5-2023

Augmented State Linear Covariance Applications for Nonlinear Missile Engagements

Jeffrey Scott Clawson
Utah State University

AUGMENTED STATE LINEAR COVARIANCE APPLICATIONS FOR NONLINEAR MISSILE ENGAGEMENTS

by

Jeffrey Scott Clawson

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Aerospace Engineering

Approved:

David Geller, Ph.D. Major Professor	Randy Christensen, Ph.D. Committee Member
Matt Harris, Ph.D. Committee Member	Stephen Whitmore, Ph.D. Committee Member
Greg Droge, Ph.D. Committee Member	Doug Hunsaker, Ph.D. Committee Member

D. Richard Cutler, Ph.D.
Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023

ABSTRACT

Augmented State Linear Covariance Applications for Nonlinear Missile Engagements

by

Jeffrey Scott Clawson, Doctor of Philosophy

Utah State University, 2023

Major Professor: David Geller, Ph.D.
Department: Mechanical and Aerospace Engineering

Sustained actuator saturation is a common nonlinearity for the terminal homing phase of a missile engagement. This nonlinearity creates some difficulty for high-fidelity linear analysis methods. This dissertation investigates three methods of modeling actuator saturation in an augmented state linear covariance simulation. In contrast with sample-based methods, the augmented state linear covariance tools from this dissertation can calculate the truth state dispersion covariance and the estimation error covariance throughout the engagement in a single run. They also provide several advantages over other linear missile engagement analysis methods such as the adjoint technique or traditional covariance analysis. First, a linear covariance simulation framework is developed and validated for a target engagement scenario without actuator saturation. The engagement comprises a single interceptor equipped with an inertial navigation system aided by absolute position measurements, as well as range, angle, and range rate measurements relative to the incoming target. The interceptor uses a proportional navigation guidance law to engage the target. The linear covariance framework is developed by linearizing the differential and measurement equations about the nominal trajectory and forming an augmented system comprising truth and navigation state dispersions. Next, saturations are introduced to the problem, along with the first analysis method: statistical linear covariance analysis. This method combines the augmented state linear covariance framework with the statistical linearization technique. The second method considered is tunable linear covariance analysis. Tunable linear covariance analysis utilizes a switching parameter to determine when to switch the dynamics of the problem

from unsaturated mode to saturated mode. The final method is called event trigger linear covariance analysis. This method involves switching GN&C modes using a constraint equation and a covariance shaping matrix. All three analysis methods are validated using Monte Carlo methods, and statistical linear covariance analysis is found to be the most robust and accurate of the three methods. The computational efficiency of this method is exploited to rapidly analyze missile engagement performance under varying levels of saturation. The parameters of the analysis include guidance laws, sensor accuracy levels, target evasive maneuvers, and actuator responsiveness levels.

(219 pages)

PUBLIC ABSTRACT

Augmented State Linear Covariance Applications for Nonlinear Missile Engagements

Jeffrey Scott Clawson

Sustained actuator saturation is a common occurrence for missile engagements. The saturation nonlinearity creates some difficulty for high-fidelity linear analysis methods. This dissertation investigates three methods of modeling actuator saturation in an advanced linear analysis. The linear covariance tools from this dissertation run extremely fast and provide several advantages over other linear missile engagement analysis methods. First, a simulation is developed and validated for a target engagement scenario without actuator saturation. Next, saturations are introduced to the problem, along with the first analysis method: statistical linear covariance analysis. This method combines the augmented state linear covariance framework with the statistical linearization technique. The second method considered is tunable linear covariance analysis. Tunable linear covariance analysis utilizes a switching parameter to determine when to switch the dynamics of the problem. The final method is called event trigger linear covariance analysis. This method involves switching GN&C modes using a constraint equation and a covariance shaping matrix. All three analysis methods are validated using Monte Carlo methods, and statistical linear covariance analysis is found to be the most robust and accurate of the three methods. This method is utilized to rapidly analyze missile engagement performance under varying levels of saturation. The parameters of the analysis include guidance laws, sensor accuracy levels, target evasive maneuvers, and actuator responsiveness.

To my teammate, my buddy, my ray of sunshine, and the little joys of my life

ACKNOWLEDGMENTS

Splendid, rising o'er the mountains
Glowing with celestial cheer
Streaming from eternal fountains
Rays of living light appear

-John Jaques

As I come to the end of this research, I have had time to reflect on my academic and life journey thus far. I think the great navigator, Klaes Ashford, said it well when he said a man is "not just himself, he is everyone who cares for him." Along these lines, it would be remiss of me not to recognize the caregivers, mentors, motivators, inspirations, and precious people in my life.

I was motivated at a young age to explore math and science by my paternal Grandfather Alden Ernest Clawson, whom the angels laid to rest in the year 2021. Each time I visited him he gave me math challenges to solve, while pretending not to know the answers so that way I would teach him about the subject. Over the years he told me fascinating stories about the early space and missile programs on which he worked.

Around age 7, I remember a few specific career conversations with my maternal Grandfather, Delbert Ray Bills, whom the angels laid to rest in the year 2022. Grandpa Bills told me all about his electrical engineering career, and how engineering is a great path to a rewarding career. From that time forward I knew I wanted to be an engineer when I grew up.

I would like to acknowledge the influence that my mother has had in my life. She nurtured me in my youth, made me feel loved during hard times, and taught me the importance of helping the less fortunate and downtrodden. This has led directly to my success in many areas of life. I always try to emulate her parenting style and love my children as much as she loved hers.

I love my wife, Sarah, and my two children, Leo and Lani. Home for me is experiencing the glowing radiance of my wife's countenance, a run-and-jump hug from my son, or a

smile-and-giggle from my daughter. I hope that as my children grow up they know that being a caregiver to them has brought me a transcendent joy that is imprinted on my soul.

Dr. David Geller played a vital role in my academic development over the past ten years. His availability, demeanor, and excitement for teaching led me to choose USU for grad school and eventually move to Logan. I learned everything I know about space and satellites from his classes. He always went out of his way to be a good mentor, even if it meant extra work for him. After our research meetings, I always felt uplifted and excited to get started on the next chapter.

I owe a debt of gratitude to my Space Dynamics Lab mentor, John Brewer. He motivated me to progress in my studies and enjoy the journey along the way. Not only did he involve me in rewarding work projects, he became my best friend. I enjoyed all the technical discussions, invention drawings, philosophical discussions, scouting routes from the office window, manual labor projects, back-country exploration, motorcycle adventures, and ski races.

I am grateful to God and Country for allowing me the opportunity to learn over the years. I have been raised in loving principles, instructed in productive academic methods by exceptional teachers, all while being able to live in two beautiful mountain havens. The splendid rays of light peering through a mountain meadow will always be a marvel and an inspiration to me.

Finally, I am grateful for the intellectual challenge of the unknown. I can only hope for the chance to meet and solve immeasurable unknowns in the expanse of the eternities.

Jeffrey Scott Clawson

CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOTATIONS

Variables

$\mathbf{a}$	acceleration vector
$a_{m,un}$	unlimited missile acceleration
a_m	actual missile acceleration
$\hat{a}_c$	commanded missile acceleration
$\mathbf{b}$	sensor bias vector
C_A	covariance matrix of the augmented state vector
d_{miss}	miss distance
D_{true}	covariance matrix of the truth state dispersions
$\mathbf{f}_s$	specific force vector
$\check{f}$	statistical linearization expectation vector
H	measurement equation partial derivative
k_p	proportional gain of the control law
K	Kalman gain
m	sum of the truth and nav states in the sim
N	statistical linearization describing function matrix
n	number of truth states in the sim
$\hat{n}$	number of nav states in the sim
n_y	number of continuous inertial measurements in the sim
n_z	number of discrete measurements in the sim
$\mathbf{n}_c$	missile desired acceleration vector
P_{true}	covariance matrix of the true estimation errors
$\hat{P}$	Extended Kalman Filter covariance matrix
$\mathbf{r}$	position vector

S	power spectral density
T_i^a	DCM transforming the inertial coordinate frame to the along-track frame
$\hat{\mathbf{u}}$	actuator input acceleration vector
$\overline{\mathbf{u}}$	nominal actuator input acceleration vector
$\mathbf{v}$	velocity vector
V_c	closing velocity between the missile and the target
w	process noise
$\mathbf{x}$	truth state vector
$\hat{\mathbf{x}}$	navigation state vector
$\overline{\mathbf{x}}$	nominal truth state vector
$\mathbf{x}_n$	true-navigation state vector
$\tilde{\mathbf{y}}$	continuous inertial measurement vector
$\overline{\mathbf{y}}$	nominal continuous inertial measurement vector
$\tilde{\mathbf{z}}$	discrete measurement vector
$\overline{\mathbf{z}}$	nominal discrete measurement vector
α	acceleration time constant reciprocal
η	continuous measurement noise
θ	crossing angle
λ	line-of-sight angle
ν	discrete measurement noise
σ_m^2	acceleration maneuver variance
τ	sensor bias time constant
$\mathbf{X}$	augmented state vector
ψ	heading angle

Subscripts

acc	accelerometer sensor parameter

at	along-track coordinate frame
e	expected value
f	final time value
gps	GPS sensor parameter
i	inertial coordinate frame
los	line-of-sight sensor parameter
range	range sensor parameter
rel	relative parameter
rr	range rate sensor parameter
s	singer noise parameter
ss	steady state sensor variance
t	target parameter
0	initial time value

ACRONYMS

APN	augmented proportional navigation
CADET	covariance analysis describing function technique
CSDA	complex step derivative approach
DCM	direction cosine matrix
GN&C	guidance, navigation, and control
ECRV	exponentially correlated random variable
EKF	extended Kalman filter
IMU	inertial measurement unit
LinCov	augmented state linear covariance
MC	Monte Carlo
NCV	nominal constant velocity
NRT	nominal reference trajectory
ON	optimal navigation
P_k	probability of kill
PDF	probability density function
PN	proportional navigation
PSD	power spectral density
QLSDM	the quasi-linear system dynamics matrix
RIDF	random input describing function
RPO	rendezvous and proximity operations
RSO	resident space object
SL	statistical linearization
SLAM	statistical linearization adjoint method
TCA	traditional covariance analysis
TOCA	time of closest approach
UAV	unmanned aerial vehicle
ZEM	zero effort miss

CHAPTER 1

INTRODUCTION

> "It is a characteristic of the engineering profession that there are always new problems waiting in the wings to replace the old ones as they are solved. The constant element is the challenge itself, which demands that we do our utmost and rewards us only when we do."
>
> -Mike W. Fossier

Trajectory error analysis is an important aspect of closed-loop guidance, navigation, and control (GN&C) problems. For a target engagement problem, it is critical to know how uncertainties in the initial state, dynamics, sensor models, and actuator models affect the knowledge of where the pursuing vehicle is and how far it deviates from its desired path, especially at the time of closest approach. For high fidelity analysis, this complex problem is often analyzed with the Monte Carlo method: running a simulation thousands of times and calculating the ensemble statistics of the truth dispersions and estimation errors. For analyzing a target engagement in a single run there are two fundamental methods: the adjoint technique [1] and traditional covariance analysis [2, 3]. There are also variations on these methods, including the Covariance Analysis Describing Function Technique (CADET) [4, 5], and the Statistical Linearization Adjoint Method (SLAM) [6]. This research presents an extension of traditional covariance analysis called augmented state linear covariance analysis [7] and applies it to the target engagement problem. Augmented state linear covariance analysis, referred to as LinCov in this research, provides several advantages to traditional covariance analysis methods.

Monte Carlo analysis is an important tool for closed-loop target engagement problems because it accurately simulates nonlinear, time-varying, noise driven systems. The disadvantage to Monte Carlo analysis is the extensive computing time. If a quick assessment of the preliminary system performance is desired, an adjoint analysis is effective (See ch. 3 of [8]). This technique utilizes the impulse response of the system to calculate the final miss distance of the engagement. However, adjoint analysis is limited because it does not track vehicle

state uncertainty throughout the flight and it cannot incorporate an extended Kalman filter (EKF). The CADET method is useful for target engagement simulations because it employs statistical linearization (SL), which is well-suited for actuator saturation [9]. The SLAM method combines this SL technique with the adjoint method to analyze nonlinear systems and compute miss distance error budgets.

Traditional covariance analysis addresses the gaps of adjoint analysis by calculating statistical performance for every system state throughout the flight and allowing EKFs in the system. Traditional covariance analysis is accomplished by propagating and updating a state covariance matrix for the duration of the flight. One drawback to traditional covariance analysis is that it does not model the effect of state estimation errors on truth trajectory dispersions. Nor does it account for continuous inertial measurements (gyro and accelerometer) [10]. These effects are critical for accurate analysis of modern target engagement systems. The linear covariance (LinCov) method used in this research successfully addresses these limitations.

A missile engagement has three main phases: launch phase, mid-course guidance phase, and terminal homing phase. Figure 1.1 illustrates the different phases of flight. The estimated path of flight of the target is shown in a blue dotted line. This figure assumes a nominal constant velocity (NCV) target flight. The launch phase refers to the process of launching the interceptor missile from a vehicle or from the ground, stabilizing the flight, and activating the sensors. This phase is shown in orange. It may include time to characterize the system biases, and confirm communication with external systems as well. The mid-course guidance phase of flight, shown in red, is the longest phase and it may have several segments. The general idea of this phase is to guide the interceptor toward the target by conducting a path following strategy for the desired approach angle. The interceptor maintains course and heading knowledge using gyros, accelerometers, and GPS measurements.

A common strategy for mid-course guidance is to set up a coarse "zero-effort miss" (ZEM) engagement, meaning that the vehicles will collide if they continue to fly without accelerating. This method minimizes the amount of actuation needed at the end of the

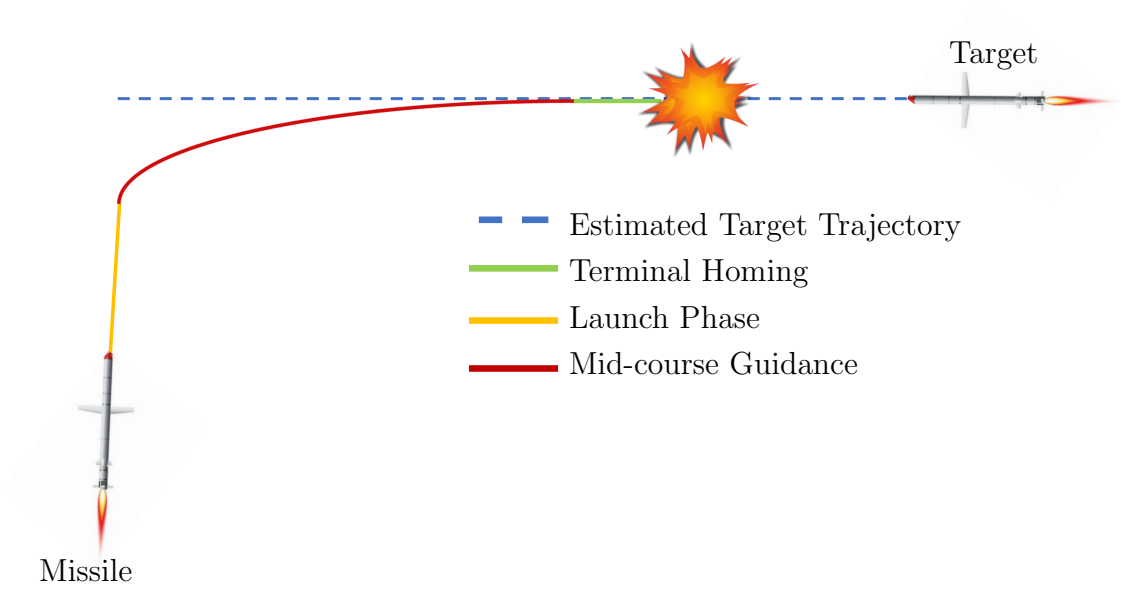

Fig. 1.1: Missile Engagement Phases

engagement. Any misalignment that remains at the end of this phase can then be eliminated during terminal homing. Another common strategy is to set up a direct head-on or tail-aligned engagement. Figure 1.1 shows a head-on engagement. This approach simplifies the dynamics of the missile flight by decoupling the actuation direction from the target velocity direction. A missile generally steers by actuating tail fins, which causes an acceleration perpendicular to the velocity vector of the missile. This acceleration input into the system is small compared to the velocity of the missile. If the engagement is a head-on engagement, the fins only have to overcome cross-track movements, which are much slower than along track movements. The mid-course guidance phase ends when the interceptor is "close" to an engagement. This can occur at a specified "time-to-go" in the engagement, when it enters a specified range to the target, or when it enters the range of a specific sensor.

The final phase of flight, terminal homing, is shown in green. It is short in length (often less than 15 seconds [11]), however, this phase is where the GN&C performance of the interceptor is the most crucial. In this phase, all parts of the GN&C hardware are actively being used, and must be working together for a successful intercept. The terminal

guidance law is activated in the system, guiding the interceptor toward a collision. The interceptor also relies on estimating the relative states of the vehicles. The accelerometers, gyros, and GPS measurements inform the interceptor state, while relative measurements like range, range rate or line-of-sight angle inform the relative states. The actuation of the interceptor must react quickly and be well characterized for a successful intercept. Figure 1.1 shows a non-maneuvering target, however, it is common for a target to implement an evasive maneuver close to the time of intercept. Figure 1.2 shows an example of an evasive maneuver by the target.

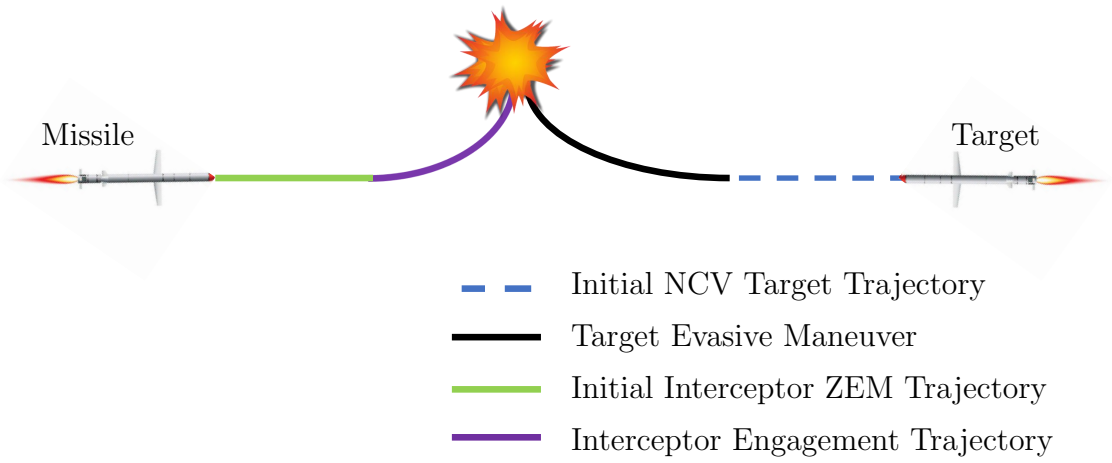

Fig. 1.2: Terminal Homing Example

Figure 1.3 illustrates the times when various GN&C sensors and processes are active during each portion of an engagement. The solid lines represent different phases of flight, while the dotted lines represent a specific sensor or process and when it is active. For example, the Extended Kalman Filter, inertial measurement unit (IMU), and GPS measurements are assumed to be running the entire flight. Note that this is just an example problem, and each engagement problem has its own GN&C configuration. This example shows two flight phases

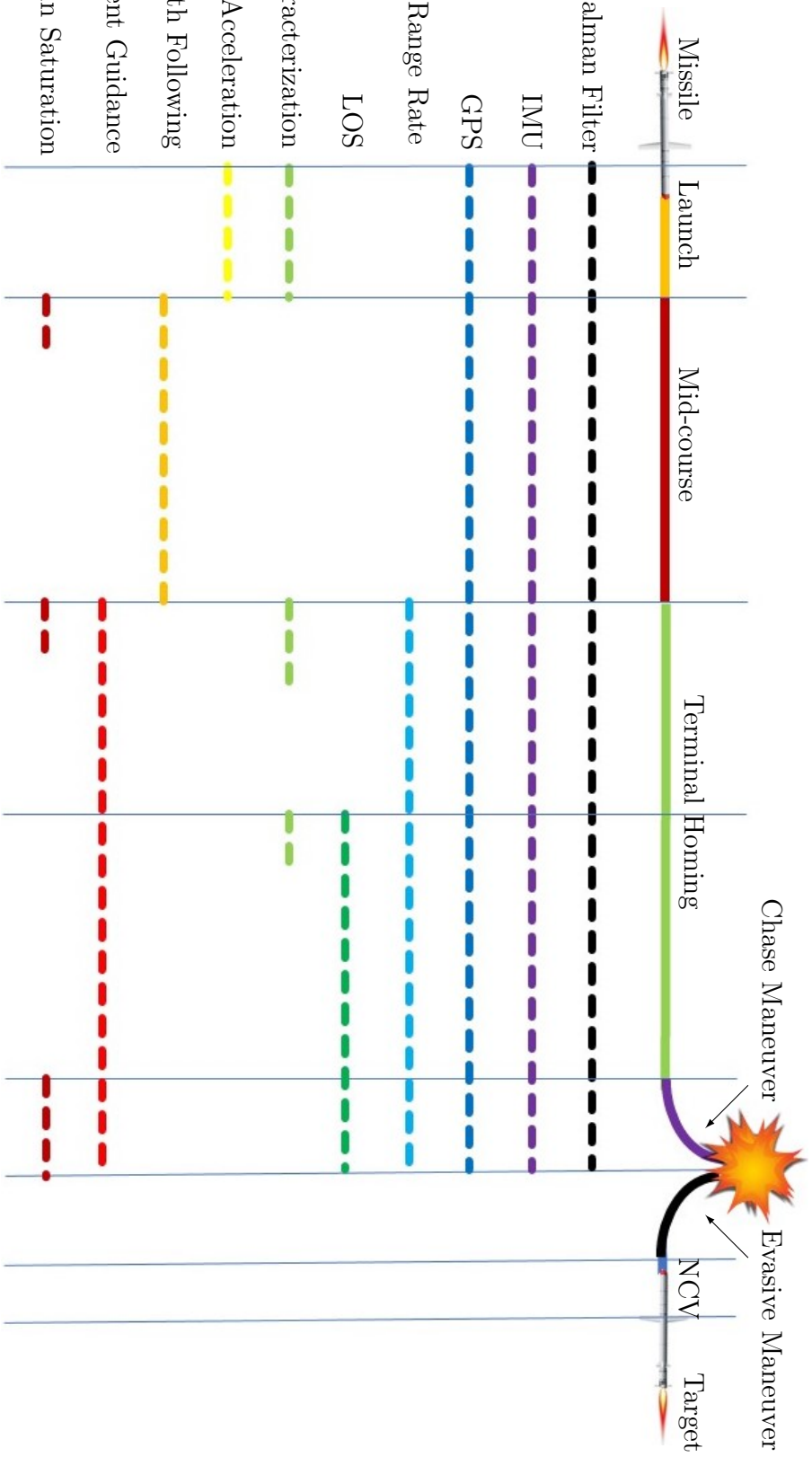

Fig. 1.3: Flight System Activation

for the target: nominal constant velocity (NCV) at first, and an evasive maneuver at the end. The interceptor has the same phases of flight as listed previously: launch, mid-course, and terminal homing. The terminal homing phase is broken up into three segments: the initial sensor phase, the full sensor phase, and the "chase maneuver" response to the target evasive maneuver. This is to simulate relative sensors being activated at different ranges or coming from different sources. In this example problem, the range and range rate measurements are available before the line-of-sight measurements.

