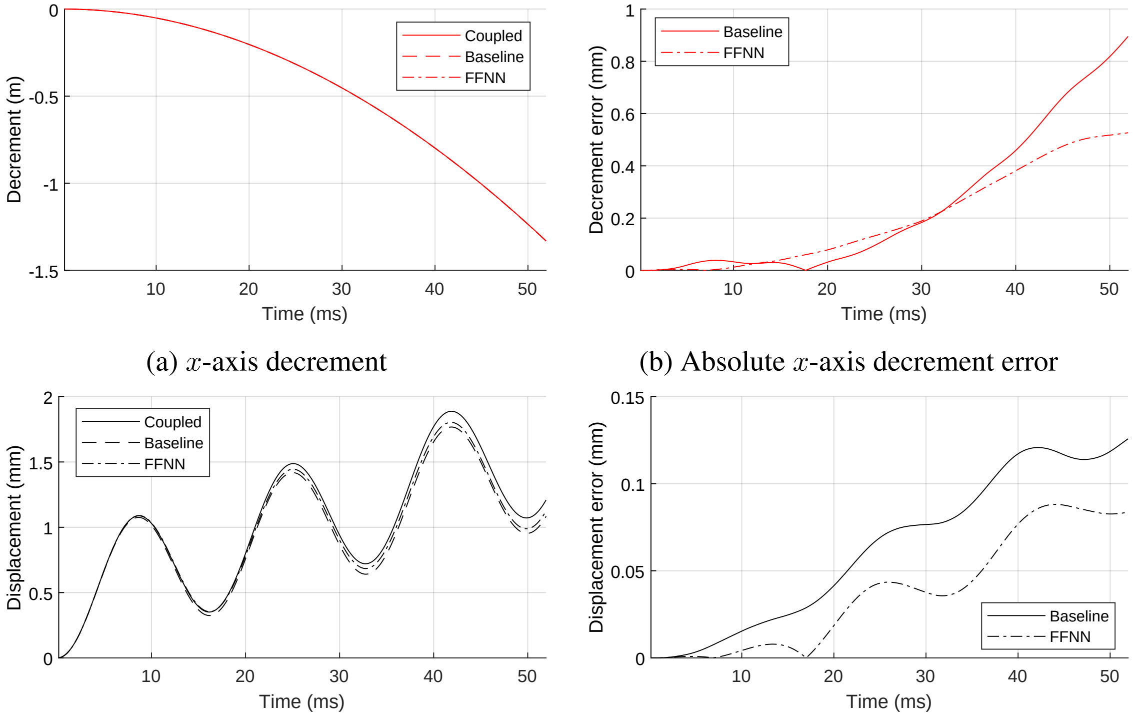
The overall error metrics for the aerodynamic predictions of Case 7 are listed in Table 8.6. All of the error metrics for both models are relatively small. The FFNN yields improvements in all metrics for C_A, and negligible improvements for C_N. The FFNN results for C_M are mixed, with a slightly worse mean error magnitude and a slightly better RMS error.

Table 8.6: Free-flight aerodynamic coefficient error comparison

	Axial Force		Normal Force		Pitch Moment	
Model	RMS (%)	Mean (%)	RMS (%)	Mean (%)	RMS (%)	Mean (%)
Baseline	0.350	0.088	2.142	0.083	0.556	-0.018
FFNN	0.186	-0.015	2.116	0.071	0.484	0.023

Trajectory Components

A comparison of the linear trajectory components is plotted in Figure 8.22. Both the x_{dec} and r_z errors have about the same magnitude and propagation over time, though the FFNN model yields slightly smaller final errors. The pitch responses plotted in Figure 8.23 show that the FFNN yields slightly improved results, most notably after 35ms.



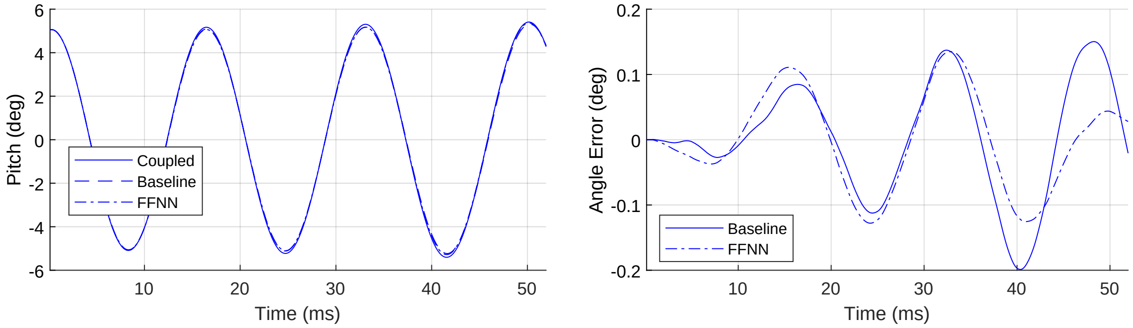
(a) x-axis decrement

(b) Absolute x-axis decrement error

(c) z-axis displacement

(d) Absolute z-axis displacement error

Figure 8.22: Free-flight translation comparison



(a) Pitch

(b) Pitch error

Figure 8.23: Free-flight orientation comparison

A comparison of the trajectory component overall error metrics in Table 8.7 confirms that the FFNN had slightly better results for all components, but both models have relatively small error.

Table 8.7: Free-flight trajectory component RMS error comparison

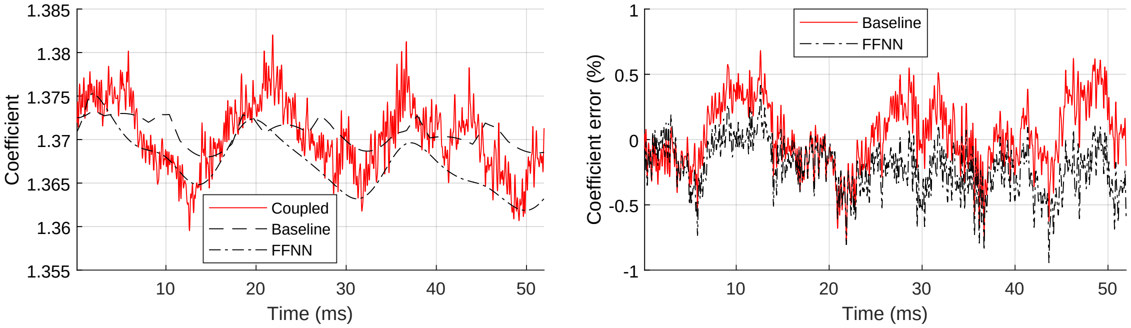
Model	x_{dec} (%)	r_z (%)	θ (%)
Baseline	0.028	4.05	0.802
FFNN	0.020	2.61	0.684

8.3.2 Open-Loop

While the comparison of the free-flight simulation in Section 8.3.1 provides a further investigation into results that were captured in Experiment 2, the FFNN model has not been trained using open-loop simulations.

Aerodynamic Coefficients

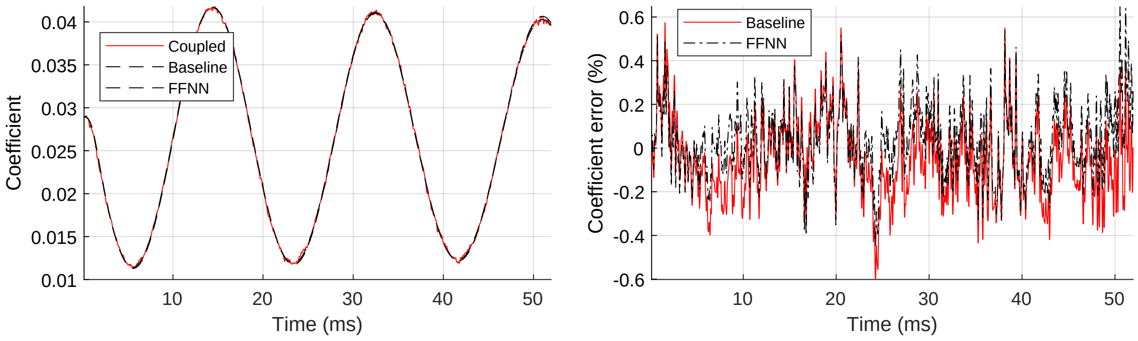
The plot of axial coefficient in Figure 8.24a shows that, as with the free-flight simulation, the open-loop FFNN response better predicts the oscillatory behavior of the coupled simulation. However, Figure 8.24b shows that the FFNN prediction has a noticeable bias that increases over time (i.e., with decreasing Mach number). Figure 8.25 shows that the normal force coefficient error is relatively similar.



(a) Coefficient

(b) Coefficient error

Figure 8.24: Open-loop axial force comparison

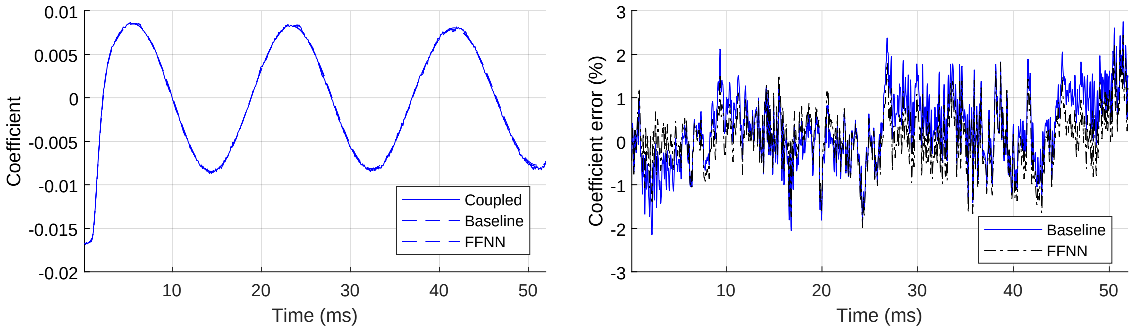


(a) Normal coefficient

(b) Normal coefficient error

Figure 8.25: Open-loop normal force comparison

The moment coefficient results in Figure 8.26 show a measurable reduction in the error for the FFNN model. The peak error amplitude is slightly reduced, but most notably the bias is significantly reduced. This is especially noticeable after about 25ms.



(a) Moment coefficient

(b) Moment coefficient error

Figure 8.26: Open-loop aerodynamic moment comparison

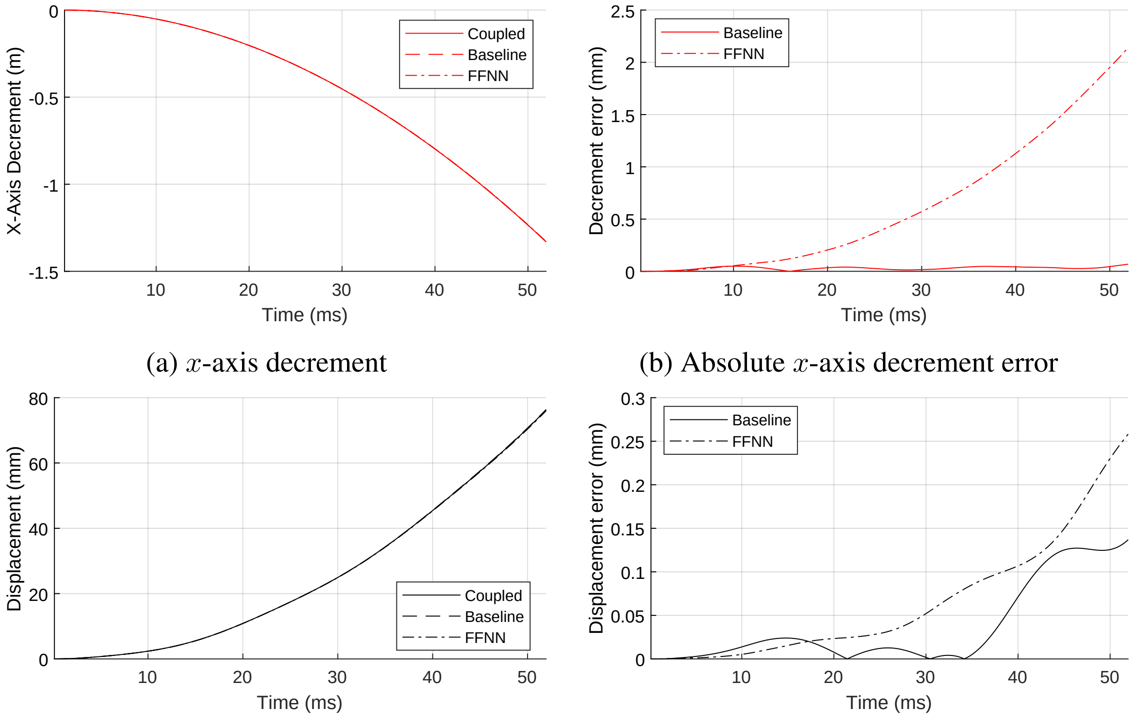
The overall error metrics listed in Table 8.8 show that all the C_A and C_N metrics are increased for the FFNN model, most significantly the mean C_A error. However, the FFNN does result in decreased C_M error metrics, with the RMS error reduced and the mean error almost entirely nullified.

Table 8.8: Open-loop aerodynamic coefficient error comparison

	Axial Force		Normal Force		Pitch Moment	
Model	RMS (%)	Mean (%)	RMS (%)	Mean (%)	RMS (%)	Mean (%)
Baseline	0.259	0.031	1.87	-0.012	0.860	0.310
FFNN	0.298	-0.217	1.87	0.082	0.600	-0.012

Trajectory Components

Figure 8.27 shows that while the x_{dec} and r_z responses are relatively similar between the two models, the FFNN model yielded increased error propagation over time as a result of the increased biases of the aerodynamic components.



(a) x-axis decrement

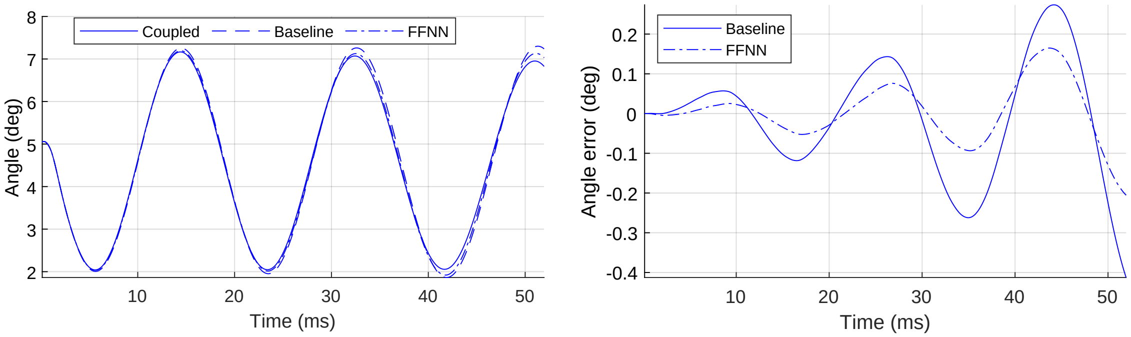
(b) Absolute x-axis decrement error

(c) z-axis displacement

(d) Absolute z-axis displacement error

Figure 8.27: Open-loop translation comparison

However, the largest change in the results from the two models is in the pitch angle, plotted in Figure 8.28. The FFNN model has measurably decreased error compared to the baseline model. Figure 8.29 shows that the FFNN model more closely predicts both the amplitude change and the frequency of oscillation.



(a) Pitch

(b) Pitch error

Figure 8.28: Open-loop orientation comparison

The overall error metrics are listed in Table 8.9. While the FFNN model had increased translation error, the pitch error was decreased by about 2%.

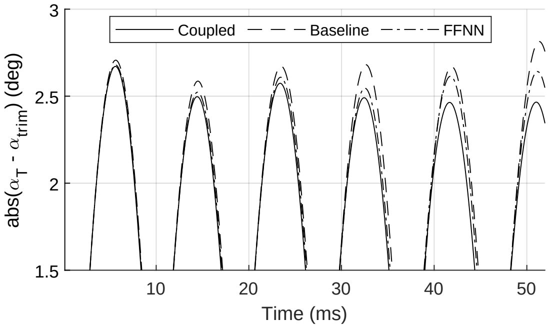


Figure 8.29: Total angle of attack difference to average trim angle of attack

Table 8.9: Open-loop trajectory component RMS error comparison

Model	x_{dec} (%)	r_z (%)	θ (%)
Baseline	0.003	0.081	3.48
FFNN	0.067	0.132	1.66

8.3.3 Findings

While the results of Experiment 2 demonstrate that the FFNN model has the best performance in the aggregate, the replication of Experiments 1.1 and 1.2 provides greater detail into the mechanisms by which the model is improved.

The aerodynamic results show that the FFNN prediction errors have weaker patterns with respect to the input variables, correcting the most obvious deficiencies noted from the baseline model. Even in cases where the error metrics increased, the qualitative trends with the input variables were decreased. While this capability is known from literature, Section 8.3 demonstrates that the effects are measurable for the problem of interest.

Finding 8.5: Without a prescribed functional form, the FFNN-based aerodynamic surrogate model better captures the response as a function of the input variables.

The most important result is that for the pitch response, the FFNN model had lower error than the baseline model for both the free-flight and open-loop simulations. The baseline model free-flight prediction was already below the threshold of $0.17°$ for precision landing with active guidance, so any improvement thereupon is not significant. However, the baseline model open-loop prediction did have significant error. The FFNN model has reduced this error to below the threshold, demonstrating that in this case, the differences in the model are significant.

Finding 8.6: The FFNN-based aerodynamic surrogate model provides error reduction in the pitch response that can be significant compared to the threshold established for instantaneous tracking error.

8.4 Conclusions

Research Question 2.1 (How can an aerodynamic surrogate model be constructed using time-series data from the coupled model as it is generated during the tuning process?) can be answered using the findings from Experiment 2. The question was designed to address the character of the data output from the coupled model and how it differs from the data used for a state-of-the-art model: much higher density along the trajectory responses, but otherwise much more sparse. The results of Experiment 2 demonstrate that the quasi-linear, FFNN, and TDNN models are capable of being trained with this data while yielding satisfactory error in extrapolation (in the case of the FFNN and TDNN models this is true once a certain threshold for the training set size has been passed).

Findings 8.1 and 8.3 prove that all of the surrogate models evaluated converge with the coupled model response with increasing training set size, or that the components already have satisfactory error with respect to the coupled model. Findings 8.2 and 8.4 prove that the FFNN model has the best performance, *ipso facto* providing better error convergence than the baseline model. On this basis, Hypothesis 2.1 is accepted. An advanced surrogate model for the aerodynamics of the problem of interest was identified and found to perform better than the baseline model.

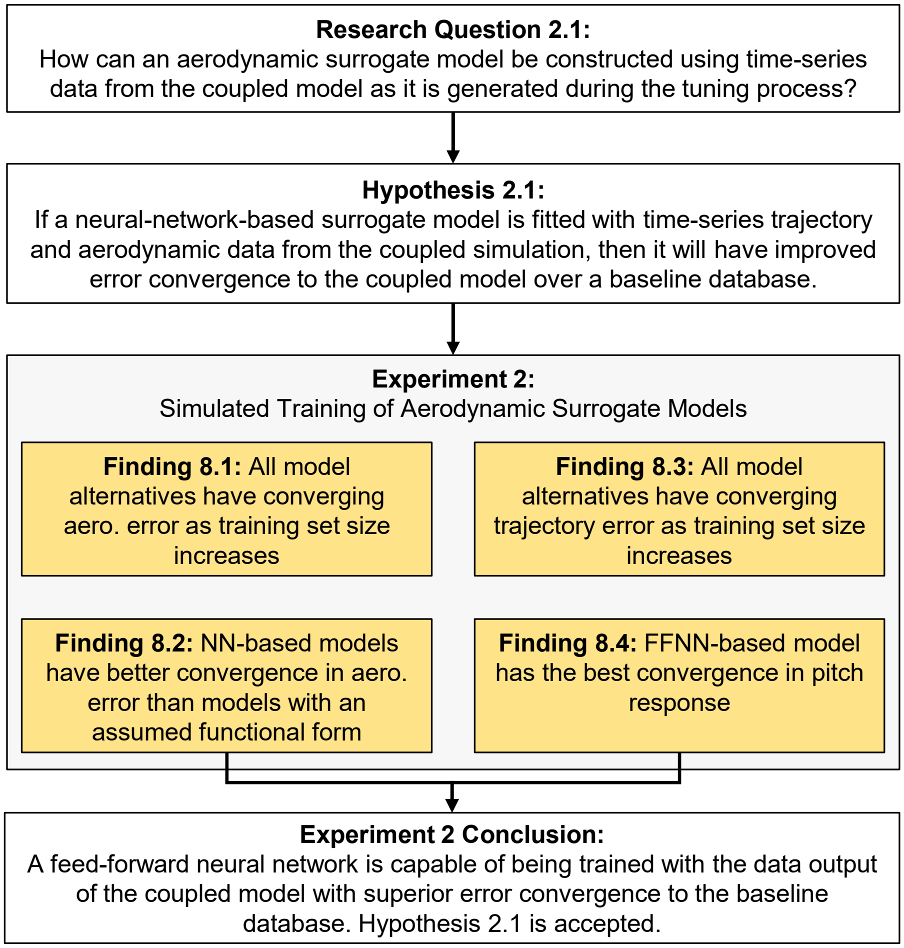
However, the performance of the other neural network model, the TDNN, bears further discussion. The TDNN did not perform better than the baseline model with respect to the pitch angle, which is the most important component. While the FFNN was demonstrated to be capable of using the time-series nature of the data, the training process treated the data as completely independent of any temporal relationship. The TDNN was the only model considered that directly utilized the time-series nature of the data.

Figure 8.30 presents a summary of the investigation performed in order to answer Research Question 2.1.

The final question to be discussed is whether the FFNN merits use in the multifidelity controller tuning process. Finding 8.5 proves that, from a scientific basis, the FFNN is a better model to use. The surrogate model chosen does correct the deficiencies identified for the baseline aerodynamic database. Finding 8.6 demonstrates that the substitution of the FFNN for the baseline model could yield significant results when used for tuning an entry controller. The execution of Experiment 2 revealed no disadvantages to using a FFNN model that outweigh the potential improved accuracy. While the k-fold results from Experiment 2 cannot be directly related to the threshold for tracking error significance estabilished for entry vehicles with active guidance, the subsequent use of the FFNN to repeat Experiment 1.2 did yield results that rose to that level of significance. Therefore, the methodology will use the FFNN for the low-fidelity aerodynamic surrogate.

These conclusions, like those for Experiments 1.1 and 1.2, have limitations that derive from the design of the problem of interest. It is possible that the surrogate model alternatives considered might not have been suitable for a vehicle with more complex aerodynamic behavior. Similarly, training data over a wider range of flight conditions might have revealed behavior which the alternatives might struggle to represent. The largest limitation is that these surrogate architectures are tailored to the control mechanism of the problem of interest. They would be incompatible with RCS actuation, for example.

The experimental procedure also results in the limitation that no control mechanism actuation was included in the training data. While this was acceptable for the problem of interest, it would likely be insufficient for a vehicle with a more complex control mechanism. The premise of the experiment, to provide aerodynamic and trajectory comparisons to the baseline database, also limited the surrogate alternatives to the paradigm of linearized aerodynamics. Without the need for this comparison, more complex surrogate models could be investigated.



Research Question 2.1: How can an aerodynamic surrogate model be constructed using time-series data from the coupled model as it is generated during the tuning process?

Hypothesis 2.1: If a neural-network-based surrogate model is fitted with time-series trajectory and aerodynamic data from the coupled simulation, then it will have improved error convergence to the coupled model over a baseline database.

Experiment 2: Simulated Training of Aerodynamic Surrogate Models

Finding 8.1: All model alternatives have converging aero. error as training set size increases

Finding 8.3: All model alternatives have converging trajectory error as training set size increases

Finding 8.2: NN-based models have better convergence in aero. error than models with an assumed functional form

Finding 8.4: FFNN-based model has the best convergence in pitch response

Experiment 2 Conclusion: A feed-forward neural network is capable of being trained with the data output of the coupled model with superior error convergence to the baseline database. Hypothesis 2.1 is accepted.

Figure 8.30: Summary of investigation into Research Question 2.1

CHAPTER 9
EVALUATION OF DATA FUSION FOR MULTIFIDELITY CONTROLLER TUNING (RESEARCH QUESTION 2.2)

This chapter documents the investigation of Research Question 2.2 and its associated Hypothesis 2.2, both of which are restated below:

Research Question 2.2

How can the multifidelity optimization method utilize the high- and low-fidelity controller metrics to reduce the computational cost of controller tuning?

Hypothesis 2.2: If a data-fusion-based multifidelity optimization method is used to tune an entry controller, then it will have less computational cost than using a quasi-hierarchical filtering-based method for an equal performance.

The proposed methodology for tuning control gains subject to the unsteady behavior and interactions captured by the coupled framework is dictated by the characteristics of the simulation. The chief consideration, computational cost, has been discussed in Section 4.2.2, motivating the shift from a Monte Carlo filtering activity to an optimization activity. Further cost reduction can be gained by exploring multi-fidelity optimization techniques, where a lower-cost, lower-fidelity model can be used to efficiently explore the design space.

Effectively, the most important decision in the selection of the optimization algorithm is whether to use data fusion methods to inform the optimizer. Experiment 3 is designed to evaluate which method has the lowest computational cost. Both methods are used to tune a controller for the problem of interest, with the results tracked in comparison to a known optimum. For reduced computational cost, the coupled model is substituted for a lower-fidelity aerodynamic surrogate, with an even lower fidelity surrogate architecture created to preserve the multifidelity nature of the problem.

As a prerequisite to the experiment, Section 9.1 details the extension of the problem of interest given the new data made available through Experiments 1.1 and 1.2, and 2. Section 9.2 describes the optimization method alternatives considered for Experiment 3, including the common features and the different data fusion alternatives. Section 9.3 documents Experiment 3, including the procedure, results, and findings. The conclusions are then presented in Section 9.4.

9.1 Controller Tuning Problem of Interest

In addition to the definitions of the vehicle and control mechanism for the problem of interest in Chapter 6, a requirement for an entry controller must be defined that is representative of the controller tuning design activity.

Since the literature review in Chapter 3 found that the instantaneous tracking error is a primary concern for designers, the problem of interest will seek to minimize the α response error with respect to α_{cmd}. As observed from the literature review, tuning a controller for multiple commanded conditions is achieved by extending or repeating the tuning methodology. As such, the problem of interest can use a single α_{cmd} to serve as a functionally complete example. An $\alpha_{cmd} = -2.5°$ is selected because the value lies within the trim angle of attack while being large enough to require driving the IMMA to a significant extent of its travel.

Since the controller performance metric is only evaluated over a short duration, the initial conditions that it may be expected to operate under can vary depending on when during entry the controller is active. Thus, the performance will be tested for multiple sets of initial conditions. A subset of the initial condition cases developed in Section 7.1 will be used. The selected cases, shown in Table 9.1, are the four cases with the lowest initial amplitude. This will minimize the total computational cost of running the coupled model until a controller achieves control.

Table 9.1: Coupled simulation initial conditions used for problem of interest

Case	Pitch (°)	Pitch Rate ($^\circ/s$)	Pitch Amplitude (°)
1	0.90	997	2.5
5	0.21	-468	1.4
6	3.23	-877	4.0
7	5.02	235	5.3

From an engineering perspective, the designers of a vehicle seek to plan for a minimum level of performance. As such, the requirements on an entry controller are written to constrain the maximum tracking error subject to a set of testing conditions. Monte Carlo simulation on the baseline model was performed while varying the initial conditions in order to explore the controller gain design space. It was found that the performance is most sensitive to the values of $k_{z,3}$ and $k_{z,6}$. Therefore, the other controller gains were defaulted at the optimum values determined from the baseline model. Thus, for the problem of interest, the overall evaluation criterion (OEC) for controller performance is found by Equation 9.1:

$$OEC = \max_{j=[1,5,6,7]} \tilde{\alpha}_{ERMS,LF}(k_{z,3}, k_{z,6}, \theta_{0,j}, q_{0,j}) \tag{9.1}$$

The OEC is a complex function depending on four input variables, which is satisfactory for the problem of interest.

9.2 Optimization Methods

Two main optimization method alternatives will be evaluated. Before describing these alternatives, the common characteristics are enumerated. The flowchart for the multifidelity tuning methodology is reprinted in Figure 9.1, and both alternatives follow the steps listed below.