During the launch phase, the vehicle undergoes its initial acceleration, and the Kalman Filter begins to estimate the system biases. This can include GPS bias, and IMU biases/misalignments. Launch phase also might include test maneuvers by the vehicle to characterize the actuator behavior and calculate trim conditions. It is assumed that once the interceptor reaches maximum speed, it continues at the same velocity for the duration of the flight. During the mid-course phase, the interceptor has an accurate estimate of its position, velocity, orientation, and angular velocity from the inertial sensors. The vehicle follows a specified path to align itself with the velocity vector of the target, setting up a head-on collision. The interceptor missile steers by actuating fins on the rear of the vehicle. This is a called a "bank to turn" or "skid to turn" vehicle steering system [12, 13]. Figure 1.3 also shows several times that the actuator fins are "saturated". This means that the rear fins are moved to the maximum range, creating the maximum angle of attack or turning at maximum g's. This can happen immediately following guidance mode changes or if the target performs an evasive maneuver. GN&C behavior before, during, and after actuator fin saturation is a crucial element of this dissertation. At the beginning of the mid-course phase, the figure shows that a maximum g turn can occur as the interceptor begins its path following trajectory.

Terminal homing is the most complex phase of the target engagement. The biggest changes are that the relative measurements are added to the filter, and the engagement guidance law is activated. These are active until the end of the engagement. It is also possible to tune the filter specifically for terminal homing performance by over-weighting or

under-weighting certain measurements. When the new measurements are activated in the system, there is a time when the filter is estimating the biases of the measurement. This might lead to decreased performance at initial implementation, but will increase if the filter can successfully estimate the bias. This is indicated by the light green dotted line at the beginning of the terminal homing phase.

The first segment of the terminal homing phase begins when relative range and range-rate measurements are activated. The engagement guidance law for terminal homing is typically some form of proportional navigation guidance law, meaning that it is dependent on the flight computer's estimate of the relative states. More advanced guidance laws include optimal terminal guidance (see [14], ch. 5 of [15] and ch. 26 of [8]) and modified proportional guidance (see ch. 5 of [16]). An actuator fin saturation can also happen at the beginning of terminal homing.

Segment two of terminal homing begins when the line-of-sight measurement is activated in the simulation. There is a short time of bias characterization at the start of this segment. The engagement guidance law is still active, but should be set up for a ZEM if the target is not maneuvering.

Segment three of the terminal homing phase, labeled the "chase maneuver" in fig. 1.3, happens as the target initiates an evasive maneuver right before impact. During this chase maneuver, the missile has all measurement systems active and calibrated. The engagement guidance commands a maximum g turn to maintain the intercept, saturating the fin actuator. Depending on the timing of the maneuver and the maneuverability of the interceptor, the target may or may not be able to escape lethal range at the time of closest approach. This final segment of flight is difficult to model with linear methods, and is discussed at length in this dissertation.

Figure 1.4 is a graphical representation of a 2-D missile/target scenario, and gives more insight into what the Monte Carlo and LinCov simulations are analyzing. The target vehicle on the right travels with a nominally-constant velocity, perturbed by accelerations, toward some desired final location. The nominal path of the target is represented by the blue curve

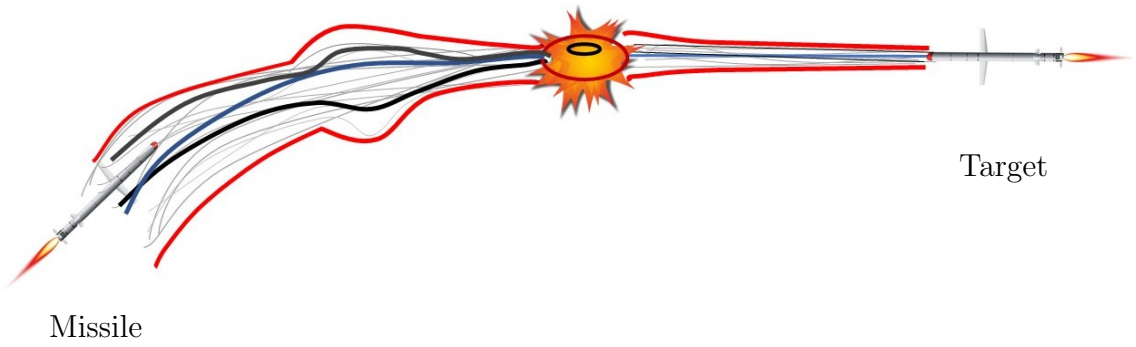

Fig. 1.4: Linear Covariance Target Engagement

on the right, while the 3σ position dispersion lines are shown in red. The position of the target is defined to have initial uncertainty, so the trajectory dispersion envelope of the target does not start at zero. The acceleration disturbances can be modeled by zero mean white noise with a specified power spectral density, which is why the position dispersions of the target grow with time. By definition, 99.7% of the target trajectories lie within the 3σ bounds. The objective of the missile on the left is to disable or destroy the target by intercepting it. The interceptor has greater maneuverability than the target and is aided by IMUs, inertial sensors, and relative sensors. Approximately midway through the flight, the missile changes to terminal homing mode, and has a short period of re-calibration before continuing to the end of the engagement. The nominal path of the interceptor is represented by the blue line on the left. This line is not affected by the mode change because the mid-course guidance puts the missile into a ZEM trajectory such that it will collide with the target.

The missile truth dispersion covariance lines are shown in red. The gray "hair lines" for both vehicles represent multiple runs of a Monte Carlo simulation. Each realization of the target follows one particular path, and the corresponding realizations of missile motion are illustrated as the gray lines on the left. The black lines represent the true estimation error covariance. The final relative position dispersions and estimation errors of the two vehicles are represented with the red and black ellipses, respectively. The ideal engagement scenario

results in small truth dispersions and small estimation errors. This corresponds to low miss distances and a higher probability of intercept. For example, a system with highly accurate measurements but low-fidelity dynamic models and low capability actuators will have small estimation error and large truth dispersions. Alternatively, a high fidelity dynamic model and high capability actuator coupled with bad sensors will have large estimation error.

During a typical Monte Carlo simulation, thousands of runs are simulated, and the end result is to calculate the covariance of the truth dispersions and true estimation error of the system throughout the engagement. By linearizing the entire closed-loop GN&C system about the nominal trajectory, an analysis tool like LinCov is able to calculate the same results with one simulation run. Among single run analysis tools, LinCov analysis is uniquely capable of analyzing how highly-coupled models, sensors, actuators, and GN&C algorithms will affect trajectories during flight.

CHAPTER 2
OVERVIEW AND THESIS STATEMENT

2.1 Research Objectives

The goal of the proposed research is to extend the application of augmented state linear covariance theory and event triggers to a saturated actuator missile engagement problem. The research objectives are as follows:

1. Show that augmented state linear covariance theory (LinCov) can effectively model actuator saturation using event triggers for missile engagement problems by:
   (a) Building and validating a missile engagement LinCov simulation
   (b) Comparing LinCov results to statistical linearization results
   (c) Investigating LinCov performance under varying levels of saturation

2.2 Scope

The research will be conducted by creating a Monte Carlo and Linear Covariance simulation of a missile engagement with actuator saturation and appropriate event triggers. The type of actuator saturation under consideration is a sustained saturation close to the intercept time. An evasive maneuver by the target triggers the missile actuator saturation. The model will incorporate process noise to model random accelerations, discrete measurement noise, continuous inertial measurement noise, and linearized dynamics and measurements for covariance propagation and updates. The event triggers will allow the system to switch acceleration modes at non-fixed times. The event trigger will be a change in the dynamics of the problem, not a change in the guidance law. For example, during an actuator saturation, the missile acceleration will become a constant instead of proportional guidance. The LinCov results will be compared to statistical linearization, which is a common approach to analyzing this type of problem.

2.3 Thesis Statement

Linear Covariance techniques can be applied to nonlinear target engagement problems to accurately model truth trajectory dispersions and miss distances, including situations where the interceptor experiences GN&C mode changes and actuator saturation.

2.4 Dissertation Overview

The dissertation is laid out as follows: Chapter 3 provides a literature survey of relevant research and briefly describes how it applies to nonlinear target engagement problems. Chapters 4-5 develop the general theory used for LinCov, event triggers, statistical linearization, and CADET models. This theory is for a general GN&C problem, and can be applied to any aerospace GN&C architecture. Chapter 6 introduces a missile engagement LinCov simulation without saturations. Chapters 7 and 8 define and validate three LinCov simulation methods for dealing with saturations in a missile engagement problem. These chapters define the dynamics, sensors, linearization technique, system partial derivatives, and validation methods for the nonlinear target engagement problem. The three LinCov simulations are called Statistical LinCov, Tunable LinCov, and Trigger LinCov. Chapter 9 utilizes the newly developed Statistical LinCov to perform a trade study of various missile engagement scenarios. The final conclusions of the research are summarized in Chapter 10, and Chapter 11 lists a few potential research applications for Statistical LinCov

CHAPTER 3
LITERATURE SURVEY AND RELATED WORK

3.1 Linear Covariance

Augmented State Linear Covariance theory (LinCov) as it is presented in this paper was developed and published by Dr. David Geller in the early 2000s [10, 17]. Since that time, LinCov theory has been applied to various aerospace systems for analysis. This section will review a few precursors to LinCov, discuss relevant LinCov literature, and describe how it can be applied to missile engagements.

3.1.1 Basic Linear Covariance Analysis

Basic linear covariance analysis (Basic LinCov) methods have been consistently used in space programs since the 1960s [18, 19]. The term "covariance analysis" however, may refer to several different analysis methods. In the literature, the names for each method are inconsistent and the differences between the various methods are not always explained. The following paragraphs will review the most common covariance analysis types and how they are used for aerospace GN&C problems.

The most straightforward application of covariance analysis, dispersion analysis, is accomplished by studying the effect of environmental disturbances on trajectory dispersions. The foundation for this method was established by Norbert Wiener in the 1940s [20]. He showed that the statistical properties of a stochastic process disturbance can be quantified by using the power spectral density (PSD) of the signal. The basic steps of this method are characterizing the disturbances, and propagating the growth of the dispersions forward in time [21]. A common starting point for this method is to assume the disturbance is zero-mean Gaussian white noise and calculate the power spectral density of the noise. Depending on the fidelity required, a first-order or second-order Markov process may also be used (see ch. 4 of [22]).

Another routine application of covariance analysis, estimation error analysis, is to investigate the estimation error covariance created in the state space by imperfect system measurements. The estimate of the state vector and its associated covariance is propagated forward in time, and updated each time a measurement is processed. This process is derived and explained extensively in [22] and [3]. Both the dispersion analysis method and the estimation error method are considered "open loop" methods, as they do not take into account the effect of closed-loop guidance and control on the system. A common implementation of estimation error analysis is the Kalman Filter for linear systems or the Extended Kalman Filter for nonlinear systems.

In the early 1960s Kalman and Bucy expanded on Wiener's research [18,20] by developing a linear state space formulation in the discrete time domain [23] and in the continuous time domain [24]. The optimal estimation filter they created, called the Kalman Filter, was able to give the optimal state estimate and covariance of a linear system. Around the same time, Stanley Schmidt from NASA Ames Research Center was working on the nonlinear circumlunar mid-course navigation and guidance problem for the Apollo program. Schmidt was able to see a connection between linear perturbation theory and Kalman's linear filtering theory [25]. By linearizing a system about a nominal trajectory, his team was able to apply the Kalman filter to a nonlinear system. This approach is called an extended Kalman filter. Using Monte Carlo tests to validate the results, Schmidt and his team were able to show that the extended Kalman filter state estimate converged to the truth state, even with large initial state uncertainty. The extended Kalman filter, EKF variations, and Monte Carlo validation method are still vital tools in the aerospace GN&C field today.

Unmodeled disturbances or bad measurements can cause a Kalman Filter to diverge in its state estimation, therefore filter tuning is another important category of covariance analysis. The art of filter tuning is a balance of maximizing filter performance and avoiding filter divergence/smugness. This method can be used both for simulated truth problems, or real-world hardware problems. In a simulated truth problem, the truth dynamics are usually higher fidelity than the design model dynamics so the Kalman filter will face unmodeled

disturbances. In a real-world hardware test, the system does not have access to truth data, only state estimates, so it will face unmodeled disturbances as well. Common tactics for filter tuning are adjusting the PSD of additive noise, adding states to the state vector, rejecting bad measurements, or over/under weighting measurements [2, 26].

It is often useful to conduct sensitivity studies in GN&C problems to discover how variations in parameters, sensor quality, navigation errors, or actuator execution will affect the trajectory dispersions. There are a few additional Basic LinCov analysis methods that can be categorized as sensitivity studies. One example is called consider analysis. Consider analysis involves evaluating the effect of uncertainty in unestimated system states, and increasing the state covariance to account for it [27, 28]. Sensor trade studies are a commonly conducted to decide which sensors meet budget and performance requirements [7]. Another extension of filter tuning is called true covariance analysis. True covariance analysis evaluates the sensitivity of the EKF to reduced order design models [29]. If the true estimation error of a system lines up well with the filter covariance, the lower order model is tuned properly. Finally, Battin developed a covariance method which studies the coupling of position and velocity estimation error to final time position dispersions (See ch. 14 of [30]).

The Basic LinCov analysis methods mentioned in this section are not able to fully evaluate the behavior of a closed-loop GN&C system. Recent publications show that for many aerospace systems, augmented state linear covariance analysis (LinCov) can accurately model the cumulative effects of navigation error, actuator error, sensor uncertainty, unmodeled disturbances, and inertial measurement errors on trajectory dispersions [10].

3.1.2 Applications of Augmented State Linear Covariance Analysis

LinCov theory can be summarized as follows: define the nonlinear system, linearize the system using perturbation theory and a first-order Taylor series approximation, define an augmented state vector containing truth dispersions and navigation dispersions, and finally, formulate the linear equations into the augmented state covariance equations [7]. By keeping both truth information and navigation information in the augmented system, LinCov is able to accurately model the cross correlation between navigation error and truth dispersions. It

fills a gap between time-consuming Monte Carlo methods and traditional simplified "single run" analysis methods. Although there is a significant amount of up front work to set up LinCov, it is a very powerful tool for modeling complex systems.

Over the past fifteen years, LinCov has been applied to many different types of space systems. LinCov is well suited to spacecraft navigation problems because these systems rely heavily on measurements when ground updates are not available. Geller's first LinCov paper shows that a flight computer can use LinCov to autonomously calculate and execute mid-course maneuvers while still maintaining dispersion requirements [10]. This led to papers on angles-only navigation orbital rendezvous [31], cislunar navigation [32], and interplanetary navigation [33]. Additional space related research topics include lunar GN&C [34–37], atmospheric entry [38], and precision pointing spacecraft systems [39].

Historically, missile engagements have been analyzed using Basic LinCov techniques, adjoint method, and Monte Carlo methods. This dissertation research extends the applications of LinCov analysis to the missile engagement scenario. The LinCov statistics can be used for error analysis throughout the flight, and miss distance analysis at the end of the flight [40, 41].

3.2 Missile Engagements

Guided missiles as we know them today originated around the time of the second world war. Early missiles were not very accurate, as they lacked sufficient sensors. They did not rely on the traditional GN&C algorithms we see today, but rather on predetermined flight programs for velocity direction and engine cutoff. The development of inertial navigation sensors from the 1920s to the 1950s spurred the advancement of missile flight and their ability to fly accurately. Walter Haeussermann, a German-born scientist who was instrumental in the development of rockets in the U.S., wrote an auto-biographical history of the developments that led to automatic missile guidance [42]. At first, there were two schools of thought on how to guide a missile: radio sensors or internal inertial sensors. Inertial sensors were less accurate than radio sensors at the time, but they eventually won out over radio sensors since they were self-contained, and not prone to signal jamming. It is notable that every missile

system relies heavily on inertial measurements, but many of the common missile analysis methods are not capable of modeling these sensors correctly [10]. Inertial sensor modeling is a strength of LinCov, and would address this oddity.

Around 1950, accelerometers and 3-axis gyros were developed enough for accurate missile guidance. The next challenge for guided missiles was to intercept maneuvering targets using homing guidance. A paper by Fossier [43] outlines the developments that led to radar homing missiles. Two important developments discussed in the paper are an internal continuous wave radar sensor and the proportional navigation guidance law.

It was a challenge for a missile to simultaneously transmit radar signals to bounce off the target and receive them back, especially with ground cluttering the signal. Additionally, an on-board radar sensor is very sensitive to "radome error", a signal disruption from the nosecone of the vehicle. Eventually, technology was developed to overcome these hurdles.

Missile guidance laws are strongly linked to line-of-sight rate between two vehicles. Figure 3.1 shows the line-of-sight angle λ. If the line-of-sight angle is not changing with

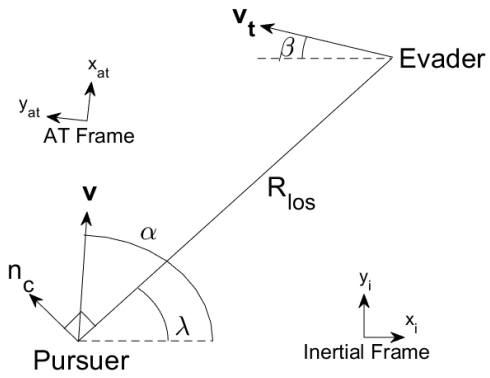

Fig. 3.1: Target Engagement Geometry

time, the interceptor is on a collision course with the target. The concept is put into practice with the following equation:

$$n_c = k_p V_c \dot{\lambda} \tag{3.1}$$

This equation is the basic proportional navigation missile guidance law and is still used today in homing missile systems. The effective navigation ratio, k_p, represents the ratio of interceptor acceleration capability to the target acceleration capability. This value should be at least three or four in order to prevent saturation at the end of flight. The other terms in equation 3.1 are related to the sensor outputs of the interceptor, line-of-sight rate $\dot{\lambda}$ and closing velocity V_c.

While the goal of the guidance equations is a direct hit on the target, in missile vs. missile engagements, a collision is a rare event [44]. Rather, a detonation is triggered when the missile passes within a specified distance of the target, and an ordnance is used to damage the threat. Lethal damage is inflicted on the target provided the missile is close enough and the geometry is appropriate for the fragmentation pattern of the ordnance. Probability of Kill (P_k) is the desired end metric, which is a function of miss distance covariance. The focus of this LinCov research is to provide an efficient means for calculating the miss distance covariance. Aerospace engagement problems have many different vehicle configurations and terminal constraints. Common vehicle configurations include single-pursuer/single-evader [45], multiple-pursuer/single-evader [46–48], and multiple-pursuer/multiple-evader [49]. The focus of this research is the statistical analysis of a single missile and single target system in the terminal homing phase of the engagement [50]. (In this dissertation, the term 'interceptor' and 'missile' are used interchangeably.) The mid-course maneuvers [51] are assumed to be finished as the simulation begins. The two vehicles are traveling toward each other on a near-collision course, with the missile maneuvering to intercept the target [52, 53] and the target maneuvering evasively [54]. The interceptor is equipped with an inertial navigation system aided by absolute position measurements, as well as range, angle, and range rate measurements relative to the target [55, 56]. The missile uses a proportional navigation guidance law to engage the target [57, 58]. Some examples of this might be a missile attempting to pursue a moving target [59–61], or an unmanned aerial vehicle (UAV) attempting to intercept another UAV [62–64]. A complex target engagement may include terminal constraints on the miss distance, crossing angle of the trajectories, and the body

pointing angle. References [65–70] provide suitable simulation equations to model these constraints.

3.3 Event Triggers

Mode changes based on state estimates are an important aspect of many GN&C systems. For missile engagements, actuator saturation is an important mode change to consider. A journal paper by Dr. Geller et al. incorporates these mode changes into LinCov by using event triggers [71]. This dissertation will refer to this type of analysis as Trigger LinCov. The example problem used in Geller's research is a terminal phase initiation maneuver for an orbital rendezvous. The maneuvering vehicle and the resident space object (RSO) are in coelliptic orbits, and the vehicle is below and behind the RSO. When the estimated elevation angle reaches the threshold value, a maneuver is executed to put the chaser on the v-bar of the RSO orbit. This type of maneuver trigger has been used in many space missions, including the Gemini and Apollo programs. The reason that mode changes can be difficult for any covariance analysis is that they do not happen at a fixed point in time. Depending on the disturbances for a given run, the trigger may happen slightly early or slightly late. For a Monte Carlo analysis, the state covariance will start to balloon once the earliest trajectory begins to change modes but the others haven't yet. It is more informative to "reset" the Monte Carlo clock to zero at the time of the mode change. A Trigger LinCov sim doesn't run multiple instances of the trajectory, so the event trigger theory linearizes the effect of resetting the clock by using a "covariance shaping matrix." The shaping matrix is calculated using the partial derivatives of the system and the constraint equation. These partial derivatives are also used to calculate the difference between nominal trigger time and specific run trigger time, and the variance of when the event could occur. For a complex system with multiple mode changes, it is possible to cascade multiple event triggers as long as the shaping matrix is applied at each event [72]. Geller's initial derivation for the event trigger equations makes the assumption that the truth state vector and the navigation state vector are the same size. Dr. Rose shows a more generic derivation of the event trigger equations in his dissertation that does not make this same assumption [36].

3.4 Basic Statistical Linear Covariance Analysis

One current method for approximating the statistics of nonlinear stochastic systems is the Covariance Analysis Describing Function Technique (CADET), which utilizes the principles of statistical linearization. Since CADET uses statistical linearization, but not an augmented state vector, it is categorized as Basic Statistical LinCov for this dissertation. Taylor's CADET handbook provides a comprehensive guide to applying the CADET method to missile guidance systems [73]. Basic Statistical LinCov methods provide a valuable comparison to Basic LinCov methods. This section summarizes statistical linearization and the CADET analysis method and the related literature.

A common nonlinearity in missile guidance problems is actuator saturation. (See fig. 3.2) One method for approximating a saturation operator is to use statistical linearization.

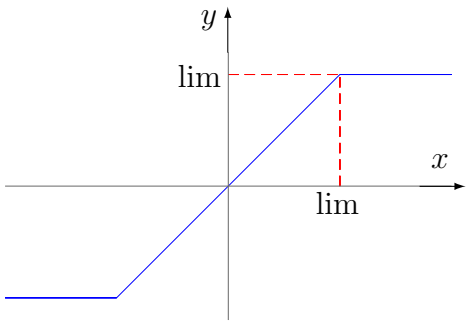

Fig. 3.2: Saturation Operator

The basic idea behind statistical linearization is to replace a nonlinearity in a system with an equivalent gain. Since the approximation still depends on properties of the input signal, it is sometimes call quasi-linearization [4]. This process is accomplished by minimizing the mean-squared error signal. A Booton paper on nonlinear control [74] derives the equations for statistical linearization, and the derivation is shown below.

Consider a nonlinear system with input $x(t)$ and output $y(t)$. It is desired to linearize the nonlinear portion of the system by replacing nonlinear equations with an equivalent gain

K_{eq}. An error equation can be defined as follows:

$$e = y - K_{eq}x \tag{3.2}$$

The next step is to solve for the gain K_{eq} to minimize the error. First, define the mean-squared value of the error equation in order to minimize the magnitude of the error:

$$e^2 = (y - K_{eq}x)^2 = y^2 - 2K_{eq}xy + K_{eq}^2x^2 \tag{3.3}$$

The value of K_{eq} that minimizes $e^2(t)$ is found by taking the derivative of e^2 with respect to K_{eq} and setting it equal zero.

$$\frac{\partial e^2}{\partial K_{eq}} = 2xy + 2K_{eq}x^2 = 0 \tag{3.4}$$

$$K_{eq} = \frac{xy}{x^2} \tag{3.5}$$

Assuming a probability density function (PDF) of $p(x)$, the expected value of K_{eq} is:

$$E[K_{eq}] = \frac{\int_{-\infty}^{\infty} xyp\,dx}{\int_{-\infty}^{\infty} x^2p\,dx} \tag{3.6}$$

Assuming that the input $x(t)$ is zero-mean Gaussian, the PDF is:

$$p(x) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left[\frac{-x^2}{2\sigma_x^2}\right] \tag{3.7}$$

If σ_x is defined as the rms value of x, then the denominator of equation 3.6 is simplified as:

$$\frac{1}{\sigma_x\sqrt{2\pi}} \int_{-\infty}^{\infty} x^2 \exp\left[\frac{-x^2}{2\sigma_x^2}\right] dx = \sigma_x^2 \tag{3.8}$$

Plugging this result into equation 3.6:

$$K_{eq} = \frac{1}{\sigma_x^3\sqrt{2\pi}} \int_{-\infty}^{\infty} xy \exp\left[\frac{-x^2}{2\sigma_x^2}\right] dx \tag{3.9}$$

Equation 3.9 is an important result because it represents the equivalent gain of a Gaussian input to a nonlinear function. It will approximate the behavior of the nonlinear function. This type of linearized gain equation is called a random input describing function (RIDF) [75]. RIDFs for many common nonlinearities are listed in [76].

Now that the general gain function has been derived, it is possible to plug in a specific nonlinear function. Zarchan shows the step-by-step process to calculating the RIDF for a limiter or saturation operator in [6]. The resultant equivalent gain equation is:

$$K_{eq} = 2\left[\frac{1}{\sigma_x\sqrt{2\pi}}\int_{-\infty}^{lim}\exp\left[\frac{-x^2}{2\sigma_x^2}\right]dx\right] - 1 \tag{3.10}$$

Equation 3.10 can be approximated by the following equations:

$$K_{eq} = 1 - \frac{2}{\sqrt{2\pi}}\exp\left[\frac{-lim^2}{2\sigma_x^2}\right]\left[c_1\omega - c_2\omega^2 + c_3\omega^3\right] \tag{3.11}$$

where

$$c_1 = .4361836 \;\; c_2 = .1201676 \;\; c_3 = .937298$$

and

$$\omega = \frac{1}{1 + \frac{.33267lim}{\sigma_x}}$$

Note that 'lim' refers to the limit of the saturation operator, shown in Fig. 3.2. Equations 3.11 and 3.10 show that the statistical linearization is dependent on the value of the limit, and the standard deviation of the input. Fig. 3.3 shows the behavior of the limiter RIDF: as the normalized standard deviation of the input $\frac{\sigma_x}{lim}$ goes to zero, the equivalent gain K_{eq} goes to one, with a roll off as the standard deviation increases.

One benefit of SL is that it does not require the derivatives of the nonlinear function [9]. This is very useful for discontinuous functions. The drawback is that PDF of the input must be known. This form of linearization is different than the LinCov linearization method, which uses a 1st-order Taylor series approximation.

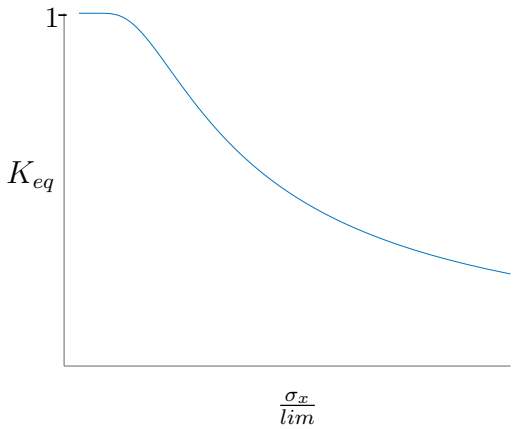

Fig. 3.3: Saturation Equivalent Gain Function

Basic Statistical LinCov, combines statistical linearization with Basic LinCov analysis. This method can also be called CADET [4]. The linearized system is fed into the covariance propagation and update equations. The CADET method is compared to the Monte Carlo method in [77], and used to estimate the P_k for an engagement in [78]. CADET can also be used in uncertainty propagation in orbital mechanics [79].

CHAPTER 4

LINCOV THEORETICAL DEVELOPMENT

This dissertation uses a Monte Carlo simulation to validate the LinCov simulations. The trajectories being studied in the target engagement problem have small dynamic uncertainties, making them impossible to model perfectly. A common way to simulate these uncertainties is to add disturbance accelerations to the dynamics, which limits the ability to analyze the scenario deterministically. Hence, sample-based (e.g., Monte Carlo) methods are employed for computing the statistics of the vehicle dispersion about the nominal trajectory. Figure 4.1 illustrates the flow diagram for a typical Monte Carlo system architecture. The

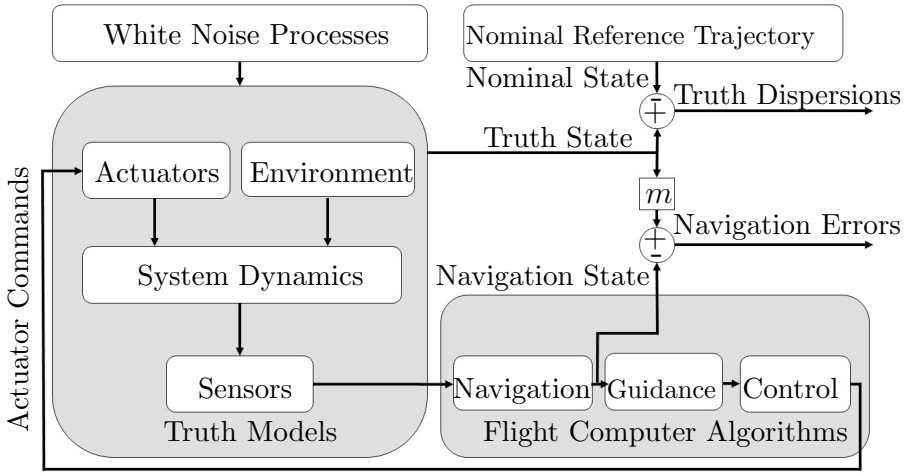

Fig. 4.1: Monte Carlo Simulation Architecture

truth state, $\mathbf{x}$, is simulated by the computer by adding noise to measurements and dynamics. The navigation state, $\hat{\mathbf{x}}$, is the flight computers' best estimate of the truth state, and is calculated by the flight computer using the output from the sensors. The truth or navigation dispersions are calculated by subtracting the truth state or navigation state from the nominal reference trajectory (NRT). The navigation errors are calculated by subtracting the truth state from the navigation state or the truth dispersions from the navigation dispersions. Note that this dissertation uses navigation error and navigation error interchangeably. If

the truth state vector has different states than the navigation state vector, the truth state is transformed to the true-navigation state, $\mathbf{x}_n$, using the mapping function $\mathbf{x}_n = \mathbf{m}(\mathbf{x})$. This allows a direct subtraction of the vectors. In the literature, the navigation error is also sometimes called the "true navigation error," indicating that the calculation directly accounts for the truth data. The flight computer maintains its own calculation of the navigation error called the filter covariance. This is calculated with an EKF, which has access to truth data through imperfect measurements. If the flight computer has an accurate model of the system dynamics and accurate measurements, the true navigation error and the filter covariance statistics are very similar. The subsystems in Figure 4.1 are further defined in this section and Chapter 6.

4.1 Nonlinear Modeling

In this research, the nonlinear equations are run in a Monte Carlo simulation and compared to a linear covariance simulation to show that the linearization of the problem is accurate. The general equations are shown below, and the specific models for the missile engagement problem are defined in Chapter 6. The dynamics are a function of the state, $\mathbf{x}$, actuator input, $\hat{\mathbf{u}}$, and disturbance accelerations, $\mathbf{w}$ in the form of process noise.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\hat{\mathbf{u}}) + B\mathbf{w} \tag{4.1}$$

Including process noise in the differential equations allows the simulation to model perturbations due to uncertainty in the dynamics. In general, the noise driving the system dynamics is zero-mean Gaussian white noise with PSD S_w.

$$E\left[\mathbf{w}(t)\right] = \mathbf{0} \tag{4.2}$$

$$E\left[\mathbf{w}(t)\mathbf{w}(t')^T\right] = S_w\delta(t-t') \tag{4.3}$$

where $\delta(t-t')$ is the Dirac delta function.

The Kalman filter has access to continuous inertial measurements (accelerometers and gyros) corrupted by zero-mean Gaussian white noise with Power Spectral Density S_η. A generic continuous inertial measurement equation is written as:

$$\tilde{\mathbf{y}} = \mathbf{c}(\mathbf{x}, \hat{\mathbf{u}}) + \boldsymbol{\eta} \tag{4.4}$$

With noise characteristics:

$$E\left[\boldsymbol{\eta}(t)\right] = \mathbf{0} \tag{4.5}$$

$$E\left[\boldsymbol{\eta}(t)\boldsymbol{\eta}^T(t')\right] = S_\eta \delta(t - t') \tag{4.6}$$

where $\delta(t - t')$ is the Dirac delta function.

The discrete measurements are composed of two parts: the measurement function $\mathbf{h}(\mathbf{x})$, and the measurement noise. The measurement noise ν is zero-mean Gaussian white noise with covariance R_ν. The general discrete measurement equations are:

$$\tilde{\mathbf{z}}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\nu}_k \tag{4.7}$$

$$E\left[\boldsymbol{\nu}_k\right] = \mathbf{0} \tag{4.8}$$

$$E\left[\boldsymbol{\nu}_k \boldsymbol{\nu}_{k'}^T\right] = R_\nu \delta(k - k') \tag{4.9}$$

where $\delta(t - t')$ is the Kronecker delta function.

The acceleration of the chaser is created by an actuator that is following a specific control law. The control law is a function of the estimated navigation state vector.

$$\hat{\mathbf{u}} = \hat{\mathbf{g}}(\hat{\mathbf{x}}) \tag{4.10}$$

4.1.1 Kalman Filter Equations

The navigation state vector propagation equation is a function of the navigation state vector and the continuous inertial measurements as is known as model replacement. The true navigation state $\mathbf{x}_n$ is the representation of the navigation state using the truth values. The function $\mathbf{m}(\mathbf{x})$ maps the truth state to the true navigation state.

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\hat{\mathbf{x}},\tilde{\mathbf{y}}) \tag{4.11}$$

$$\mathbf{x}_n = \mathbf{m}(\mathbf{x}) \tag{4.12}$$

The filter covariance is propagated with the following equation.

$$\dot{\hat{P}} = [\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}\hat{C}_{\hat{\mathbf{x}}}]\hat{P} + \hat{P}[\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}\hat{C}_{\hat{\mathbf{x}}}]^T + \hat{F}_{\tilde{\mathbf{y}}}\hat{S}_\eta\hat{F}_{\tilde{\mathbf{y}}}^T + \hat{B}\hat{S}_w\hat{B}^T \tag{4.13}$$

Where $\hat{F}_{\hat{\mathbf{x}}}$ is the partial derivative of the navigation state dynamics $\dot{\hat{\mathbf{x}}}$ with respect to the navigation state $\hat{\mathbf{x}}$ and $\hat{F}_{\tilde{\mathbf{y}}}$ is the partial derivative of the navigation state dynamics $\dot{\hat{\mathbf{x}}}$ with respect to the inertial measurements $\tilde{\mathbf{y}}$. The state estimate update equation is:

$$\hat{x}_k^+ = \hat{x}_k^- + \hat{K}_k[\tilde{z}_k - \hat{h}(\hat{\mathbf{x}}_k)] \tag{4.14}$$

The state covariance is updated after a measurement using the Joseph form (Eq. 5-82 in [22]).

$$P_k^+ = [I - \hat{K}_k\hat{H}_k]\hat{P}_k^-[I - \hat{K}_k\hat{H}_k]^T + \hat{K}_k\hat{R}_\nu\hat{K}_k^T \tag{4.15}$$

$$\hat{K}_k = \hat{P}_k^-\hat{H}_k^T[\hat{H}_k\hat{P}_k^-\hat{H}_k^T + \hat{R}_\nu]^{-1} \tag{4.16}$$

Where K is the Kalman gain and R represents the measurement noise covariance matrix of the measurement being processed.

4.2 Linear Modeling

The procedure used for linearizing a function about a nominal trajectory is defined as follows:

Define the input to a function as the sum of the nominal input plus a perturbation. Substitute the input (nominal plus perturbation) into the function and linearize using a first-order Taylor series expansion. The result is Jacobian derivatives multiplied by perturbation values. Note that the perturbations in the truth state vector and the navigation state vector are called dispersions.

The results of the linearization provide information about how perturbations in the input of a function affect the output.

4.2.1 Nominal Reference Trajectory Definition

The nominal truth state $\overline{\mathbf{x}}$ represents the nominal trajectory, or the trajectory that occurs during a simulation run without disturbance accelerations or measurement noise. The nominal continuous inertial measurement $\overline{\mathbf{y}}$ is an accelerometer reading without any uncertainty. This is the same as the true specific force on the vehicle. The nominal discrete measurement $\overline{\mathbf{z}}$ is a measurement that is not corrupted by noise. The nominal navigation state $\hat{\overline{\mathbf{x}}}$ is equal to the nominal truth state if the vectors contain the same elements. If not, the nominal navigation state is equal to navigation state calculated with the nominal truth values. This is called the nominal true navigation state, or $\overline{\mathbf{x}}_n$. The nominal actuator command $\overline{\mathbf{u}}$ is the command profile that would be used during the nominal trajectory.

4.2.2 Control Linearization

This section shows the linearization process for the control law. All the subsequent subsystems use this same technique.

First, define the perturbations:

$$\hat{\mathbf{u}} = \overline{\mathbf{u}} + \delta\hat{\mathbf{u}} \quad \hat{\mathbf{x}} = \hat{\overline{\mathbf{x}}} + \delta\hat{\mathbf{x}} \tag{4.17}$$

To linearize the control, let

$$\overline{\mathbf{u}} \equiv \hat{\mathbf{g}}(\hat{\overline{\mathbf{x}}}) \tag{4.18}$$

Substitute equation 4.18 and equation 4.17 into equation 4.10.

$$\overline{\mathbf{u}} + \delta\hat{\mathbf{u}} = \hat{\mathbf{g}}(\hat{\overline{\mathbf{x}}} + \delta\hat{\mathbf{x}}) \tag{4.19}$$

Expand to first-order using a Taylor series expansion.

$$\overline{\mathbf{u}} + \delta\hat{\mathbf{u}} \approx \hat{\mathbf{g}}(\hat{\overline{\mathbf{x}}}) + \left.\frac{\partial\hat{\mathbf{g}}}{\partial\hat{\mathbf{x}}}\right|_{\hat{\overline{\mathbf{x}}}} (\hat{\mathbf{x}} - \hat{\overline{\mathbf{x}}}) \tag{4.20}$$

Canceling like terms and simplifying:

$$\delta\hat{\mathbf{u}} = \hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} \tag{4.21}$$

Equation 4.21 is the control linearization equation.