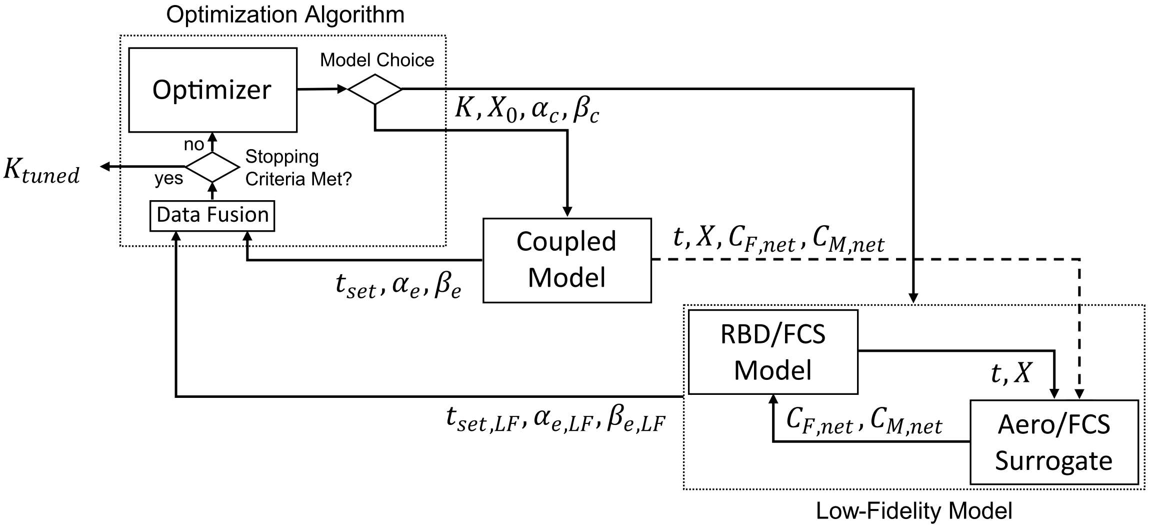


Figure 9.1: Multifidelity controller tuning process

1. The LF aerodynamic surrogate is "hot-started" by populating it with an initial set of data. This also allows for an initial guess for controller design.

2. The multi-fidelity optimizer is executed, calling the high- and low-fidelity models as dictated by the algorithm. Each time the optimizer calls the HF model, the following steps are executed:

   (a) The optimizer calls for some set of runs in the HF model. This can include runs at multiple gain designs and initial conditions.

   (b) The HF model is executed at for all of the specified runs for a set duration.

   (c) Post-processing extracts the time-series vehicle and control system state and aerodynamic coefficients. It calculates the net forces and moments on the vehicle, excluding gravitational forces and direct control system effects. This leaves aerodynamic and control system interaction effects.

   (d) The time-series vehicle state and net forces and moments are added to the pool of data available for the low-fidelity model.

   Each time the optimizer calls the LF model, the following steps are executed:

   (a) A pre-processor checks to see if any new time-series data has been made available to the LF surrogate model. If so, the surrogate is re-trained using the available data, subject to its own algorithm for partitioning the data into training, validation, and test sets. If the data set is the same, no change is made to the surrogate model.

   (b) The LF model is executed at the input initial conditions and commanded state. The trajectory propagator settings are the same as those set for the HF model.

3. After execution of either the HF or LF model, post-processing is performed to calculate the controller performance metrics, which are passed back to the optimizer.

4. The optimizer uses the output performance metrics data according to its own algorithm to generate the next guess for the controller design.

5. The optimizer evaluates the results from the HF model and determines whether the control performance requirements are satisfied. If so, then the optimization is terminated, yielding the set of satisfactory control gains. If not, the optimization continues.

The differences between the Experiment 3 alternatives are constrained to within the optimization algorithm. The following subsections discuss the notable features that are the same between the two.

Quasi-Hierarchical Relationship

Recall that in the multifidelity controller tuning methodology, every time the high-fidelity model is called it produces new time-series aerodynamic and trajectory data. This data is used to re-train the low-fidelity aerodynamic surrogate. Experiment 2 demonstrated that the LF aerodynamic surrogate therefore provides converging error with respect to the high-fidelity model. Since the model is adapted over time, the multifidelity method is considered to be quasi-hierarchical even with no data fusion.

Low-Fidelity Error Metric Surrogate

For both methods, a surrogate model is made of the low-fidelity error metric. The surrogate model predicts the RMS error:

$$\tilde{\alpha}_{RMSE,LF} = f(k_{z,3}, k_{z,6}, \theta_0, q_0) \tag{9.2}$$

where $k_{z,3}$ and $k_{z,6}$ are the free controller gains and θ_0 and q_0 are the initial pitch and pitch rate.

The model takes the form of a single-layer feed-forward neural network (FFNN). The hidden layer uses perceptron neurons with a sigmoid activation function, the same as the FFNN architecture described in Section 8.1.3. This LF surrogate[1] provides a continuous prediction for use by the optimizer.

To construct the LF surrogate, the low-fidelity model is sampled using a DOE that spans the design space of controller gains, plotted in Figure 9.2. An outer, 40×40-point grid design is used to explore a region identified in Section 9.1. This region contains the minimum as evaluated in the baseline model constructed in Section 7.2 from the free-flight coupled simulations. A higher-density Latin hypercube design with 800 points is used to explore a smaller region around the minimum. The change in density is acceptable since

[1] Unless otherwise indicated, "LF surrogate" refers in this chapter to the low-fidelity surrogate of α_{RMSE}

the global minimum, which is the feature of interest, probably lies in this area. At each controller design, the LF model is sampled at all four cases stipulated in the requirement, for a total of 9,940 design points.

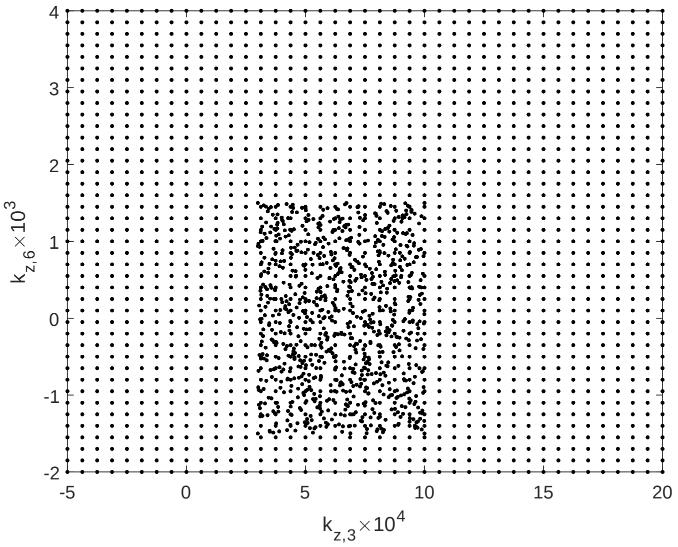
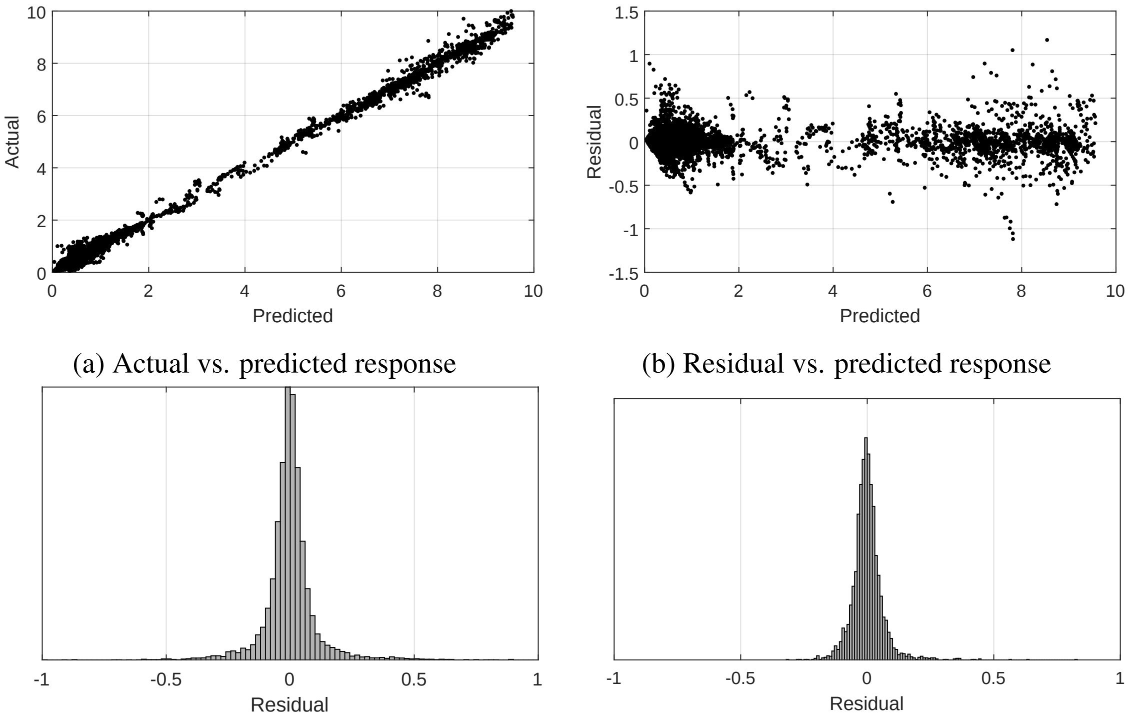


Figure 9.2: DOE in controller gain dimensions for low-fidelity model sampling

An example of the training is shown for the LF surrogate produced on the first iteration (which is the same for both alternatives). A LF surrogate with 75 hidden layer nodes is trained using the Levenberg Marquardt training algorithm with the same settings as described in Section 8.1.3. The goodness of fit metrics are plotted in Figure 9.3. Figures 9.3a and 9.3b so not show any noticeable trends in the actual or residual values as a function of the predicted value. The clumping along the predicted values is unavoidable since the actual values are not evenly distributed. Figure 9.3c shows that the MFE has a mean close to zero, with a standard deviation of $0.11°$, about 1.2% the total range of the response. The MFE for the Latin hypercube design is shown in Figure 9.3d, demonstrating that the fit is better for the region which probably contains the global minimum. Therefore the LF surrogate is judged to be a good fit to the generated data.



(a) Actual vs. predicted response

(b) Residual vs. predicted response

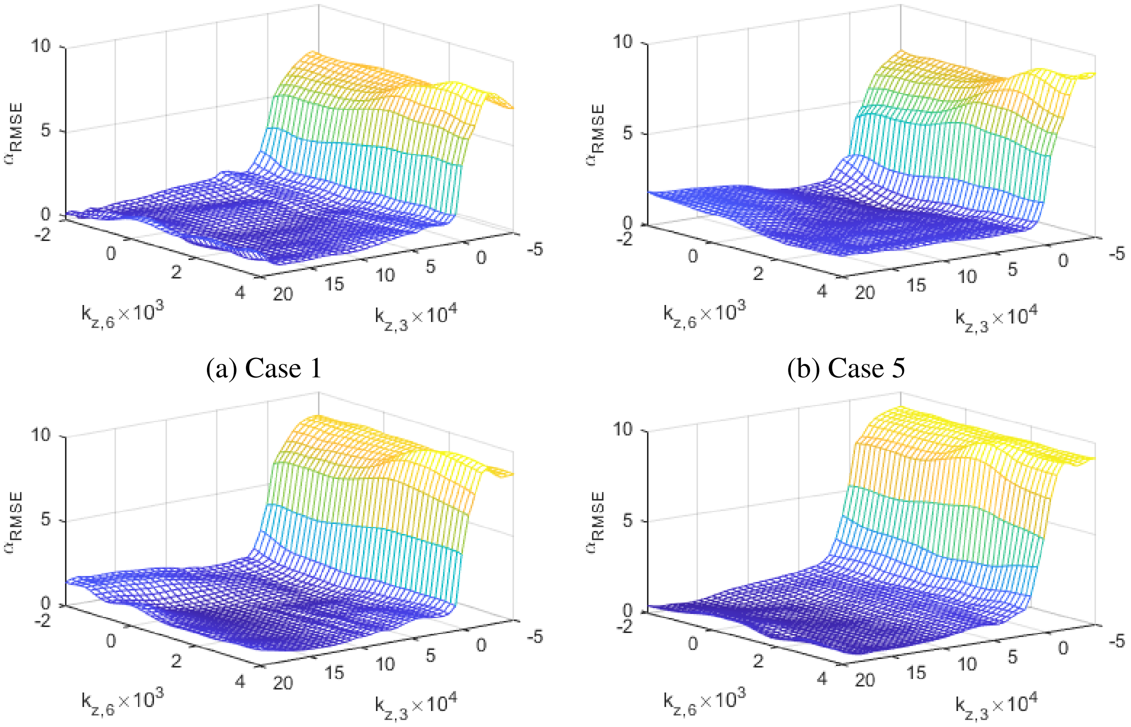
(c) Residual histogram

(d) Residual histogram (Latin hypercube DOE)

Figure 9.3: Goodness of fit metrics for LF surrogate

An example of the LF surrogate is shown in Figure 9.4. The RMSE surface is plotted as a function of the controller gains for each initial condition case. For each case, the largest driving factor is the value of $k_{z,3}$. The RMSE is significantly decreased with values increasing over zero.

The OEC is visualized in Figure 9.5, where the surface is colored according to which input case is driving the OEC. The OEC is largely dominated by cases 5 and 7, and to a lesser extent case 6 at the extremes of $k_{z,6}$. Case 1 is not dominant within these bounds. Within the region of interest, there are several local minima. The Monte Carlo simulation performed in Section 9.1 to narrow the scope of the controller design show that these likely include the global optimum for a wider range of $k_{z,3}$ and $k_{z,6}$.



(a) Case 1

(b) Case 5

(c) Case 6

(d) Case 7

Figure 9.4: LF RMSE surfaces at input cases

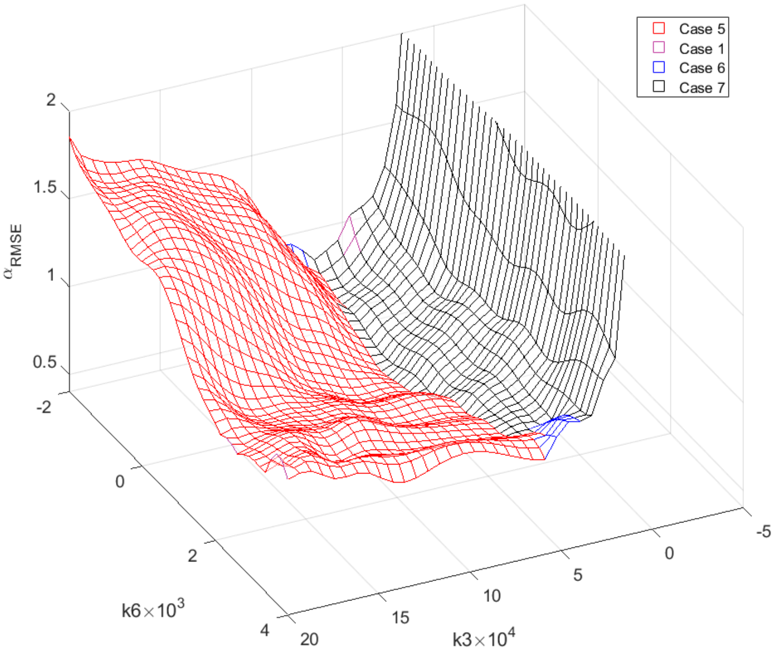


Figure 9.5: OEC surface calculated from LF surrogate

Initialization Procedure

Both optimization processes are started with the same initial guess for controller design. The alternatives are also started with the same four high-fidelity runs to "hot start" the aerodynamic surrogate.

Optimizer

Whenever the methods are searching for the minimum, they will use the same optimization process. The optimizer minimizes the OEC, reprinted below:

$$OEC = \max_{j=[1,5,6,7]} \tilde{\alpha}_{RMSE}(k_{z,3}, k_{z,6}, \theta_{0,j}, q_{0,j}) \tag{9.3}$$

A genetic algorithm (GA) was selected to perform the optimization. A GA optimizer allows for the exploration of the global minimum in the presence of many local minima [191]. The MATLAB® Optimization Toolbox™ was used to execute the optimization [235]. The optimizer used generations of 50 individuals, with a crossover fraction of 0.8.

Within the optimization algorithms, the model choice and stopping criteria is also held constant. Model selection alternates between sets of runs in the high- and low-fidelity models. This is done because of the low cost of the LF model. There is no instance where the algorithm, when calling the LF model, would not run it for as many points as possibly desired. It also follows that the LF model would be called after every set of HF model calls, to take advantage of the quasi-hierarchical relationship. Thus, one iteration of the optimization algorithm is considered to include the calls to the LF model, then the HF model.

The optimization algorithm will also sample all four cases, allowing the optimizer to have completely accurate knowledge of the OEC at each design guess.

9.2.1 Filtering-Based Method

The filtering-based approach is designed as an adaptation of the state-of-the-art gain scheduling method. The optimizer performs the following steps:

1. The LF aerodynamic surrogate models are trained using the available HF results. This new LF aerodynamic surrogate is then incorporated into the LF model.

2. A DoE of the LF model is performed to explore the control gain design space.

3. The LF surrogate is fitted using the LF model results.

4. The optimizer is run on the LF surrogate to find the controller design with the mininmum OEC.

5. The high-fidelity model is executed using the gain design selected from the previous step. The new high-fidelity results are added to the training data available for the LF aerodynamic surrogate. The controller performance metrics are evaluated to determine whether the design satisfies the requirements.

6. If the design does not satisfy the requirements, return to step 1. The algorithm can also be stopped if it does not meet some threshold for improvement over a certain number of iterations.

Each iteration of this methodology is equivalent to gain scheduling with the low-fidelity model, the results of which are then validated using the high-fidelity model.

As iterations are executed, the new HF runs made available for training the aerodynamic surrogate can be thought of as the decision to populate the low-fidelity model with more data before performing the gain scheduling. However, the DoE for populating this data is determined as a byproduct of the optimization, rather than an intentional design architecture such as Latin hypercube. Thus it has no guarantee of sampling any particular new space, but it could have a higher probability of sampling data in the region of interest.

From the optimizer's persepctive, the results of each iteration are independent from the previous ones. Additional checks are therefore made to ensure that the optimizer does not choose a design to run in the HF model that has already been used.

9.2.2 Fusion-Based Method

The fusion-based method applies a corrector to the LF surrogate in order to produce a multifidelity (MF) surrogate of the controller performance. The use of a corrector function was chosen due to the large number of LF design points. The number of points would make a deterministic method such as co-Kriging much more computationally expensive. An additive correction δ_{MF} was selected, giving the MF surrogate the form:

$$\tilde{\alpha}_{RMSE,MF} = \delta_{MF}(k_{z,3}, k_{z,6}, \theta_0, q_0) + \tilde{\alpha}_{RMSE,LF}(k_{z,3}, k_{z,6}, \theta_0, q_0) \tag{9.4}$$

Given the variation in magnitude of the response, an additive correction was chosen over a multiplicative correction since it is predicted to result in a function with greater uniformity, thus performing better in extrapolation.

δ_{MF} will use a Kriging model fitted against the error between the LF surrogate and the HF response. Kriging is a common surrogate modeling method that represents the response as a combination of mean behavior with deviations from the mean [236]. The mean behavior is determined by a set of basis functions, and stochastic functions capture deviations from the mean. Gaussian processes are commonly used stochastic functions.

Assume the response y is sampled at n points as a function of k independent variables. For $i = 1, ..., n$, each response $y^{(i)}$ has the corresponding sample $x^{(i)} = (x_1^{(i)}, ..., x_k^{(i)})$. Kriging is used to find the approximate value of the response, $\hat{y}$, at an unsampled location x^* using the function:

$$\hat{y}(x^*) = \mu + \epsilon \tag{9.5}$$

where μ is the mean term, and ϵ is the error term.

In Universal Kriging, μ is a linear combination of functions $f_0(x), ..., f_k(x)$, each with an estimated coefficient.

$$\mu = f(x^*)^T \hat{\beta} \tag{9.6}$$

where $f(x^*)$ is a $(k+1) \times 1$ vector of basis functions, and $\hat{\beta}$ is a $(k+1) \times 1$ vector of coefficients.

ϵ is normally distributed with zero mean and some variance σ^2.

$$\epsilon = c(x^*)^T C^{-1}(y - F\hat{\beta}) \tag{9.7}$$

where $c(x^*)$ is an $n \times 1$ vector of covariance between x^* and all of the sample points, C is an $n \times n$ covariance matrix between each of the sample points, and F is an $n \times (k+1)$ experimental matrix.

The covariance vector $c(x^*)$ and matrix C rely on a function to calculate the correlation between points. For example, the Gaussian correlation function for points u and v is:

$$d(u, v) = \prod_{i=1}^{k} exp\left[-\left(\frac{|u_i - v_i|}{\theta_i}\right)^2\right] \tag{9.8}$$

where θ_i is the weight for each dimension. Thus, $c(x^*)$ and C are calculated as:

$$c(x^*) = \begin{bmatrix} d(x^*, x^1) & d(x^*, x^2) & \dots & d(x^*, x^n) \end{bmatrix}^T \tag{9.9}$$

$$C = \begin{bmatrix} d(x^1, x^1) & d(x^1, x^2) & \dots & d(x^1, x^n) \\ d(x^2, x^1) & d(x^2, x^2) & \dots & d(x^2, x^n) \\ \vdots & \vdots & \ddots & \vdots \\ d(x^n, x^1) & d(x^n, x^2) & \dots & d(x^n, x^n) \end{bmatrix} \tag{9.10}$$

The experimental matrix F is calculated as:

$$F = \begin{bmatrix} f_0 & f_1(x_1^1) & \dots & f_k(x_k^1) \\ f_0 & f_1(x_1^2) & \dots & f_k(x_k^2) \\ \vdots & \vdots & \ddots & \vdots \\ f_0 & f_1(x_1^n) & \dots & f_k(x_k^n) \end{bmatrix} \tag{9.11}$$

$\hat{\beta}$ is calculated by generalized least-squares estimate

$$\hat{\beta} = (F^T C^{-1} F)^{-1} F^T C^{-1} Y \tag{9.12}$$

Fitting the covariance requires inverting an $n \times n$ matrix, at a computational cost on the order of $O(n^3)$, where n is the number of data points [237]. Since the number of HF points is much more limited than the number of LF points, Kriging is feasible for the corrector where it was not feasible for the LF surrogate.

The Design and Analysis of Computer Experiments (DACE) package for MATLAB® is used to train and execute the Kriging model [238].

The fusion-based algorithm is executed using the following steps:

1. The LF aerodynamic surrogate models are trained using the available HF results. This new LF aerodynamic surrogate is then incorporated into the LF model.

2. A Monte Carlo simulation of the LF model is performed to explore the control gain design space.

3. The LF surrogate is fitted using the Monte Carlo results.

4. A Kriging surrogate model is trained using the error between the LF surrogate and the available HF results. The fitted corrector is paired with the LF surrogate to produce the MF surrogate.

5. The optimizer is run on the LF surrogate to find the controller design with the mininmum OEC.

6. The algorithm alternates between exploration and exploitation of the space on each iteration. On exploitation, the HF model is executed using the gain design selected from the previous step. During exploration, the expected improvement function is calculated for a region around the minimum. A space-filling Latin hypercube design is generated that maximizes the sum of the expected improvement (in order to maintain similarity with the filtering-based method, the design is limited to four points). The HF model is executed using these points. In both cases, the new high-fidelity results are added to the training data available for the LF aerodynamic surrogate.

7. On the exploitation iterations, the controller performance metrics are evaluated to determine whether the design satisfies the requirements. If the design does not satisfy the requirements, return to step 1. The algorithm can also be stopped if it does not meet some threshold for improvement over a certain number of iterations.

The fusion-based method utilizes the performance results from the low- and high-fidelity models to select the next high-fidelity design point. The filtering-based method does not use this information, relying solely on the adaptive update to the low-fidelity model aerodynamics. By leveraging the additional information, the fusion-based method is designed to result in a change in the computational expense for tuning the control gains.

9.2.3 Fusion-Based Method Without Updating Low-Fidelity Model

In addition to the two methods described in the sections above, optimization will also be attempted with a method that does not use the quasi-adaptive nature of the LF model; instead, it will rely solely on data fusion. This method will be the same as the fusion-based method in Section 9.2.2 except that the LF aerodynamic surrogate, LF model, and LF surrogate will not be updated after the first iteration. These will be held fixed while new

HF runs are only used to update the correction.

By only updating the corrector, this alternative provides a point of comparison to the fusion-based method. The effect of the changing LF response at each iteration can thereby be investigated.

9.3 Simulated Performance of Multifidelity Methods (Experiment 3)

9.3.1 Experimental Procedure

The experimental procedure is to simulate the multifidelity optimization of a controller design using each alternative. All of the methods are started with the same initial guess and are allowed to proceed for 15 iterations, regardless of the stopping criteria described in Sections 9.2.1 and 9.2.2. Cost will be evaluated in terms of the iteration, since each alternative uses the same number of calls to the high-fidelity model on each iteration.

However the high cost of the coupled model makes it infeasible to perform the experiment directly using the proposed methodology, or to perform further investigation. The FFNN-based aerodynamic surrogate — what would nominally be the LF aerodynamic surrogate — is used as an analogue to the coupled model. The critical factor in Experiment 3 is the investigation into the effectiveness of the optimization method given two models with higher and lower fidelity. A reduction in the absolute fidelity is acceptable if a relative difference in fidelity can be preserved. Therefore, an aerodynamic surrogate design must be selected that deliberately has lower fidelity than the previously-generated FFNN, but still provides the same categorical capabilities (training with the time-series data and convergence to the HF model with increasing training set size).

Low-Fidelity Aerodynamic Surrogate Model

The LF aerodynamic surrogate is constructed as an FFNN-based model with the same architecture as described in Section 8.1.3, but with much smaller hidden node sizes. The number of hidden layer nodes for each coefficient are listed for both models in Table 9.2.

The LF model has a significantly reduced number of unknowns for each coefficient. This will prevent the LF model from achieving the accuracy of the HF model.

Table 9.2: FFNN hidden layer nodes

Coefficient	High-fidelity model		Low-fidelity model	
	Nodes	Total unknowns	Nodes	Total unknowns
C_A	8	41	1	6
C_N	16	73	1	6
C_M	16	73	2	11

Generation of High-Fidelity Data

The use of the FFNN-based model as the HF model also allows for the HF data to be generated *a priori*. The model is executing using the same DOE constructed for the LF surrogate in Figure 9.2. When calling the HF model at some guessed design x_{guess}, the data will be returned from the design point with the closest euclidian distance (normalized by the ranges of $k_{z,3}$ and $k_{z,6}$). A contour plot of the HF OEC is shown in Figure 9.6, also indicating the location of the true optimum. The surface is somewhat noisy, with individual designs disrupting the general shape of the surface.