4.2.3 Linearization of Truth State Dynamics

The truth dynamics are a function of the truth state vector and the actuator.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \hat{\mathbf{u}}) \tag{4.22}$$

Defining these perturbations:

$$\mathbf{x} = \overline{\mathbf{x}} + \delta\mathbf{x} \qquad \hat{\mathbf{u}} = \overline{\mathbf{u}} + \delta\hat{\mathbf{u}} \tag{4.23}$$

The nominal dynamics are a function of the nominal truth state and the nominal control equation:

$$\dot{\mathbf{x}} = \dot{\overline{\mathbf{x}}} + \delta\dot{\mathbf{x}} \tag{4.24}$$

where

$$\dot{\overline{\mathbf{x}}} \equiv \mathbf{f}(\overline{\mathbf{x}}, \overline{\mathbf{u}}) \tag{4.25}$$

Substitute this into equation 4.1

$$\dot{\overline{\mathbf{x}}} + \delta\dot{\mathbf{x}} = \mathbf{f}(\overline{\mathbf{x}} + \delta\mathbf{x}, \overline{\mathbf{u}} + \delta\hat{\mathbf{u}}) + \mathrm{B}\mathbf{w} \tag{4.26}$$

The first-order Taylor series expansion is written as

$$\dot{\overline{\mathbf{x}}} + \delta\dot{\mathbf{x}} \approx \mathbf{f}(\overline{\mathbf{x}}, \overline{\mathbf{u}}) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\overline{\mathbf{x}},\overline{\mathbf{u}}} (\mathbf{x} - \overline{\mathbf{x}}) + \left.\frac{\partial \mathbf{f}}{\partial \hat{\mathbf{u}}}\right|_{\overline{\mathbf{x}},\overline{\mathbf{u}}} (\hat{\mathbf{u}} - \overline{\mathbf{u}}) + B\mathbf{w} \tag{4.27}$$

Canceling like terms and simplifying

$$\delta\dot{\mathbf{x}} = F_{\mathbf{x}}\delta\mathbf{x} + F_{\hat{\mathbf{u}}}\delta\hat{\mathbf{u}} + B\mathbf{w} \tag{4.28}$$

Substituting equation 4.21 into this result yields the linearized truth dynamics.

$$\delta\dot{\mathbf{x}} = F_{\mathbf{x}}\delta\mathbf{x} + F_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + B\mathbf{w} \tag{4.29}$$

4.2.4 Inertial Measurements

Defining the perturbations:

$$\tilde{\mathbf{y}} = \overline{\mathbf{y}} + \delta\tilde{\mathbf{y}} \quad \hat{\mathbf{u}} = \overline{\mathbf{u}} + \mathrm{d}\hat{\mathbf{u}} \quad \mathbf{x} = \overline{\mathbf{x}} + \delta\mathbf{x} \tag{4.30}$$

Next, linearize the inertial measurements. Let

$$\overline{\mathbf{y}} \equiv \mathbf{c}(\overline{\mathbf{x}}, \overline{\mathbf{u}}) \tag{4.31}$$

Following the linearization process in the previous sections,

$$\overline{\mathbf{y}} + \delta\tilde{\mathbf{y}} = \mathbf{c}(\overline{\mathbf{x}} + \delta\mathbf{x}, \overline{\mathbf{u}} + \delta\hat{\mathbf{u}}) + \boldsymbol{\eta} \tag{4.32}$$

The first-order Taylor series expansion is as follows,

$$\overline{\mathbf{y}} + \delta\tilde{\mathbf{y}} \approx \mathbf{c}(\overline{\mathbf{x}}, \overline{\mathbf{u}}) + \left.\frac{\partial \mathbf{c}}{\partial \mathbf{u}}\right|_{\overline{\mathbf{u}}} (\mathbf{u} - \overline{\mathbf{u}}) + \left.\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right|_{\overline{\mathbf{x}}} (\mathbf{x} - \overline{\mathbf{x}}) + \boldsymbol{\eta} \tag{4.33}$$

Canceling like terms and simplifying:

$$\delta\tilde{\mathbf{y}} = C_{\hat{\mathbf{u}}}\delta\hat{\mathbf{u}} + C_{\mathbf{x}}\delta\mathbf{x} + \boldsymbol{\eta} \tag{4.34}$$

$$\delta\tilde{\mathbf{y}} = C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + C_{\mathbf{x}}\delta\mathbf{x} + \boldsymbol{\eta} \tag{4.35}$$

4.2.5 Discrete Measurements

The measurements are linearized and measurement noise is added:

$$\tilde{\mathbf{z}} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\nu} \tag{4.36}$$

Defining the perturbations:

$$\mathbf{x} = \overline{\mathbf{x}} + \delta\mathbf{x} \tag{4.37}$$

$$\tilde{\mathbf{z}} = \overline{\mathbf{z}} + \delta\mathbf{z} \tag{4.38}$$

Defining the nominal measurement:

$$\overline{\mathbf{z}} = \mathbf{h}(\overline{\mathbf{x}}) \tag{4.39}$$

Substituting these into the discrete measurement equation 4.36 yields

$$\overline{\mathbf{z}} + \delta\mathbf{z} = \mathbf{h}(\overline{\mathbf{x}} + \delta\mathbf{x}) + \boldsymbol{\nu} \tag{4.40}$$

This equation is expanded with a first-order Taylor series approximation.

$$\mathbf{h}(\overline{\mathbf{x}}) + \delta\mathbf{z} \approx \mathbf{h}(\overline{\mathbf{x}}) + \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\overline{\mathbf{x}}} (\mathbf{x} - \overline{\mathbf{x}}) + \boldsymbol{\nu} \tag{4.41}$$

$$H_{\mathbf{x}} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\overline{\mathbf{x}}} \tag{4.42}$$

$$\delta \mathbf{z} = H_{\mathbf{x}} \delta \mathbf{x} + \boldsymbol{\nu} \tag{4.43}$$

Equation 4.43 is the form that is needed to linearize the measurements about a nominal trajectory.

4.2.6 Navigation State Dynamics

The navigation state dynamics are a function of the navigation state vector and the continuous inertial measurement vector. In other words, the navigation filter uses model replacement to estimate the dynamics of the states.

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\hat{\mathbf{x}}, \tilde{\mathbf{y}}) \tag{4.44}$$

Defining these perturbations:

$$\hat{\mathbf{x}} = \hat{\overline{\mathbf{x}}} + \delta \hat{\mathbf{x}} \quad \tilde{\mathbf{y}} = \overline{\mathbf{y}} + \delta \tilde{\mathbf{y}} \tag{4.45}$$

The nominal navigation state dynamics are a function of the nominal navigation state vector and the nominal continuous inertial measurement vector.

$$\dot{\hat{\mathbf{x}}} = \dot{\overline{\mathbf{x}}} + \delta \dot{\hat{\mathbf{x}}} \tag{4.46}$$

where

$$\dot{\hat{\overline{\mathbf{x}}}} = \dot{\overline{\mathbf{x}}}_n \equiv \hat{\mathbf{f}}(\hat{\overline{\mathbf{x}}}, \overline{\mathbf{y}}) \tag{4.47}$$

Now linearize the navigation dynamics following the linearization process in the previous sections:

$$\dot{\overline{\mathbf{x}}}_n + \delta \dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\hat{\overline{\mathbf{x}}} + \delta \hat{\mathbf{x}}, \overline{\mathbf{y}} + \delta \tilde{\mathbf{y}}) \tag{4.48}$$

The first-order Taylor series expansion is:

$$\dot{\overline{\mathbf{x}}}_n + \delta\dot{\hat{\mathbf{x}}} \approx \hat{\mathbf{f}}(\hat{\overline{\mathbf{x}}},\overline{\mathbf{y}}) + \left.\frac{\partial\hat{\mathbf{f}}}{\partial\hat{\mathbf{x}}}\right|_{\overline{\mathbf{x}},\overline{\mathbf{y}}} (\hat{\mathbf{x}} - \hat{\overline{\mathbf{x}}}) + \left.\frac{\partial\hat{\mathbf{f}}}{\partial\tilde{\mathbf{y}}}\right|_{\overline{\mathbf{x}},\overline{\mathbf{y}}} (\tilde{\mathbf{y}} - \overline{\mathbf{y}}) \tag{4.49}$$

Canceling like terms,

$$\delta\dot{\hat{\mathbf{x}}} \approx \left.\frac{\partial\hat{\mathbf{f}}}{\partial\hat{\mathbf{x}}}\right|_{\overline{\mathbf{x}},\overline{\mathbf{y}}} (\hat{\mathbf{x}} - \hat{\overline{\mathbf{x}}}) + \left.\frac{\partial\hat{\mathbf{f}}}{\partial\tilde{\mathbf{y}}}\right|_{\overline{\mathbf{x}},\overline{\mathbf{y}}} (\tilde{\mathbf{y}} - \overline{\mathbf{y}}) \tag{4.50}$$

Simplifying,

$$\delta\dot{\hat{\mathbf{x}}} = \hat{F}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}}\delta\tilde{\mathbf{y}} \tag{4.51}$$

Substituting in equation 4.35 and simplifying,

$$\delta\dot{\hat{\mathbf{x}}} = \hat{F}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}}(C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + C_{\mathbf{x}}\delta\mathbf{x} + \boldsymbol{\eta}) \tag{4.52}$$

$$\delta\dot{\hat{\mathbf{x}}} = \hat{F}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\mathbf{x}}\delta\mathbf{x} + \hat{F}_{\tilde{\mathbf{y}}}\boldsymbol{\eta} \tag{4.53}$$

Combining like terms yields

$$\delta\dot{\hat{\mathbf{x}}} = (\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}})\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\mathbf{x}}\delta\mathbf{x} + \hat{F}_{\tilde{\mathbf{y}}}\boldsymbol{\eta} \tag{4.54}$$

4.2.7 Measurement Update Equations

The truth states and nominal truth states are unaffected by discrete measurements. In equation form, this is written as

$$\mathbf{x}^+ = \mathbf{x}^- \tag{4.55}$$

Expanding this,

$$\overline{\mathbf{x}}^+ + \delta\mathbf{x}^+ = \overline{\mathbf{x}}^- + \delta\mathbf{x}^- \tag{4.56}$$

Thus, the truth state dispersion is unchanged as well

$$\delta\mathbf{x}^+ = \delta\mathbf{x}^- \tag{4.57}$$

The navigation state dispersions are changed by a discrete measurement update. Introducing truth and navigation state dispersions:

$$\mathbf{x}_k = \overline{\mathbf{x}}_k + \delta\mathbf{x}_k \tag{4.58}$$

$$\hat{\mathbf{x}}_k = \hat{\overline{\mathbf{x}}}_k + \delta\hat{\mathbf{x}}_k \tag{4.59}$$

Substituting these equations into equations 4.14 and 4.36, and noting that $\overline{\mathbf{x}}^+_{n,k} = \overline{\mathbf{x}}_{n,k} = \overline{\mathbf{x}}^-_{n,k}$

$$\overline{\mathbf{x}}_{n,k} + \delta\hat{\mathbf{x}}^+_k = \overline{\mathbf{x}}_{n,k} + \delta\hat{\mathbf{x}}^-_k + \hat{K}_k\big(\mathbf{h}(\overline{\mathbf{x}}_k + \delta\mathbf{x}_k) + \boldsymbol{\nu}_k - \hat{\mathbf{h}}(\overline{\mathbf{x}}_{n,k} + \delta\hat{\mathbf{x}}_k)\big) \tag{4.60}$$

The first-order Taylor series expansion is:

$$\overline{\mathbf{x}}_{n,k} + \delta\hat{\mathbf{x}}^+_k \approx \overline{\mathbf{x}}_{n,k} + \delta\hat{\mathbf{x}}^-_k + \hat{K}_k\left(\mathbf{h}(\overline{\mathbf{x}}_k) + \left.\frac{\partial\mathbf{h}}{\partial\mathbf{x}}\right|_{\overline{\mathbf{x}}}(\mathbf{x} - \overline{\mathbf{x}})) + \boldsymbol{\nu_k} - \hat{\mathbf{h}}(\overline{\mathbf{x}}_{n,k}) + \left.\frac{\partial\hat{\mathbf{h}}}{\partial\hat{\mathbf{x}}}\right|_{\overline{\mathbf{x}}_n}(\hat{\mathbf{x}} - \hat{\overline{\mathbf{x}}})\right) \tag{4.61}$$

Canceling like terms and simplifying (assuming $h(\overline{\mathbf{x}}_k) = \hat{h}(\overline{\mathbf{x}}_{n,k})$)

$$\delta\hat{\mathbf{x}}^+_k = [I - \hat{K}_k\hat{H}_{\hat{\mathbf{x}},k}]\delta\hat{\mathbf{x}}^-_k + \hat{K}_k H_{\mathbf{x},k}\delta\mathbf{x}^-_k + \hat{K}_k\boldsymbol{\nu}_k \tag{4.62}$$

4.2.8 True Navigation Errors

The true navigation state mapping function is linearized with equation 4.12.

$$\mathbf{x}_n = \mathbf{m}(\mathbf{x}) \tag{4.63}$$

Defining the perturbations:

$$\mathbf{x}_n = \overline{\mathbf{x}}_n + \delta\mathbf{x}_n \quad \mathbf{x} = \overline{\mathbf{x}} + \delta\mathbf{x} \tag{4.64}$$

Substituting the equations from 4.64 into 4.63:

$$\overline{\mathbf{x}}_n + \delta\mathbf{x}_n = \mathbf{m}(\overline{\mathbf{x}} + \delta\mathbf{x}) \tag{4.65}$$

The first-order Taylor series expansion is written as:

$$\overline{\mathbf{x}}_n + \delta\mathbf{x}_n \approx \mathbf{m}(\overline{\mathbf{x}}) + \left.\frac{\partial \mathbf{m}}{\partial \mathbf{x}}\right|_{\overline{\mathbf{x}}} \delta\mathbf{x} \tag{4.66}$$

Simplifying:

$$\delta\mathbf{x}_n = M_{\mathbf{x}}\delta\mathbf{x} \tag{4.67}$$

Rearranging:

$$M_{\mathbf{x}} = \frac{\delta\mathbf{x}_n}{\delta\mathbf{x}} \tag{4.68}$$

The true navigation error is defined as

$$\mathbf{e} = \hat{\mathbf{x}} - \mathbf{m}(\mathbf{x}) \tag{4.69}$$

Adding in the nominal values and dispersions:

$$\overline{\mathbf{e}} + \delta\mathbf{e} = \overline{\mathbf{x}}_n + \delta\hat{\mathbf{x}} - \mathbf{m}(\overline{\mathbf{x}} + \delta\mathbf{x}) \tag{4.70}$$

The first-order Taylor series expansion

$$\overline{\mathbf{e}} + \delta\mathbf{e} = \overline{\mathbf{x}}_n + \delta\hat{\mathbf{x}} - \mathbf{m}(\overline{\mathbf{x}}) - M_{\mathbf{x}}\delta\mathbf{x} \tag{4.71}$$

Simplifying

$$\delta\mathbf{e} = \delta\hat{\mathbf{x}} - M_{\mathbf{x}}\delta\mathbf{x} \tag{4.72}$$

4.3 Augmented State Linear Covariance Theory

The control, dynamics, and measurement equations are linearized about the NRT to calculate the truth and navigation dispersion from the NRT. The truth dispersion $\delta\mathbf{x}$ is the difference between the state values of the NRT and the actual truth values. The position dispersion indicates how far away the vehicle is from the nominal position. The navigation state dispersion $\delta\hat{\mathbf{x}}$ is the difference between the estimated state and the nominal

estimated state. Equations 4.1, 4.4, 4.10, and 4.11 are linearized using a 1st-order Taylor series approximation to produce the following propagation equations:

$$\delta\dot{\mathbf{x}} = F_{\mathbf{x}}\delta\mathbf{x} + F_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + B\mathbf{w} \tag{4.73}$$

$$\delta\dot{\hat{\mathbf{x}}} = \left[\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\right]\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\mathbf{x}}\delta\mathbf{x} + \hat{F}_{\tilde{\mathbf{y}}}\boldsymbol{\eta} \tag{4.74}$$

The upper case letters represent a partial derivative taken with respect to the subscript vector and evaluated at the nominal trajectory. For example, $F_{\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}}\big|_{\overline{\mathbf{x}}}$, $\hat{G}_{\hat{\mathbf{x}}} = \frac{\partial \hat{g}}{\partial \hat{\mathbf{x}}}\big|_{\overline{\mathbf{x}}}$, and $\hat{F}_{\tilde{\mathbf{y}}} = \frac{\partial \hat{f}}{\partial \tilde{\mathbf{y}}}\big|_{\overline{\mathbf{y}}}$.

The linear discrete update equations are the following:

$$\delta\mathbf{x}_k^+ = \delta\mathbf{x}_k^- \tag{4.75}$$

$$\delta\hat{\mathbf{x}}_k^+ = [I - \hat{K}_k\hat{H}_{\hat{x},k}]\delta\hat{\mathbf{x}}_k^- + \hat{K}_k H_{x,k}\delta\mathbf{x}_k^- + \hat{K}_k\boldsymbol{\nu} \tag{4.76}$$

Note that the truth dispersions are not changed by a measurement update.

4.3.1 Augmented State Covariance Equations

With the linearized system defined, it is possible to write the truth and navigation state propagation and update equations in a compact form using the augmented state vector. The augmented state vector and associated differential equations allow the simulation to efficiently propagate and update the estimation error covariance and the truth dispersion covariance.

$$\mathbf{X} = \begin{bmatrix} \delta\mathbf{x} \\ \\ \delta\hat{\mathbf{x}} \end{bmatrix} \tag{4.77}$$

The augmented state vector X is zero mean with covariance C_A:

$$E[\mathbf{X}] = \mathbf{0} \quad E[\mathbf{X}\mathbf{X}^T] = C_A \tag{4.78}$$

The dynamics and update equations for the augmented state vector $\mathbf{X}$ are shown below.

$$\dot{\mathbf{X}} = \mathcal{F}\mathbf{X} + \mathcal{G}\boldsymbol{\eta} + \mathcal{W}\mathbf{w} \tag{4.79}$$

$$\mathbf{X}_k{}^{+} = \mathcal{A}_k\mathbf{X}_k^{-} + \mathcal{B}_k\boldsymbol{\nu}_k \tag{4.80}$$

where

$$\mathcal{F} = \begin{bmatrix} F_{\mathbf{x}} & F_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}} \\ \\ \hat{F}_{\tilde{\mathbf{y}}}C_{\mathbf{x}} & \hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}} \end{bmatrix}_{m\times m} \quad \mathcal{G} = \begin{bmatrix} 0_{n\times n_y} \\ \\ \hat{F}_{\tilde{\mathbf{y}}} \end{bmatrix}_{m\times n_y} \quad \mathcal{W} = \begin{bmatrix} B_{n\times n_w} \\ \\ 0_{\hat{n}\times n_w} \end{bmatrix}_{m\times n_w} \tag{4.81}$$

$$\mathcal{A}_k = \begin{bmatrix} I_{n\times n} & 0_{n\times\hat{n}} \\ \\ \hat{K}_k H_{\mathbf{x},k} & I_{\hat{n}\times\hat{n}} - \hat{K}_k\hat{H}_{\hat{\mathbf{x}},k} \end{bmatrix}_{m\times m} \quad \mathcal{B}_k = \begin{bmatrix} 0_{n\times n_z} \\ \\ \hat{K}_{k,\,\hat{n}\times n_z} \end{bmatrix}_{m\times n_z} \tag{4.82}$$

The C_A matrix contains the information needed for performance evaluation of the simulation. Note that the propagation and update equations below have a similar form to the Kalman filter covariance equations.

$$\dot{C}_A = \mathcal{F}C_A + C_A\mathcal{F}^T + \mathcal{G}S_\eta\mathcal{G}^T + \mathcal{W}S_w\mathcal{W}^T \tag{4.83}$$

$$C_A(t_i^{+}) = \mathcal{A}_i C_A(t_i^{-})\mathcal{A}_i^T + \mathcal{B}_i R_\nu \mathcal{B}_i^T \tag{4.84}$$

4.3.2 Performance Evaluation

The truth dispersion covariance D_{true} and the true estimation error covariance P_{true} equations are used to evaluate the overall performance of the closed-loop system. These quantities are obtained from the augmented state covariance matrix using identity matrices, zero matrices, and the $M_{\mathbf{x}}$ matrix. The $M_{\mathbf{x}}$ matrix is the partial derivative of the mapping function $\mathbf{m}(x)$ that maps the truth states to the true-navigation states. D_{true} is the covariance

of the truth state dispersions.

$$D_{\text{true}} = E[\delta\mathbf{x}\delta\mathbf{x}^T] = [I_{n\times n} \;\; 0_{n\times\hat{n}}]C_A[I_{n\times n} \;\; 0_{n\times\hat{n}}]^T \tag{4.85}$$

The covariance of the true estimation errors is given by the following equations.

$$P_{\text{true}} = E[\mathbf{e}\mathbf{e}^T]$$

$$P_{\text{true}} = E[\{\delta\hat{\mathbf{x}} - M_{\mathbf{x}}\delta\mathbf{x}\}\{\delta\hat{\mathbf{x}} - M_{\mathbf{x}}\delta\mathbf{x}\}^T]$$

$$P_{\text{true}} = [-M_{\mathbf{x}} \;\; I_{\hat{n}\times\hat{n}}]C_A[-M_{\mathbf{x}} \;\; I_{\hat{n}\times\hat{n}}]^T \tag{4.86}$$

4.4 Event Trigger Theory

The goal of event trigger theory is to incorporate the effect of a discrete event into the LinCov computed statistics. The theory relies on a constraint equation that is a function of the navigation state vector $\hat{\mathbf{x}}$. This equation is linearized along with truth and navigation state dynamics to calculate the time dispersion δt_e of the event, the variance of the event time σ_{te}^2, and the covariance shaping matrix Φ. The augmented state covariance matrix is pre- and post- multiplied by the shaping matrix to incorporate the discrete event into the LinCov simulation. This section will derive the general event trigger equations for an augmented state linear covariance simulation.

The constraint of the event trigger Ψ is a function of the navigation state at event time t_e.

$$\Psi[\hat{\mathbf{x}}(t_e)] = 0 \tag{4.87}$$

The nominal event time $\bar{t}_e$ is the simulation time when the constraint equation is met for the NRT. At this nominal event time $\bar{t}_e$, the truth state and navigation state for a given Monte Carlo run is different than the reference trajectory state. The truth and navigation states at time $\bar{t}_e$ are written as follows:

$$\mathbf{x}(\bar{t}_e) = \bar{\mathbf{x}}(\bar{t}_e) + \delta\mathbf{x}(\bar{t}_e) \tag{4.88}$$

$$\hat{\mathbf{x}}(\bar{t}_e) = \mathbf{m}\big(\bar{\mathbf{x}}(\bar{t}_e)\big) + \delta\hat{\mathbf{x}}(\bar{t}_e) \tag{4.89}$$

The derivative of equation 4.88 with respect to time is written as:

$$\dot{\mathbf{x}}(\bar{t}_e) = \dot{\bar{\mathbf{x}}}(\bar{t}_e) + \delta\dot{\mathbf{x}}(\bar{t}_e) \tag{4.90}$$

Using the chain rule, the time derivative of equation 4.89 is:

$$\dot{\hat{\mathbf{x}}}(\bar{t}_e) = M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e) + \delta\dot{\hat{\mathbf{x}}}(\bar{t}_e) \tag{4.91}$$

Since the discrete event constraint equation will not be met at the nominal time $\bar{t}_e$, it is useful to know what the truth and navigation states is at the actual event time t_e. Utilizing a first-order Taylor series approximation, the truth and navigation states at the time of the event is by written with the following equations:

$$\mathbf{x}(t_e) \approx \mathbf{x}(\bar{t}_e) + \dot{\mathbf{x}}(\bar{t}_e)[t_e - \bar{t}_e] = \mathbf{x}(\bar{t}_e) + \dot{\mathbf{x}}(\bar{t}_e)\delta t_e \tag{4.92}$$

$$\hat{\mathbf{x}}(t_e) \approx \hat{\mathbf{x}}(\bar{t}_e) + \dot{\hat{\mathbf{x}}}(\bar{t}_e)[t_e - \bar{t}_e] = \hat{\mathbf{x}}(\bar{t}_e) + \dot{\hat{\mathbf{x}}}(\bar{t}_e)\delta t_e \tag{4.93}$$

The time dispersion equation represents the difference between the nominal event time and the truth event time:

$$\delta t_e = t_e - \bar{t}_e \tag{4.94}$$

Combining equations 4.94, 4.90, and 4.88 with equation 4.92:

$$\mathbf{x}(t_e) \approx \bar{\mathbf{x}}(\bar{t}_e) + \delta\mathbf{x}(\bar{t}_e) + \dot{\bar{\mathbf{x}}}(\bar{t}_e)\delta t_e + \delta\dot{\mathbf{x}}(\bar{t}_e)\delta t_e \tag{4.95}$$

Dropping the second order term yields:

$$\mathbf{x}(t_e) \approx \bar{\mathbf{x}}(\bar{t}_e) + \delta\mathbf{x}(\bar{t}_e) + \dot{\bar{\mathbf{x}}}(\bar{t}_e)\delta t_e \tag{4.96}$$

The navigation state at the time of the event is approximated similarly:

$$\hat{\mathbf{x}}(t_e) \approx \mathbf{m}\big(\bar{\mathbf{x}}(\bar{t}_e)\big) + \delta\hat{\mathbf{x}}(\bar{t}_e) + M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\delta t_e \tag{4.97}$$

This navigation state equation is substituted back into the original constraint equation, equation 4.87.

$$\Psi[\hat{\mathbf{x}}(t_e)] = \Psi\big[\mathbf{m}\big(\bar{\mathbf{x}}(\bar{t}_e)\big) + \delta\hat{\mathbf{x}}(\bar{t}_e) + M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\delta t_e\big] = 0 \tag{4.98}$$

Linearizing this with a first-order Taylor series approximation:

$$\Psi\big[m\big(\bar{\mathbf{x}}(\bar{t}_e)\big) + \delta\hat{\mathbf{x}}(\bar{t}_e) + M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\delta t_e\big] \approx$$

$$\Psi\big[m\big(\bar{\mathbf{x}}(\bar{t}_e)\big)\big] + \frac{\partial\Psi[\hat{\mathbf{x}}(t_e)]}{\partial\hat{\mathbf{x}}(t_e)}\Big|_{\bar{\mathbf{x}}}\big[\delta\hat{\mathbf{x}}(\bar{t}_e) + M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\delta t_e\big] \tag{4.99}$$

Note that this linearization includes the non-contemporaneous term and the contemporaneous term (See Fig. 1 of [80]).

By definition, the constraint equation of the nominal trajectory is zero at the nominal event time:

$$\Psi\big[\mathbf{m}\big(\bar{\mathbf{x}}(\bar{t}_e)\big)\big] = 0 \tag{4.100}$$

Applying this to equation 4.99:

$$\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)\delta\hat{\mathbf{x}}(\bar{t}_e) + \Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\delta t_e = 0 \tag{4.101}$$

Solving for the time dispersion δt_e:

$$\delta t_e = -\big[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\big]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)\delta\hat{\mathbf{x}}(\bar{t}_e) \tag{4.102}$$

Since the constraint equation, i.e. the event is always a scalar, this inverse is just a scalar inversion. This time dispersion equation is important because it is the first-order

approximation of the difference between the NRT event time and the actual event time in a given Monte Carlo run. Rewriting the time dispersion equation in terms of the augmented state vector form and substituting it into the expectation operator will yield the variance of the time dispersions. The augmented state form of the time dispersion equation is written as:

$$\delta t_e = \begin{bmatrix} \mathbf{0}_{1\times n} & -\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) M_{\mathbf{x}}(\bar{t}_e)\dot{\hat{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}(\bar{t}_e) \\ \\ \delta\hat{\mathbf{x}}(\bar{t}_e) \end{bmatrix} \tag{4.103}$$

Or,

$$\delta t_e = \Omega(\bar{t}_e)\mathbf{X}(\bar{t}_e) \tag{4.104}$$

Substituting in $\delta t_e \delta t_e^T$ into the expectation operator yields:

$$E[(\delta t_e)^2] = E[\{\Omega(\bar{t}_e)\mathbf{X}(\bar{t}_e)\}\{\Omega(\bar{t}_e)\mathbf{X}(\bar{t}_e)\}^T] \tag{4.105}$$

Simplify this expression by removing the non-stochastic elements from the expectation operator:

$$E[(\delta t_e)(\delta t_e)^T] = \Omega(\bar{t}_e)E[\mathbf{X}(\bar{t}_e)\mathbf{X}(\bar{t}_e)^T]\Omega(\bar{t}_e)^T \tag{4.106}$$

LinCov Equation 4.78 states that the covariance of augmented state vector at the nominal event time is $C_A(\bar{t}_e)$:

$$C_A(\bar{t}_e) = E[\mathbf{X}(\bar{t}_e)\mathbf{X}(\bar{t}_e)^T] \tag{4.107}$$

Therefore, to first-order, the time variance of a defined discrete event is written as:

$$\sigma^2_{\delta t_e} = \Omega(\bar{t}_e)C_A(\bar{t}_e)\Omega(\bar{t}_e)^T \tag{4.108}$$

Note that the variance in event time is a function of the constraint partial derivative, the dynamics partial derivative, and the augmented state vector covariance, all evaluated at time $\bar{t}_e$.

Next, the covariance shaping matrix Φ is derived. The first step is to find the truth and navigation state at the time of the event by substituting equation 4.102 into equations

4.96 and 4.97.

$$\mathbf{x}(t_e) = \bar{\mathbf{x}}(\bar{t}_e) + \delta\mathbf{x}(\bar{t}_e) - \dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)\delta\hat{\mathbf{x}}(\bar{t}_e) \tag{4.109}$$

$$\hat{\mathbf{x}}(t_e) = m\big(\bar{\mathbf{x}}(\bar{t}_e)\big) + \delta\hat{\mathbf{x}}(\bar{t}_e) - M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)\delta\hat{\mathbf{x}}(\bar{t}_e) \tag{4.110}$$

Note that by definition the expected value of the truth and navigation states at the event time are equal to the expected values of the nominal states at the nominal event time:

$$E[\mathbf{x}(t_e)] = E[\bar{\mathbf{x}}(\bar{t}_e)] \tag{4.111}$$

$$E[\hat{\mathbf{x}}(t_e)] = E[\mathbf{m}(\bar{\mathbf{x}}(\bar{t}_e))] \tag{4.112}$$

Breaking down the augmented state vector:

$$\mathbf{X}(t_e) = \begin{bmatrix} \delta\mathbf{x}(t_e) \\ \\ \delta\hat{\mathbf{x}}(t_e) \end{bmatrix} = \begin{bmatrix} \mathbf{x}(t_e) - \bar{\mathbf{x}}(\bar{t}_e) \\ \\ \hat{\mathbf{x}}(t_e) - \mathbf{m}(\bar{\mathbf{x}}(\bar{t}_e)) \end{bmatrix} \tag{4.113}$$

Equations 4.109 and 4.110 are arranged into augmented state space form by substituting them into equation 4.113:

$$\mathbf{X}(t_e) =$$

$$= \begin{bmatrix} \bar{\mathbf{x}}(\bar{t}_e) + \delta\mathbf{x}(\bar{t}_e) - \dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)\delta\hat{\mathbf{x}}(\bar{t}_e) - \bar{\mathbf{x}}(\bar{t}_e) \\ \mathbf{m}(\bar{\mathbf{x}}(\bar{t}_e)) + \delta\hat{\mathbf{x}}(\bar{t}_e) - M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)\delta\hat{\mathbf{x}}(\bar{t}_e) - \mathbf{m}(\bar{\mathbf{x}}(\bar{t}_e)) \end{bmatrix} \tag{4.114}$$

Cancelling like terms:

$$\mathbf{X}(t_e) = \begin{bmatrix} \delta\mathbf{x}(\bar{t}_e) - \dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)\delta\hat{\mathbf{x}}(\bar{t}_e) \\ \delta\hat{\mathbf{x}}(\bar{t}_e) - M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)\delta\hat{\mathbf{x}}(\bar{t}_e) \end{bmatrix} \tag{4.115}$$

Each term is multiplied by $\delta\mathbf{x}$ or $\delta\hat{\mathbf{x}}$, therefore, $\mathbf{X}(t_e)$ is expressed in state space form:

$$\mathbf{X}(t_e) = \Phi(\bar{t}_e)\mathbf{X}(\bar{t}_e) \tag{4.116}$$

Where Φ is the covariance shaping matrix

$$\mathbf{\Phi}(\bar{t}_e) = \begin{bmatrix} I_{n\times n} & -\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \\ 0_{\hat{n}\times n} & I_{\hat{n}\times\hat{n}} - M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \end{bmatrix} \tag{4.117}$$

In summary:

$$\mathbf{X}(t_e) = \begin{bmatrix} I_{n\times n} & -\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \\ 0_{\hat{n}\times n} & I_{\hat{n}\times\hat{n}} - M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e)M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}(\bar{t}_e) \\ \delta\hat{\mathbf{x}}(\bar{t}_e) \end{bmatrix} \tag{4.118}$$

This expression is important because is shows that the augmented state vector at the event time is a function of the partial derivatives of the system, evaluated at the nominal event time. This covariance shaping matrix is used to update the LinCov simulation at the time of the discrete event. The augmented state covariance C_a is updated by Φ at the time of the NRT event time $\bar{t}_e$ with the following equation:

$$C_a^+(\bar{t}_e) = \Phi(\bar{t}_e)C_a^-(\bar{t}_e)\Phi(\bar{t}_e)^T \tag{4.119}$$

Trigger LinCov is a powerful LinCov tool: it is used to fully incorporate the effects of navigation based discrete events into a LinCov simulation.

CHAPTER 5
STATISTICAL LINEARIZATION THEORETICAL DEVELOPMENT

> We can see one distinct advantage that statistical linearization has over the Taylor series expansion; it does not require the existence of derivatives of f(x). Thus, a large number of nonlinearities–relays, saturation, threshold, etc. can be treated by this method without having to approximate discontinuities at corners in f(x) by smooth functions. ... The resulting statistical approximation for f(x) is considerably more accurate than the Taylor series, from a statistical point of view.
>
> -Price [9]

Estimation of a strictly linear system is a straightforward and well documented process. Assuming the probability density function (PDF) of the input is known, an optimal estimator can be derived [22] and the performance statistics of the system can be calculated exactly. Corporeal systems, however, often have nonlinear elements. Thus, for a GN&C problem, it is often necessary to use Monte Carlo analysis methods to calculate performance statistics, or approximate the behavior of a nonlinear system by linearizing it. This chapter first defines the traditional linearization process used in LinCov analysis, then defines the process for statistical linearization. An example problem is then developed where a nonlinear saturation equation is linearized using both methods.

Statistical linearization, also sometimes called "quasi-linearization" [4], is sometimes confused with unscented Kalman filtering or particle filtering, as these methods both provide statistical data by approximating nonlinear systems with methodically chosen data points. The Unscented Kalman Filter is well suited for systems with nonlinear dynamics [81] and the particle filter is well suited for nonlinear systems with non-Gaussian inputs [82]. Both of these nonlinear analysis techniques are computation heavy [83]. This dissertation is focused on computation efficient linear covariance applications, therefore the Unscented Kalman Filter and the particle filter will not be considered.

5.1 Probability Theory

Before developing the linearization process traditionally used in covariance analysis, it is important to review the role that random variables and probability density functions play in statistical analysis. According to Maybeck, a random variable is a real-valued point function which assigns a real scalar value to each point in the domain of the function (ch. 3.2 of [22]). This dissertation focuses on Gaussian random variable inputs, as they provide a suitable model for many random processes found in nature, and they provide manageable models upon which to base estimators and controllers (pg. 101 of [22]). This assumption is not particularly restrictive for most missile engagement problems. While it is true that the output of a nonlinear function with non-Gaussian random variable inputs is not Gaussian, it is often approximately valid to make this assumption (ch. 1 of [73]). According to the central limit theorem, a random variable becomes more Gaussian as it passes through a low-pass filter. Therefore, by filtering the nonlinear dynamics of the system, it can be assumed that the Gaussian input assumption is valid. Additionally, if a system output is a sum of random variables, its distribution approaches a Gaussian distribution, regardless of the PDFs of the inputs (ch. 3.9 of [22]). This is often true for systems with as few as three additive random variables [84].

Gaussian random variable inputs are frequently used in aerospace GN&C problems to represent uncertainty, disturbances, or biases in the system dynamics, control, or measurements. The random variable x is governed by a probability density function f_x that specifies the probability that the random variable will take a given range of values. One notable property of a PDF is:

$$prob[x_1 < x < x_2] = \int_{x_1}^{x_2} f_x(\xi)\, d\xi \tag{5.1}$$

This calculation is important for missile engagement analysis, as it can used to calculate probability of a successful engagement based on the miss distance statistics of the engagement. The expectation operator is used with the PDF of a random variable to calculate the mean and variance of the output of the random variable. The expectation operator returns the

average value of a function of the random variable. The expectation operator is defined as:

$$E[g(x)] = \int_{-\infty}^{\infty} g(\xi) f_x(\xi)\, d\xi \tag{5.2}$$

One reason Gaussian random variables are useful is that specifying the first two moments completely describes the PDF. The first moment of a random variable, also called the mean, is written as

$$\mu_x = E[x] = \int_{-\infty}^{\infty} \xi f_x(\xi)\, d\xi \tag{5.3}$$

The second moment, or variance of a random variable is

$$P_x = \sigma_x^2 = E[(x - \mu_x)^2] = \int_{-\infty}^{\infty} (\xi - \mu_x)^2 f_x(\xi)\, d\xi \tag{5.4}$$

Figure 5.1 illustrates the PDF for a Gaussian random variable with mean value μ_x and variance σ_x^2. Note that the total area under the curve of a PDF is 1. The equation for the PDF of a Gaussian random variable is:

$$f_x(\xi) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{1}{2\sigma_x^2}(\xi - \mu_x)^2} \tag{5.5}$$

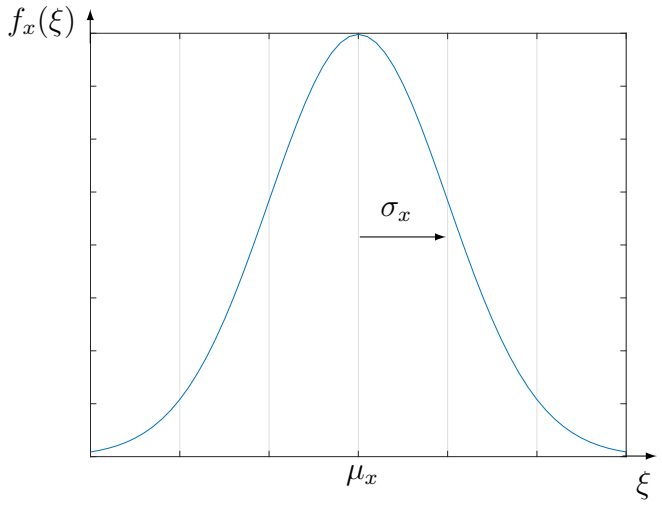

Fig. 5.1: Gaussian Probability Density Function

Nonlinear differential equations govern the motion of many types of aerospace vehicles. Examining the effect linear operators have on Gaussian random variables gives insight into how nonlinear operators affect them. If $H(x)$ is a linear transformation of random variable input x with mean μ_x and variance P_x, the output y is written as

$$y = H(x) = A + Bx \tag{5.6}$$

Note that a linear transformation of a Gaussian random variable x outputs a Gaussian random variable y (ch. 3.10 of [22]). The statistics of y can be computed with the following equations

$$\mu_y = E[y] = \int_{-\infty}^{\infty} (A + B\xi) f_x(\xi)\, d\xi = A + B\mu_x \tag{5.7}$$

$$P_y = E[(y - \mu_y)^2] = \int_{-\infty}^{\infty} (A + B\xi - A - B\mu_x)^2 f_x(\xi)\, d\xi = \int_{-\infty}^{\infty} (B\xi - B\mu_x)^2 f_x(\xi)\, d\xi$$

$$P_y = B^2 P_x \tag{5.8}$$

If a nonlinear operation is performed on a Gaussian random variable x, the output y is not strictly Gaussian. The statistics of y can be computed with Monte Carlo methods, or approximated with linear methods. If the nonlinearity does not dominate the system dynamics, it is common to calculate a 1st-order Taylor series approximation to the function. As long as the linearization is performed in a somewhat linear region, then the statistics of the output y are still accurate. However, if the derivative of the nonlinear function is discontinuous, a more robust linearization technique called statistical linearization can be used.