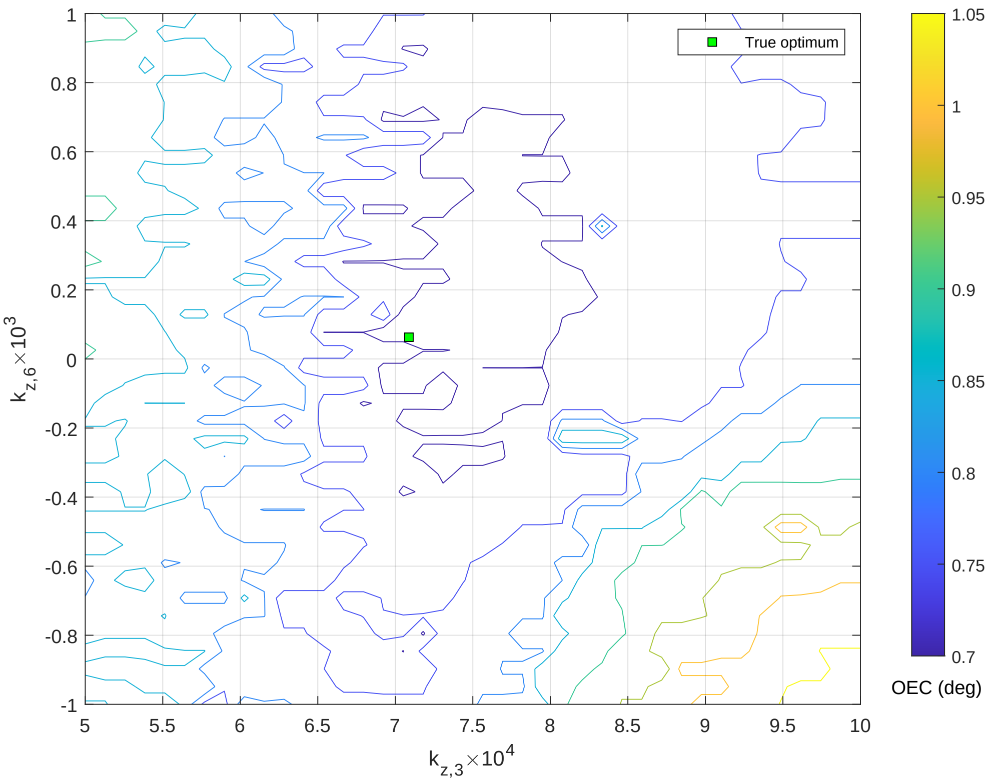


Figure 9.6: High-fidelity OEC response

Initialization Procedure

As stated in Section 9.2, each alternative is started with the same HF runs used to train the LF aerodynamic surrogate. Based on the results of Experiment 2, a set of four runs was judged to provide sufficient training data such that the LF model would perform in a physically believable manner, but would still insufficiently span the design space of flight conditions, allowing for improvement on later iterations. The conditions four runs chosen are listed in Table 9.3.

Table 9.3: Initial high-fidelity run conditions

$k_{z,3} \times 10^4$	$k_{z,6} \times 10^3$	Case	θ_0 (deg)	q_0 (deg/s)
-5.0	-2.0	5	0.21	-468
20.0	-2.0	1	0.90	997
-5.0	-2.0	6	3.23	-877
-5.0	4.0	7	5.02	235

The initial runs include gains that bound the design space, and has one run from each initial condition case. The resulting trajectories are plotted in Figure 9.7.

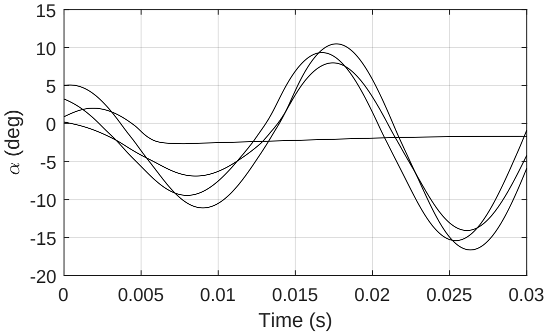
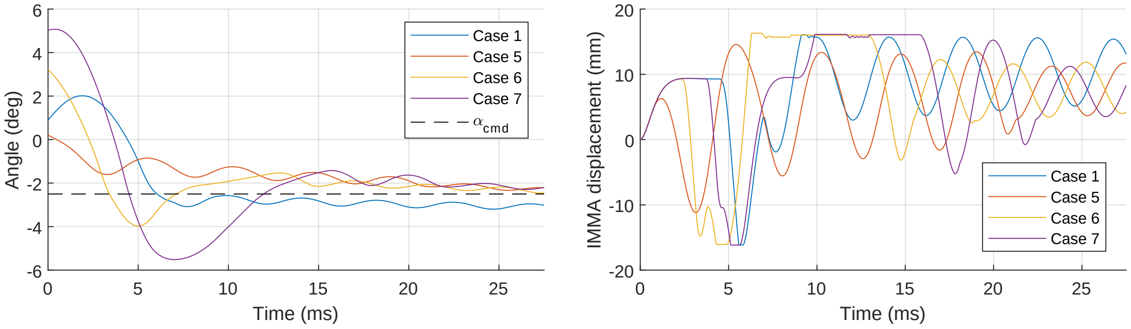


Figure 9.7: Initial high-fidelity runs provided for optimization algorithms

Only one of these designs managed to bring the vehicle somewhat under control, while the other three actually result in increased oscillation amplitude. Several of the runs exceed the maximum α_T of the coupled free-flight simulations. Since the accuracy of the behavior relative to the coupled model is not strictly important in this experiment, the extrapolation was judged to be acceptable.

9.3.2 Filtering-Based Method Execution

Figure 9.8 shows an example response generated by the HF model for all four input cases for a design which achieves control. The α response is plotted in Figure 9.8a, which drives the IMMA motion plotted in Figure 9.8b. Figure 9.8b shows that the IMMA travels to its limits for each case except Case 5, which has the smallest initial error. The largest variation in motion occurs at the start of the simulations, where the controller is working to damp out the angular velocity. The initial oscillatory motion is damped out, and the angle of attack comes under control around the α_{cmd}. Note that small amplitude oscillations continue, as the control mechanism has no means of imparting a moment to cancel out rotational motion other than by moving the mass and therefore inducing rotational motion. Note also that the oscillation period of about 4-5ms for the closed-loop simulation is much shorter than the period measured for free-flight, at about 17ms. The responses for a single design run at all four cases span a large range of the α design space for higher Mach numbers due to the initial conditions. However since α comes under control, the data spans a much smaller region of the α design space for lower Mach numbers. .



(a) Angle of attack

(b) IMMA displacement

Figure 9.8: Aerodynamic coefficient RMS error history

The results of the filtering-based method are examined in the order in which they are produced, starting with the LF aerodynamic surrogate. Coefficient errors are measured relative to the response of the FFNN surrogate being used for the HF model. The convergence of the aerodynamic surrogates are plotted in Figure 9.9.

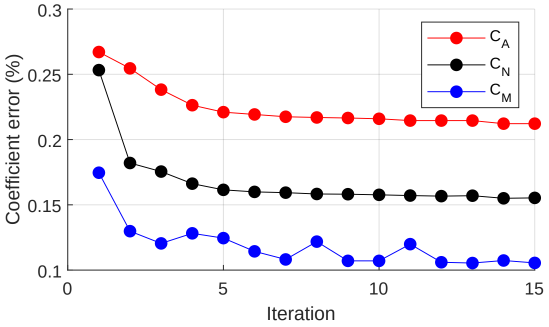


Figure 9.9: Aerodynamic coefficient RMS error during filtering-based method execution

The RMSE of each coefficient surrogate on the first iteration is compared in Table 9.4 to the k-fold results of the FFNN from Experiment 2 for the same number of runs. The error magnitudes between the LF aerodynamic surrogate (with artificially-reduced fidelity) and the FFNN is similar to the error between the FFNN and the coupled model. This provides validation for the sizing of the LF aerodynamic surrogate made in Section 9.3.1.

Table 9.4: Comparison of aerodynamic surrogate error

Coefficient	Experiment 3 LF surrogate	k-fold FFNN from Experiment 2
C_A	0.27	0.65
C_N	0.25	0.22
C_M	0.17	0.27

Figure 9.9 shows that each of the coefficient surrogates have converging error with increasing training set size. Both the C_A and C_N surrogates are largely converged by iteration 6, although both have slight improvement through to the final iteration. While C_M does converge, there is more noise, including noticeable increases on iterations 4, 8, and 11.

Figure 9.10 plots the RMS error of the LF surrogate with respect to the HF data. Two measurements are included: one measured at every HF data point, and one measured for the points within a certain region near the true optimum design. This region is defined as being within 5% of the normalized range of the controller gains:

$$|\frac{k_{z,3} - k_{z,3,opt}}{k_{z,3,max} - k_{z,3,min}}| \leq 0.05, \quad |\frac{k_{z,6} - k_{z,6,opt}}{k_{z,6,max} - k_{z,6,min}}| \leq 0.05 \tag{9.13}$$

Figure 9.10 shows that the error in the region near the true optimum is about half that of the error over the whole design space. However, neither measurement significantly decreases with increasing iteration. The variation does not correlate with coefficient error, which is consistent with the results from Experiment 2.

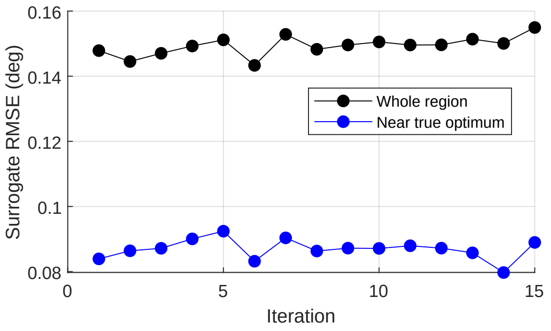


Figure 9.10: LF surrogate RMSE during filtering-based method execution

The OEC response history is plotted in Figure 9.11, showing the value of the queried response at each iteration as well as the cumulative minimum. Each predicted optimum design from the filtering-based method yields some improvement on the initial guess. The filtering-based method is able to obtain the true minimum on iteration 5. This equates to an improvement in the OEC of 15.8%.

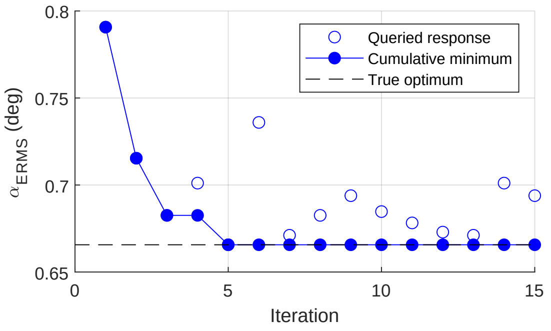


Figure 9.11: Filtering-based method response history

Figure 9.12 plots the history of predicted optimum design over a contour plot of the HF response. The figure shows how the predicted optimum does move circuitously towards the true optimum, but then moves away. For the remainder of the iterations, the predicted optimum remains within the lowest contour on the plot, but does not converge on a particular location.

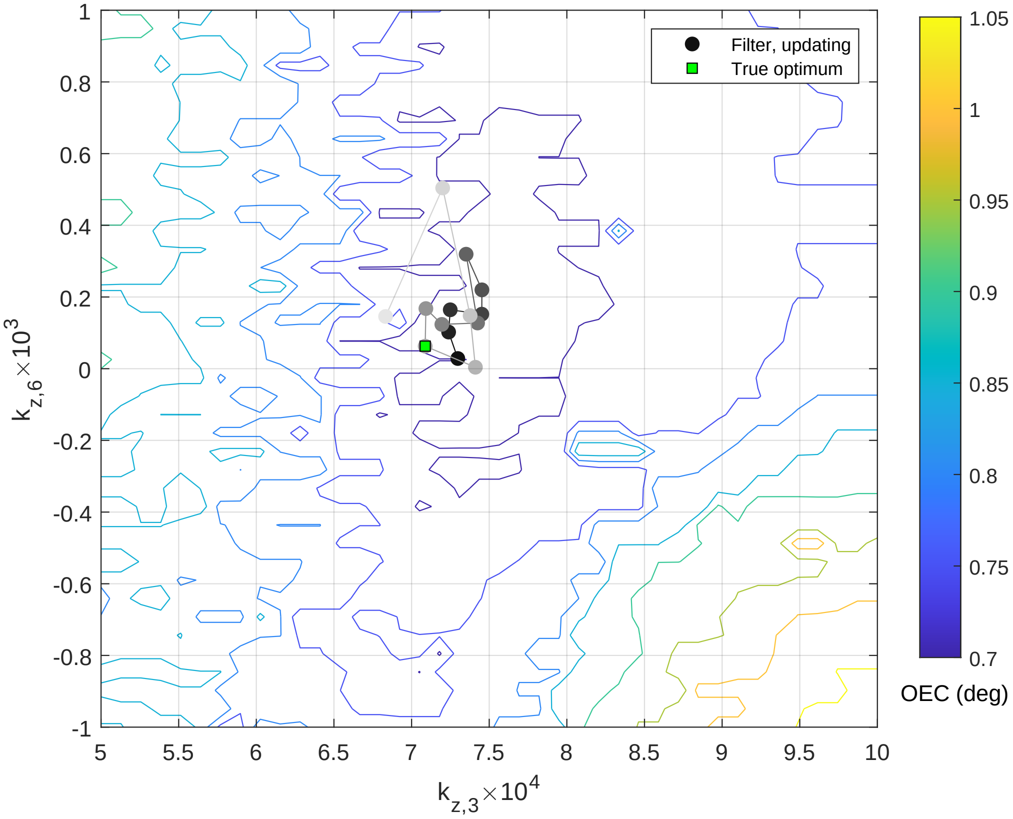


Figure 9.12: Filtering-based method predicted optimum design history. Lighter to darker color of the points indicates increasing iteration.

Findings

The filtering-based method does successfully yield the optimum design within the number of iterations tested in the experiment:

Finding 9.1: The filtering-based method succeeds in finding the optimum design.

Furthermore, the LF aerodynamic surrogate constructed at each iteration does converge in error even though the ratio of model unknowns to data points is much higher than in Experiment 2.

Finding 9.2: The filtering-based method yields an LF aerodynamic surrogate that has converging error with increasing training set size.

While this provides validation for the specific problem developed for Experiment 3, further application of these results is limited to the finding that the results do not disprove Hypothesis 2.1.

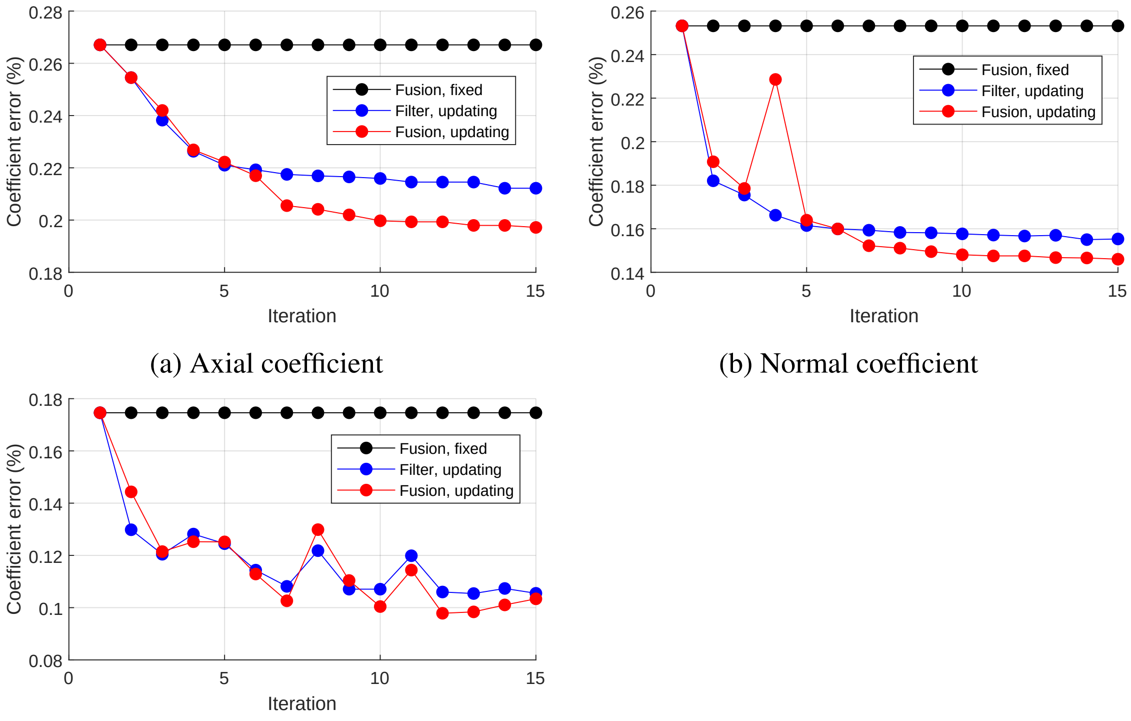
Examining the LF surrogate error results brings into question the efficacy of the filtering-based method. Whereas in Experiment 2, the increasing training set size resulted in a general convergence of error for both the aerodynamic and trajectory components, for the filtering-based method the additional data does not increase the accuracy of the LF response. As the LF model incorporates additional data, it moves the predicted minimum away from the true minimum. Thus, it cannot be disproved that the filtering-based method only succeeded in finding the true optimum due to random chance.

Finding 9.3: The filtering-based method does not yield an LF model that has converging error against the HF response with increasing training set size.

9.3.3 Comparison of Fusion-Based Method Execution

The fusion-based method was executed, both with and without updating the LF aerodynamic surrogate, to enable comparisons with the filtering-based method.

A comparison of the aerodynamic coefficient convergence for all three methods is plotted in Figure 9.13. As expected, the fusion method with fixed LF aerodynamic surrogate has constant error across all iterations. Figures 9.13a and 9.13b clearly show that the fusion-based method with updating LF aerodynamic surrogate has better error convergence than the filtering-based method for both the axial and normal coefficients. The results are similar through iteration 6 (barring the outlier in C_N on the 4th iteration), whereupon the fusion-based results diverge. Figure 9.13c shows that the two methods yield results with similar noisy behavior for C_M, but the fusion-based method achieves the lowest error.



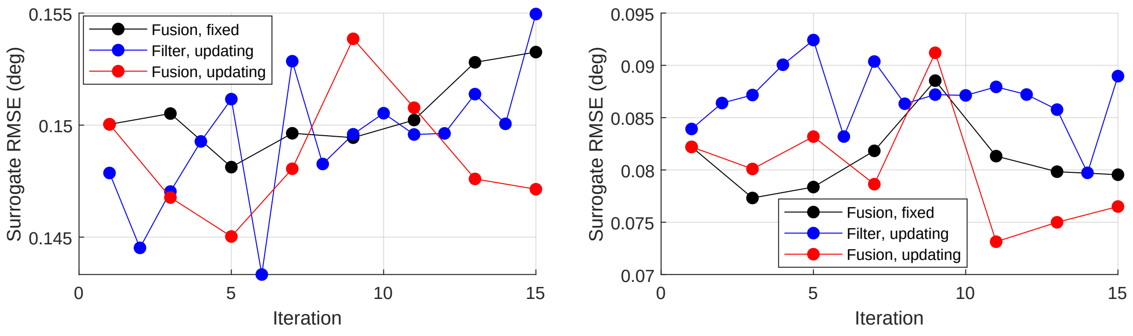
(a) Axial coefficient

(b) Normal coefficient

(c) Moment coefficient

Figure 9.13: Aerodynamic coefficient RMS error history

The LF surrogate error convergence is plotted in Figure 9.14, again measured across the whole design space and in a region near the true optimum. Figure 9.14a shows that, as with the filtering-based method, both fusion-based methods do not have converging error with increasing iteration. In the case of the fusion-based method with updating LF aerodynamic surrogate, this is in spite of the decreased error in the aerodynamic coefficients. However, Figure 9.14b shows that in the region near the true optimum, the fusion-based methods do have a slight decrease in error. The results are not monotonic, with particularly high values on iteration 9.



(a) Whole design space

(b) Near true optimum

Figure 9.14: LF surrogate RMS error history

Examining the behavior of the MF correction shows a possible cause. Figure 9.15 plots the MF correction for one initial condition case. The figure shows that while the surface corrects the results at the known HF points, the Kriging surface extends across the whole design space, with greater expected error as the distance from the sampled points increases. This explains why, with an increasing number of HF points, the near region could improve while causing no improvement when measured across the whole space. Furthermore, the recalculation of the Kriging surface can result in changes to the response magnitude that are significant with respect to the LF model response.

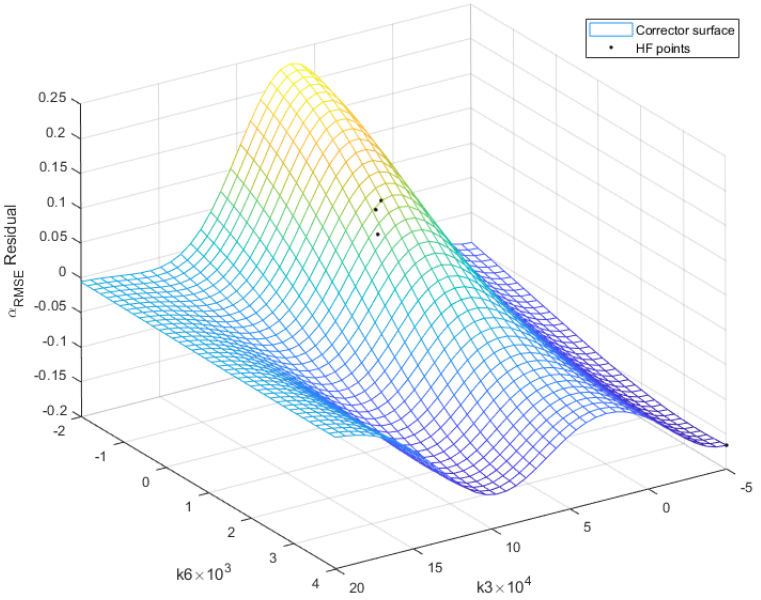


Figure 9.15: MF correction surface for Case 7, iteration 3

Figure 9.16 plots the response history for each method. The fusion-based method with fixed LF aerodynamic surrogate yielded the worst convergence, and yielded the worst results at the end of the execution. The convergence of the fusion-based method with updating LF aerodynamic surrogate was on par with the filtering-based method. However, the fusion-based method did not achieve the true optimum.

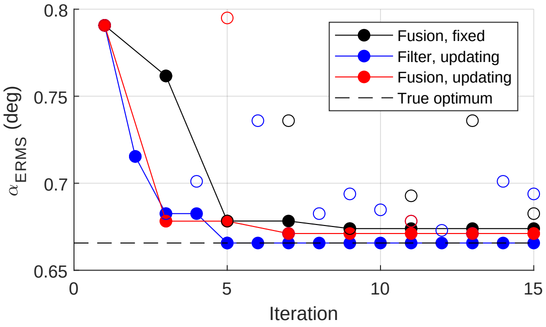


Figure 9.16: Response history

Figure 9.17 plots the history of predicted optimum design for each method over a contour plot of the HF response. Both of the fusion-based methods had guesses that were much farther away from the true optimum than the filtering-based method. As shown by the contour plot, these were in areas with a significantly higher true response.

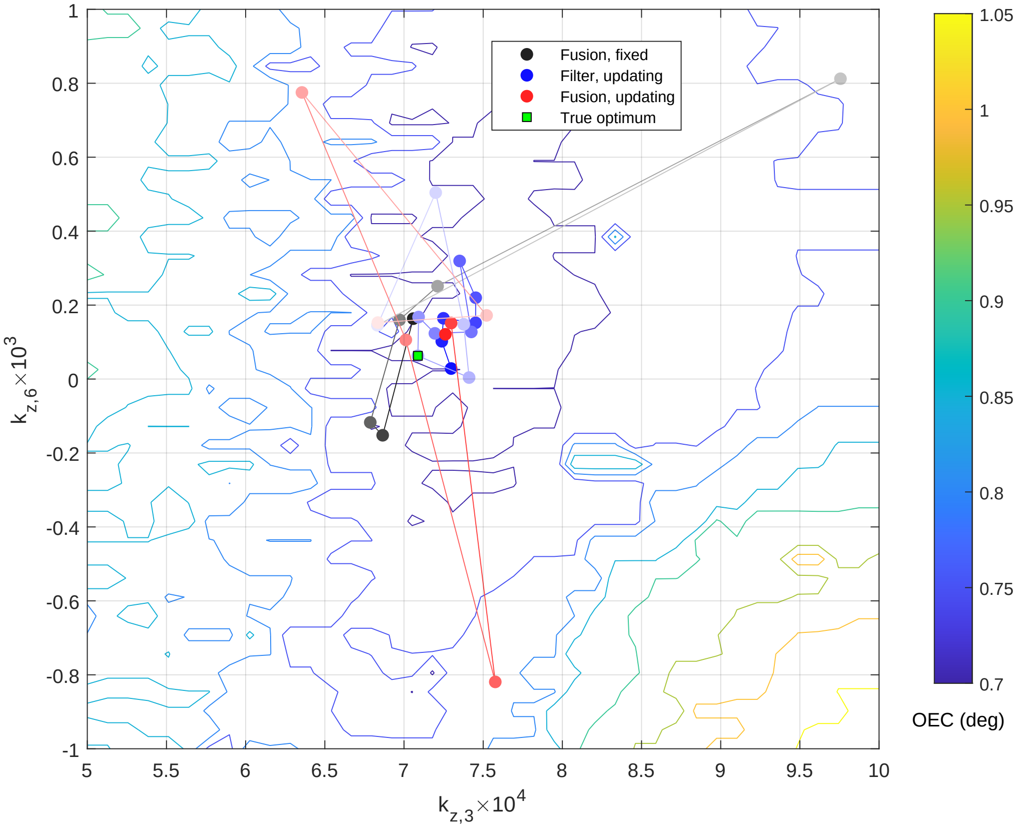


Figure 9.17: Predicted optimum design history. Lighter to darker color of the points indicates increasing iteration.

Findings

The results from both fusion-based methods do not achieve the true optimum within the number of iterations executed for Experiment 3:

Finding 9.4: The fusion-based method achieves a minimum result that is worse than the minimum result for the filtering-based method.

However, beyond the top-level results for the response convergence, the fusion-based method had characteristics that could be more beneficial than the filtering-based method. In terms of the aerodynamics, the fusion-based method resulted in simulations that better spanned the flight condition design space, with a corresponding decrease in the aerodynamic coefficient error.

Finding 9.5: The fusion-based method yields decreased error in the LF aerodynamic surrogate compared to the filtering-based method.

Furthermore, the use of data fusion in both fusion-based methods resulted in a slight but measurable decrease in response error near the true optimum:

Finding 9.6: The fusion-based method has decreased error with respect to the HF data in the region close to the true optimum compared to the filtering-based method.

9.4 Conclusions

Research Question 2.2 (How can the multifidelity optimization method utilize the high- and low-fidelity controller metrics to reduce the computational cost of controller tuning?) can be answered using the findings from Experiment 3.

Based on Findings 9.1 and 9.4, Hypothesis 2.2 is rejected. The fusion-based method did not achieve an equal performance while tuning the PID-based controller when compared to the filtering-based method, within the number of iterations executed in Experiment 3.

Findings 9.3 and 9.6 paint an unfavorable picture of the effectiveness of either optimization method in their current implementations. The filtering-based method showed no improvement in the response prediction, even with a large number of data points in the viscinity of the true optimum. The fusion-based method did yield a slight but insignificant improvement in the knowledge about the response in the vicinity of the true optimum. While both methods had lackluster performance, the results do indicate where improvements can be made. Therefore, the multifidelity controller tuning methodology will use a synthesis of the filtering- and fusion-based methods evaluated for Experiment 3. Note that this is an improvement on the fusion-based method is selected for implementation rather than the filtering-based method, even though the latter techically had better results. This is justified since the results of the filtering-based method were not substantially better than for the fusion-based method.

First, Finding 9.5 indicates that the fusion-based method, by virtue of attempting to explore the design space, results in a significant improvement in the aerodynamic surrogates. This is a beneficial result that should be kept. Based on the failures in extrapolation, the size of the design space to be covered by the DoE can also be expanded. In addition, without the need to maintain equivalence between the methods, the number of points queried can be increased.

Second, the procedure of the filtering-based method — using the minimum of the LF model — is, by definition, insensitive to the training of the multifidelity correction. During the first few iterations, this model could be used to anchor the guess for optimum design, and avoid the deviations that were caused by the extrapolation of the Kriging model.

Third, examining the relative results between the different initial condition cases shows that the LF models are generally able to identify the relative importance of each case with respect to the OEC. This information can be used to preferentially sample the input cases that are driving the OEC, and sampling the dominated cases with some lower frequency so as to maintain the accuracy of the relative predictions.

Figure 9.18 shows the multifidelity controller tuning process as it has been developed through the conclusions made from Experiments 1.1, 1.2, 2, and 3.