5.2 Traditional Linearization

Previous LinCov research utilizes a 1st-Order Taylor series approximation for linearizing the dynamics, control, and measurements of a system [7, 10, 36]. This linearization can be demonstrated with the following equations:

$$y = f(x) \approx f(\mu_x) + \left.\frac{\partial f}{\partial x}\right|_{\mu_x} (x - \mu_x)$$

$$y \approx f(\mu_x) + F_x(x - \mu_x) \tag{5.9}$$

where $f(x)$ is a nonlinear function of the Gaussian random variable x and F_x is the partial derivative of the function $f(x)$ with respect to x evaluated at the mean value of x, μ_x. Substituting this result into equations 5.3 and 5.4 yields the approximated mean and variance of the output y.

$$\mu_y = E[y] \approx \int_{-\infty}^{\infty} [f(\mu_x) + F_x(\xi - \mu_x)] f_x(\xi)\, d\xi$$

$$= f(\mu_x) \int_{-\infty}^{\infty} f_x(\xi)\, d\xi + F_x \int_{-\infty}^{\infty} (\xi - \mu_x) f_x(\xi)\, d\xi$$

$$= f(\mu_x) \int_{-\infty}^{\infty} f_x(\xi)\, d\xi + F_x\mu_x - F_x\mu_x$$

$$\boxed{\mu_y \approx f(\mu_x)} \tag{5.10}$$

The second moment, or variance of a y is approximated as:

$$P_y = E[(y - \mu_y)^2] \approx \int_{-\infty}^{\infty} [f(\mu_x) + F_x(\xi - \mu_x) - f(\mu_x)]^2 f_x(\xi)\, d\xi$$

$$= \int_{-\infty}^{\infty} [F_x(\xi - \mu_x)]^2 f_x(\xi)\, d\xi = F_x^2 \int_{-\infty}^{\infty} [(\xi - \mu_x)]^2 f_x(\xi)\, d\xi$$

$$\boxed{P_y \approx F_x^2 P_x} \tag{5.11}$$

5.3 Statistical Linearization

Statistical linearization is the process of calculating the optimal linear coefficients that approximate a nonlinear function with a random variable input. The goal of the optimization is to calculate the linear function that most accurately approximates the nonlinear function statistically. This process is derived and explained in [9,73,74,85,86]. It is called "statistical" linearization because in addition to relying on the behavior of the nonlinear function, it also relies on the PDF of the random variable input (and thereby its statistical quantities mean and variance). A traditional linearization only takes into account the slope of the nonlinear function.

The statistical linearization equations are derived in Section 3.4 of this dissertation. However, a more general derivation is shown in this section. The derivation in Section 3.4 assumed that the input to the nonlinear function is zero-mean, and that the nonlinearity is anti-symmetric, i.e., $f(-x) = -f(x)$. In this specific case, one of the nonlinear coefficients cancelled out to zero. The derivation in this section applies to any nonlinear function. Since different reference documents for Statistical Linearization use different notations, table 5.1 lists a few of the different notations styles according to the document author.

Table 5.1: Statistical Linearization Gain Nomenclature Reference

Reference	Expectation Gain	Expectation Vector	RIDF Gain	QLSDM
Clawson	$\check{f}$	$\check{\mathbf{f}}$	n	N
Taylor [73]	$\hat{f}$	$\hat{\mathbf{f}}$	n	N
Gelb [4]		$N_m\mathbf{m}$		N_r
Zarchan [6]			K_{eq}	
Booten [74]			K_{eq}	
Price [9]	$\hat{f}$	$\hat{\mathbf{f}}$	n_1	N_f
Example Problem	a	$\mathbf{a}$	b	B

As an example to show where the statistical linearization gains/coefficients fit into the linearized equations, consider the generic nonlinear system $y = f(x)$ which is approximated

with statistical linearization by the scalar equation

$$y \approx a + b(x - \mu_x) \tag{5.12}$$

or the vector form

$$\mathbf{y} \approx \mathbf{a} + B(\mathbf{x} - \boldsymbol{\mu}_x) \tag{5.13}$$

In the vector form, $\mathbf{a}$ is a vector, and B is a partial derivative matrix that is multiplied by the vector $(\mathbf{x} - \boldsymbol{\mu}_x)$. In statistical linearization, the first term $\mathbf{a}$ is a vector called the expectation vector (or expectation gain a if it is a scalar). It represents the expected value of the nonlinear function. The second term B is a multiplicative term called the quasi-linear system dynamics matrix (QLSDM) (or describing function gain b for a scalar [9, 73]). The B term represents the partial derivative of $\mathbf{a}$ with respect to $\boldsymbol{\mu}_x$ (See Eq. 1.2-10 of [73]). This dissertation uses a modified version of Taylor's notation [73]. Note that the statistical linearization equations 5.12 and 5.13 are similar in form to the traditional linearization equation 5.9.

In this dissertation, the expectation vector is written as $\check{\mathbf{f}}$, or $\check{f}$ for a scalar. The quasi-linear system dynamics matrix is referred to as N, and the scalar describing function gain is referred to as n. For a state vector $\mathbf{x}$ with multiple elements, this distinction between scalar form and vector form is made because not every element of a system is driven by nonlinear dynamics. The statistical linearization derivations in this chapter assume the scalar form, and Section 5.5 will address the vector form.

The process for deriving the statistical linearization equations is as follows: define a nonlinear equation $y = f(x)$, set a linear approximation function $f_a(x)$, combine them into an error function $e(x)$, minimize the error function e given f_a, and then solve for the coefficients of f_a.

Consider the nonlinear function $y = f(x)$ and its approximation $f_a(x)$

$$y = f(x) \approx f_a(x) = \check{f} + n(x - \mu_x) \tag{5.14}$$

It is desired to calculate the values of $\check{f}$ and n that minimize the error with the nonlinear function $f(x)$, therefore the error function is proposed:

$$e = f(x) - f_a(x) = f(x) - \check{f} - n(x - \mu_x) \tag{5.15}$$

The cost function J to minimize is

$$J = E[e^2] = E\left[\left(f(x) - \check{f} - n(x - \mu_x)\right)^2\right] \tag{5.16}$$

Expanding terms and simplifying:

$$J = E[f(x)^2] + E[\check{f}^2] + E[n^2(x - \mu_x)^2] - E[2f(x)\check{f}] - E[2f(x)n(x - \mu_x)] + E[2\check{f}n(x - \mu_x)]$$

$$J = E[f(x)^2] + \check{f}^2 + n^2E[(x-\mu_x)^2] - 2\check{f}E[f(x)] - 2nE[f(x)(x-\mu_x)] + 2\check{f}nE[(x-\mu_x)] \tag{5.17}$$

The following conditions must be met for J to be a minimum: the Hessian matrix J_H must be positive definite and the following equations must be true:

$$\frac{\partial J}{\partial \check{f}} = 0 \tag{5.18}$$

$$\frac{\partial J}{\partial n} = 0 \tag{5.19}$$

Noting that $E[(x - \mu_x)]$=0 and utilizing the variance equation 5.4, the partial derivatives of 5.17 are

$$\frac{\partial J}{\partial \check{f}} = 2\check{f} - 2E[f(x)] = 0 \tag{5.20}$$

$$\frac{\partial J}{\partial n} = 2nP_x - 2E[f(x)(x - \mu_x)] = 0 \tag{5.21}$$

$$J_H = \begin{bmatrix} \frac{\partial^2 J}{\partial \check{f}^2} & \frac{\partial^2 J}{\partial \check{f} \partial n} \\ \frac{\partial^2 J}{\partial n \partial \check{f}} & \frac{\partial^2 J}{\partial n^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2P_x \end{bmatrix} \tag{5.22}$$

Since the Hessian matrix J_H is positive definite, and the first derivatives are set to zero, it is assured that the system of equations is at a minimum. Solving 5.20 and 5.21 for $\check{f}$ and n,

$$\boxed{\check{f} = E[f(x)]} \tag{5.23}$$

$$nP_x = E[f(x)(x - \mu_x)]$$

$$\boxed{n = E[f(x)(x - \mu_x)]P_x^{-1}} \tag{5.24}$$

These results agree with the results from [87]. In summary, y is approximated by $f_a(x)$

$$y = f(x) \approx f_a(x) = \check{f} + n(x - \mu_x) \tag{5.25}$$

and the approximate statistics of y are (compare with equations 5.7 and 5.8)

$$\mu_y \approx E[f(x)] = \check{f} \tag{5.26}$$

$$P_y \approx n^2 P_x = \Big(E[f(x)(x - \mu_x)]\Big)^2 P_x^{-1} \tag{5.27}$$

Equations 5.23-5.27 are the general equations for statistical linearization. They show that the linearization of the nonlinear function y depends on the mean, variance, and PDF of the random variable input x. Solving these equations using a specific PDF and nonlinear function yields the random input describing function, or RIDF. Tables of RIDFs for common nonlinearities can be found in [76].

5.4 Statistical Linearization Example: The Saturation Function

This section derives the statistical linearization equations for the saturation function shown in Figure 5.2. This function is also called a limiter function. The variable a_c represents the commanded acceleration, and a_m represents the realized acceleration. The saturation

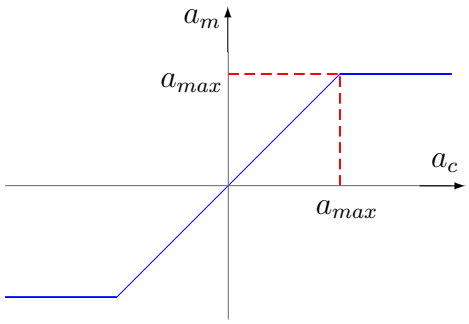

Fig. 5.2: Saturation Function

equation is written as

$$y = f(x) = \left\{ \begin{array}{ll} x, & \text{if } |x| \leq a_{max} \\ a_{max} sgn(x), & \text{if } |x| > a_{max} \end{array} \right\} \tag{5.28}$$

Gaussian random variable theory has been extensively developed, and it is common to use the probability function $PF(x)$, the probability integral $PI(x)$, and a third function $G(x)$ as auxiliary functions for Gaussian random variables (See ch. 4 of [73]). The equation for the probability function $PF(x)$, illustrated in Figure 5.3a, is:

$$PF(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{5.29}$$

The probability integral $PI(x)$ is shown in Figure 5.3b and is written as:

$$PI(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{\xi^2}{2}} \, d\xi = \frac{1}{2}\left(1 + erf(\frac{x}{\sqrt{2}})\right) \tag{5.30}$$

The third auxiliary function $G(x)$, illustrated in Figure 5.4, is defined as:

$$G(x) = xPI(x) + PF(x) = \int_{-\infty}^{x} PI(\xi) \, d\xi \tag{5.31}$$

The function $G(x)$ goes to zero as $x \rightarrow -\infty$ and approaches the line $y = x$ as $x \rightarrow \infty$.

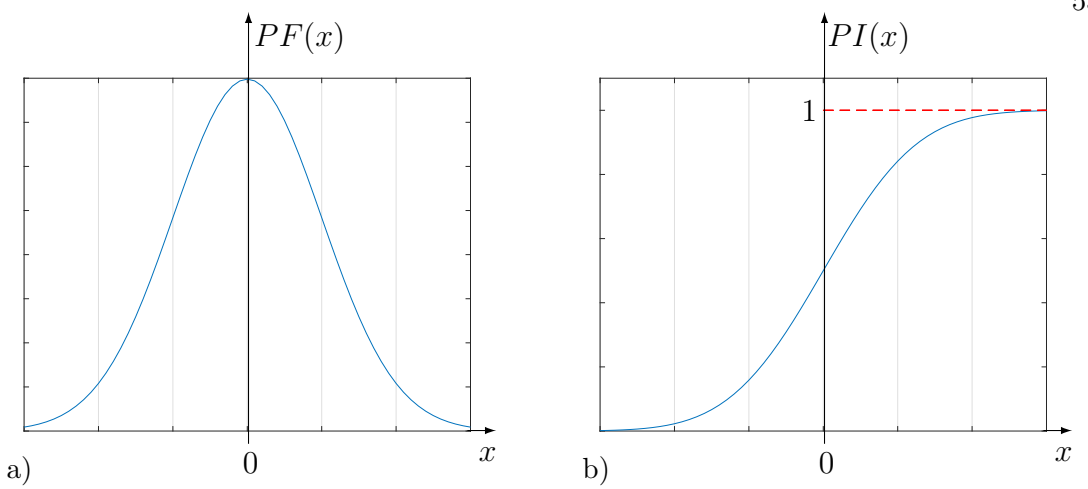

Fig. 5.3: Probability Function And Probability Integral

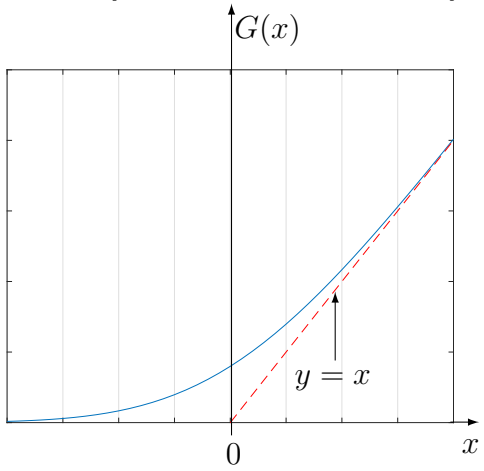

Fig. 5.4: Probability Auxiliary Function $G(x)$

Properties of the Gaussian auxiliary functions include:

$$PF(-x) = PF(x) \tag{5.32}$$

$$PI(-x) = 1 - PI(x) \tag{5.33}$$

$$G(-x) = G(x) - x \tag{5.34}$$

$$erf(x) = 2PI(\sqrt{2}x) - 1) = \frac{2}{\sqrt{\pi}} \int_0^x e^{\frac{-\xi^2}{2}} \, d\xi \tag{5.35}$$

5.4.1 Saturation Statistical Linearization Derivation

The statistical linearization equations from Section 5.3 are:

$$y \approx \check{f} + n(x - \mu_x) \tag{5.36}$$

$$\check{f} = E[f(x)] \tag{5.37}$$

$$n = E[f(x)(x - \mu_x)]P_x^{-1} \tag{5.38}$$

Substituting equations 5.2 and 5.5 into equation 5.37:

$$\check{f} = E[f(x)] = \int_{-\infty}^{\infty} f(\xi) \frac{1}{\sigma_x \sqrt{2\pi}} e^{\frac{-(\xi-\mu_x)^2}{2\sigma_x^2}} \, d\xi \tag{5.39}$$

Substituting equation 5.28 into equation 5.39 and separating into three integral terms:

$$\check{f} = \frac{1}{\sigma_x \sqrt{2\pi}} \left\{ \int_{-\infty}^{-a_{max}} -a_{max} e^{\frac{-(\xi-\mu_x)^2}{2\sigma_x^2}} \, d\xi + \int_{-a_{max}}^{a_{max}} \xi e^{\frac{-(\xi-\mu_x)^2}{2\sigma_x^2}} \, d\xi + \int_{a_{max}}^{\infty} a_{max} e^{\frac{-(\xi-\mu_x)^2}{2\sigma_x^2}} \, d\xi \right\}$$

$$= \frac{-a_{max}}{\sigma_x \sqrt{2\pi}} \int_{-\infty}^{-a_{max}} e^{\frac{-(\xi-\mu_x)^2}{2\sigma_x^2}} \, d\xi + \frac{1}{\sigma_x \sqrt{2\pi}} \int_{-a_{max}}^{a_{max}} \xi e^{\frac{-(\xi-\mu_x)^2}{2\sigma_x^2}} \, d\xi + \frac{a_{max}}{\sigma_x \sqrt{2\pi}} \int_{a_{max}}^{\infty} e^{\frac{-(\xi-\mu_x)^2}{2\sigma_x^2}} \, d\xi \tag{5.40}$$

Next, utilize the following substitution:

$$u = \frac{x - \mu_x}{\sigma_x} \tag{5.41}$$

Calculating the integration limits of 5.40 with the substitution yields

$$x_1 = -a_{max} \rightarrow u_1 = -\frac{a_{max} + \mu_x}{\sigma_x} \tag{5.42}$$

$$x_2 = a_{max} \rightarrow u_2 = \frac{a_{max} - \mu_x}{\sigma_x} \tag{5.43}$$

Substitute equations 5.41-5.43 into equation 5.40:

$$\check{f} = \frac{-a_{max}}{\sqrt{2\pi}} \int_{-\infty}^{u_1} e^{-\frac{u^2}{2}}\, du + \frac{1}{\sqrt{2\pi}} \int_{u_1}^{u_2} (u\sigma_x + \mu_x) e^{\frac{-u^2}{2}}\, du + \frac{a_{max}}{\sqrt{2\pi}} \int_{u_2}^{\infty} e^{-\frac{u^2}{2}}\, du \tag{5.44}$$

Substituting equation 5.30 into 5.44 and utilizing property 5.32 yields

$$\check{f} = -a_{max} PI(u_1) + \frac{1}{\sqrt{2\pi}} \int_{u_1}^{u_2} (u\sigma_x + \mu_x) e^{\frac{-u^2}{2}}\, du + a_{max} PI(-u_2) \tag{5.45}$$

Breaking up the integral term:

$$= a_{max} PI(-u_2) - a_{max} PI(u_1) + \frac{\sigma_x}{\sqrt{2\pi}} \int_{u_1}^{u_2} u e^{\frac{-u^2}{2}}\, du + \frac{\mu_x}{\sqrt{2\pi}} \int_{u_1}^{u_2} e^{\frac{-u^2}{2}}\, du \tag{5.46}$$

Evaluating the integrals:

$$\check{f} = a_{max} PI(-u_2) - a_{max} PI(u_1) - \frac{\sigma_x}{\sqrt{2\pi}} \left[e^{\frac{-u_2^2}{2}} \right] + \frac{\sigma_x}{\sqrt{2\pi}} \left[e^{\frac{-u_1^2}{2}} \right]$$

$$+ \frac{\mu_x}{2} \left[erf(\frac{u_2}{\sqrt{2}}) \right] - \frac{\mu_x}{2} \left[erf(\frac{u_1}{\sqrt{2}}) \right] \tag{5.47}$$

Utilizing properties 5.35 and 5.33:

$$\check{f} = a_{max} - a_{max} PI(u_2) - a_{max} PI(u_1) - \sigma_x PF(u_2) + \sigma_x PF(u_1) + \mu_x PI(u_2) - \mu_x PI(u_1)$$

$$= \frac{a_{max}}{\sigma_x} + G(u_1) - G(u_2) \tag{5.48}$$

Applying property 5.34 and simplifying yields the final result for $\check{f}$

$$\check{f} = \frac{a_{max}}{\sigma_x} + G(-u_1) + u_1 - G(u_2) \tag{5.49}$$

$$\boxed{\check{f} = \sigma_x \left[G\left(\frac{a_{max} + \mu_x}{\sigma_x} \right) - G\left(\frac{a_{max} - \mu_x}{\sigma_x} \right) \right] - \mu_x} \tag{5.50}$$

Next, the describing function gain n is calculated using Corollary 1.2-10 from [73]. The corollary is written as:

$$n = \frac{\partial \check{f}}{\partial \mu_x} \tag{5.51}$$

This simplifies the process of calculating the expectation operator in equation 5.24. According to equation 5.31,

$$\frac{\partial G(x)}{\partial x} = PI(x) \tag{5.52}$$

Substituting equation 5.52 and the corollary from 5.51 into equation 5.50:

$$\boxed{n = \left[PI\left(\frac{a_{max} + \mu_x}{\sigma_x} \right) + PI\left(\frac{a_{max} - \mu_x}{\sigma_x} \right) \right] - 1} \tag{5.53}$$

Equations 5.50 and 5.53 match the results from Taylor chapter 4 [73] and the RIDF table found in Gelb Appendix E [76]. The gains are used to approximate equation 5.28 with the linear function

$$y \approx \check{f} + n(x - \mu_x) \tag{5.54}$$

Although the derivations for these RIDFs can be extensive, they are very powerful once they are computed. They do not require the derivatives of the nonlinear function, which means that they can handle functions with discontinuous derivatives and periodic functions. Statistical linearization yields similar results to traditional linearization in linear regions, and provides a minimum error linearization in regions with a sharp corner, discontinuity, or otherwise strong nonlinearity (e.g. trigonometric functions). Nonlinearities such as a relay, saturation, or dead zone, are particularly well suited to statistical linearization. For these functions, the statistical linearization is significantly more accurate than traditional linearization.

5.5 CADET Theoretical Development

The Covariance Analysis Describing Function Technique (CADET) is a method of analysis that was developed at The Analytic Sciences Corporation in the 1970s. CADET utilizes statistical linearization to perform traditional covariance analysis on a system. The purpose of CADET is to efficiently calculate the output statistics of nonlinear systems that have stochastic inputs. CADET was originally developed for analyzing missile engagement problems, but it is applicable for any nonlinear system. This chapter reviews the traditional linear and nonlinear system analysis equations, and develops the general CADET equations. The equations in this chapter can be found in [4, 73, 77, 85].

5.5.1 Traditional Linear Analysis

Consider the general linear system where $\mathbf{x}$ is the state vector and $\dot{\mathbf{x}}$ is sufficient to represent the system behavior.

$$\dot{\mathbf{x}} = F\mathbf{x} + B\mathbf{w} \tag{5.55}$$

The vector $\mathbf{w}$ is assumed to be a zero-mean Gaussian white noise vector with power spectral density (PSD) Q. The differential equation for the state covariance of the system is

$$\dot{P} = FP + PF^T + BQB \tag{5.56}$$

Equation 5.56 represents a direct method of analysis, meaning the results can be calculated without simulating multiple trajectories and computing ensemble statistics. Since the system is linear, the exact statistics of the system are computed. This form of analysis is the basis for approximating the statistics of nonlinear systems.