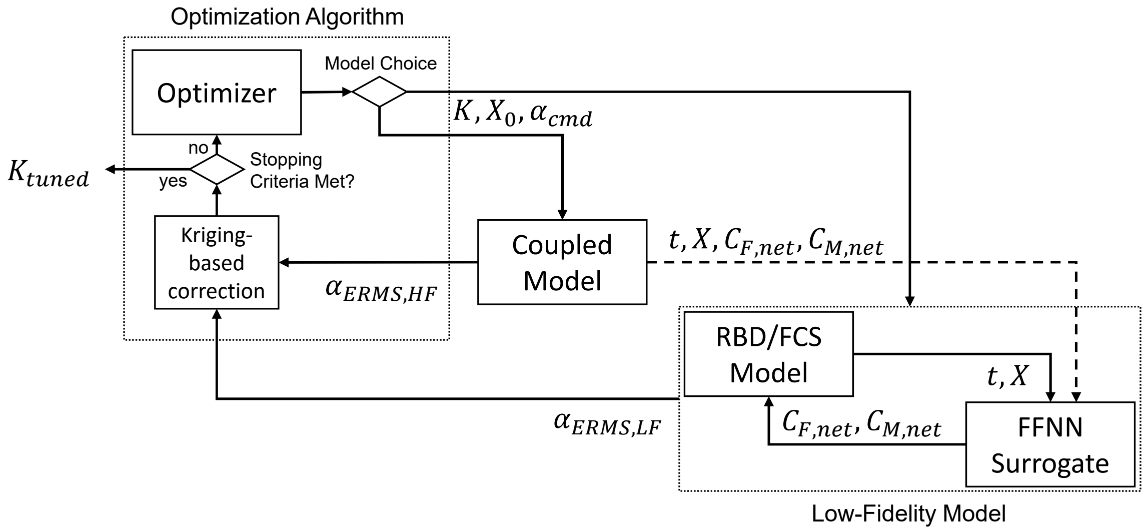


Figure 9.18: Multifidelity controller tuning process as implemented

The limitations of Experiment 3 arise from the problem of interest, the scope of the alternatives considered, and the experimental procedure. The controller for the problem of interest was selected base on literature review after establishing that it met the requirement that it could bring the vehicle under control when simulated with the baseline model. There are likely other control systems that could achieve better theoretical performance, allowing the tuning methods to be exercised over a greater range of improvement. Defaulting all but two of the controller gains has a similar effect.

The experiment is also limited by the fact that a large number of architectural choices were defaulted for the two optimization methods considered. While the alternatives did test the largest architectural difference, the explicit inclusion of data fusion, choices such as the optimizer (e.g., investigating alternatives to the GA-based algorithm) could potentially result in significant changes to the convergence. These design choices also include the multifidelity surrogate architecture, method of adaptive sampling, etc.

The largest limitation of the conclusions is the decision to run the experiment using an aerodynamic surrogate instead of the coupled model. While necessary to limit the computational cost, using the surrogate necessarily decreases the fidelity of the vehicle behavior. This could also change the theoretical performance of the controller.

Figure 9.19 presents a summary of the investigation performed in order to answer Research Question 2.2.

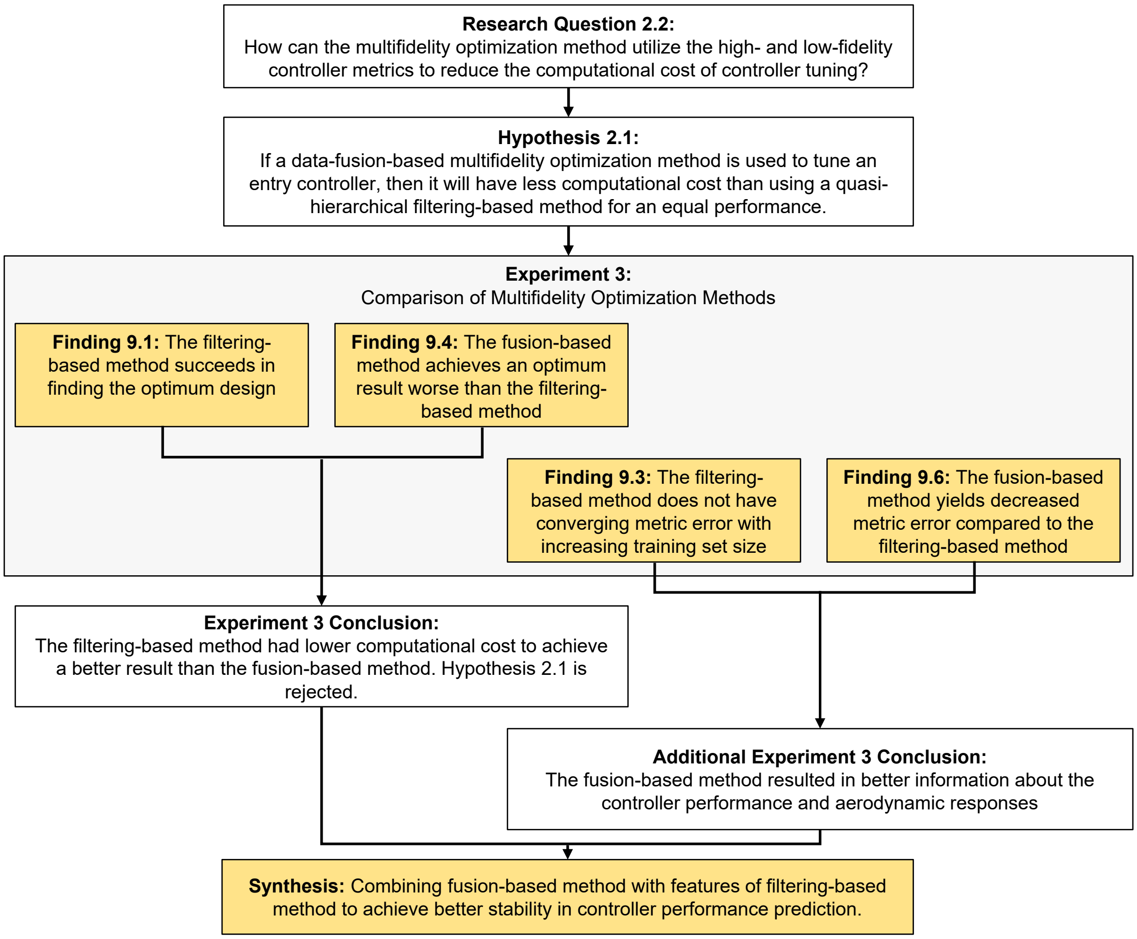

Research Question 2.2: How can the multifidelity optimization method utilize the high- and low-fidelity controller metrics to reduce the computational cost of controller tuning?
Hypothesis 2.1: If a data-fusion-based multifidelity optimization method is used to tune an entry controller, then it will have less computational cost than using a quasi-hierarchical filtering-based method for an equal performance.
Experiment 3: Comparison of Multifidelity Optimization Methods
Finding 9.1: The filtering-based method succeeds in finding the optimum design
Finding 9.4: The fusion-based method achieves an optimum result worse than the filtering-based method
Finding 9.3: The filtering-based method does not have converging metric error with increasing training set size
Finding 9.6: The fusion-based method yields decreased metric error compared to the filtering-based method
Experiment 3 Conclusion: The filtering-based method had lower computational cost to achieve a better result than the fusion-based method. Hypothesis 2.1 is rejected.
Additional Experiment 3 Conclusion: The fusion-based method resulted in better information about the controller performance and aerodynamic responses
Synthesis: Combining fusion-based method with features of filtering-based method to achieve better stability in controller performance prediction.

Figure 9.19: Summary of investigation into Research Question 2.2

CHAPTER 10
EVALUATION OF A CONTROLLER TUNED USING THE COUPLED FLIGHT MODEL (OVERARCHING RESEARCH QUESTION)

This chapter documents the testing of the Overarching Hypothesis, restated below:

Overarching Hypothesis: If the entry controller is tuned using a coupled, unsteady aerodynamic and control system model, the performance of the entry controller will be improved compared to a controller tuned using the state-of-the-art methodology.

Experiment 4 is designed as a numerical demonstration of the Overarching Hypothesis. Two entry controllers are trained: one using the state-of-the-art methodology described in Section 3.3, and one using the multifidelity methodology developed and in Chapters 7 to 9. These controllers will be simulated using the coupled CFD-RBD-FCS model developed in Chapter 5 to establish their performance at the highest level of fidelity. The experiment will also therefore demonstrate the implementation of the multifidelity controller tuning methodology.

Section 10.1 describes the controller alternatives and their method of construction. Section 10.2 documents the experiment, and Section 10.3 presents the conclusions.

10.1 Controller Alternatives

The two entry controllers will use the form described in Section 6.4, reprinted below:

$$u_z = k_{z,0}\alpha_{cmd} + k_{z,1}\alpha_{cmd}^2 + k_{z,2}\alpha_e + k_{z,3}\dot{\alpha} + k_{z,4}\alpha_{\mathcal{I}} + k_{z,5}\dot{z} + k_{z,6}\ddot{z} \tag{10.1}$$

The gains for each controller are listed in Table 10.1:

Table 10.1: Controller gains

Gain	DB-tuned	MF-tuned
$k_{z,0}$	-3.758×10^{-3}	
$k_{z,1}$	1.735×10^{-5}	
$k_{z,2}$	3.910×10^{-4}	
$k_{z,3}$	6.780×10^{-4}	6.565×10^{-4}
$k_{z,4}$	1.213	
$k_{z,5}$	-1.779×10^{-2}	
$k_{z,6}$	-3.600×10^{-4}	-1.735×10^{-4}

10.1.1 Database-Tuned Controller

The "Database-tuned" (DB-tuned) controller is named as such because its tuning process used the aerodynamic database constructed in Section 7.2. The process for tuning $k_{z,0}$, $k_{z,1}$, $k_{z,2}$, $k_{z,4}$, and $k_{z,5}$ is documented in Section 9.1. The remining gains, $k_{z,3}$ and $k_{z,6}$ (the gains for which the performance was determined to be most sensitive), were tuned using a separate Monte Carlo over the dimensional ranges established within the same section. A histogram of the OEC results for this Monte Carlo are shown in Figure 10.1, demonstrating that the selected design is likely to very close to the optimum within the design space.

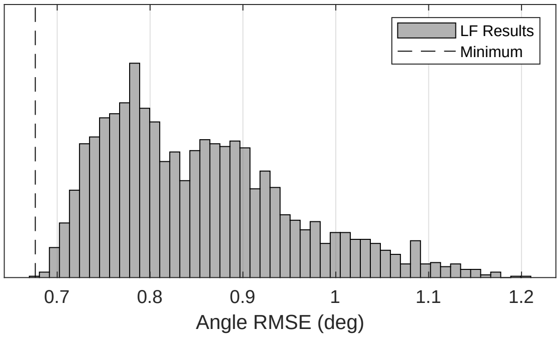


Figure 10.1: LF Monte Carlo OEC results

Since the process and model are the same, this is equivalent to tuning all of the gains concurrently. It is also important to note that, since all of the gains were tuned, this controller design is likely close to the true optimal according to the state-of-the-art methodology.

10.1.2 Multifidelity-Method-Tuned Controller

The “Multifidelity-tuned” (MF-tuned) controller is named as such because it is tuned using the multifidelity controller tuning methodology. Figure 10.2 shows the flowchart for the process as it has been developed and implemented for Experiment 4.

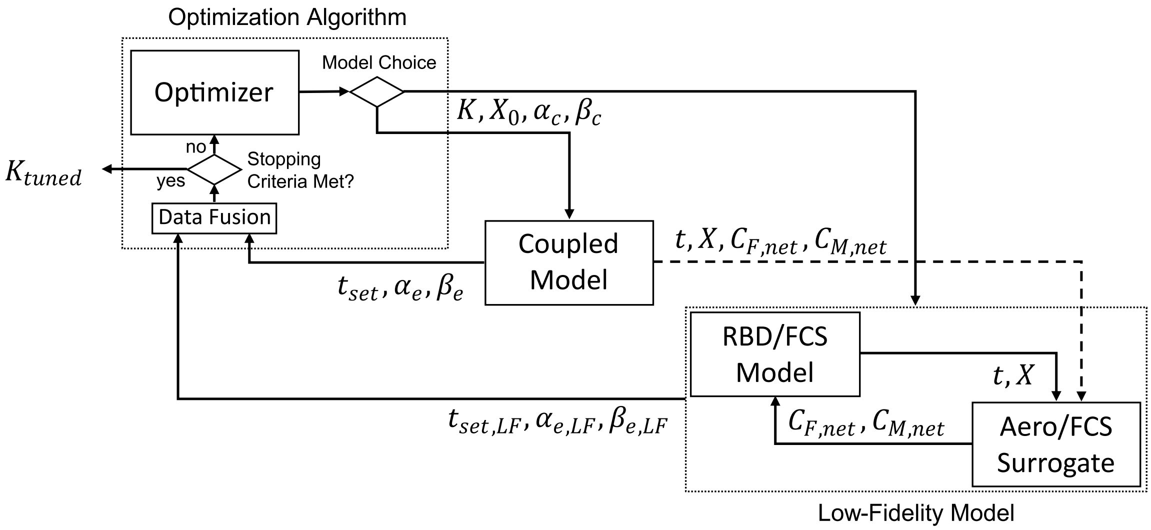


Figure 10.2: Multifidelity controller tuning process

First, a DoE was constructed to sample the design space for the generation of a set of initial coupled simulations. The DoE is listed in Table 10.2. The designs explore the dimensions of $k_{z,3}$ and $k_{z,6}$, the free controller gains, as well as spanning the initial condition cases listed in Table 9.1 The designs were chosen to include the extremes of the controller gain design space for better performance of the Kriging correction. Thus, this DoE is designed to ameliorate some of the problems noted with the Kriging correction in Experiment 3.

Table 10.2: Initial high-fidelity run conditions

$k_{z,3} \times 10^4$	$k_{z,6} \times 10^3$	Case	θ_0 (deg)	q_0 (deg/s)
5.791	3.983	1	0.902	997
-5.000	4.000	1	0.902	997
2.031	-0.117	1	0.902	997
20.00	4.000	1	0.902	997
-2.380	-1.391	5	0.209	-468
19.83	0.070	5	0.209	-468
-5.000	-2.000	5	0.209	-468
1.321	1.392	5	0.209	-468
20.00	-2.000	5	0.209	-468
20.00	4.000	6	3.24	-877
-0.053	1.913	6	3.24	-877
2.720	1.787	6	3.24	-877
20.00	-2.000	7	5.03	236
20.00	4.000	7	5.03	236

The methodology was started using the DB-tuned gains as an initial guess. This means that, in total, 18 HF simulations were available for initial training of the LF aerodynamic surrogate and the MF correction. The revised procedure for each iteration is enumerated below:

1. The available HF data is used to construct the LF aerodynamic surrogate — in this instance, the FFNN model design chosen in Experiment 2. This model is used to construct the LF model.

2. A DoE is executed using the LF model to explore the design space and generate the LF response. A FFNN surrogate model is constructed for the LF response. The MF

correction Kriging model is then fitted using the available HF data points, resulting in the MF surrogate model.

3. The GA-based optimizer finds the minimum OEC for both the LF and MF models, resulting in two predicted minimums. If the Euclidian distance between the two designs is greater than 20% of the normalized gain ranges, the LF prediction is substituted for the MF prediction. As noted in the Experiment 3 conclusions, this is designed to limit the effect of a poorly-shaped MF correction.

4. The optimization loop proceeds using the alternating exploration/exploitation described in Section 9.2.2.

   (a) For each exploitation step, the predicted minimum is executed in the coupled model at all four input cases.

   (b) For each exploration step, a DoE of 8 designs is chosen to maximize the expected improvement function. The designs are centered around the predicted minimum, with a range of $\pm 10\%$ of the normalized $k_{z,3}$ and $k_{z,6}$ ranges. Initial condition cases are chosen to favor runs with a higher predicted α_{RMSE}. For each DoE point, an input case is randomly selected. $\tilde{\alpha}_{RMSE}$ is calculated for each input case, and the probability of selecting each case is found using Equation 10.2:

$$p_i = \frac{\tilde{\alpha}_{RMSE,i} - 0.75\tilde{\alpha}_{RMSE,min}}{\sum_{j=[1,5,6,7]} \left(\tilde{\alpha}_{RMSE,j} - 0.75\tilde{\alpha}_{RMSE,min}\right)} \tag{10.2}$$

   Where $\tilde{\alpha}_{RMSE,min}$ is the minimum error:

$$\tilde{\alpha}_{RMSE,min} = \min_{j=[1,5,6,7]} \tilde{\alpha}_{RMSE,j} \tag{10.3}$$

   The adjustment by 75% of the minimum value more heavily weights the cases away from the case with the lowest error. These designs are then executed in the HF model.

The optimization was executed for six iterations, for a total of 36 coupled simulations. The resulting OEC surface is plotted in Figure 10.3. The HF points that lie on the OEC surface are shown. These points are clustered around the intersection between the regions of the OEC dominated by Cases 5 and 7, similar to the behavior found in Experiment 3. This provides validation for the definition of the controller performance metric, which was designed to ensure that the OEC would not be dominated by a single initial condition case.

Overall, the method achieved good coverage of the region around the minimum. 21 of the qureied points appear on the OEC, including 75% of the exploration points.

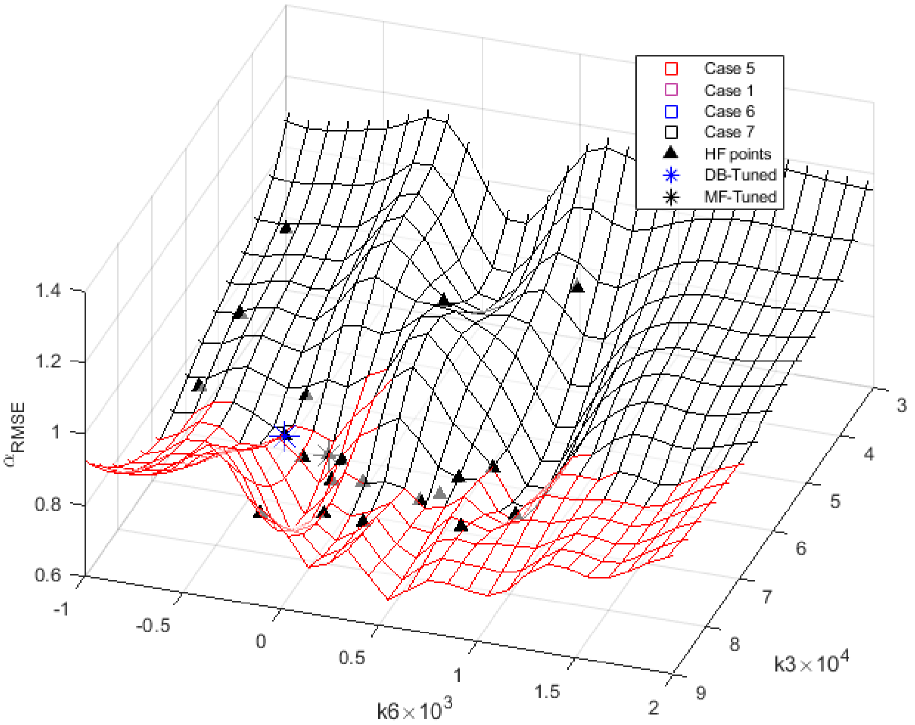


Figure 10.3: Multifidelity predicted OEC surface. The OEC responses for the DB- and MF-tuned controller designs are also indicated.

The final MF-tuned controller design is listed in Table 10.1. Recall that for the problem of interest, all gains except for $k_{z,3}$ and $k_{z,6}$ are defaulted to values found for the DB-tuned controller. Thus, the MF-tuned controller is likely not the optimum controller design as evaluated in the coupled model, when accounting for the constraints instituted in Section 9.1.

10.2 Evaluation of Controllers in Baseline and Coupled Models (Experiment 4)

10.2.1 Experimental Procedure

The experimental procedure is to execute both the DB- and MF-tuned controller designs in two flight models: the baseline model constructed according to the state of the art (i.e., using a database as the aerodynamic surrogate model), and the coupled model. Each controller is evaluated at all four initial condition cases listed in Table 9.1. Thus, the experiment requires a total of eight coupled simulations and an equal number of runs in the baseline model.

Three sets of comparisons can be made between these four sets of simulations. First, comparing the results for the DB-tuned controller as executed in both the coupled and baseline models will establish the change in performance for that controller when it is evaluated with a higher-fidelity model than the one in which it was tuned. This comparison will determine if the conclusions made about the difference in model fidelity between the baseline and coupled models for both free-flight and open-loop simulations in Experiments 1.1 and 1.2 can be extended to include closed-loop simulations.

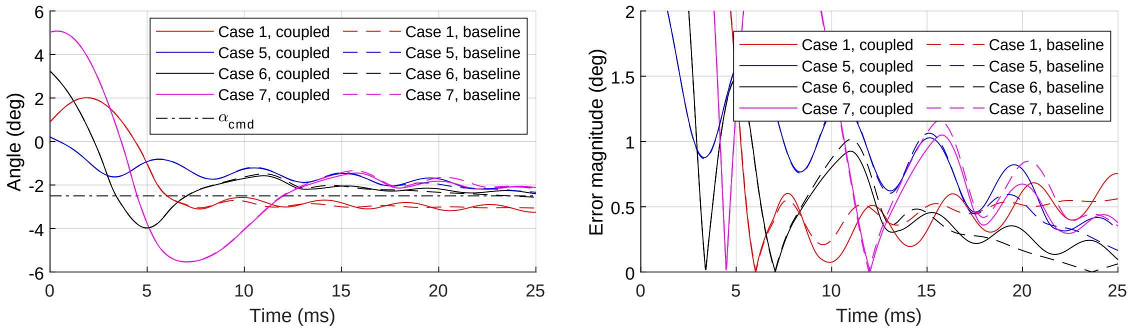
Second, comparing the performance of the DB- and MF-tuned controllers in the baseline model will verify whether the DB-tuned controller is the better controller as evaluated in that model. Since the MF-tuned controller design is not guaranteed to have been included in the Monte Carlo run in Section 10.1.1, there exists a possibility that the MF-tuned controller is simply dominant in both the baseline and coupled models, in which case no findings can be drawn from the results.

Finally, the performance of the DB- and MF-tuned controllers will be evaluated in the coupled simulation. This comparison will determine whether the multifidelity tuning process yielded a controller with improved performance over the DB-tuned controller. The first two points of comparison are necessary but not sufficient to test the Overarching Hypothesis; this point provides that final determination.

The coupled simulations required for this experiment are already completed as a result of the multifidelity tuning process documented in Section 10.1.2. The DB-tuned controller has already been run in the baseline model for Section 10.1.1, so the procedure only includes execution of the MF-tuned controller in the baseline model.

10.2.2 Comparison of Database-Tuned Controller as Evaluated in the Baseline and Coupled Models

Figure 10.4 plots the α response for the DB-tuned controller executed in both the baseline and coupled models. The qualitative behavior of the results is similar to the behavior noted in Figure 9.8: the large-amplitude oscillation due to the initial conditions is damped out, and the α response comes under control about α_{cmd} with some remaining small-amplitude oscillation at a higher frequency. The qualitative behavior of the baseline responses has some differences to the coupled responses. Notably starting around 16ms the Case 6 response appears to have no oscillatory component.



(a) Response

(b) Error to α_{cmd}

Figure 10.4: Comparison of DB-tuned controller in baseline and coupled models

Figure 10.5 plots the IMMA displacement for each set of simulations. As noted previously, all of the simulations except for Case 5 result in the IMMA traveling to its limits.

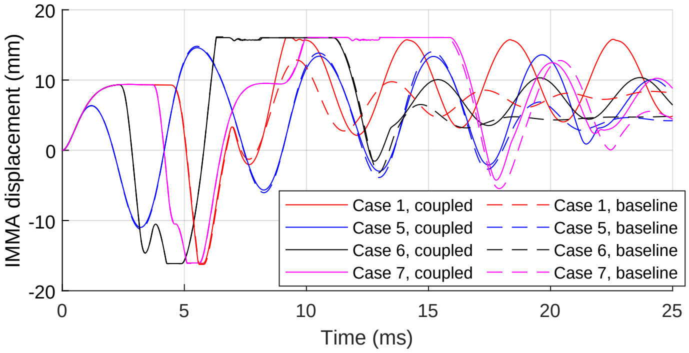


Figure 10.5: IMMA displacement of DB-tuned controller

The error between the responses predicted by the baseline model and the responses from the coupled model are plotted in Figure 10.6. The error in each case begins to grow from the start of the simulation due to error propagation, but each case shows a large increase in error magnitude starting around 8ms, which roughly corresponds to the time when the large-amplitude oscillatory motion gives way to the small-amplitude motion about α_{cmd}.

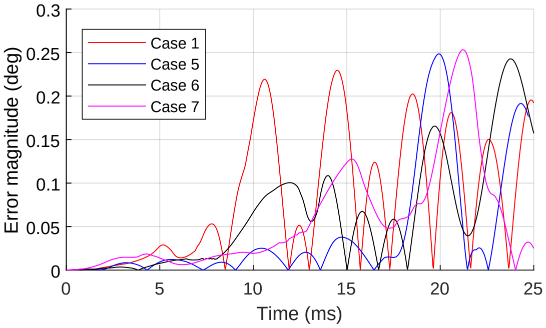


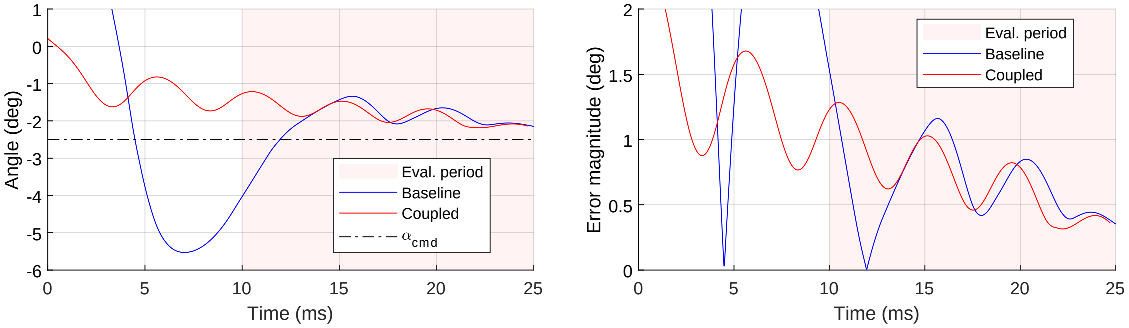
Figure 10.6: Error between baseline α prediction and coupled response for DB-tuned controller

Table 10.3 lists the controller performance metrics for each simulation, along with the difference between the baseline and coupled model responses. The results for Cases 1, 6, and 7 show a slight improvement as evaluated in the coupled model, but the results for Case 5 show a larger increase in error. The constraining runs were Case 7 and Case 5 in the baseline and coupled models, respectively. As a result, the OEC for the coupled model increased by 11.8%.

Table 10.3: Performance of DB-tuned controller in baseline and coupled models

Case	Baseline (°)	Coupled (°)	Change (°)	Change (%)
1	0.4987	0.4890	-0.0097	-1.95
5	0.6692	0.7565	0.0873	13.0
6	0.4136	0.3974	-0.0162	-3.92
7	0.6764	0.6455	-0.0309	-4.57
OEC	0.6764	0.7565	0.0801	11.8

The responses for the worst-case results are plotted in Figure 10.7.



(a) Response

(b) Error

Figure 10.7: Cases driving OEC for DB-tuned controller in baseline and coupled models

Findings

The comparison of the two preceeding sets of runs shows that the DB-tuned controller does have increased error in the coupled simulations:

Finding 10.1: The database-tuned controller exhibits a performance decrease from the baseline model prediction when evaluated in the coupled model.

This conforms with the conclusions of Experiments 1.1 and 1.2, which showed a discrepancy when simulating free flight and open-loop control, respectively[1]. However, the

[1] It is not possible to identify any findings regarding the relative magnitude magnitude of the error (as was done between the results of Experiment 1.1 and Experiment 1.2) since the results of Experiment 4 follow a significantly different trajectory.

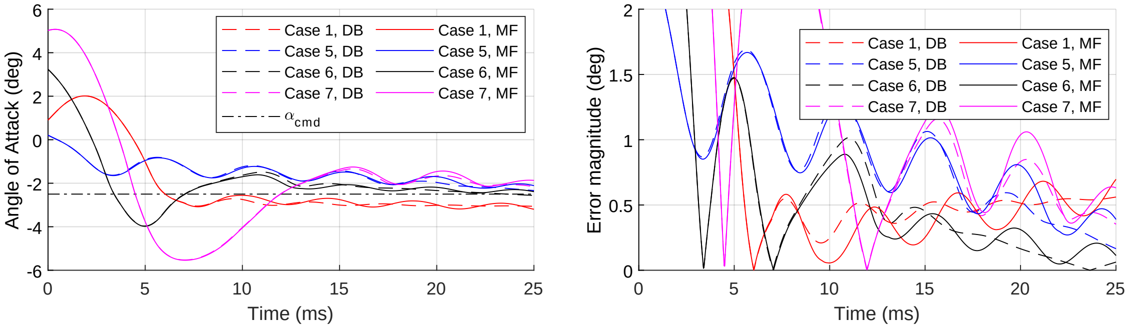
magnitude of this change is only $0.08°$, which is less than the $0.17°$ threshold of significance for tracking error of an entry vehicle with active guidance during precision landing.

The change in the case dominating the OEC between the two models serves as an indication that the construction of the problem of interest in Section 9.1 was successful in ensuring that the tuning process will require querying at more than one set of initial conditions.

It is also interesting to note that, as shown in Figure 10.6, the increase in the error between the two models occurs when the vehicle is largely under control (i.e., not during the period of greatest IMMA motion). This indicates that it is possible the unsteady aerodynamic behavior is still the largest driving in the difference in fidelity between the models, as was indicated in Experiment 1.2 for open-loop simulations.

10.2.3 Comparison of Database- and Multifidelity-Tuned Controllers in the Baseline Model

The responses for the DB- and MF-tuned controllers in the baseline model are plotted in Figure 10.8.



(a) Response

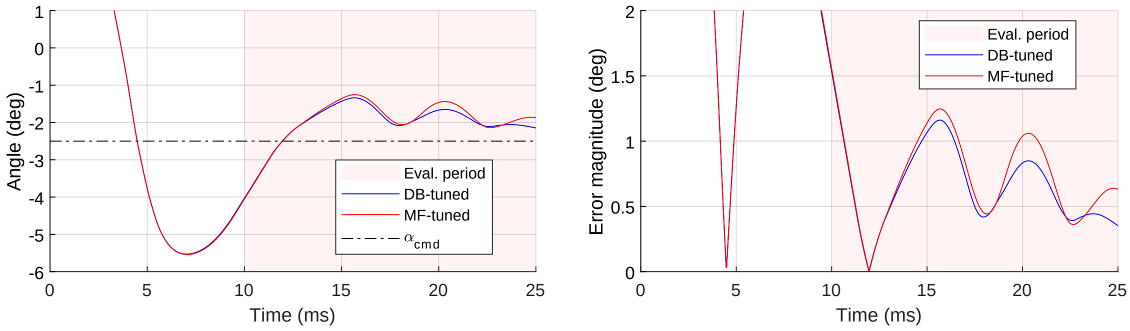
(b) Error

Figure 10.8: Comparison of database- and MF-tuned controller in baseline model

The performance results for each simulation are listed in Table 10.4, and the responses for the worst-case results are plotted in Figure 10.9. The MF-tuned controller has better performance than the DB-tuned controller for Cases 1 and 6, but worse performance for Cases 5 and 7. The worst-case results are both from Case 7, which shows a 12% increase in error for the MF-tuned controller. Figure 10.9b shows how change between the simulations increases with increasing time.

Table 10.4: Performance of DB- and MF-tuned controllers in baseline model

Case	DB-tuned (°)	MF-tuned (°)	Change (°)	Change (%)
1	0.4987	0.4915	-0.0072	-1.44
5	0.6692	0.6979	0.0287	4.28
6	0.4136	0.3565	-0.0571	-13.8
7	0.6764	0.7578	0.0814	12.0
OEC	0.6764	0.7578	0.0814	12.0



(a) Response

(b) Error

Figure 10.9: Cases driving OEC for database- and MF-tuned controller in baseline model

Findings

The MF-tuned controller was found to have worse overall performance than the DB-tuned controller:

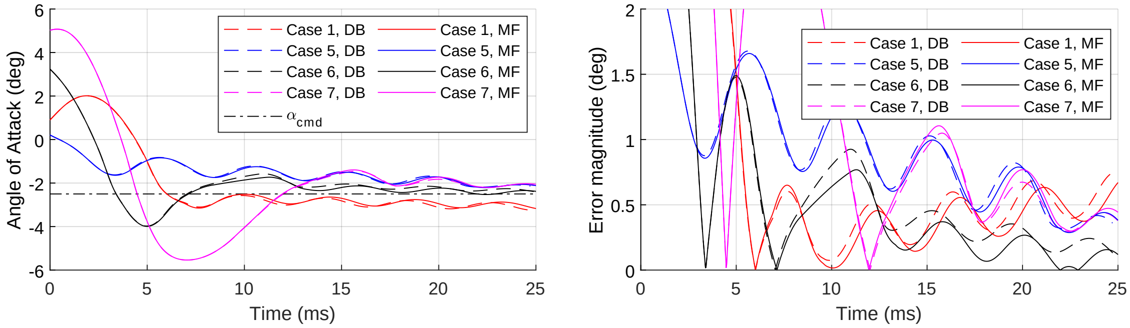
Finding 10.2: The multifidelity-tuned controller exhibits worse performance than the database-tuned controller when each is evaluated in the baseline model.

When combined with the Monte Carlo run in Section 10.1.1, the results provide further verification that the DB-tuned controller is the optimum as evaluated in the baseline model.

However, it is only necessary to prove that the DB-tuned controller does not have bad performance due to incomplete optimization.

10.2.4 Comparison of Database- and Multifidelity-Tuned Controllers in the Coupled Model

The results for the DB- and MF-tuned controllers executed in the coupled model are shown in Figure 10.10, with the performance values listed in Table 10.5.



(a) Response

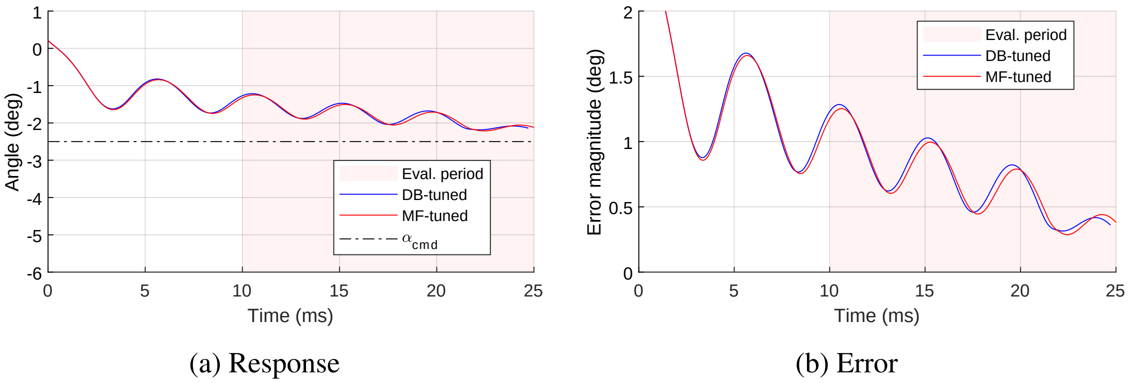
(b) Error

Figure 10.10: Comparison of database- and MF-tuned controller in coupled model

Table 10.5: Performance of DB- and MF-tuned controllers in the coupled model

Case	DB-tuned (°)	MF-tuned (°)	Change (°)	Change (%)
1	0.4890	0.4229	-0.0661	-13.5
5	0.7565	0.6907	-0.0658	-8.69
6	0.3974	0.3174	-0.0800	-20.43
7	0.6455	0.6374	-0.0081	-1.25
OEC	0.7565	0.6907	-0.0658	-8.69

The MF-tuned controller achieves a reduced error for all four input cases, with the greatest improvement in Case 6. Case 5 yielded the worst-case results for both controllers, meaning that the MF-tuned controller achieved an OEC reduction of 8.69%. The plots of the worst-case results in Figure 10.11 shows that the MF-tuned controller has a slightly reduced error across the whole range of the metric evaluation period.



(a) Response

(b) Error

Figure 10.11: Cases driving OEC for database- and MF-tuned controller in coupled model

Findings

The results show that the MF-tuned controller has reduced error compared to the DB-tuned controller:

Finding 10.3: The multifidelity-tuned controller exhibits better performance than the database-tuned controller when each is evaluated in the coupled model.

However, the magnitude of the improvement is about $0.066°$, which is less than the $0.17°$ established as the threshold for significance for precision landing.

The best controller is able to bring the α response under control within $\pm 1°$ of α_{cmd}. While no Mars entry vehicles have flown with controllers that directly control the angle of attack or sideslip, the performance is comparable in magnitude to the controllers developed by Atkins and Queen for an IMMA-equipped MPL [15]. However regardless of the performance of the controller, its capability of bringing the vehicle under control as simulated in the coupled model verifies its suitability for the problem of interest, since the test of the results is being made based on relative performance.

Since the DB-tuned controller design was used as the starting point for the optimization in Section 10.1.2, this also demonstrates that the optimization process was successful. Experiment 3 showed that the method could be successful with a lower-fidelity set of models, but Experiment 4 shows that the method is also able to utilize the coupled model:

Finding 10.4: The multifidelity controller tuning process was successful in optimizing an entry controller while operating with the coupled model.

10.2.5 Sensitivity of Results to Perturbation of Overall Evaluation Criterion

The controller performance metric (and similarly the OEC) was defined in Section 9.1 as a calculation of α_{RMSE} over the interval $t = t_i..t_f$, where $t_i = 10\text{ms}$ and $t_f = 27\text{ms}$. The t_f was selected to limit the simulation time necessary for each execution, but the t_i was selected arbitrarily (with no regard to the literature) in order to shape the OEC such that the true optimum occurred in a region that would not be dominated by one single initial condition case.

An investigation of the results was performed in order to validate the rationale that adjustment of the metric definition could result in over-sensitivity to the initial conditions (thus under-sensitivity to the α response when brought under control). The post-processing for all of the runs performed in Experiment 4 was recalculated while varying t_i from 0ms to 20ms. At $t_i = 20\text{ms}$, this leaves 7ms of simulation time for calculation of the OEC. At this point, the duration of metric calculation is smaller than two times the average period observed in the controlled closed-loop response.

No additional optimization was performed, nor were any additional coupled simulations executed. Each of the coupled controllers evaluated during the optimization were simulated in the baseline model. This resulted in a set of controller designs that have been simulated in both the coupled and baseline models, shown in Figure 10.12. Thus, neither controller will be the true optimum for any t_i except $t_i = 10\text{ms}$. Instead, they will be identified from the best database and coupled results for these designs. A trend for the performance change between the baseline and coupled models can then be found.

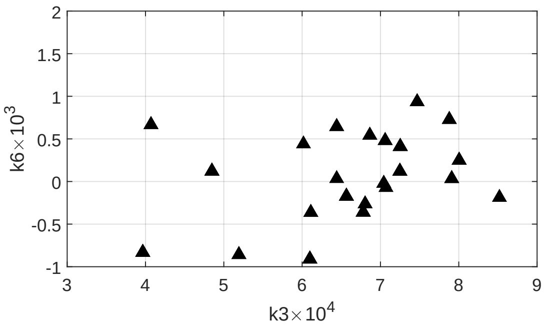


Figure 10.12: Gain designs used for OEC perturbation

The results of the perturbation are shown in Figure 10.13 in terms of the relative improvement from the identified DB-tuned to the MF-tuned controllers. The post-processing calculations were performed in increments of 1ms over the range[2]. For each of the shown OEC calculations, the existing data yielded an MF-tuned controller that had improved performance over the best DB-tuned controller. The smallest change occurred for $t_i = 0\text{ms}$ with 0.2%, and the largest change of 31.3% was found at $t_i = 20\text{ms}$. Only three different MF-tuned controller designs were found to be optimal for any calculation. “Design C” is the design found for the MF tuning process in Section 10.1.2.

The figure shows that the relative change is small from $t_i = 0\text{ms}$ to 7ms. This shows the region for which the results are dominated by the large-amplitude oscillatory motion resulting from the initial conditions. After this point, the results trend toward a plateau around 8% to 9% improvement up through $t_i = 16\text{ms}$. Starting around 17ms, the results show a sharp improvement in the performance. At $t_i = 20\text{ms}$, the best available controller selected from the coupled model results has an error reduced by about 30% from the best available controller selected in the baseline model. This could be an indication that the relative improvements identified by the use of the coupled model increase with increasing t_i. In absolute terms, the error for $t_i = 20\text{ms}$ is reduced by $0.18°$, in which case it would be significant with respect to the threshold for instantaneous tracking error. However, this can only be speculated since the optimization process was not re-executed for each OEC. The

[2] Results at $t_i = 12\text{ms}$, 13ms, and 15ms are not shown because simulation failures meant that the optimal simulation only had either Case 5 or 7 available, and thus the OEC could not be calculated

general trend of the perturbation results indicate that an OEC that is engineered to focus on the long-term error of the controller without regard for the settling time could result in more significant differences in the baseline and coupled results. However investigating this response would require additional simulation time beyond 27ms.

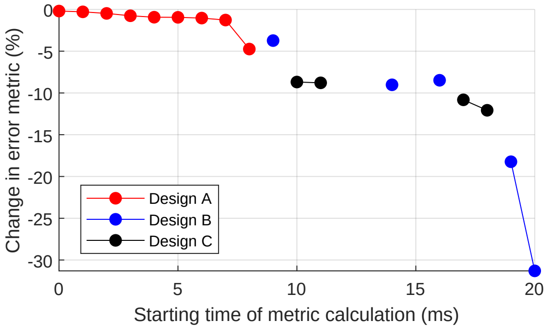


Figure 10.13: Variation of performance change vs. metric calculation

The constraining initial condition case is plotted for each metric in Figure 10.14. As expected, Case 7 is the constraint for small values of t_i because it has the largest oscillation amplitude. This gives way to OECs where Case 5, and some instances Case 1, are the constraining factor. Note that this crossover point occurs between $t_i = 8\text{ms}$ and 10ms, corresponding to the jump in performance in Figure 10.13.

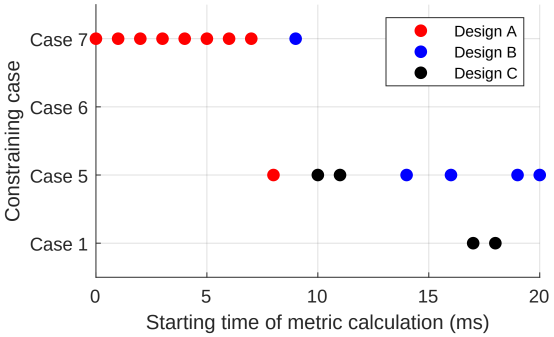


Figure 10.14: Variation of constraining input condition case vs. metric calculation

The results demonstrate that the selected controller performance metric, using $t_i = 10\text{ms}$, is outside the range dominated by the initial conditions, and that the magnitude of the results are typical for a range of metric calculation durations. Therefore, the suitability of the chosen metric and OEC are validated.

10.3 Conclusions

Finding 10.1 determined that the DB-tuned controller suffers from decreased performance when evaluated in the higher-fidelity coupled model. This provides specific, numerical evidence for the possibility that tuning a controller with the state-of-the-art methodology could result in a discrepancy due to the difference in fidelity between the models used and the source of truth, though in this instance the change was not determined to be significant.

Finding 10.3 established that the MF-tuned controller had better performance than the DB-tuned controller when evaluated in the coupled model. Since Finding 10.2 verified that the performance of the DB-tuned controller is better than the MF-tuned controller in the baseline model, the resulting improvement in the coupled model must arise from the multifidelity controller tuning process, rather than simply due to the DB-tuned controller being sub-optimal.

The progression of design knowledge can also be considered, starting from the DB-tuned controller and its performance prediction in the baseline model. This is the result that would be delivered in the state-of-the-art methodology. Validating this result in the coupled model reveals a performance that is 11.8% worse. By then applying the multifidelity controller tuning methodology, a controller design is found which recoups some of that performance loss, with a decrease in error of about 8.7% from the true value. Though the magnitudes of these changes do not meet the established threshold for significance, the methodolgy results in a measurably improved controller performance over the state of the art. There is also evidence to suggest that perturbation of the OEC could result in a significant improvement.

The Overarching Hypothesis is accepted. When the controller for the problem of interest was tuned using the multifidelity method developed from Experiment 3, the controller had better performance than one trained using the baseline methodology. Figure 10.15 presents a summary of the investigation performed through the demonstration experiment.

The limitations of the demonstration are largely due to the scope of the problem of interest. As with the conclusions from Experiment 3, the selection of a controller architecture and the defaulting of control gains limits the maximum performance that can be achieved. Furthermore, the number of coupled simulations was limited. On one hand this prevents the OEC perturbation from re-running the experiment at different definitions with optimization, limiting the conclusions there to speculation. On the other hand it means that the experiment could not determine that the MF-tuned controller was the true optimum, meaning that the multifidelity method could have potentially resulted in more improvement.

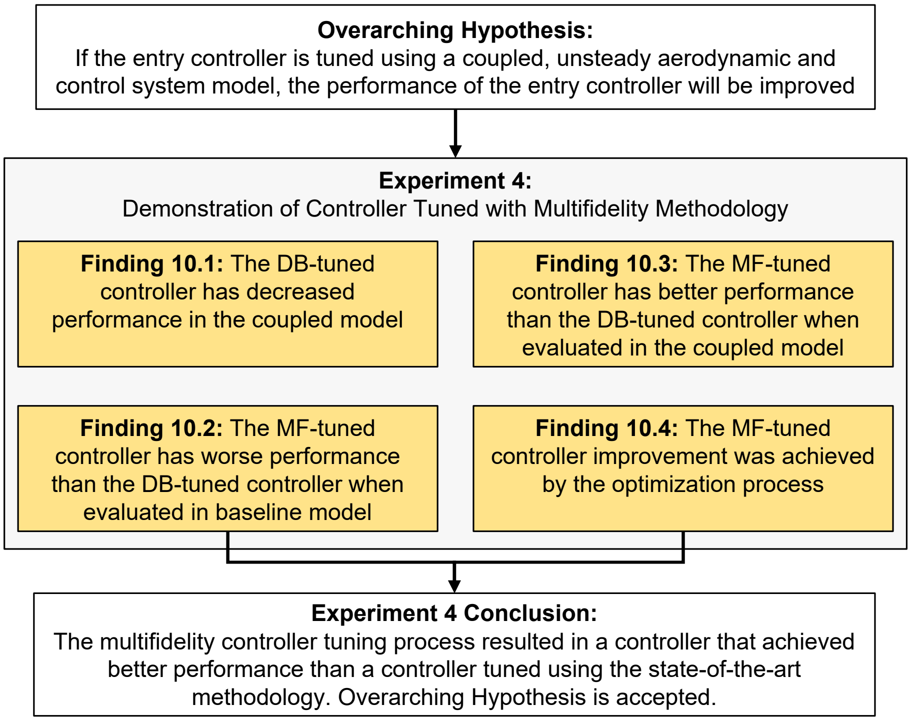

Overarching Hypothesis:
If the entry controller is tuned using a coupled, unsteady aerodynamic and control system model, the performance of the entry controller will be improved
Experiment 4:
Demonstration of Controller Tuned with Multifidelity Methodology
Finding 10.1: The DB-tuned controller has decreased performance in the coupled model
Finding 10.3: The MF-tuned controller has better performance than the DB-tuned controller when evaluated in the coupled model
Finding 10.2: The MF-tuned controller has worse performance than the DB-tuned controller when evaluated in baseline model
Finding 10.4: The MF-tuned controller improvement was achieved by the optimization process
Experiment 4 Conclusion:
The multifidelity controller tuning process resulted in a controller that achieved better performance than a controller tuned using the state-of-the-art methodology. Overarching Hypothesis is accepted.

Figure 10.15: Summary of investigation performed through Experiment 4

CHAPTER 11
CONCLUSIONS AND RECOMMENDATIONS

This chapter presents the conclusions of the dissertation. The conclusions from the numerical experiments performed in previous chapters are summarized, and their consequences to the overarching research objective are discussed. Also highlighted are the primary academic contributions from the development and implementation of this work. The chapter ends with a discussion of future work that can extend the developed methodology or leverage the primary contributions.

11.1 Summary of Conclusions

Active guidance and control during EDL is a key enabler for reducing landing uncertainty and achieving precision landing on Mars, especially with the advent of entry technologies that will invalidate the heritage of Mars entry vehicle design. However the guidance algorithm can be a double-edged sword: while it can correct for error propagation in downrange and crossrange position, it is also susceptible to uncertainty in the unsteady behavior of the aerodynamics, and interactions between the controller, the control system, the aerodynamics, and flight dynamics.

Traditional methods for controller tuning have historically relied on aerodynamic and control interaction models populated by quasi-static aerodynamic coefficients. The uncertainty model is used as a catch-all for the time-dependent effects beyond the moment damping coefficient, either from the naturally unsteady aerodynamics and control system behavior, or from the unsteady control system operation as a result of commands from the controller. While satisfactory for the common design heritage of Mars entry vehicles, this runs contrary to the desire for reduced uncertainty when anticipating the development of new vehicles with increased design and technological uncertainty.

Coupled simulation of aerodynamics, flight dynamics, and guidance and control systems allows for the unsteady interaction to be directly modeled and evaluated. Advances in computing power are constantly increasing the feasibility of running more and longer-duration multiphysics simulations. However, this cost is still incompatible with the thousands of runs necessary for the Monte Carlo simulations used to determine satisfactory controller gains.

As such, multi-fidelity optimization is a key enabler for adapting the controller tuning methodology to leverage these coupled simulations. The high-fidelity coupled simulations provide quantitative information about controller performance, while a lower-fidelity model is constructed to explore the design space. Conveniently, the byproducts of the high-fidelity model — the trajectory and aerodynamic data — can be used to populate the low-fidelity model.

The overarching research objective, which guided the problem formulation, is restated below:

Research Objective

Develop an entry controller tuning methodology that directly accounts for coupled, unsteady entry vehicle aerodynamic and control system behavior.

This overarching objective was decomposed into the following sub-objectives:

1. Develop a method for modeling the coupled, unsteady behavior of entry vehicle aerodynamics, control system, flight dynamics, and guidance and control systems.

2. Develop a method for incorporating this model into the process of entry controller tuning.

3. Quantify the change in entry controller performance between the developed methodology and a baseline representing the state of the art.

In order to achieve that overarching objective, a multifidelity controller tuning methodology was developed to feasibly enable the tuning of a controller using a model that coupled high-fidelity aerodynamics, flight dynamics, control mechanism, and guidance & control modeling. A series of numerical experiments was conducted to quantify the capability gap, inform the implementation of the methodology, and quantify its performance for the problem of interest.

Experiments 1.1 and 1.2, documented in Chapter 7, sought to verify the capabilities of the coupled model to predict the unsteady, coupled behavior exhibited by a blunt-body entry vehicle in the supersonic regime. This was achieved by running time-accurate simulations of the vehicle both in free flight (Experiment 1.1) and with open-loop control actuation (Experiment 1.2), and comparing the trajectory and aerodynamic results to the predictions made in a baseline model designed to represent the current state of the art for atmospheric entry vehicles.

Both experiments found that, for the problem of interest, the coupled model successfully modeled the unsteady, coupled behavior expected from the literature review. In gross terms, the trajectories showed deceleration, heaving in the vertical direction, and pitch oscillation subject to changes in amplitude due to aerodynamic damping. Importantly, the high-frequency behavior of the coefficients reflected the unsteady nature of the subsonic wake on the afterbody of the vehicle. Comparisons of the trajectory and aerodynamic components to the results of the baseline model showed deficiencies in the latter resulting from the inability to represent the time-dependent behavior of the flow. In particular, prediction of the effect of pitch damping — a significant concern for bluff bodies in the low supersonic regime — was found to produce a significant enough difference in the pitch response to be of potential impact on the precision of the guidance and control.

A comparison between the results of Experiments 1.1 and 1.2 found that the discrepancies between the coupled and baseline models were exacerbated for the simulations with open-loop control actuation, i.e., with the introduction of the control mechanism interaction with aerodynamics and flight dynamics. Thus the experiments provided confirmation that the use of the coupled model could be beneficial for the problem of interest. Even for a vehicle and control mechanism with a more benign interaction than is found in the literature (i.e., a vehicle with RCS), the discrepancies elicited between the coupled and baseline models were measurable and could be evaluated in terms of their significance to real-world landing precision criteria.

While Experiments 1.1 and 1.2 had demonstrated the capability of the coupled model to support design activities, they did not address the issues of feasibility inherent in the use of high-cost time-accurate CFD simulation for activities that require many flight simulations. This would require the multifidelity methodology that paired execution of the high-fidelity coupled model with a lower-fidelity model to allow inexpensive exploration of the design space. The methodology itself prompted two research questions on its implementation: the design of the low-fidelity model, and the method for using the multiple levels of fidelity available to the optimizer.

Experiment 2 sought to answer the first question: how can the low-fidelity model effectively utilize the unique data products produced by the coupled model: the new, time-accurate aerodynamic and trajectory data generated from each simulation? The experiment investigated the performance of candidate aerodynamic surrogate models under the simulated process of training that they would experience in the multifidelity tuning methodology. A quasi-linear parametric model, a feed-forward neural network (FFNN), and a time-dependent neural network (TDNN) were evaluated against the baseline aerodynamic database in terms of their error in predicting the aerodynamic and (when paired with a flight mechanics model) the trajectory response for free-flight simulations made in the coupled model.

Each surrogate model was found to have error convergence with the coupled simulation response, in cases with both interpolation and extrapolation in the flight conditions. General characteristics were determined about the relative performance of the surrogate models.

The neural-network-based models, being non-parametric, were found to have superior error convergence in the aerodynamic predictions than the parametric quasi-linear and database models, given a large enough set of training data. However, it was noted that with an insufficient set of training data, the neural networks suffered from increased error in the trajectory predictions due to extrapolation. Since, the pitch response was of paramount importance for the problem of interest, the surrogate with the best performance for that response, the FFNN, was selected for implementation into the methodology. This model had the best balance between stability in terms of error propagation (a problem noted for the TDNN) and leveraging the non-parametric nature. Subsequent evaluation of the FFNN through its use in a repetition of Experiments 1.1 and 1.2 verified its suitability in both free-flight and open-loop simulations for the problem of interest.

Experiment 3 was designed to answer the second of the research questions related to the methodology of controller tuning: can the methodology effectively utilize the results of two flight models — each with a different level of fidelity — to reduce the cost of controller tuning? Expressed in another way: given a fixed number of optimization operations, which method would result in the better controller? The controller tuning process was executed using two alternatives of the methodology: a filtering-based method which relied on the quasi-hierarchical relationship between the two models (a relationship which was validated in Experiment 2) to passively improve the accuracy of the controller performance prediction, and a fusion-based method which combined the controller performance responses from the high- and low-fidelity models to further correct the prediction .

The experiment found that both alternatives were able to tune the PID-based controller selected for the problem of interest. The filtering-based method was determined to have yielded a better result with fewer iterations than the fusion-based method. However, this was found to be the result of the random nature of sampling, not due to the behavior of the filtering-based method. The fusion-based method actually yielded a more accurate understanding of the controller gain design space within the region most heavily sampled by both methods. In addition, the fusion-based model also yielded a more accurate aerodynamic surrogate model. Based on these results, a synthesis of the two alternatives was used for implementation into the methodology: one which leveraged the ability of the fusion-based method to explore the design space, but which had the stability of the filtering-based method to account for sensitivity of the data fusion.

Based on the findings of Experiments 2 and 3, the components of the multifidelity controller tuning methodology were identified and selected. This enabled the evaluation of the methodology with respect to the overarching research objective.

Experiment 4 served as a demonstration of the methodology using the problem of interest. Two entry controllers were tuned for the vehicle: the “DB-tuned” controller tuned using the state-of-the-art methodology (using an aerodynamic database and a filtering-based method to identify the best design), and the “MF-tuned” controller tuned using the developed methodology. Each of these controllers was evaluated in both the coupled and baseline models to compare their relative performance. The experiment found that, for the selected performance requirement, the DB-tuned controller suffered from a decrease in performance from the prediction made in the baseline model to the results of the coupled simulation (thus extending the conclusions of Experiments 1.1 and 1.2 for simulations with closed-loop control actuation). This performance loss was ameliorated by the MF-tuned controller.

Since the multifidelity controller tuning methodology uses the coupled model, the methodology can be said to directly account for coupled, unsteady aerodynamic and control system behavior that is exhibited by the IMMA-equipped SIAD vehicle. Furthermore, the results of Experiment 4 show that the methodology yielded measurably better results than the state-of-the-art methodology. Therefore, the overarching research objective has been achieved, when applied to the problem of interest.