5.5.2 Nonlinear Analysis and CADET Equations

The fundamental idea of CADET is similar to other covariance analysis methods (See Section 3.1.1). Using linearization methods, the mean and covariance of a state vector with random variable inputs are propagated. CADET is different than other methods because it uses statistical linearization methods rather than traditional linearization methods. With the statistical linearization coefficients $\check{f}$ and n already derived in Chapter 5, the CADET equations are straightforward. The CADET technique assumes the state variables are approximately jointly normal [73], and usually assumes a Gaussian input.

Consider the general nonlinear system where $\mathbf{x}$ is the state vector and $\dot{\mathbf{x}}$ is sufficient to represent the system behavior.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + B\mathbf{w} \tag{5.57}$$

where $\mathbf{f}(\mathbf{x})$ contains a linear component and a nonlinear component

$$\mathbf{f}(\mathbf{x}) = F_{lin}\mathbf{x} + Af_{nl}(\mathbf{x}) \tag{5.58}$$

In this example, F_{lin} is the partial derivative of the linear dynamics with respect to the state vector and is called the linear dynamics matrix. The function $f_{nl}(\mathbf{x})$ is the nonlinear function in the state, and the matrix A assigns the nonlinear function to the correct state values. Note that the noise vector $\mathbf{w}$ is a zero mean Gaussian white noise process with PSD Q. This system is approximated using statistical linearization with the equation

$$\mathbf{f}(\mathbf{x}) \approx \check{\mathbf{f}} + N(\mathbf{x} - \boldsymbol{\mu}) \tag{5.59}$$

where $\boldsymbol{\mu}$ is the mean value of the state vector and $\check{\mathbf{f}}$ contains the linear and nonlinear expected values:

$$\check{\mathbf{f}} = E[\mathbf{f}(\mathbf{x})] = F_{lin}\boldsymbol{\mu} + A\check{f} \tag{5.60}$$

The matrix N, which also contains linear and nonlinear information, is used in the function approximation in equation 5.57, as well as in the covariance propagation. N is called the

quasi-linear system dynamics matrix (QLSDM), and it represents the linearized version of the system. N is comparable in a linear system to the partial derivative of the dynamics with respect to the state vector (e.g., F from equation 5.55). The QLSDM is formally defined as (see equation 1.2-6 of [73]):

$$N = E[\mathbf{f}(\mathbf{x})(\mathbf{x} - \boldsymbol{\mu})^T]P^{-1} \tag{5.61}$$

The matrix N identical to F_{lin}, but with the scalar describing function gain n substituted in for the element with the nonlinearity. Or in other words, the scalar statistical linearization coefficient n is an element of the matrix N, and N contains linearized dynamics from the nonlinear function $\mathbf{f}(\mathbf{x})$. The coefficients $\check{f}$ and n are calculated using the random input describing function for the nonlinearity (see equations 5.50 and 5.53).

Using the statistically linearized dynamics, the differential equations for the mean and state covariance are given by equations 5.62 and 5.63 below.

$$\dot{\boldsymbol{\mu}} = \check{\mathbf{f}} \tag{5.62}$$

$$\dot{P} = NP + PN^T + BQB^T \tag{5.63}$$

Implementing equations 5.62 and 5.63 in a simulation is similar to traditional covariance analysis, with the extra step of calculating $\check{f}$ and n prior to calculating $\dot{m}$ and $\dot{P}$. Extra care must be taken to apply $\check{f}$ and n to the correct elements of the state vector $\mathbf{x}$. A flow chart for implementing CADET is shown in Figure 5.5.

5.5.3 CADET Example: Missile Engagement Problem

This section provides a step-by-step walk through of implementing CADET into a missile engagement covariance analysis. The example under consideration is from the original CADET journal paper by Gelb and Warren [2]. (See also ch. 2 of [73]) This problem is a dispersion analysis problem, without any measurements or navigation states. The state

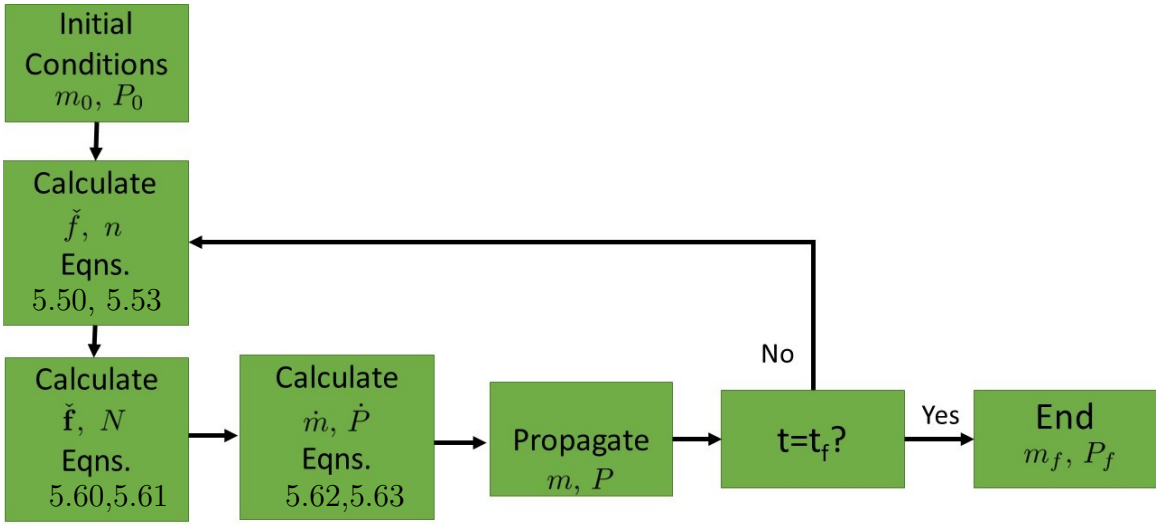

Fig. 5.5: CADET Flow Chart

vector for this problem is

$$\mathbf{x} = \begin{bmatrix} y \\ \dot{y} \\ a_{m,un} \\ a_t \end{bmatrix}_{4\times 1} \tag{5.64}$$

where y is the lateral relative separation, $\dot{y}$ is the lateral relative velocity, $a_{m,un}$ is the unlimited lateral missile acceleration, and a_t is the target lateral acceleration. The dynamics of the system are:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ \dot{a}_{m,un} \\ \dot{a}_t \end{bmatrix} = \begin{bmatrix} \dot{y} \\ a_t - a_m \\ \frac{a_c - a_{m,un}}{\tau_m} \\ -\frac{1}{\tau_t} + w_t \end{bmatrix} \tag{5.65}$$

The realized missile acceleration a_m has a maximum value of a_{max}. This is implemented with a saturation function on $a_{m,un}$. The realized missile acceleration a_m is related to the

unlimited missile acceleration $a_{m,un}$ with the equation

$$a_m = \left\{ \begin{array}{ll} a_{m,un}, & \text{if } |a_{m,un}| \leq a_{max} \\ a_{max} sgn(a_{m,un}), & \text{if } |a_{m,un}| > a_{max} \end{array} \right\} \tag{5.66}$$

The missile guidance law is defined as

$$a_c = \frac{k}{t_{go}} \left(\dot{y} + \frac{y}{t_{go}} \right) \tag{5.67}$$

where the time remaining in the engagement is t_{go}. The missile acceleration is driven by a low pass filter on the acceleration command a_c, and the target acceleration is driven by an exponentially correlated random variable (ECRV). The noise input for the target random walk w_t is zero mean Gaussian white noise with PSD Q. The time-constant of the missile actuator is τ_m, and the time-constant of the target acceleration is τ_t. The parameters for this example problem are defined in table 5.2.

Table 5.2: Example Problem Parameter Values

Parameter	Value	Units	Description
t_f	10	s	length of simulation
τ_m	1	s	missile acceleration time-constant
τ_t	1	s	target acceleration time-constant
$\sigma_{ss,at}$	$15g$	$\frac{m}{s^2}$	3σ steady state target ECRV acceleration
k	3		control law gain
a_{max}	$10g$	$\frac{m}{s^2}$	missile max acceleration

The first step for the simulation is to define the initial mean $\boldsymbol{\mu}_0$ and covariance P_0 of the state vector (See Figure 5.3). The next step for CADET is to begin the integration loop of the simulation. Each subsequent step is repeated until $t = t_f$. First, the scalar statistical linearization coefficients $\check{f}$ and n are calculated using equations 5.50 and 5.53. Next, the vector $\check{\mathbf{f}}$ and matrix N are calculated with equations 5.60 and 5.61. Then, $\dot{\boldsymbol{\mu}}$ and $\dot{P}$ are

Table 5.3: Example Problem Initial Values

State	m_0	Units	$3\sigma_x$
y	0	m	1
$\dot{y}$	0	$\frac{m}{s}$	0.1
$a_{m,un}$	0	$\frac{m}{s^2}$	0.01
a_m	0	$\frac{m}{s^2}$	0.01

computed with equations 5.62 and 5.63. The equation substitutions are shown below:

$$\check{\mathbf{f}} = F_{lin}\boldsymbol{\mu} + A\check{f}$$

$$\check{\mathbf{f}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{t_{go}^2\tau_m} & \frac{k}{t_{go}\tau_m} & -\frac{1}{\tau_m} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_t} \end{bmatrix} \boldsymbol{\mu} + \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \check{f} \tag{5.68}$$

The matrix F_{lin} is the 4×4 linear dynamics matrix and the 4×1 matrix A is the nonlinear gain assignment matrix. The combined expectation vector $\check{\mathbf{f}}$ represents the linearized system, with both linear and approximated nonlinear elements included. The mean propagation equation $\dot{\boldsymbol{\mu}}$ is:

$$\boxed{\dot{\boldsymbol{\mu}} = \check{\mathbf{f}}} \tag{5.69}$$

The state covariance is propagated with the quasi-linear system dynamics matrix N. Note the similarity between the QLSDM N and the system dynamics matrix F_{ideal} for a system without the saturation.

$$N = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -n & 1 \\ \frac{k}{t_{go}^2\tau_m} & \frac{k}{t_{go}\tau_m} & -\frac{1}{\tau_m} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_t} \end{bmatrix} \tag{5.70}$$

$$F_{ideal} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ \frac{k}{t_{go}^2 \tau_m} & \frac{k}{t_{go} \tau_m} & -\frac{1}{\tau_m} & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau_t} \end{bmatrix} \tag{5.71}$$

The state covariance propagation equation is:

$$\boxed{\dot{P} = NP + PN^T + BQB^T} \tag{5.72}$$

With $\dot{\boldsymbol{\mu}}$ and $\dot{P}$ calculated, the mean and covariance are propagated forward using an rk4 integrator.

The results of the CADET simulation are shown in Figure 5.6. The CADET results are compared to a 1000 run Monte Carlo simulation and to traditional covariance analysis (TCA). The figure illustrates the 3σ relative position dispersions of the engagement. The dispersions grow in the first few seconds as the target accelerates, and shrink toward the end as the missile attempts to engage. The 3σ dispersions at t_f represent the 3σ miss distance of the engagement. The a_{max} value chosen for this problem was chosen to represent a moderate saturation, therefore, the 3σ miss distance for this engagement is somewhat high. The TCA results in Figure 5.6 show that traditional linearization does not account for the missile saturation, and is overly-optimistic on the final miss distance. The CADET results line up well with the Monte Carlo results.

An extension to this example problem is to examine the performance of CADET under varying levels of saturation. The results of this analysis are shown in Figure 5.7. The missile maximum acceleration a_{max} is varied from 1g to 40g. The statistical linearization results agree with the Monte Carlo results, highlighting the accuracy of the statistical linearization for the full domain of acceleration inputs. The TCA results show that traditional linearization is not able to handle saturations, predicting a low miss distance even during high saturation runs. The Monte Carlo simulation takes about 2 hours to run while the CADET simulation runs in about 8 seconds.

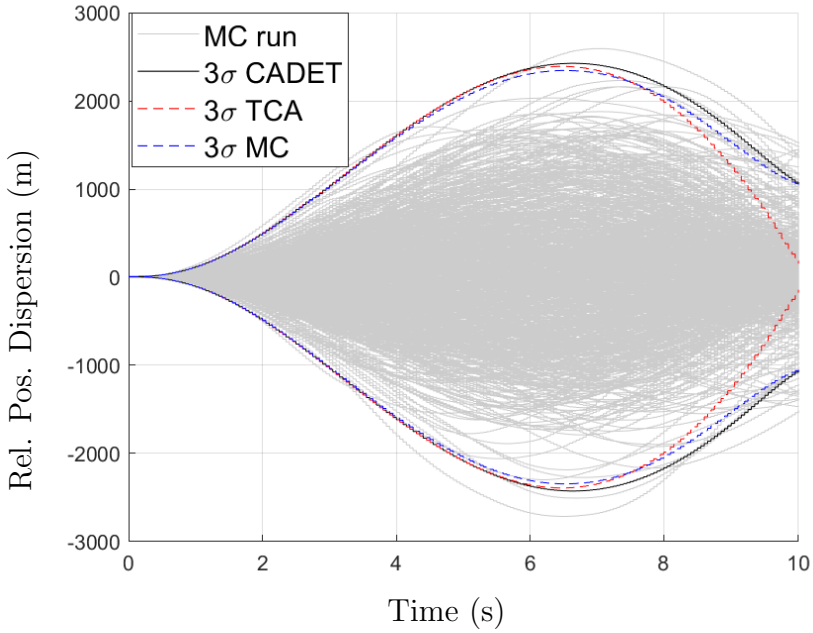

Fig. 5.6: Relative Lateral Position Dispersions

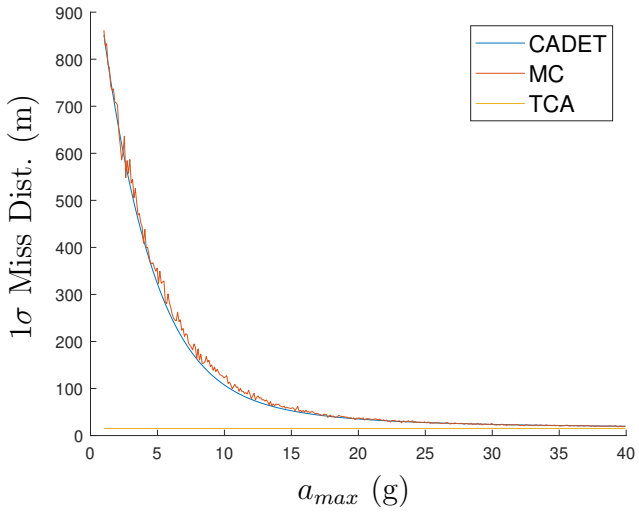

Fig. 5.7: CADET Miss Distance Vs. a_{max}

This chapter defined the statistical linearization method for approximating a nonlinear system, and showed the process for calculating the linear coefficients. Next, the Covariance Analysis Describing Function Technique (CADET) for applying statistical linearization to traditional covariance analysis was described. For a simple missile engagement problem, the CADET method replicates nonlinear Monte Carlo simulation results with a fraction of the

computing time. In Chapter 7 of this dissertation, this method will be applied to augmented state linear covariance analysis.

CHAPTER 6
LINCOV FOR MISSILE ENGAGEMENTS

The missile engagement problem is a new application for augmented state linear covariance analysis (LinCov). Therefore, it is important to establish a baseline simulation upon which future research can expand from. Once it is shown that the LinCov framework can accurately model the standard setup of sensors and dynamics in the missile engagement problem, it is possible to conduct further LinCov research on the actuator saturation problem, advanced guidance laws, game theory, cluttered environment target assessment, or numerous other areas of interest. This chapter introduces the specifics of the missile engagement problem and how to model it for analysis. In Section 6.1, the engagement problem is introduced, and a few characteristic parameters of missile engagements, crossing angle and closest approach distance (miss distance), are defined. Next, in sections 6.2-6.4, the equations and models used in the Monte Carlo and the LinCov simulation are defined. Section 6.5 details the steps required to validate a LinCov simulation against a Monte Carlo simulation, and shows the validation results for the LinCov simulation. Finally, Section 6.6 presents the results of an error budget study that investigates the sensitivity of the miss distance to sensor parameters and initial conditions.

6.1 Engagement Problem Introduction

As discussed in Chapter 3, the goal of an engagement is usually not a direct collision, but to pass within a certain distance of the target with the correct crossing angle and body angle. The maximum range that will still result in lethal damage to the target depends on the vulnerabilities of the target and fragmentation pattern of the ordnance, which properties are not publicly available. For purposes of demonstration, this chapter uses 10 meters as the threshold for lethality. Therefore, in order to meet the design requirement of a successful engagement, the 3σ miss distance of the Monte Carlo simulation must remain under 10 meters. Figure 6.1 illustrates the crossing angle γ of the engagement. The blue arrow represents the interceptor velocity vector $\mathbf{v}$, while the red arrow represents the target

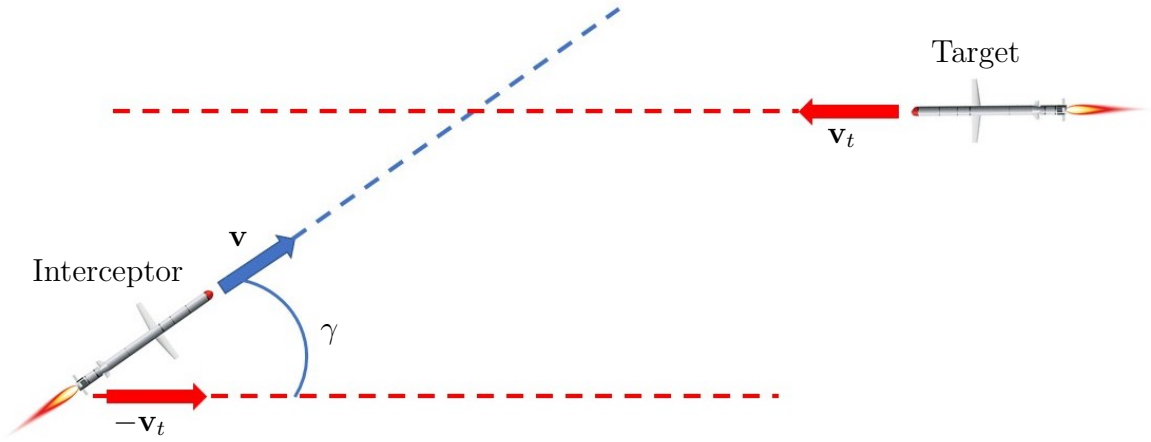

Fig. 6.1: Crossing Angle Of An Engagement

velocity vector $\mathbf{v}_t$. The crossing angle γ is defined as the angle between the interceptor velocity vector and the negative target velocity vector. In this dissertation, the crossing angle γ is always acute for near head-on engagements, and a direct head-on engagement has a crossing angle of 0 degrees. The equation for crossing angle is written as follows:

$$|\sin\gamma| = \frac{\|\mathbf{v}_t \times (-\mathbf{v})\|}{\|\mathbf{v}_t\|\|\mathbf{v}\|} \qquad \frac{-\pi}{2} \leq \gamma \leq \frac{\pi}{2} \tag{6.1}$$

The time of closest approach (TOCA) happens when the magnitude of the relative position vector starts increasing. This happens when the relative position vector $\mathbf{r}_{rel}$ is perpendicular to the relative velocity vector $\mathbf{v}_{rel}$. This is written in equation form as $\mathbf{r}_{rel} \cdot \mathbf{v}_{rel} = 0$. This concept is illustrated in figures 6.2 and 6.3. Figure 6.2 shows the engagement before TOCA. The yellow arrow represents the current relative position vector, and the purple arrow represents the current relative velocity vector. The green line shows what the miss distance will be if both vehicles remain on the same trajectory. As the engagement progresses to TOCA, it transitions to look like Figure 6.3, where $\mathbf{r}_{rel}$ and $\mathbf{v}_{rel}$ are perpendicular. Mathematically, the closest approach distance, or miss distance, is defined as the magnitude of the relative position vector at TOCA. The equation for miss distance is written as

$$d_{miss} = \|\mathbf{r}_t(t_f) - \mathbf{r}(t_f)\| \tag{6.2}$$

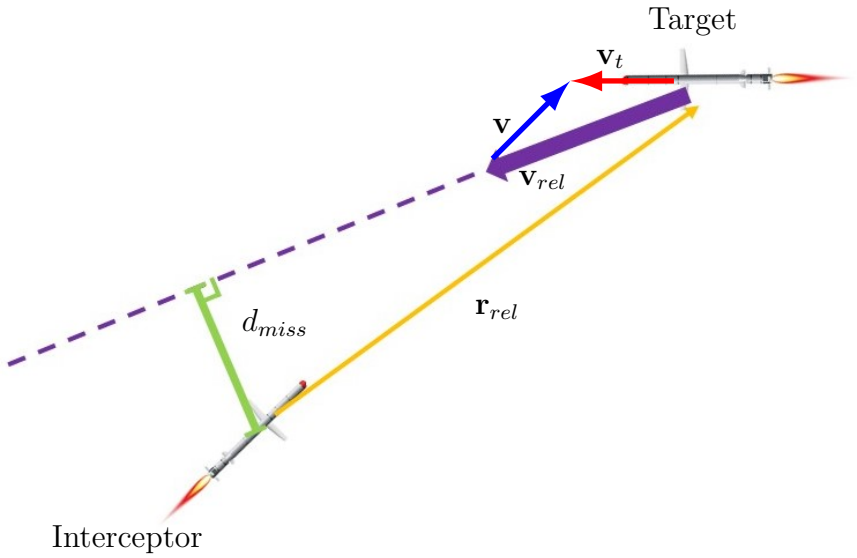

Fig. 6.2: Missile Configuration Before Time Of Closest Approach

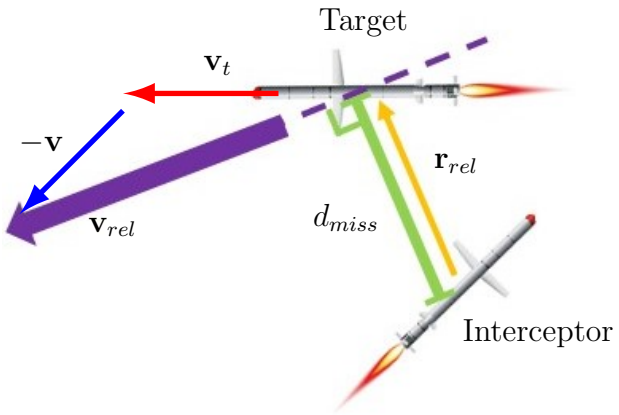

Fig. 6.3: Missile Configuration At Time Of Closest Approach

where t_f is the time of closest approach.