Beyond the conclusions reached within the bounds of the overarching research objective and the assumptions laid out in Section 4.1, further discussion can be made for two points related to significance in real-world applications.

First, where the results of Experiments 1.1, 1.2, 2, and 4 can be directly evaluated against the threshold for tracking error significance established from literature, only in Experiments 1.2 and 2 was this threshold met. The perturbations in OEC performed alongside Experiment 4 indicate that the results may be significant for evaluations that deemphasize the initial conditions, but there is not enough evidence to prove that this is the case. However, if the complexity of the vehicle had resulted in a greater difference between the coupled and baseline models, there is no indication that the developed methodology would not be capable of converting that into performance benefits for an MF-tuned controller.

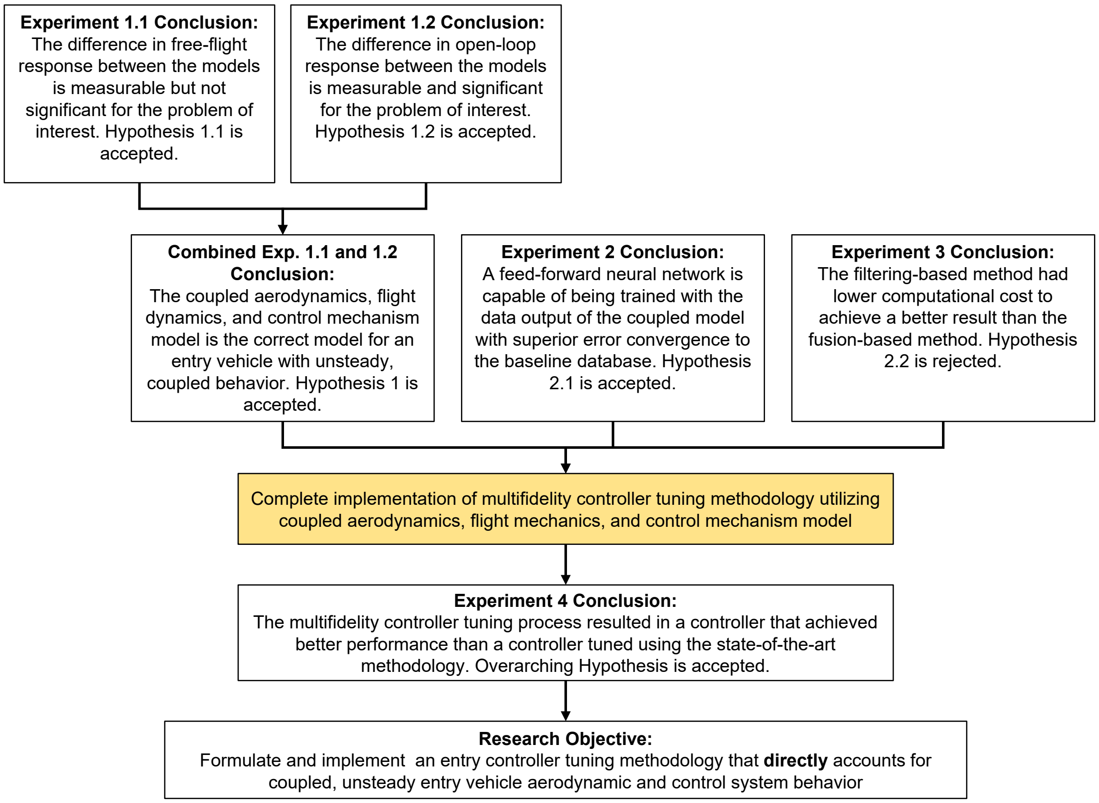
The second point is computational cost. The number of time-accurate CFD simulations used in the demonstration of the methodology can be compared to the change in performance from the state of the art. From an engineering standpoint, using the methodology to tune a controller for the vehicle of interest is feasible but not affordable. However, the results of the experimentation provide a quantified starting point using a vehicle with a relatively benign aerodynamic/control mechanism interaction. This indicates that the methodology can be utilized for vehicles with more complex interactions, such as entry vehicles with active RCS.

Figure 11.1 presents a summary of the experimental conclusions used to develop the methodology, and the demonstration that verifies that the methodology meets the Overarching Research Objective.

11.2 Contributions

The primary contributions resulting from the development of the technical approach and execution of the experiments are listed below:

1. This work provided the first demonstration of the POST2/FUN3D coupled flight model environment executing a simulation with trajectory-propagator-centric guidance and control modeling. The improved functionality beyond CFD-RBD simula-



Experiment 1.1 Conclusion:
The difference in free-flight response between the models is measurable but not significant for the problem of interest. Hypothesis 1.1 is accepted.

Experiment 1.2 Conclusion:
The difference in open-loop response between the models is measurable and significant for the problem of interest. Hypothesis 1.2 is accepted.

Combined Exp. 1.1 and 1.2 Conclusion:
The coupled aerodynamics, flight dynamics, and control mechanism model is the correct model for an entry vehicle with unsteady, coupled behavior. Hypothesis 1 is accepted.

Experiment 2 Conclusion:
A feed-forward neural network is capable of being trained with the data output of the coupled model with superior error convergence to the baseline database. Hypothesis 2.1 is accepted.

Experiment 3 Conclusion:
The filtering-based method had lower computational cost to achieve a better result than the fusion-based method. Hypothesis 2.2 is rejected.

Complete implementation of multifidelity controller tuning methodology utilizing coupled aerodynamics, flight mechanics, and control mechanism model

Experiment 4 Conclusion:
The multifidelity controller tuning process resulted in a controller that achieved better performance than a controller tuned using the state-of-the-art methodology. Overarching Hypothesis is accepted.

Research Objective:
Formulate and implement an entry controller tuning methodology that **directly** accounts for coupled, unsteady entry vehicle aerodynamic and control system behavior

Figure 11.1: Summary of experimental conclusions

tion serves to demonstrate the usefulness of the environment for modeling problems of interest in atmospheric entry.

2. An evaluation made into aerodynamic surrogate alternatives for entry vehicles found that models without prescribed form achieved better trajectory and aerodynamic results, especially in pitch damping. The use of the surrogate models themselves is possible, but this also provides evidence that further advantage can be made by effectively applying the methodology for a design activity that is less complex than controller tuning.

3. Related to the construction of the aerodynamic surrogates, the work demonstrated the sampling of a flight condition design space using combined free-flight, open-loop, and closed-loop time-accurate simulations for entry vehicles.

4. A comparison was established between state-of-the-art flight simulation method and coupled CFD-RBD-FCS model for entry vehicles with IMMA control mechanisms. The measurement of the differences serves as a baseline for investigating more complex control mechanism interaction (e.g., RCS).

5. The execution of the methodology demonstrated the adaptation of design activities to enable the use of high-cost coupled simulation where it would previously have been infeasible. This result furthers the goal of applying computational multiphysics simulation laid out in the CFD Vision 2030 [106].

11.3 Recommendations for Future Research

The previously-discussed findings, conclusions, and limitations of the experimental campaign form the basis of recommendations for future The coupled model and multifidelity controller tuning methodology, in their current iteration, represent a minimum viable product whose use has only been established for the problem of interest. As such, the recommendations for future work focus on tasks that can expand the scope and utility of these two elements.

Evaluation of a Vehicle Exhibiting More Complex Behavior

A limitation noted in many of the experiments was the small magnitude in difference between the coupled and baseline models. This is understood to be the result of the vehicle and control mechanism selection for the problem of interest. A logical continuation of the work presented in this dissertation would be to investigate the application of the methodology to a vehicle with more complex interactions between aerodynamics, control mechanism, and flight dynamics. As noted in Section 6.1, the flat afterbody of the SIAD results in benign separation and attachment behavior, where a more traditional conical afterbody would excite more complex dynamic attachment behavior. All currently proposed Mars entry vehicles, as well as many Earth entry vehicles, use RCS for directional control and oscillation damping. As found in the literature, the interaction between RCS plumes and the fluid flow around the vehicle can result in complex behavior. As such, an RCS-equipped vehicle could have a greater deficiency between the state-of-the-art modeling and the use of coupled CFD-RBD-FCS models. Thus, an entry controller could have the potential to be further improved by the multifidelity controller tuning methodology. This work would also necessitate the investigation of new surrogate model alternatives that are compatible with the RCS control mechanism state.

Refinement of Multifidelity Data Fusion and Optimization Methods

As noted in the Experiment 3 conclusions, in addition to the binary investigation made in Experiment 3 of whether to use data fusion, the optimization methodology presents other design choices that can be investigated. Even within the architecture proposed, there are more design choices that can be explored: variation of the number of high- and low-fidelity model executions, the choice in exploration/exploitation of the design space, or evaluation based on minimizing error in the aerodynamic responses in addition to the controller metric response. Other multifidelity corrections can also be investigated; resolving the noted deficiencies in the implementation of the Kriging-based corrector could increase the cost of the optimization process. Beyond the architecture of the developed methodology, further investigation can be made into other optimizers and multifidelity surrogate alternatives.

Application to Aerodynamic Surrogate Modeling Beyond Scalar Coefficients

As noted in the Experiment 2 conclusions, the chosen aerodynamic surrogate models were limited to architectures that predicted the scalar aerodynamic coefficients. This scope was applied to the alternatives in order to enable full comparison with the baseline aerodynamic database. The true results of the coupled model — a fully converged flow volume at each time step — were reduced to at most six coefficients. However, without this limitation, the coupled model can be used to populate more complex reduced-order model forms. For example, ROMs with proper orthogonal decomposition (POD) can be used to construct a surrogate of the pressure and shear distribution over the surface of the vehicle, preserving the spatial relationships from the flow solution. The ROMs can be used to reconstruct coherent structures , giving a much more physically relevant representation of flow phenomena such as attachment/detachment, or impingement/entrainment due to RCS.

Leveraging Advanced Control Systems

As noted in the conclusions for Experiments 3 and 4, the controller architecture selected for the problem of interest, while capable of controlling the vehicle, is not the optimum. Beyond allowing all of the controller gains to be tuned, different controller architectures can potentially provide more room for improvement and thus more significant differences compared to a DB-tuned controller.

The motivation for the multifidelity tuning methodology lies in the need to reconcile the high cost of the coupled model with the large number of simulations required by the state-of-the-art methodology for tuning the chosen controller architecture. However, more advanced control systems may both inherently require fewer simulations, and have the potential to extract more information out of each coupled simulation. Using an architecture like model predictive control [169] could be tuned in a way that is more automatic and less dependent on the expertise of controls engineers. Furthermore, adaptive control systems could leverage the unique ability of the coupled simulation to run a high-fidelity simulation with no *a priori* aerodynamic model. As an example, the methodology behind the Learn to Fly development effort [239] could be extended, allowing the entry vehicle to determine its own excitation of the control system in order to identify the system.

APPENDIX A

DETAILED COUPLED MODEL ARCHITECTURE

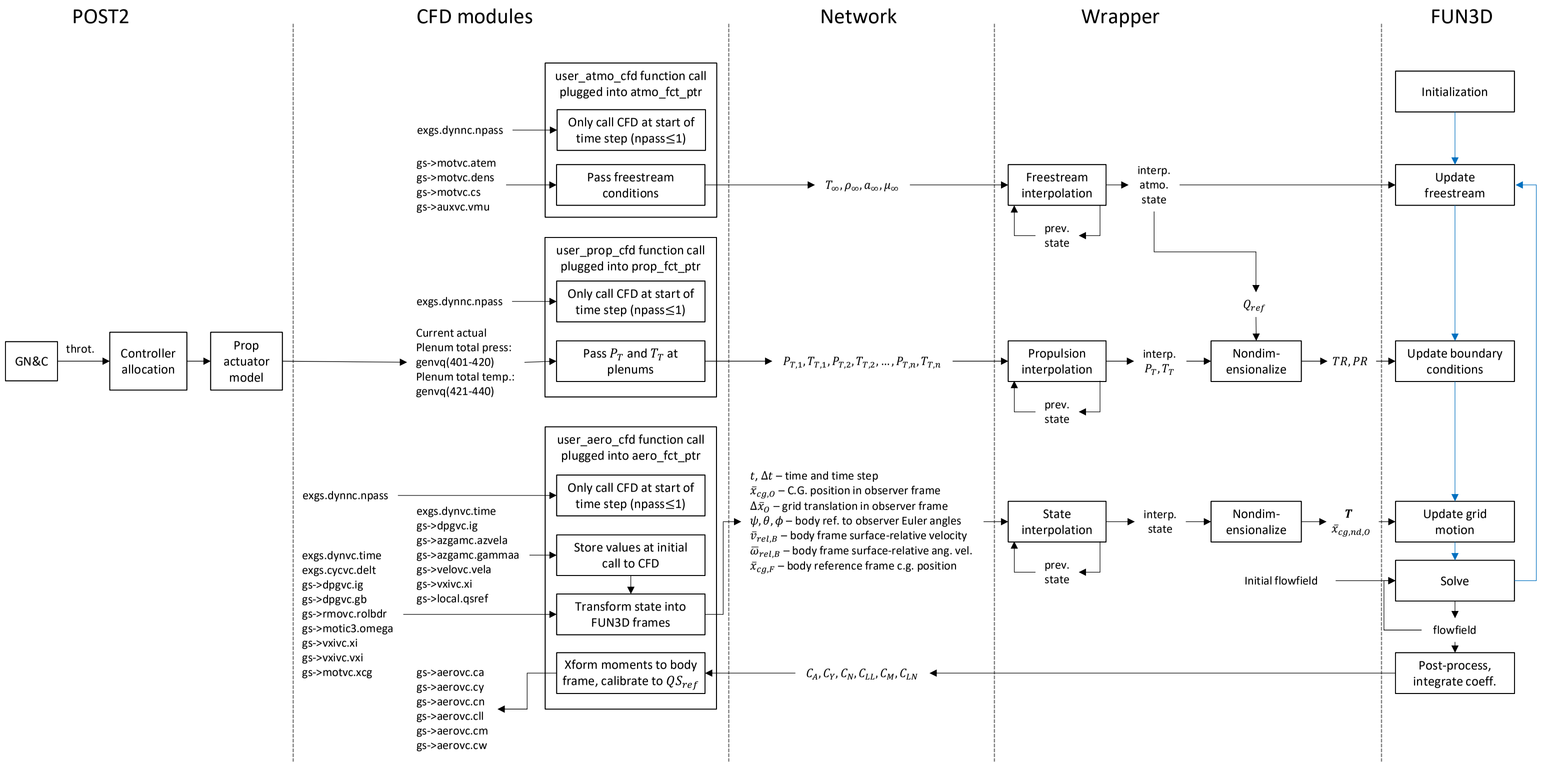

POST2
CFD modules
Network
Wrapper
FUN3D
GN&C
throt.
Controller allocation
Prop actuator model
user_atmo_cfd function call plugged into atmo_fct_ptr
exgs.dynnc.npass
Only call CFD at start of time step (npass≤1)
gs->motvc.atem
gs->motvc.dens
gs->motvc.cs
gs->auxvc.vmu
Pass freestream conditions
$T_\infty, \rho_\infty, a_\infty, \mu_\infty$
user_prop_cfd function call plugged into prop_fct_ptr
exgs.dynnc.npass
Only call CFD at start of time step (npass≤1)
Current actual
Plenum total press:
genvq(401-420)
Plenum total temp.:
genvq(421-440)
Pass P_T and T_T at plenums
$P_{T,1}, T_{T,1}, P_{T,2}, T_{T,2}, \dots, P_{T,n}, T_{T,n}$
user_aero_cfd function call plugged into aero_fct_ptr
exgs.dynnc.npass
Only call CFD at start of time step (npass≤1)
exgs.dynvc.time
gs->dpgvc.ig
gs->azgamc.azvela
gs->azgamc.gammaa
gs->velovc.vela
gs->vxivc.xi
gs->local.qsref
Store values at initial call to CFD
exgs.dynvc.time
exgs.cycvc.delt
gs->dpgvc.ig
gs->dpgvc.gb
gs->rmovc.rolbdr
gs->motic3.omega
gs->vxivc.xi
gs->vxivc.vxi
gs->motvc.xcg
Transform state into FUN3D frames
$t, \Delta t$ – time and time step
$\vec{x}_{cg,O}$ – C.G. position in observer frame
$\Delta\vec{x}_O$ – grid translation in observer frame
ψ, θ, ϕ – body ref. to observer Euler angles
$\vec{v}_{rel,B}$ – body frame surface-relative velocity
$\vec{\omega}_{rel,B}$ – body frame surface-relative ang. vel.
$\vec{x}_{cg,F}$ – body reference frame c.g. position
Xform moments to body frame, calibrate to QS_{ref}
gs->aerovc.ca
gs->aerovc.cy
gs->aerovc.cn
gs->aerovc.cll
gs->aerovc.cm
gs->aerovc.cw
$C_A, C_Y, C_N, C_{LL}, C_M, C_{LN}$
Freestream interpolation
prev. state
interp. atmo. state
Q_{ref}
Propulsion interpolation
prev. state
interp. P_T, T_T
Nondim-ensionalize
TR, PR
State interpolation
prev. state
interp. state
Nondim-ensionalize
$\boldsymbol{T}$
$\vec{x}_{cg,nd,O}$
Initial flowfield
Initialization
Update freestream
Update boundary conditions
Update grid motion
Solve
flowfield
Post-process, integrate coeff.

Figure A.1: Detailed flowchart of the coupled model implementation, including the POST2 variables used

APPENDIX B
INPUT FILE EXAMPLES

The following is an example of the input files used to execute a coupled simulation. Note that the input file is configured for a 2-axis IMMA, though in all of the simulations the y-axis mass was not actuated. The examples shown are for initial condition case 7 from Table 7.1.

B.1 POST2 Input Files

B.1.1 POST2 .inp file

This is the main POST2 input file. It includes variables that are not changed with different initial condition cases. It also models the controller and control mechanism.

```
srchm = 0,              //no search or optimization
ioflag = 3,             //use metric input//metric output
ipro = -1,              //print final trajectory only
title='SIAD IMMA',

*declare CONST_G 9.80665,         //gravitational accel, m/s^2
*declare CONST_SOS 343.349754,    //speed of sound, m/s
*declare CONST_ATMDENS 0.269,     //freestream density, kg/m^3
*declare CONST_ATMTEMP 292.8,     //freestream temperature, K

*declare INP_MACH 2.1,            //initial Mach number
*declare INP_TMAX 0.06,           //maximum simulation time, s
*declare INP_DT 2.35381473e-05,   //physical time step, s
*declare INP_XAEROD 0.1577,       //aerodynamic reference
                                  //point x-position/d
*declare INP_D 0.03556,           //body diameter, m
*declare INP_MB 0.08727384,       //body mass, kg
*declare INP_IXXB {5.52500E-06},  //body inertia components
*declare INP_IYYB {3.00700E-06},
*declare INP_XCGD 0.1577,         //body xcg/d
*declare INP_MI 0.00229668,       //IMMA mass
```

```
*declare INP_XCGY 0.1527,        //y-axis mass xcg/d
*declare INP_XCGZ 0.1627,        //z-axis mass xcg/d
*declare INP_LIM 0.016,          //movement limit of imma.
*declare INP_IXXM 6.7352472e-09, //imma inertia
*include ’SIADimma.dat’

*declare TRIM_ON 1,         //turn on imma trim
*declare CONTROLLER_ON 1,   //turn on imma controller
*declare INTER_MODEL_ON 1,  //imma interaction model
*declare OPEN_LOOP_ON 0,    //drive imma by open-loop
*declare AERO_ON 1,         //turn on aero

event = 1,              //event or phase number
fesn = 100,             //final event number
maxtim = {INP_TMAX},    //maximum allowable time
dt = {1 * INP_DT},      //timestep (s)
altmax = 5.0e6,         //maximum allowable altitude
pinc = {1 * INP_DT},    //print interval
prnc = {1 * INP_DT},    //plot interval, 0=dt
npc(2) = 1,             //Integrator, 1=RK4
npc(19) = 1,            //print input conditions
npc(24) = 1,            //activate generalized integral vars
CIXXD = 1,              //full effect of inertia
                        //derivative on c.o.a.m.
PRNT(97) = AMXB,AMYB,AMZB,CM,CLL,CW,CMQ,
           XL,YL,ZL,VXL,VYL,VZL,ROLI,YAWI,PITI,OMGBD,
           IXX,IYY,IZZ,IXY,IXZ,IYZ,IXXD,IYYD,IZZD,IXYD,
           IXZD,IYZD,XCG,YCG,ZCG,DXCG,DYCG,DZCG,
           GENVQ1,GENVQ2,GENVQ3,GENVQ4,GENVQ5,GENVQ6,
           GENVQ7,GENVQ8,GENVQ9,GENVQ11,GENVQ12,GENVQ13,
           GENVQ14,GENVQ15,GENVQ16,GENVQ21,GENVQ22,GENVQ23,
           GENVQ24,GENVQ25,GENVQ26,GENVQ31,GENVQ32,
           GENVQ100,GENVQ101,GENVQ102,GENVQ103,GENVQ104,
           GENVQ105,GENVQ106,GENVQ110,GENVQ111,GENVQ112,
           GENVQ113,GENVQ114,GENVQ115,GENVQ116,
           GINT1,GINT2,GINT3,GINT4,GINT5,GINT6,
           TMXBENG1,TMXBENG2,TMYBENG1,TMYBENG2,TMZBENG1,
           TMZBENG2,THRFX1,THRFX2,THRFX3,THRFX4,THRFY1,
           THRFY2,THRFY3,THRFY4,THRFZ1,THRFZ2,THRFZ3,THRFZ4,
           TMXB,TMYB,TMZB,PSTOP//variables to be printed

c Input environmental characteristics
npc(5) = 1,                              //atmo model
prest = pres constant 0 2.2584809E+04,   //pressure, Pa
atemt = atem constant 0 {CONST_ATMTEMP}, //atmo temp, K
```

```
cst = cs constant 0 {CONST_SOS},         //sonic speed, m/s
denst = dens constant 0 {CONST_ATMDENS}, //density, kg/m^3
atmosk(1) = 4.026265490842E+02,          //constant 1
atmosk(2) = 3.487441492199E-03,          //constant 2

c Mass Properties
npc(30) = 4, //N-stage weight calculation
nstpl = 1,   //(we have just one stage)
nstph = 3,
wstpd(1) = {INP_MB * CONST_G}, //body
xcgs1t = 'xcgs1t',constant,0,{INP_XCGD * INP_D},
ixxs1t = 'ixxs1t',constant,0,{INP_IXXB},
iyys1t = 'iyys1t',constant,0,{INP_IYYB},
izzs1t = 'izzs1t',constant,0,{INP_IYYB},
wstpd(2) = {INP_MI * CONST_G}, //y-axis mass
xcgs2t = 'xcgs2t',constant,0,{INP_XCGY * INP_D},
ycgs2t = 'ycgs2t',monovar,gint2,0,lin_inp,noxt,
{-INP_LIM},{-INP_LIM},{INP_LIM},{INP_LIM},
zcgs2t = 'zcgs2t',constant,0,0.0,
ixxs2t = 'ixxs2t',constant,0,{INP_IXXM},
iyys2t = 'iyys2t',constant,0,{INP_IXXM},
izzs2t = 'izzs2t',constant,0,{INP_IXXM},
wstpd(3) = {INP_MI * CONST_G}, //z-axis mass
xcgs3t = 'xcgs3t',constant,0,{INP_XCGZ * INP_D},
ycgs3t = 'ycgs3t',constant,0,0.0,
zcgs3t = 'zcgs3t',monovar,gint4,0,lin_inp,noxt,
{-INP_LIM},{-INP_LIM},{INP_LIM},{INP_LIM},
ixxs3t = 'ixxs3t',constant,0,{INP_IXXM},
iyys3t = 'iyys3t',constant,0,{INP_IXXM},
izzs3t = 'izzs3t',constant,0,{INP_IXXM},

c controller equations
*declare INP_CMDMAX {2 * INP_LIM} //maximum commanded zcmd
*declare INP_DB 0.5, //deadband for aoa error
*declare Kz00 -0.003758128376633,
*declare Kz01 1.735238577535138e-05,
*declare Ky00 0.0,
*declare Ky01 0.0,

genveqn(1) = "alpha - CMD_A"    //aoa error
genveqn(2) = "beta - CMD_B"    //beta error
gderv(5) = "genvq1"            //alpha error integral
gderv(6) = "genvq2"            //beta error integral
//state is aerr, berr, adot, bdot, aint, bint, dz, dy
```

```
*if def TRIM_ON
    //commanded trim position of masses
    genveqn(11) = "Kz00*CMD_A + Kz01*sign(CMD_A*CMD_A,CMD_A)"
    genveqn(21) = "Ky00*CMD_B + Ky01*sign(CMD_B*CMD_B,CMD_B)"
*endif
*if def CONTROLLER_ON
    \\building controller gain states
    genveqn(12) = "Kz02*genvq1"
    genveqn(13) = "Kz03*alpdot"
    genveqn(14) = "Kz04*gint5"
    genveqn(15) = "Kz05*genvq116"
    genveqn(16) = "Kz06*genvq115/1000"

    genveqn(22) = "Ky02*genvq2"
    genveqn(23) = "Ky03*betdot"
    genveqn(24) = "Ky04*gint6"
    genveqn(25) = "Ky05*genvq106"
    genveqn(26) = "Ky06*genvq105/1000"
*endif

*if def OPEN_LOOP_ON
    \\commanding open-loop IMMA actuation
    genveqn(3) = "(time >= 0.001)? -0.016 : 0.0"
    genveqn(4) = "(time >= 0.010)? -0.016 : genvq3"
    genveqn(5) = "(time >= 0.035)? -0.016 : genvq4"
    genveqn(7) = "genvq5"
*else
    //if z mass is around trim position
    genveqn(31) = "(abs(gint4-genvq11)<=(0.4*INP_LIM))? 1:0"
    //if alpha error is within deadband
    genveqn(32) = "(abs(genvq1)<=INP_DB)? 1 : 0"
    //z cmd ramp for non-trim terms
    genveqn(3) = "(genvq31*genvq32>0)? 1 : min(max(0.1/(
                  abs(genvq1)-0.4) - 0.025*abs(genvq1)
                  + 0.0125,0.0),1.0)"
    genveqn(4) = "1" //y cmd ramp for non-trim terms
    //commanded z-mass position
    genveqn(5) = "genvq11 + genvq3*(genvq12 + genvq13
                  + genvq14 + genvq15 + genvq16)"
    //commanded y-mass position
    genveqn(6) = "genvq21 + genvq4*(genvq22 + genvq23
                  + genvq24 + genvq25 + genvq26)"
    //limiting maximum commanded values
    genveqn(7)="sign(1.0,genvq5)*min(abs(genvq5),INP_CMDMAX)"
    genveqn(8)="sign(1.0,genvq6)*min(abs(genvq6),INP_CMDMAX)"
```

```
*endif

c IMMA equations
//force on y-imma
genveqn(100) = "(INP_K * (genvq8 - gint2) - INP_C * gint1)"
genveqn(101) = "genvq100 / INP_MI"     //ddy ideal
//is y-mass at the limit?
genveqn(102) = "(abs(gint2) >= INP_LIM)? 1 : 0"
//do the signs of ddy and y match?
genveqn(103)="(sign(1.0,genvq101)*sign(1.0,gint2) > 0)? 1:0"
//do the signs of dy and y match?
genveqn(104) = "(sign(1.0,gint1) * sign(1.0,gint2) > 0)? 1:0"
genveqn(105) = "(genvq102 * genvq103 > 0)? 0 : genvq101"
genveqn(106) = "(genvq102 * genvq104 > 0)? 0 : gint1"
gderv(1) = ’genvq105’ // integrate to get dy
gderv(2) = ’genvq106’ // integrate to get y

//force on z-imma
genveqn(110) = "(INP_K * (genvq7 - gint4) - INP_C * gint3)"
genveqn(111) = "genvq110 / INP_MI" //ddz ideal
//is z-mass at the limit?
genveqn(112) = "(abs(gint4) >= INP_LIM)? 1 : 0"
//do the signs of ddz and z match?
genveqn(113) = "(sign(1.0,genvq111)
                * sign(1.0,gint4) > 0)? 1 : 0"
//do the signs of dz and z match?
genveqn(114) = "(sign(1.0,gint3)
                * sign(1.0,gint4) > 0)? 1 : 0"
genveqn(115) = "(genvq112 * genvq113 > 0)? 0 : genvq111"
genveqn(116) = "(genvq112 * genvq114 > 0)? 0 : gint3"
gderv(3) = ’genvq115’ // integrate to get dz
gderv(4) = ’genvq116’ // integrate to get z

c Engines to model internal reaction forces
npc(9) = 1, //finite-burn n-engine prop
neng = 4,   //number of engines
iengmf(1) =  1 1 1 1,
iwdf(1) =    1 1 1 1,
ithrvec(1) = 1 1 1 1,
menstp(1) =  1 1 1 1,
mentnk(1) =  1 1 1 1,
gxp(1) = {INP_XCGY * INP_D} {(2*INP_XCGD-INP_XCGY) *INP_D}
         {INP_XCGZ * INP_D} {(2*INP_XCGD-INP_XCGZ) *INP_D},
thrvecx(1) =   0  0  0  0,
thrvecy(1) =  -1  1  0  0,
thrvecz(1) =   0  0 -1  1,
thrclkvecx(1) = 1 1 1 1,
thrclkvecy(1) = 0 0 0 0,
thrclkvecz(1) = 0 0 0 0,
//create as table, zero because not accounting for fuel
wd1t = 'wd1t' constant 0 0.0
wd2t = 'wd2t' constant 0 0.0
wd3t = 'wd3t' constant 0 0.0
wd4t = 'wd4t' constant 0 0.0
ae1t = 'ae1t' constant 0 0.0
ae2t = 'ae2t' constant 0 0.0
ae3t = 'ae3t' constant 0 0.0
ae4t = 'ae4t' constant 0 0.0
*if def INTER_MODEL_ON
    //thrust table
    tvc1t = 'tvc1t',monovar,genvq100,0,lin_inp,xtrap,
    -1.0,0.5,1.0,-0.5,
    tvc2t = 'tvc2t',monovar,genvq100,0,lin_inp,xtrap,
    -1.0,-0.5,1.0,0.5,
    tvc3t = 'tvc3t',monovar,genvq110,0,lin_inp,xtrap,
    -1.0,0.5,1.0,-0.5,
    tvc4t = 'tvc4t',monovar,genvq110,0,lin_inp,xtrap,
    -1.0,-0.5,1.0,0.5,
    tvc1m(1) = 1,
    tvc2m(1) = 1,
    tvc3m(1) = 1,
    tvc4m(1) = 1,
*else
    tvc1m(1) = 0,
    tvc2m(1) = 0,
    tvc3m(1) = 0,
    tvc4m(1) = 0,
*endif

c Input initial state
npc(4) = 2,   //input position in spherical coords
gdalt = 0.0,  //geodetic altitude
long = 0.0,   //initial east long. w.r.t. prime meridian
gclat = 0.0,  //initial geocentric latitude
npc(3) = 3,   //input vel., azimuth, and flight path angle
vela = {INP_MACH * CONST_SOS}, //atmo-relative velocity
azvela = 0.0, //azimuth angle (from north)
gammaa = 0.0, //flight path angle (from horizontal)
dof = 6       //turn on 6-degree-of-freedom calculation
iguid(1) = -2 //type of guidance (-2 for no steering)
iguid(12) = 3 //attitude initialization
iguid(22) = 1 //attitude rate (1=body rates)
rolr = 0      //initial attitude (deg)
yawr = 0.0
rolbd = 0.0   //initial attitude rate (deg/s)
yawbd = 0.0

c Aerodynamic Properties
*if def AERO_ON
    npc(8) = 5, //custom aero module
*else
    npc(8) = 1,
*endif
//aerodynamic reference point in body ref frame
xreft = xref constant 0 {INP_XAEROD * INP_D},
sref = {0.7853982 * INP_D^2}, //ref area (m^2)
lref = {INP_D},  //ref length (m)
drefr = {INP_D}, //roll moment reference length
drefp = {INP_D}, //pitch moment ref. length
drefy = {INP_D}, //yaw moment ref. length

c *include ’aero_range.dat’
c *include ’aero_range_cmq_at.dat’
c c *include ’aero_range_cmq_qh.dat’

event = 100,        //final event
critr = ’time’,     //variable to initiate phase
value = {INP_TMAX}, //value of critr to start phase
endprb = 1,         //end the problem
endjob = 1,         //end the job
```

B.1.2 inputs.dat

This file contains the variables that are changed depending on the input condition case or controller design.

```
*declare INP_K 10000, //IMMA spring constant, N/m
*declare INP_C 7.95,  //IMMA damping constant, N*s/m
*declare CMD_A -2.5,  //commanded AoA, deg
*declare CMD_B 0,     //commanded bega, deg
*declare Ky02 0,      //controller gains
*declare Ky03 0,
*declare Ky04 0,
*declare Ky05 0,
*declare Ky06 0,
*declare Kz02 0.000390994,
*declare Kz03 0.000519268822586,
*declare Kz04 1.2132474121,
*declare Kz05 -0.017790936,
*declare Kz06 -0.000858182379027,
gint2 = 0, //y imma initial position, m
gint4 = 0, //z imma initial position, m
pitr = 5.024988627326E+00, //initial pitch, deg
pitbd = 2.356090702830E+02, //initial pitch rate, deg/s
```

B.2 FUN3D Input Files

B.2.1 fun3d.nml

The FUN3D namelist file specifies flow solver settings.

```
&project
 project_rootname = 'SIAD'
/
&raw_grid
 grid_format = 'aflr3'
/
&governing_equations
 eqn_type = 'compressible'
 viscous_terms = 'turbulent'
/
&reference_physical_properties
 mach_number = 2.1
 reynolds_number = 10545840.865
 temperature = 292.8
 angle_of_attack = 0
 angle_of_yaw = 0
/
&force_moment_integ_properties
 area_reference=9.931466590311E-04
 x_moment_center=5.71711E-03
 x_moment_length=3.556E-02
 y_moment_length=3.556E-02
/
&turbulent_diffusion_models
```

```
 turbulence_model = 'des'
 reynolds_stress_model = 'qcr2000'
/
&spalart
 sarc = .true.
 ddes = .false.
 ddes_mod1 = .false.
 cddes = 0.9750
/
&special_parameters
 large_angle_fix = 'on'
/
&inviscid_flux_method
 flux_limiter = 'hvanalbada'
 flux_construction = 'dldfss'
 first_order_iterations = 0
/
&nonlinear_solver_parameters
 time_accuracy        = '2ndorderOPT'
 schedule_iteration(1)   = 1
 time_step_nondim     = 8.0818182E-03
 subiterations        = 75
 schedule_iteration(2)   = 75
 schedule_cfl(1)         = 60.0
 schedule_cflturb(1)     = 45.0
 schedule_cfl(2)         = 60.0
 schedule_cflturb(2)     = 45.0
 temporal_err_control = .true.
 temporal_err_floor = 0.1
/
&code_run_control
 steps              = 2000
 restart_write_freq = 2000
 restart_read       = 'on'
/
&sampling_output_variables
 primitive_variables = .true.
 mach                = .true.
 temperature         = .true.
/
&massoud_output
 funtofem_include_skin_friction = .true.
/
&global
 boundary_animation_freq=-1
 moving_grid = .true.
/
&boundary_output_variables
 number_of_boundaries = 4
 boundary_list = '1,2,3,4'
 primitive_variables = .true.
 residuals = .true.
 entropy = .true.
/
```

B.2.2 moving_body.input

This file sets the FUN3D simulation to be driven by the Python-based API.

```
&body_definitions
 n_moving_bodies = 1
 body_name(1) = 'forced07'
 body_frame_forces = .true.
 dimensional_output = .true.
 ref_velocity = 3.433498E+02
 ref_density = 2.69E-01
 ref_length = 1
 n_defining_bndry(1) = 4
 defining_bndry(1,1) = 1
 defining_bndry(2,1) = 2
 defining_bndry(3,1) = 3
 defining_bndry(4,1) = 4
 motion_driver(1) = 'external'
 mesh_movement(1) = 'rigid'
 output_transform = .true.
 x_mc(1)=5.71711E-03
 y_mc(1)=0.0
 z_mc(1)=0.0
/
```

REFERENCES

[1] M. D. Griffin and J. R. French, *Space vehicle design*. AIAA, 2004.

[2] H. W. Gehman, *Columbia accident investigation board report*. Columbia Accident Investigation Board, 2003, vol. 6.

[3] S. Gran, *The Voskhod 2 mission revisited*, Apr. 2015.

[4] A. Albee, S. Battel, R. Brace, G. Burdick, J. Casani, J. Lavell, C. Leising, D. MacPherson, P. Burr, and D. Dipprey, "Report on the loss of the Mars Polar Lander and Deep Space 2 missions", 2000.

[5] P. N. Desai and D. T. Lyons, "Entry, descent, and landing operations analysis for the Genesis entry capsule", *Journal of Spacecraft and Rockets*, vol. 45, no. 1, pp. 27–32, 2008.

[6] R. M. Lightfoot, *EM-1 crew study results summary*, Jul. 2017.

[7] N. Barlow, *Mars: An introduction to its interior, surface and atmosphere*. Cambridge Planetary Science, 2014.

[8] D. R. Williams, *Chronology of Mars exploration*, https://nssdc.gsfc.nasa.gov/planetary/chronologymars.html, Accessed 1/21/2021, NASA Goddard Space Flight Center, Dec. 2020.

[9] A. A. Dyakonov, C. E. Glass, K. T. Edquist, M. Schoenenberger, P. Chwalowski, J. Van-Norman, W. I. Scallion, C. Tang, M. J. Wright, F. M. Cheatwood, B. R. Hollis, V. R. Lessard, and N. Takashima, "Design considerations for reaction control systems", in *Fifth International Planetary Probe Workshop*, 2007.

[10] A. Nelessen, C. Sackier, I. Clark, P. Brugarolas, G. Villar, A. Chen, A. Stehura, R. Otero, E. Stilley, D. Way, *et al.*, "Mars 2020 entry, descent, and landing system overview", in *2019 IEEE Aerospace Conference*, IEEE, 2019, pp. 1–20.

[11] A. A. Dyakonov, C. E. Glass, P. N. Desai, and J. W. Van Norman, "Analysis of effectiveness of Phoenix entry reaction control system", *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 746–755, 2011.

[12] A. Dyakonov, M. Schoenenberger, W. Scallion, J. Van Norman, L. Novak, and C. Tang, "Aerodynamic interference due to MSL reaction control system", in *41st AIAA Thermophysics Conference*, 2009, p. 3915.

[13] A. M. Cassell, C. A. Brivkalns, J. V. Bowles, J. A. Garcia, D. J. Kinney, P. F. Wercinski, A. D. Cianciolo, and T. T. Polsgrove, “Human Mars mission design study utilizing the adaptive deployable entry and placement technology”, in *2017 IEEE Aerospace Conference*, IEEE, 2017, pp. 1–16.

[14] B. Yount, A. Cassell, and S. D’Souza, “Pterodactyl: Mechanical designs for integrated control design of a mechanically deployable entry vehicle”, in *AIAA SciTech 2020 Forum*, 2020, p. 1009.

[15] B. M. Atkins and E. M. Queen, “Internal moving mass actuator control for Mars entry guidance”, *Journal of Spacecraft and Rockets*, vol. 52, no. 5, pp. 1294–1310, 2015.

[16] J. Harpold and C. Graves Jr, “Shuttle entry guidance”, *American Astronautical Society*, 1978.

[17] C. Paez, “The development of the X-37 re-entry vehicle”, in *40th AIAA/ASME/SAE/ ASEE Joint Propulsion Conference and Exhibit*, 2004, p. 4186.

[18] S. Dussy, J.-P. Preaud, G. Malucchi, V. Marco, E. Zaccagnino, and A. Drocco, “Intermediate eXperimental Vehicle (IXV), the ESA re-entry demonstrator”, in *AIAA Guidance, Navigation, and Control Conference*, 2011, p. 6340.

[19] D. W. Way, J. L. Davis, and J. D. Shidner, “Assessment of the Mars Science Laboratory entry, descent, and landing simulation”, NASA, Tech. Rep., 2013.

[20] R. D. Braun, *Planetary entry, descent and landing*, Mar. 2014.

[21] D. A. Spencer, R. C. Blanchard, R. D. Braun, P. H. Kallemeyn, and S. W. Thurman, “Mars Pathfinder entry, descent, and landing reconstruction”, *Journal of Spacecraft and Rockets*, vol. 36, no. 3, pp. 357–366, 1999.

[22] P. N. Desai, M. Schoenenberger, and F. Cheatwood, “Mars Exploration Rover six-degree-of-freedom entry trajectory analysis”, *Journal of Spacecraft and Rockets*, vol. 43, no. 5, pp. 1019–1025, 2006.

[23] E. Euler, G. Adams, and F. Hopper, “Design and reconstruction of the Viking lander descent trajectories”, *Journal of Guidance and Control*, vol. 1, no. 5, pp. 372–378, 1978.

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

[25] G. Mendeck and L. Craig, “Entry guidance for the 2011 Mars Science Laboratory mission”, in *AIAA Atmospheric Flight Mechanics Conference*, 2011, p. 6639.

[26] P. B. Brugarolas, A. M. San Martin, and E. C. Wong, “Entry attitude controller for the Mars Science Laboratory”, in *2007 IEEE Aerospace Conference*, IEEE, 2007, pp. 1–6.

[27] P. Brugarolas, A. San Martin, and E. Wong, “Attitude controller for the atmospheric entry of the Mars Science Laboratory”, in *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, p. 6812.

[28] D. E. Smith, M. T. Zuber, H. V. Frey, J. B. Garvin, J. W. Head, D. O. Muhleman, G. H. Pettengill, R. J. Phillips, S. C. Solomon, H. J. Zwally, *et al.*, “Mars Orbiter Laser Altimeter: Experiment summary after the first year of global mapping of Mars”, *Journal of Geophysical Research: Planets*, vol. 106, no. E10, pp. 23 689–23 722, 2001.

[29] R. D. Braun and R. M. Manning, “Mars exploration entry, descent and landing challenges”, in *2006 IEEE Aerospace Conference*, IEEE, 2006.

[30] L. Toups and S. Hoffman, “Pioneering objectives and activities on the surface of Mars”, in *AIAA SPACE 2015 Conference and Exposition*, 2015, p. 4410.

[31] T. Polsgrove, J. Chapman, S. Sutherlin, B. Taylor, E. Robertson, B. Studak, S. Vitalpur, L. Fabisinski, A. Y. Lee, T. Collins, *et al.*, “Human Mars lander design for NASA’s evolvable Mars campaign”, in *2016 IEEE Aerospace Conference*, IEEE, 2016, pp. 1–15.

[32] E. Venkatapathy, K. Hamm, I. Fernandez, J. Arnold, D. Kinney, B. Laub, A. Makino, M. McGuire, K. Peterson, D. Prabhu, *et al.*, “Adaptive deployable entry and placement technology (ADEPT): A feasibility study for human missions to Mars”, in *21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2011, p. 2608.

[33] S. Hughes, F. Cheatwood, R. Dillman, A. Calomino, H. Wright, J. DelCorso, and A. Calomino, “Hypersonic inflatable aerodynamic decelerator (HIAD) technology development overview”, in *21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2011, p. 2524.

[34] A. M. Korzun, J. R. Cruz, and R. D. Braun, “A survey of supersonic retropropulsion technology for Mars entry, descent, and landing”, in *2008 IEEE Aerospace Conference*, IEEE, 2008, pp. 1–15.

[35] A. A. Wolf, B. Acikmese, J. Casoliva, S. Ploen, and J. B. Manrique, “Improving the landing precision of an MSL-class vehicle”, in *2012 IEEE Aerospace Conference*, IEEE, 2012, pp. 1–10.

[36] D. A. Craig, P. Troutman, and N. Herrmann, “Pioneering space through the evolvable Mars campaign”, in *AIAA Space 2015 Conference and Exposition*, 2015, p. 4409.

[37] Langley Technology Council, “Langley strategic technology investment plan”, NASA Langley Research Center, Tech. Rep., Apr. 2020.

[38] E. P. Bonfiglio, D. Adams, L. Craig, D. A. Spencer, R. Arvidson, and T. Heet, “Landing-site dispersion analysis and statistical assessment for the Mars Phoenix Lander”, *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 784–797, 2011.

[39] A. A. Wolf, C. Graves, R. Powell, and W. Johnson, “Systems for pinpoint landing at Mars”, 2004.

[40] M. Schoenenberger, A. Dyakonov, P. Buning, B. Scallion, and J. Van Norman, “Aerodynamic challenges for the Mars Science Laboratory entry, descent, and landing”, in *41st AIAA Thermophysics Conference*, 2009, p. 3914.

[41] E. Stern, A. Schwing, J. M. Brock, and M. Schoenenberger, “Dynamic CFD simulations of the MEADS II ballistic range test model”, in *AIAA Atmospheric Flight Mechanics Conference*, 2016, p. 3243.

[42] A. Dyakonov, M. Schoenenberger, and J. Van Norman, “Hypersonic and supersonic static aerodynamics of Mars Science Laboratory entry vehicle”, in *43rd AIAA Thermophysics Conference*, 2012, p. 2999.

[43] C. D. Kazemba, R. D. Braun, I. G. Clark, and M. Schoenenberger, “Survey of blunt-body supersonic dynamic stability”, *Journal of Spacecraft and Rockets*, vol. 54, no. 1, pp. 109–127, Jan. 2017.

[44] T. Abe, S.-i. Sato, Y. Matsukawa, K. Yamamoto, and K. Hiraoka, “Study for dynamically unstable motion of reentry capsule”, in *34th Thermophysics Conference*, 2000, p. 2589.

[45] C. D. Kazemba, R. D. Braun, M. Schoenenberger, and I. G. Clark, “Dynamic stability analysis of blunt-body entry vehicles using time-lagged aftbody pitching moments”, *Journal of Spacecraft and Rockets*, vol. 52, no. 2, pp. 393–403, 2015.

[46] J. Allison, *Complex system optimization: A review of analytical target cascading, collaborative optimization, and other formulations*, 2004.

[47] S. Ghoreishi and D. Allaire, “Adaptive uncertainty propagation for coupled multidisciplinary systems”, *AIAA journal*, vol. 55, no. 11, pp. 3940–3950, 2017.

[48] M. Salas, “A review of hypersonics aerodynamics, aerothermodynamics and plasmadynamics activities within NASA’s fundamental aeronautics program”, in *39th AIAA Thermophysics Conference*, 2007, p. 4264.

[49] A. R. Vasavada, A. Chen, J. R. Barnes, P. D. Burkhart, B. A. Cantor, A. M. Dwyer-Cianciolo, R. L. Fergason, D. P. Hinson, H. L. Justh, D. M. Kass, *et al.*, “Assessment of environments for Mars Science Laboratory entry, descent, and surface operations”, *Space Science Reviews*, vol. 170, no. 1-4, pp. 793–835, 2012.

[50] S. Dutta and D. W. Way, “Comparison of the effects of velocity and range triggers on trajectory dispersions for the Mars 2020 mission”, in *AIAA Atmospheric Flight Mechanics Conference*, 2017, p. 0245.

[51] H. Bijl, D. Lucor, S. Mishra, and C. Schwab, *Uncertainty quantification in computational fluid dynamics*. Springer Science & Business Media, 2013, vol. 92.

[52] J. L. Davis, J. D. Shidner, and D. W. Way, “Mars Science Laboratory post-landing location estimation using POST2 trajectory simulation”, 2013.

[53] S. A. Striepe, D. Way, A. Dwyer, and J. Balaram, “Mars Science Laboratory simulations for entry, descent, and landing”, *Journal of Spacecraft and Rockets*, vol. 43, no. 2, pp. 311–323, 2006.

[54] M. Schoenenberger, J. V. Norman, C. Karlgaard, P. Kutty, and D. Way, “Assessment of the reconstructed aerodynamics of the Mars Science Laboratory entry vehicle”, *Journal of Spacecraft and Rockets*, vol. 51, no. 4, pp. 1076–1093, 2014.

[55] R. Shotwell, “Phoenix—the first Mars scout mission”, *Acta Astronautica*, vol. 57, no. 2-8, pp. 121–134, 2005.

[56] K. Edquist, P. Desai, and M. Schoenenberger, “Aerodynamics for the Mars Phoenix entry capsule”, *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008, AIAA2008G7219.

[57] J. L. Prince, P. N. Desai, E. M. Queen, and M. R. Grover, “Mars Phoenix entry, descent, and landing simulation design and modeling analysis”, *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 756–764, 2011.

[58] A. M. Korzun, R. W. Maddock, M. Schoenenberger, K. T. Edquist, C. H. Zumwalt, and C. D. Karlgaard, “Aerodynamic performance of the 2018 InSight Mars lander”, in *AIAA Scitech 2020 Forum*, 2020, p. 1272.

[59] R. Sammonds and R. Kruse, “Viking entry vehicle aerodynamics at M=2 in air and some preliminary test data for flight in co2 at M=11”, 1975.

[60] B. E. Robertson, “A hybrid probabilistic method to estimate design margin”, Ph.D. dissertation, Georgia Institute of Technology, 2013.

[61] K. W. Iliff and M. F. Schafer, *Space Shuttle hypersonic aerodynamic and aerothermodynamic flight research and the comparison to ground test results*. National Aeronautics and Space Administration, Office of Management . . ., Jun. 1993, vol. 4499.

[62] J. Maus, B. Griffith, K. Szema, and J. Best, “Hypersonic Mach number and real gas effects on Space Shuttle Orbiter aerodynamics”, *Journal of Spacecraft and Rockets*, vol. 21, no. 2, pp. 136–141, 1984.

[63] M. Schoenenberger, M. Rhode, J. Paulson, and J. Van Norman, “Characterization of aerodynamic interactions with the Mars Science Laboratory reaction control system using computation and experiment”, in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 971.

[64] B. Etkin, *Dynamics of Atmospheric Flight*. Courier Corporation, 2012.

[65] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 2016.

[66] J. T. Betts, “Survey of numerical methods for trajectory optimization”, *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[67] R. Koomullil and N. Prewitt, “A library based approach for rigid body dynamics simulation”, in *18th AIAA Computational Fluid Dynamics Conference*, 2007, p. 4476.

[68] G. Brauer, D. Cornick, and R. Stevenson, “Capabilities and applications of the program to optimize simulated trajectories (POST). program summary document”, NASA, Washington, United States, Tech. Rep., 1977.

[69] S. A. Striepe, R. W. Powell, P. N. Desai, E. M. Queen, D. W. Way, J. L. Prince, A. M. Cianciolo, J. L. Davis, D. K. Litton, R. M. Maddock, J. D. Shidner, R. G. Winski, S. A. O’Keefe, A. G. Bowes, J. T. Aguirre, C. A. Garrison, J. A. Hoffman, A. D. Olds, S. Dutta, C. H. Zumwalt, J. P. White, G. L. Brauer, S. M. Marsh, R. A. Lugo, and J. S. Green, *Program to optimize simulated trajectories II (POST2): Utilization manual*, 2019.

[70] J. Warner, S. C. Niemoeller, L. Morrill, G. Bomarito, P. Leser, W. Leser, R. A. Williams, and S. Dutta, “Multi-model Monte Carlo estimators for trajectory simulation”, in *AIAA Scitech 2021 Forum*, 2021, p. 0761.

[71] P. Gallais, *Atmospheric re-entry vehicle mechanics*. Springer Science & Business Media, 2007.

[72] J. D. Anderson Jr, *Hypersonic and high-temperature gas dynamics*. American Institute of Aeronautics and Astronautics, 2006.

[73] K. Edquist, A. Dyakonov, M. Wright, and C. Tang, "Aerothermodynamic design of the Mars Science Laboratory heatshield", in *41st AIAA Thermophysics Conference*, 2009, p. 4075.

[74] P. N. Desai, J. L. Prince, E. M. Queen, M. M. Schoenenberger, J. R. Cruz, and M. R. Grover, "Entry, descent, and landing performance of the Mars Phoenix Lander", *Journal of Spacecraft and Rockets*, vol. 48, no. 5, pp. 798–808, 2011.

[75] P. A. Gnoffo, R. D. Braun, K. J. Weilmuenster, R. A. Mitcheltree, W. C. Engelund, and R. W. Powell, "Prediction and validation of Mars Pathfinder hypersonic aerodynamic database", *Journal of Spacecraft and Rockets*, vol. 36, no. 3, pp. 367–373, 1999.

[76] S. Teramoto and K. Fujii, "Mechanism of dynamic instability of a reentry capsule at transonic speeds", *AIAA journal*, vol. 40, no. 12, pp. 2467–2475, 2002.

[77] T. Yoshinaga, A. Tate, M. Watanabe, and T. Shimoda, "Orbital re-entry experiment vehicle ground and flight dynamic test results comparison", *Journal of spacecraft and rockets*, vol. 33, no. 5, pp. 635–642, 1996.

[78] M. Schoenenberger, T. G. Brown, and L. Yates, "Surface pressure ballistic range test of Mars 2020 capsule in support of MEDLI2", in *35th AIAA Applied Aerodynamics Conference*, 2017, p. 4079.

[79] J. D. Anderson, *Fundamentals of Aerodynamics*, 4th. McGraw Hill, 2011.

[80] R. McGhee, P. Siemers III, and R. Pelc, "Transonic aerodynamic characteristics of the Viking entry and lander configurations", Oct. 1971.

[81] W. W. Blake, "Experimental aerodynamic characteristics of the Viking entry vehicle over the Mach range 1.5-10.0", Martin Marietta Corporation, Denver Division, Tech. Rep., Apr. 1971.

[82] K. Murphy, S. Borg, A. Watkins, D. Cole, and R. Schwartz, "Testing of the Crew Exploration Vehicle in NASA Langley's Unitary Plan Wind Tunnel", in *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007, p. 1005.

[83] M. N. Rhode and W. L. Oberkampf, "Estimation of uncertainties for a model validation experiment in a wind tunnel", *Journal of Spacecraft and Rockets*, vol. 54, no. 1, pp. 155–168, Jan. 2017.

[84] J. Gregory, K. Asai, M. Kameda, T. Liu, and J. Sullivan, “A review of pressure-sensitive paint for high-speed and unsteady aerodynamics”, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 222, no. 2, pp. 249–290, 2008.

[85] C. W. Hirt, A. A. Amsden, and J. Cook, “An arbitrary Lagrangian-Eulerian computing method for all flow speeds”, *Journal of computational physics*, vol. 14, no. 3, pp. 227–253, 1974.

[86] R. Biedron and J. Thomas, “Recent enhancements to the FUN3D flow solver for moving-mesh applications”, in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, 2009, p. 1360.

[87] W. K. Anderson and D. L. Bonhaus, “An implicit upwind algorithm for computing turbulent flows on unstructured grids”, *Computers & Fluids*, vol. 23, no. 1, pp. 1–21, 1994.

[88] R. Biedron, V. Vatsa, and H. Atkins, “Simulation of unsteady flows using an unstructured navier-stokes solver on moving and stationary grids”, in *23rd AIAA Applied Aerodynamics Conference*, 2005, p. 5093.

[89] M. Schoenenberger, F. M. Cheatwood, and P. Desai, “Static aerodynamics of the Mars Exploration Rover entry capsule”, in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005, p. 56.

[90] S. Steinberg, “Experimental pitch damping derivatives for candidate Viking entry configurations at Mach numbers from 0.6 through 3.0”, *Martin Marietta Corporation, Denver Division, TR-3709005*, 1970.

[91] B. Uselton, T. Shadow, and A. Mansfield, “Damping-in-pitch derivatives of 120- and 140-deg blunted cones at Mach numbers from 0.6 through 3”, ARO INC ARNOLD AFS TN, Tech. Rep., 1970.

[92] M. C. Wilder, D. W. Bogdanoff, and C. J. Cornelison, “Hypersonic testing capabilities at the NASA Ames ballistic ranges”, in *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 1339.

[93] R. Lugo, R. Tolson, and M. Schoenenberger, “Trajectory reconstruction and uncertainty analysis using Mars Science Laboratory pre-flight scale model aeroballistic testing”, in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013, p. 1132.

[94] L. Yates and G. Chapman, “Aerodynamic ballistic range analysis using generalized math models”, in *21st Atmospheric Flight Mechanics Conference*, 1996, p. 3360.

[95] G. T. Chapman and D. B. Kirk, “A method for extracting aerodynamic coefficients from free-flight data”, *AIAA Journal*, vol. 8, no. 4, pp. 753–758, 1970.

[96] J. Brown, L. Yates, D. Bogdanoff, G. Chapman, M. Loomis, and T. Tam, “Free-flight testing in support of the Mars Science Laboratory aerodynamics database”, *Journal of spacecraft and Rockets*, vol. 43, no. 2, pp. 293–302, 2006.

[97] M. C. Wilder, J. D. Brown, D. W. Bogdanoff, L. A. Yates, A. A. Dyakonov, I. G. Clark, and J. H. Grinstead, “Aerodynamic coefficients from aeroballistic range testing of deployed-and stowed-SIAD SFDT models”, 2017.

[98] M. Schoenenberger, L. Yates, and W. Hathaway, “Dynamic stability testing of the Mars Science Laboratory entry capsule”, in *41st AIAA Thermophysics Conference*, 2009, p. 3917.

[99] M. Schoenenberger, D. E. Cox, T. Schott, A. Mackenzie, O. Ramirez, C. P. Britcher, C. Neill, M. Wienmann, and D. Johnson, “Preliminary aerodynamic measurements from a magnetic suspension and balance system in a low-speed wind tunnel”, in *2018 Applied Aerodynamics Conference*, 2018, p. 3323.

[100] H. Tanno, T. Komuro, K. Sato, K. Itoh, and T. Yamada, “Free-flight tests of reentry capsule models in free-piston shock tunnel”, in *43rd AIAA Fluid Dynamics Conference*, 2013, p. 2979.

[101] I. Clark and M. Adler, “Summary of the second high-altitude, supersonic flight dynamics test for the LDSD project”, in *2016 IEEE Aerospace Conference*, IEEE, 2016, pp. 1–24.

[102] R. Dillman, J. DiNonno, R. Bodkin, V. Gsell, N. Miller, A. Olds, and W. Bruce, “Flight performance of the Inflatable Reentry Vehicle Experiment 3”, in *10th International Planetary Probe Workshop (IPPW-10), San Jose, CA*, 2013.

[103] R. Ingoldby, F. Michel, T. Flaherty, M. Doryand, B. Preston, K. Villyard, and R. Steele, “Entry data analysis for Viking landers 1 and 2 final report”, *NASA CR-159388*, 1976.

[104] M. M. Munk, A. Little, C. Kuhl, D. Bose, and J. Santos, “The Mars Science Laboratory (MSL) entry, descent, and landing instrumentation (MEDLI) hardware”, *AAS Paper*, pp. 13–310, 2013.

[105] K. Edquist, “Computations of Viking lander capsule hypersonic aerodynamics with comparisons to ground and flight data”, in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2006, p. 6137.

[106] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, “CFD vision 2030 study: A path to revolutionary computational aerosciences: NASA”, CR-2014-218178Washington, DC: NASA, Tech. Rep., 2014.

[107] D. M. Schuster, “CFD 2030 grand challenge: CFD-in-the-loop Monte Carlo flight simulation for space vehicle design”, in *AIAA Scitech 2021 Forum*, 2021, p. 0957.

[108] E. Stern, V. Gidzak, and G. Candler, “Estimation of dynamic stability coefficients for aerodynamic decelerators using CFD”, in *30th AIAA Applied Aerodynamics Conference*, American Institute of Aeronautics and Astronautics, Jun. 2012.

[109] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2014.

[110] J. M. Brock, E. C. Stern, and M. C. Wilder, “Computational fluid dynamics simulations of Supersonic Inflatable Aerodynamic Decelerator ballistic range tests”, *Journal of Spacecraft and Rockets*, vol. 56, no. 2, pp. 526–535, Mar. 2019.

[111] A. Cenko, “Store separation lessons learned during the last 30 years”, Naval Air Systems Command, Patuxent River MD, Tech. Rep., 2010.

[112] R. Meakin, “Computations of the unsteady flow about a generic wing/pylon/finned-store configuration”, in *Astrodynamics Conference*, 1992, p. 4568.

[113] J. Kokes, M. Costello, and J. Sahu, “Generating an aerodynamic model for projectile flight simulation using unsteady time accurate computational fluid dynamic results”, *Computational Ballistics III*, vol. 45, p. 11 131, 2007.

[114] M. Costello, S. Gatto, and J. Sahu, “Using CFD/RBD results to generate aerodynamic models for projectile flight simulation”, in *AIAA Atmospheric Flight Mechanics Conference*, 2007.

[115] C. Montalvo and M. Costello, “Estimation of projectile aerodynamic coefficients using coupled CFD/RBD simulation results”, in *AIAA Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, Aug. 2010.

[116] J. Sahu, “Numerical computations of dynamic derivatives of a finned projectile using a time-accurate CFD method”, in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2007, p. 6581.

[117] J. DeSpirito and K. Heavey, “CFD computation of Magnus moment and roll-damping moment of a spinning projectile”, in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2004, p. 4713.

[118] J. Sahu and F. Fresconi, "Flight behaviors of a complex projectile using a coupled CFD-based simulation technique: Open-loop control", in *54th AIAA Aerospace Sciences Meeting*, 2016, p. 2025.

[119] R. Meakin, C. Atwood, and N. Hariharan, "Development, deployment, and support of a set of multi-disciplinary, physics-based simulation software products", in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011, p. 1104.

[120] S. A. Morton, B. Tillman, D. R. McDaniel, D. R. Sears, and T. R. Tuckey, "Kestrel–a fixed wing virtual aircraft product of the CREATE program", in *2009 DoD High Performance Computing Modernization Program Users Group Conference*, IEEE, 2009, pp. 148–152.

[121] S. Morton, D. McDaniel, D. Sears, B. Tillman, and T. Tuckey, "Kestrel v2-6DOF and control surface additions to a CREATE simulation tool", in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2011, p. 511.

[122] S. Morton, T. Eymann, D. McDaniel, D. Sears, B. Tillman, and T. Tuckey, "Rigid and maneuvering results with control surface and 6DoF motion for Kestrel v2", in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, includes summary of kAVUS CFD flow solver (formerly cobalt60), 2011, p. 1106.

[123] B. Roget, J. Sitaraman, N. Peters, A. M. Wissink, and J. R. Forsythe, "Overset moving body 6-DOF simulations using HPCMP CREATE A/V Helios", in *AIAA Scitech 2021 Forum*, 2021, p. 0840.

[124] S. M. Murman and M. Aftosmis, "Dynamic analysis of atmospheric-entry probes and capsules", in *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007, p. 74.

[125] S. M. Murman, "Dynamic simulations of atmospheric-entry capsules", *Journal of Spacecraft and Rockets*, vol. 46, no. 4, pp. 829–835, Jul. 2009.

[126] S. M. Murman, M. Aftosmis, and M. Berger, "Simulations of 6-DOF motion with a cartesian method", in *41st Aerospace Sciences Meeting and Exhibit*, 2003, p. 1246.

[127] P. R. Spalart, S. Deck, M. L. Shur, K. D. Squires, M. K. Strelets, and A. Travin, "A new version of detached-eddy simulation, resistant to ambiguous grid densities", *Theoretical and computational fluid dynamics*, vol. 20, no. 3, p. 181, 2006.

[128] E. Stern, J. Brock, Q. McKown, and C. Kazemba, "A method for deriving capsule pitch damping coefficients from free-flight CFD data", *Submitted to AIAA Journal*, 2022.

[129] A. Ben-Yakar, M. Mungal, and R. Hanson, “Time evolution and mixing characteristics of hydrogen and ethylene transverse jets in supersonic crossflows”, *Physics of Fluids*, vol. 18, no. 2, p. 026 101, 2006.

[130] J. F. Foote, “Apollo SM-LM RCS engine development program summary report, volume 4 final report (development, performance, and production acceptance tests for Apollo R-4 D engine)”, 1969.

[131] G. Buck, A. Watkins, P. Danehy, J. Inman, D. Alderfer, and A. Dyakonov, “Experimental measurement of RCS jet interaction effects on a capsule entry vehicle”, in *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008, p. 1229.

[132] A. Mazaheri, P. A. Gnoffo, C. O. Johnston, and W. L. Kleb, “LAURA users manual”, 2013.

[133] R. T. Biedron, J. R. Carlson, J. M. Derlaga, P. A. Gnoffo, D. P. Hammond, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielsen, M. A. Park, *et al.*, *FUN3D manual: 13.6*, 2019.

[134] A. M. Korzun, C. Tang, Y. Rizk, F. Canabal, R. Childs, J. W. Van Norman, J. A. Tynis, and K. Bibb, “Powered descent aerodynamics for low and mid lift-to-drag human Mars entry, descent and landing vehicles”, in *AIAA Scitech 2020 Forum*, 2020, p. 1510.

[135] J. Sahu, “Coupled CFD and rigid body dynamics modeling of a spinning body with flow control”, in *2nd AIAA Flow Control Conference*, 2004, p. 2317.

[136] J. Sahu, F. Fresconi, and K. R. Heavey, “Unsteady aerodynamic simulations of a finned projectile at a supersonic speed with jet interaction”, in *32nd AIAA Applied aerodynamics conference*, 2014, p. 3024.

[137] J. Sahu, M. Costello, and C. Montalvo, “Development and application of multidisciplinary coupled computational techniques for projectile aerodynamics”, in *7th International Conference on Computational Fluid Dynamics*, 2012, pp. 1–24.

[138] J. Sahu and F. Fresconi, “Aeromechanics and control of projectile roll using coupled simulation techniques”, *Journal of Spacecraft and Rockets*, vol. 52, no. 3, pp. 944–957, 2015.

[139] ——, “Validation of coupled computations for pitching motions of a canard-controlled body with dynamic wind tunnel data”, in *2018 Applied Aerodynamics Conference*, 2018, p. 4126.

[140] ——, "Flight behaviors of a complex projectile using a coupled CFD-based simulation technique: Closed-loop control", in *34th AIAA Applied Aerodynamics Conference*, 2016, p. 4332.

[141] D. McDaniel, D. Sears, T. Tuckey, B. Tillman, and S. Morton, "Aerodynamic control surface implementation in Kestrel v2. 0", in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011, p. 1175.

[142] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, "Surrogate-based analysis and optimization", *Progress in aerospace sciences*, vol. 41, no. 1, pp. 1–28, 2005.

[143] M. Giselle Fernández-Godino, C. Park, N. H. Kim, and R. T. Haftka, "Issues in deciding whether to use multifidelity surrogates", *AIAA Journal*, vol. 57, no. 5, pp. 2039–2054, 2019.

[144] B. R. Hollis and A. S. Collier, "Turbulent aeroheating testing of Mars Science Laboratory entry vehicle", *Journal of Spacecraft and Rockets*, vol. 45, no. 3, pp. 417–427, 2008.

[145] B. R. Hiller, "An unsteady aerodynamics reduced-order modeling method for maneuvering, flexible flight vehicles", Ph.D. dissertation, Georgia Institute of Technology, 2019.

[146] M. Schoenenberger, W. Hathaway, L. Yates, and P. Desai, "Ballistic range testing of the Mars Exploration Rover entry capsule", in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005, p. 55.

[147] "Space technology research grants program,early stage innovations appendix", NASA Space Technology Mission Directorate, Tech. Rep., Apr. 2022.

[148] N. R. Draper and H. Smith, *Applied regression analysis*. John Wiley & Sons, 1998, vol. 326.

[149] X. Cheng, R. Xin, L. Wu, and Z. Guo, "RCS and aerodynamic parameters identification of the reentry spacecraft from Chang'E flight test", in *21st AIAA International Space Planes and Hypersonics Technologies Conference*, 2017, p. 2204.

[150] Y. Hong, Z. Xiong, J. Wang, and H. Chen, "Reaction control system of hypersonic vehicle and its moment parameter identification", in *Proceedings of the 33rd Chinese Control Conference*, IEEE, 2014, pp. 6693–6697.

[151] K. Du and M. N. S. Swamy, *Neural networks and statistical learning*. Springer Science & Business Media, 2013.

[152] W. E. Faller and S. J. Schreck, “Neural networks: Applications and opportunities in aeronautics”, *Progress in aerospace sciences*, vol. 32, no. 5, pp. 433–456, 1996.

[153] D. J. Linse and R. F. Stengel, “Identification of aerodynamic coefficients using computational neural networks”, *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 6, pp. 1018–1025, 1993.

[154] S. Singh and A. Ghosh, “Parameter estimation from flight data of a missile using maximum likelihood and neural network method”, in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2006, p. 6284.

[155] L. K. Li, “Approximation theory and recurrent networks”, in *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, IEEE, vol. 2, 1992, pp. 266–271.

[156] S. N. Kumpati, P. Kannan, *et al.*, “Identification and control of dynamical systems using neural networks”, *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[157] D. I. Ignatyev and A. N. Khrabrov, “Neural network modeling of unsteady aerodynamic characteristics at high angles of attack”, *Aerospace Science and Technology*, vol. 41, pp. 106–115, 2015.

[158] D. Ignatyev and A. Khrabrov, “Experimental study and neural network modeling of aerodynamic characteristics of canard aircraft at high angles of attack”, *Aerospace*, vol. 5, no. 1, p. 26, 2018.

[159] Q. Wang, C. E. Cesnik, and K. Fidkowski, “Multivariate recurrent neural network models for scalar and distribution predictions in unsteady aerodynamics”, in *AIAA Scitech 2020 Forum*, 2020, p. 1533.

[160] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[161] R. N. Ingoldby, “Guidance and control system design of the Viking planetary lander”, *Journal of Guidance and Control*, vol. 1, no. 3, pp. 189–196, 1978.

[162] P. B. Brugarolas, A. M. San Martin, and E. C. Wong, “The RCS attitude controller for the exo-atmospheric and guided entry phases of the Mars Science Laboratory”, in *International planetary probe workshop*, 2010.

[163] P. C. Calhoun and E. M. Queen, “Entry vehicle control system design for the Mars Science Laboratory”, *Journal of Spacecraft and Rockets*, vol. 43, no. 2, pp. 324–329, 2006.

[164] P. Caravani, *Modern linear control design*. Springer, 2013.

[165] W. J. Rugh and J. S. Shamma, “Research on gain scheduling”, *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[166] P. Ordóñez, A. R. Mills, T. Dodd, and J. Liu, “Formal verification of a gain scheduling control scheme”, in *2017 25th Mediterranean Conference on Control and Automation (MED)*, IEEE, 2017, pp. 259–264.

[167] B. M. Atkins, “Mars precision entry vehicle guidance using internal moving mass actuators”, Ph.D. dissertation, Virginia Tech, 2014.

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

[169] W. R. van Soest, Q. P. Chu, and J. A. Mulder, “Combined feedback linearization and constrained model predictive control for entry flight”, *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 2, pp. 427–434, 2006.

[170] M. L. Steinberg, “Comparison of intelligent, adaptive, and nonlinear flight control laws”, *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 4, pp. 693–699, 2001.

[171] C. Restrepo and J. Valasek, “Structured adaptive model inversion controller for Mars atmospheric flight”, *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 937–953, 2008.

[172] H. K. Schoenwetter, “High-accuracy settling time measurements”, *IEEE Transactions on Instrumentation and Measurement*, vol. 32, no. 1, pp. 22–27, 1983.

[173] S. J. McNamara, C. I. Restrepo, E. A. Medina, R. J. Whitley, J. M. Madsen, and R. W. Proud, “Gain scheduling for the Orion launch abort vehicle controller”, in *AIAA Guidance, Navigation, and Control Conference*, 2011.

[174] C. I. Marrison and R. F. Stengel, “Design of robust control systems for a hypersonic aircraft”, *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 58–63, 1998.

[175] E. M. Queen, F. M. Cheatwood, R. W. Powell, R. D. Braun, and C. T. Edquist, “Mars polar lander aerothermodynamic and entry dispersion analysis”, *Journal of Spacecraft and Rockets*, vol. 36, no. 3, pp. 421–428, 1999.

[176] B. D. Anderson, J. Pren, and S. Dickerson, *Linear optimal control*, 1971.

[177] B. Datta, *Numerical methods for linear control systems*. Academic Press, 2004, vol. 1.

[178] L. R. Ray and R. F. Stengel, “A Monte Carlo approach to the analysis of control system robustness”, *Automatica*, vol. 29, no. 1, pp. 229–236, 1993.

[179] Y. Miyazawa and T. Motoda, “Stochastic parameter tuning applied to space vehicle flight control design”, *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 3, pp. 597–604, 2001.

[180] T. Motoda, R. F. Stengel, and Y. Miyazawa, “Robust control system design using simulated annealing”, *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 2, pp. 267–274, 2002.

[181] R. R. Schaller, “Moore’s law: Past, present and future”, *IEEE spectrum*, vol. 34, no. 6, pp. 52–59, 1997.

[182] R. Stevens, J. Ramprakash, P. Messina, M. Papka, and K. Riley, “Aurora: Argonne’s next-generation exascale supercomputer”, ANL (Argonne National Laboratory (ANL), Argonne, IL (United States)), Tech. Rep., 2019.

[183] ORNL, *Frontier*, https://www.olcf.ornl.gov/frontier, Accessed Febuary 2021, 2020.

[184] LLNL, *LLNL and HPE to partner with AMD on El Capitan, projected as world’s fastest supercomputer*, https://www.llnl.gov/news/llnl-and-hpe-partner-amd-el-capitan-projected-worlds-fastest-supercomputer, Accessed Febuary 2021, Mar. 2020.

[185] A. Walden, E. Nielsen, A. Korzun, B. Jones, J.-R. Carlson, C. Henze, P. Moran, T. Sandstrom, J. Luitjens, and M. Zubair, “Landing on Mars: Petascale unstructured computational fluid dynamics on summit”, in *International Workshop on OpenPOWER for HPC*, 2019.

[186] B. Peherstorfer, K. Willcox, and M. Gunzburger, “Survey of multifidelity methods in uncertainty propagation, inference, and optimization”, *Siam Review*, vol. 60, no. 3, pp. 550–591, 2018.

[187] M. G. Fernández-Godino, C. Park, N.-H. Kim, and R. T. Haftka, “Review of multifidelity models”, *arXiv preprint arXiv:1609.07196*, 2016.

[188] M. Lickenbrock, M. P. Rumpfkeil, P. S. Beran, and R. M. Kolonay, “Multi-fidelity, multidisciplinary analysis of an efficient supersonic air vehicle”, in *AIAA Scitech 2020 Forum*, 2020, p. 2223.

[189] H. Shah, S. Hosder, S. Koziel, Y. A. Tesfahunegn, and L. Leifsson, “Multi-fidelity robust aerodynamic design optimization under mixed uncertainty”, *Aerospace Science and Technology*, vol. 45, pp. 17–29, 2015.

[190] J. Tao and G. Sun, “Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization”, *Aerospace Science and Technology*, vol. 92, pp. 722–737, 2019.

[191] L. Brevault, M. Balesdent, J. Morio, *et al.*, *Aerospace System Analysis and Optimization in Uncertainty*. Springer, 2020.

[192] H. G. Matthies and J. Steindorf, “Partitioned but strongly coupled iteration schemes for nonlinear fluid–structure interaction”, *Computers & structures*, vol. 80, no. 27-30, pp. 1991–1999, 2002.

[193] R. T. Biedron, J. R. Carlson, J. M. Derlaga, P. A. Gnoffo, D. P. Hammond, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielsen, M. A. Park, *et al.*, *FUN3D manual: 13.7*, 2019.

[194] N. Alexandrov, H. Atkins, K. Bibb, R. Biedron, M. Carpenter, P. Gnoffo, D. Hammond, W. Jones, W. Kleb, and E. Lee-Rausch, “Team software development for aerothermodynamic and aerodynamic analysis and design”, 2003.

[195] A. M. Korzun, S. Dutta, R. D. McDaniel, C. D. Karlgaard, and J. A. Tynis, “Aerodynamics for the ADEPT SR-1 flight experiment”, in *AIAA Aviation 2019 Forum*, 2019, p. 2897.

[196] R. Noack, D. Boger, R. e. Kunz, and P. Carrica, “Suggar++: An improved general overset grid assembly capability”, in *19th AIAA Computational Fluid Dynamics*, 2009, p. 3992.

[197] S. Striepe, R. Powell, P. Desai, E. Queen, D. Way, J. Prince, A. Cianciolo, J. Davis, D. Litton, R. Maddock, J. D. Shidner, R. G. Winski, S. A. O’Keefe, A. G. Bowes, J. T. Aguirre, C. A. Garrison, J. A. Hoffman, A. D. Olds, S. Dutta, G. L. Brauer, M. C. Engel, and S. M. Marsh, *Program to optimize simulated trajectories II (POST2): Utilization manual*, 2015.

[198] R. A. Lugo, J. D. Shidner, R. W. Powell, S. M. Marsh, J. A. Hoffman, D. K. Litton, and T. L. Schmitt, “Launch vehicle ascent trajectory simulation using the Program to Optimize Simulated Trajectories II”, 2017.

[199] R. P. Kornfeld, R. Prakash, A. S. Devereaux, M. E. Greco, C. C. Harmon, and D. M. Kipp, “Verification and validation of the Mars Science Laboratory/curiosity rover entry, descent, and landing system”, *Journal of spacecraft and rockets*, vol. 51, no. 4, pp. 1251–1269, 2014.

[200] J. White, A. L. Bowes, S. Dutta, M. C. Ivanov, and E. M. Queen, “LDSD POST2 modeling enhancements in support of SFDT-2 flight operations”, 2016.

[201] L. MacAllister, “The aerodynamic properties of a simple non rolling finned cone-cylinder configuration between Mach numbers 1.0 and 2.5”, Army Ballistic Research Lab, Aberdeen Proving Ground, MD, Tech. Rep., 1955.

[202] A. D. Dupuis and W. Hathaway, “Aeroballistic range tests of the basic finner reference projectile at supersonic velocities”, Defence Research Establishment Valcartier (Québec), Tech. Rep., 1997.

[203] J. Sahu and K. R. Heavey, “Parallel CFD computations of projectile aerodynamics with a flow control mechanism”, *Computers & Fluids*, vol. 88, pp. 678–687, 2013.

[204] J. Dykes, C. Montalvo, M. Costello, and J. Sahu, “Use of microspoilers for control of finned projectiles”, *Journal of Spacecraft and Rockets*, vol. 49, no. 6, pp. 1131–1140, 2012.

[205] Cadence Design Systems Inc., *Pointwise 18.0R2 user manual*, Dec. 2016.

[206] A. Dupuis, “Aeroballistic range and wind tunnel tests of the basic finner reference projectile from subsonic to high supersonic velocities”, *Defense R&D Canada, Technical Memorandum TM 2002-136*, 2002.

[207] Z. J. Ernst, B. R. Hiller, C. L. Johnson, B. E. Robertson, and D. N. Mavris, “Coupling computational fluid dynamics with 6DOF rigid body dynamics for unsteady, accelerated flow simulations”, in *2018 AIAA Atmospheric Flight Mechanics Conference*, 2018, p. 0291.

[208] D. C. Wilcox, “Formulation of the kw turbulence model revisited”, *AIAA journal*, vol. 46, no. 11, pp. 2823–2838, 2008.

[209] PACE, *Partnership for an Advanced Computing Environment (PACE)*, 2017.

[210] I. G. Clark, T. Rivellini, and M. Adler, “Development and testing of a new family of low-density supersonic decelerators”, in *AIAA Aerodynamic Decelerator Systems (ADS) Conference*, 2013, p. 1252.

[211] L. Giersch, T. Rivellini, I. G. Clark, C. Sandy, G. Sharpe, L. S. Shook, J. S. Ware, J. Welch, J. Mollura, and M. Dixon, “SIAD-R: A supersonic inflatable aerodynamic decelerator for robotic missions to Mars”, in *AIAA Aerodynamic Decelerator Systems (ADS) Conference*, 2013, p. 1327.

[212] P. R. Spalart, “Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach”, in *Proceedings of first AFOSR international conference on DNS/LES*, Greyden Press, 1997.

[213] G. D. Van Albada, B. Van Leer, and W. Roberts, “A comparative study of computational methods in cosmic gas dynamics”, in *Upwind and high-resolution schemes*, Springer, 1997, pp. 95–103.

[214] V. Vatsa, M. Carpenter, and D. Lockard, “Re-evaluation of an optimized second order backward difference (bdf2opt) scheme for unsteady flow applications”, in *48th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, 2010, p. 122.

[215] Z. J. Ernst, A. M. Hickey, B. E. Robertson, and D. Mavris, “The effect of roll rate on simulated entry vehicle ballistic range tests”, *Journal of Spacecraft and Rockets*, pp. 1–12, 2022.

[216] J. Davis, A. Dwyer Cianciolo, R. Powell, J. Shidner, and E. Garcia-Llama, “Guidance and control algorithms for the Mars entry, descent and landing systems analysis”, in *AIAA/AAS Astrodynamics Specialist Conference*, 2010, p. 7972.

[217] R. L. Nelson, D. A. Price, and F. H. Delpino, “A new concept for controlled lifting entry flight experiments”, 1967.

[218] T. Petsopoulos, F. J. Regan, and J. Barlow, “Moving-mass roll control system for fixed-trim re-entry vehicle”, *Journal of Spacecraft and Rockets*, vol. 33, no. 1, pp. 54–60, 1996.

[219] Y. Wang, J. Yu, Y. Mei, L. Wang, and X. Su, “Nonlinear dynamics of fixed-trim reentry vehicles with moving-mass roll control system”, *Journal of Systems Engineering and Electronics*, vol. 27, no. 6, pp. 1249–1261, 2016.

[220] L. Wang, J. Yu, Y. Mei, and Y. Wang, “Guidance law with deceleration control for moving-mass reentry warhead”, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 230, no. 14, pp. 2639–2653, 2016.

[221] R. D. Robinett III, B. R. Sturgis, and S. A. Kerr, “Moving mass trim control for aerospace vehicles”, *Journal of Guidance, control, and Dynamics*, vol. 19, no. 5, pp. 1064–1070, 1996.

[222] P. Menon, G. Sweriduk, E. Ohlmeyer, and D. Malyevac, “Integrated guidance and control of moving-mass actuated kinetic warheads”, *Journal of Guidance, control, and Dynamics*, vol. 27, no. 1, pp. 118–126, 2004.

[223] J. Balaram, “Sherpa moving mass entry descent landing system”, in *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection, 2005, pp. 63–79.

[224] R. Mukherjee and J. Balaram, “Attitude dynamics and control of moving mass multibody aeromaneuver vehicle”, in *AIAA atmospheric flight mechanics conference and exhibit*, 2008, p. 6390.

[225] C. H. Murphy, “Influence of moving internal parts on angular motion of spinning projectiles”, *Journal of Guidance and Control*, vol. 1, no. 2, pp. 117–122, 1978.

[226] J. Rogers and M. Costello, “Control authority of a projectile equipped with a controllable internal translating mass”, *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 5, pp. 1323–1333, 2008.

[227] ——, “A variable stability projectile using an internal moving mass”, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 223, no. 7, pp. 927–938, 2009.

[228] ——, “Cantilever beam design for projectile internal moving mass systems”, *Journal of dynamic systems, measurement, and control*, vol. 131, no. 5, 2009.

[229] R. V. Jategaonkar, *Flight vehicle system identification: a time domain methodology*. American Institute of Aeronautics and Astronautics, 2006.

[230] M. Schoenenberger and E. M. Queen, “Limit cycle analysis applied to the oscillations of decelerating blunt-body entry vehicles”, 2008.

[231] J. J. Moré, “The Levenberg-Marquardt algorithm: Implementation and theory”, in *Numerical analysis*, Springer, 1978, pp. 105–116.

[232] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Deep learning toolbox user’s guide*, MathWorks, 2022.

[233] P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-validation.”, *Encyclopedia of database systems*, vol. 5, pp. 532–538, 2009.

[234] SAS Institute, *JMP 13 Predictive and Specialized Modeling*. SAS Institute, 2017.

[235] *Optimization toolbox user’s guide*, Mathworks, Mar. 2022.

[236] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, “Design and analysis of computer experiments”, *Statistical science*, pp. 409–423, 1989.

[237] T. J. Santner, B. J. Williams, W. I. Notz, and B. J. Williams, *The design and analysis of computer experiments*. Springer, 2003, vol. 1.

[238] S. N. Lophaven, H. B. Nielsen, J. Søndergaard, *et al.*, *DACE: a Matlab kriging toolbox*. Citeseer, 2002, vol. 2.

[239] J. V. Foster, “Autonomous guidance algorithms for NASA learn-to-fly technology development”, in *2018 Atmospheric Flight Mechanics Conference*, 2018, p. 3310.