6.2 Simulation Assumptions

This simulation in this chapter relies on several assumptions that simplify the engagement problem. The first major assumption is that the engagement is a 2D point mass problem. The 2D assumption is not unrealistic, as most modern guidance laws are implemented in a single maneuver plane at a time [57]. Additionally, a point mass assumption is sufficient to present most missile engagement theory [8]. For examples of 6DOF LinCov applications, see [34, 36, 37]. Gravity is neglected for this simulation, as it is assumed that by the time the

interceptor reaches homing phase, it has found trim conditions to fly horizontally. Drag is also neglected for this simulation, which is representative of missile engagement dynamics during terminal homing (see page 3-3 of [73]). By design, this chapter assumes unlimited actuator acceleration capability for the missile, as actuator saturation is the subject of interest in upcoming chapters.

6.3 Monte Carlo Simulation

This section defines the analytical models for the Monte Carlo simulation, which are used to validate the LinCov simulation.

6.3.1 Truth State

The truth state vector contains position and velocity states for both vehicles, acceleration for the target, and bias states for the sensors. The bold font indicates a column vector, the length of which is described for clarity. For this 2-D problem, each position, velocity, and acceleration vector has two elements. The state vector has a total of 19 elements. The state vector is defined as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_{2\times1} \\ \mathbf{v}_{2\times1} \\ \mathbf{r}_{t_{2\times1}} \\ \mathbf{v}_{t_{2\times1}} \\ \mathbf{a}_{t_{2\times1}} \\ \mathbf{b}_{7\times1} \end{bmatrix}_{17\times1} \tag{6.3}$$

where **b** is the sensor bias vector with the following elements (see Section 6.3.3):

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_{acc_{2\times2}} \\ \mathbf{b}_{gps_{2\times2}} \\ b_r \\ b_{rr} \\ b_{los} \end{bmatrix}_{7\times1} \tag{6.4}$$

6.3.2 Truth State Dynamics

The state vector dynamics are generalized with the following nonlinear function:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\hat{\mathbf{u}}) + B\mathbf{w}$$

The dynamics are a function of the state **x**, the actuator commands $\hat{\mathbf{u}}$, and the noise vector **w**. The time derivative equations of the state are:

$$\dot{\mathbf{r}} = \mathbf{v} \tag{6.5}$$

$$\dot{\mathbf{v}} = \hat{\mathbf{u}} + \mathbf{w}_m \tag{6.6}$$

$$\dot{\mathbf{r}}_t = \mathbf{v}_t \tag{6.7}$$

$$\dot{\mathbf{v}}_t = \mathbf{a}_t \tag{6.8}$$

$$\dot{\mathbf{a}}_t = -\alpha\mathbf{a}_t + \mathbf{w}_s \tag{6.9}$$

$$\dot{\mathbf{b}}_{acc} = -\frac{\mathbf{b}_{acc}}{\tau_{acc}} + \mathbf{w}_{acc} \tag{6.10}$$

$$\dot{\mathbf{b}}_{gps} = -\frac{\mathbf{b}_{gps}}{\tau_{gps}} + \mathbf{w}_{gps} \tag{6.11}$$

$$\dot{b}_r = -\frac{b_r}{\tau_r} + w_r \tag{6.12}$$

$$\dot{b}_{rr} = -\frac{b_{rr}}{\tau_{rr}} + w_{rr} \tag{6.13}$$

$$\dot{b}_{los} = -\frac{b_{los}}{\tau_{los}} + w_{los} \tag{6.14}$$

The missile acceleration dynamics are modeled as the guidance law acceleration $\hat{\mathbf{u}}$ with added disturbance acceleration noise $\mathbf{w}_m$. The sensors biases are modeled as exponentially correlated random variables (ECRVs) [22]. The target acceleration is modeled as Singer motion [88]. Singer motion is similar to an ECRV, but with a different PDF. It relies on a time-constant τ_α and the acceleration maneuver variance σ_{sing}. Note that the original Singer paper [88] uses the reciprocal of the time-constant, $\alpha = \frac{1}{\tau_\alpha}$. The time-constant τ_α corresponds with the amount of time a typical maneuver would take. This can vary from $\tau_\alpha = 2$ seconds to $\tau_\alpha = 60$ seconds or more depending on whether the evasive maneuver is modeling high frequency turbulence, a quick maneuver, or a slow turn (in this case, $\frac{1}{2} > \alpha > \frac{1}{60}$). The target can accelerate at maximum acceleration a_{max} in either direction with probability p_{max}, or it can encounter zero acceleration with probability p_0. Additionally, it can accelerate at a level between $+a_{max}$ and $-a_{max}$ according to a uniform probability distribution. The probability density function of the Singer acceleration is shown in Fig. 6.4. The acceleration maneuver variance σ_{sing} is calculated with the following equation.

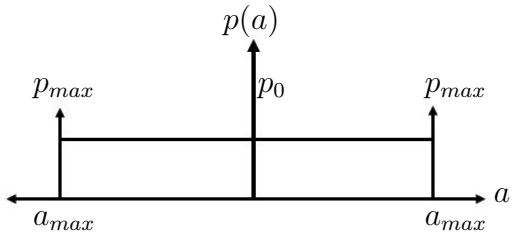

Fig. 6.4: Singer Noise Probability Density Function

$$\sigma^2_{sing} = \frac{a^2_{max}}{3}\Big(1 + 4p_{max} - p_0\Big) \tag{6.15}$$

The noise vector of the full system is written as:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_{m_{2\times1}} \\ \mathbf{w}_{s_{2\times1}} \\ \mathbf{w}_{acc_{2\times1}} \\ \mathbf{w}_{gps_{2\times1}} \\ w_r \\ w_{rr} \\ w_{los} \end{bmatrix}_{11\times1} \tag{6.16}$$

The noise is modeled as zero-mean Gaussian white noise, except for the target acceleration maneuver noise, which is non-Gaussian zero-mean white noise. The power spectral density values of the noise vector are found in equation 6.17 below.

$$S_{\mathbf{w}11\times11} = blkdiag\Big(S_{w_p}I_{2\times2}, 2\alpha\sigma^2_{sing}I_{2\times2}, \frac{2\sigma^2_{acc,ss}}{\tau_{acc}}I_{2\times2}, \frac{2\sigma^2_{gps,ss}}{\tau_{gps}}I_{2\times2}, ... \\ \frac{2\sigma^2_{r,ss}}{\tau_r}, \frac{2\sigma^2_{rr,ss}}{\tau_{rr}}, \frac{2\sigma^2_{los,ss}}{\tau_{los}}\Big) \tag{6.17}$$

The steady state covariance of each sensor is σ_{ss}, and the acceleration maneuver variance of the target is σ_{sing}. The 'blkdiag' function creates an 11×11 diagonal matrix with the input matrices. The noise assignment matrix B is

$$B = \begin{bmatrix} 0_{2\times2} & 0_{2\times2} & 0_{2\times7} \\ I_{2\times2} & 0_{2\times2} & 0_{2\times7} \\ 0_{4\times2} & 0_{4\times2} & 0_{4\times7} \\ 0_{2\times2} & I_{2\times2} & 0_{2\times7} \\ 0_{7\times2} & 0_{7\times2} & I_{7\times7} \end{bmatrix}_{17\times11} \tag{6.18}$$

The guidance law for the simulation $\hat{\mathbf{u}}$ is the standard proportional navigation law [8] and is computed with the navigation state vector $\hat{\mathbf{x}}$. The law calculates $\hat{n}_c$, the magnitude of the acceleration perpendicular to the line of sight vector that is needed for the interceptor to maintain a collision course with the target. The guidance law equation is:

$$\hat{\mathbf{u}} = \hat{n}_c \begin{bmatrix} -sin(\hat{\lambda}) \\ cos(\hat{\lambda}) \end{bmatrix}_{2\times 1} \qquad where \quad \hat{n}_c = \hat{k}_p \hat{V}_c \dot{\hat{\lambda}} \tag{6.19}$$

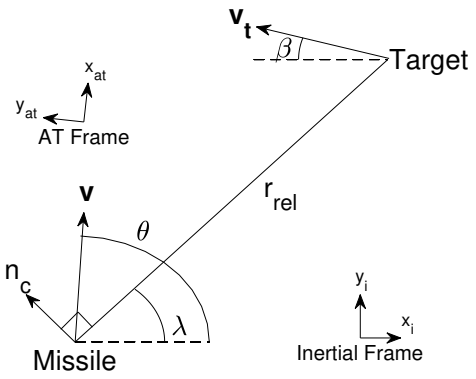

Fig. 6.5: Target Engagement Geometry

The scalar $\hat{k}_p$ is a unit-less gain, $\hat{V}_c$ is the scalar closing velocity, $\hat{\lambda}$ is the line-of-sight angle, and $\dot{\hat{\lambda}}$ is the scalar line-of-sight angle rate. Figure 6.5 illustrates the target engagement geometry.

6.3.3 Sensor Truth Models

The flight computer has access to an accelerometer that measures the true specific force on the vehicle $\dot{\mathbf{v}}$ corrupted by a first-order Markov process bias $\mathbf{b}_{acc}$ and zero-mean Gaussian white noise $\boldsymbol{\eta}$. The accelerometer is modeled as a continuous measurement. The Gaussian white noise in the sensor model $\boldsymbol{\eta}$ has a power spectral density (PSD) S_η.

$$\tilde{\mathbf{y}} = \hat{\mathbf{u}} + \mathbf{w}_m + \mathbf{b}_{acc} + \boldsymbol{\eta} \tag{6.20}$$

The remaining measurements update the state vector at discrete time intervals. In general, the measurement $\tilde{z}_k = h(x_k) + b_k + \nu_k$ represents a discrete measurement at time t_k corrupted by by a first-order Markov process bias b_k and a zero-mean Gaussian white noise ν_k. The GPS measurement gives the 2-D inertial position of the interceptor.

$$\tilde{\mathbf{z}}_{gps} = \mathbf{r} + \mathbf{b}_{gps} + \boldsymbol{\nu}_{gps} \tag{6.21}$$

The range measurement provides the distance from the interceptor to the target.

$$\tilde{z}_r = ||(\mathbf{r}_t - \mathbf{r})|| + b_r + \nu_r \tag{6.22}$$

The range rate measurement provides the time rate of change of the range from the interceptor to the target.

$$\tilde{z}_{rr} = \frac{\partial}{\partial t}||(\mathbf{r}_t - \mathbf{r})|| = \mathbf{i}_{rel} \cdot (\mathbf{v}_t - \mathbf{v}) + b_{rr} + \nu_{rr} \tag{6.23}$$

where $\mathbf{i}_{rel}$ is the unit relative position vector. The line-of-sight angle measurement is the angle from the inertial x-axis to the line-of-sight vector from the interceptor to the target.

$$\tilde{z}_{los} = \arctan\left(\frac{i_{rel_y}}{i_{rel_x}}\right) + b_{los} + \nu_{los} \tag{6.24}$$

where components of the unit relative position vector from the interceptor to the target are i_{rel_x} and i_{rel_y}. The noise covariance for each sensor is found in the following equation:

$$R_{\nu 5x5} = diag\Big(R_{\nu_{gpsx}}, R_{\nu_{gpsy}}, R_{\nu_r}, R_{\nu_{rr}}, R_{\nu_{los}}\Big) \tag{6.25}$$

6.3.4 Navigation Filter Design Model

The navigation filter design model provides the basis for the filter propagation equations. The design model is often simpler than the truth model, and the noise parameters of the filter are sometimes different from the truth model as well. In this problem, the design model differs from the truth model in several areas. The design model lacks a interceptor or target acceleration state, and models the target velocity dynamics with Gaussian white noise. Additionally, the design model uses model replacement for the interceptor velocity dynamics instead of traditional dynamics [89, 90].

$$\mathbf{x}^{dm} = \begin{bmatrix} \mathbf{r}^{dm}_{2\times1} \\ \mathbf{v}^{dm}_{2\times1} \\ \mathbf{r}^{dm}_{t_{2\times1}} \\ \mathbf{v}^{dm}_{t_{2\times1}} \\ \mathbf{b}^{dm}_{7\times1} \end{bmatrix}_{15\times1} \tag{6.26}$$

where

$$\mathbf{b}^{dm} = \begin{bmatrix} \mathbf{b}^{dm}_{acc} \\ \mathbf{b}^{dm}_{gps} \\ b^{dm}_{r} \\ b^{dm}_{rr} \\ b^{dm}_{los} \end{bmatrix}_{7\times1} \tag{6.27}$$

The design model dynamics are generalized with the nonlinear function

$$\dot{\mathbf{x}}^{dm} = \mathbf{f}^{dm}(\mathbf{x}^{dm}, \tilde{y}^{dm}) + B^{dm}\mathbf{w}^{dm}$$

Breaking down the design model dynamics:

$$\dot{\mathbf{r}}^{dm} = \mathbf{v}^{dm} \tag{6.28}$$

$$\dot{\mathbf{v}}^{dm} = \tilde{\mathbf{y}}^{dm} - \mathbf{b}_{acc}^{dm} - \boldsymbol{\eta}^{dm} \tag{6.29}$$

$$\dot{\mathbf{r}}_t^{dm} = \mathbf{v}_t^{dm} \tag{6.30}$$

$$\dot{\mathbf{v}}_t^{dm} = \mathbf{w}_t^{dm} \tag{6.31}$$

$$\dot{\mathbf{b}}_{acc}^{dm} = -\frac{\mathbf{b}_{acc}^{dm}}{\tau_{acc}^{dm}} + \mathbf{w}_{acc}^{dm} \tag{6.32}$$

$$\dot{\mathbf{b}}_{gps}^{dm} = -\frac{\mathbf{b}_{gps}^{dm}}{\tau_{gps}^{dm}} + \mathbf{w}_{gps}^{dm} \tag{6.33}$$

$$\dot{b}_r^{dm} = -\frac{b_r^{dm}}{\tau_r^{dm}} + w_r^{dm} \tag{6.34}$$

$$\dot{b}_{rr}^{dm} = -\frac{b_{rr}^{dm}}{\tau_{rr}^{dm}} + w_{rr}^{dm} \tag{6.35}$$

$$\dot{b}_{los}^{dm} = -\frac{b_{los}^{dm}}{\tau_{los}^{dm}} + w_{los}^{dm} \tag{6.36}$$

The design model noise vector includes white noise from the sensor biases and disturbance accelerations in the velocity dynamics. The design model noise vector is:

$$\mathbf{w}^{dm} = \begin{bmatrix} \mathbf{w}_{t_{2\times1}}^{dm} \\ \mathbf{w}_{acc_{2\times1}}^{dm} \\ \mathbf{w}_{gps_{2\times1}}^{dm} \\ w_r^{dm} \\ w_{rr}^{dm} \\ w_{los}^{dm} \end{bmatrix}_{9\times1} \tag{6.37}$$

The noise characteristics are as follows:

$$E\big[\mathbf{w}^{dm}(t)\big] = \mathbf{0} \tag{6.38}$$

$$E\big[\mathbf{w}^{dm}(t)\mathbf{w}^{dm}(t')^T\big] = S_{\mathbf{w}}^{dm}\delta(t-t') \tag{6.39}$$

$$S_{\mathbf{w}}^{dm}{}_{9\times9} = blkdiag\Big(S_{w_t}^{dm} I_{2\times2}, \frac{2\sigma_{acc,ss}^{dm^2}}{\tau_{acc}^{dm}} I_{2\times2}, \frac{2\sigma_{gps,ss}^{dm^2}}{\tau_{gps}^{dm}} \tau_{gps}^{dm} I_{2\times2}, \frac{2\sigma_{r,ss}^{dm^2}}{\tau_{r}^{dm}}, \frac{2\sigma_{rr,ss}^{dm^2}}{\tau_{rr}^{dm}}, \frac{2\sigma_{los,ss}^{dm^2}}{\tau_{los}^{dm}}\Big) \tag{6.40}$$

The design model noise mapping matrix B^{dm} is

$$B^{dm} = \begin{bmatrix} 0_{6\times9} \\ I_{9\times9} \end{bmatrix}_{15\times9} \tag{6.41}$$

The design model continuous accelerometer measurement is

$$\tilde{\mathbf{y}}^{dm} = \dot{\mathbf{v}}^{dm} + b_{acc}^{dm} + \eta^{dm} \tag{6.42}$$

The design model accelerometer noise $\boldsymbol{\eta}^{dm}$ has the following noise characteristics:

$$E[\boldsymbol{\eta}^{dm}(t)] = 0 \tag{6.43}$$

$$E[\boldsymbol{\eta}^{dm}(t)\boldsymbol{\eta}^{dm}(t')] = S_{\eta}^{dm}\delta(t-t') \tag{6.44}$$

6.3.5 Navigation State and Dynamics

The navigation state and dynamics equations are used by the flight computer to propagate the navigation state. The navigation states include estimates of the position and velocity of each vehicle, and the sensor biases.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{r}}_{2\times1} \\ \hat{\mathbf{v}}_{2\times1} \\ \hat{\mathbf{r}}_{t_{2\times1}} \\ \hat{\mathbf{v}}_{t_{2\times1}} \\ \hat{\mathbf{b}}_{7\times1} \end{bmatrix}_{15\times1} \tag{6.45}$$

where

$$\hat{\mathbf{b}} = \begin{bmatrix} \hat{\mathbf{b}}_{acc} \\ \hat{\mathbf{b}}_{gps} \\ \hat{b}_r \\ \hat{b}_{rr} \\ \hat{b}_{los} \end{bmatrix}_{7\times 1} \tag{6.46}$$

The dynamics of the navigation states are equal to the design model dynamic equations without the noise.

$$\dot{\hat{\mathbf{r}}} = \hat{\mathbf{v}} \tag{6.47}$$

$$\dot{\hat{\mathbf{v}}} = \tilde{\mathbf{y}} - \hat{\mathbf{b}}_{acc} \tag{6.48}$$

$$\dot{\hat{\mathbf{r}}}_t = \hat{\mathbf{v}}_t \tag{6.49}$$

$$\dot{\hat{\mathbf{v}}}_t = \mathbf{0} \tag{6.50}$$

$$\dot{\hat{\mathbf{b}}}_{acc} = -\frac{\hat{\mathbf{b}}_{acc}}{\hat{\tau}_{acc}} \tag{6.51}$$

$$\dot{\hat{\mathbf{b}}}_{gps} = -\frac{\hat{\mathbf{b}}_{gps}}{\hat{\tau}_{gps}} \tag{6.52}$$

$$\dot{\hat{b}}_r = -\frac{\hat{b}_r}{\hat{\tau}_r} \tag{6.53}$$

$$\dot{\hat{b}}_{rr} = -\frac{\hat{b}_{rr}}{\hat{\tau}_{rr}} \tag{6.54}$$

$$\dot{\hat{b}}_{los} = -\frac{\hat{b}_{los}}{\hat{\tau}_{los}} \tag{6.55}$$

6.3.6 Relative Position Covariance

For a target engagement, the final miss distance and crossing angle depend on the relative states of the vehicles. The relative position covariance is calculated from inertial position covariance using a linear transformation matrix A (See Section 3.10 of [22]). Equations 6.56 and 6.57 illustrate how to transform the navigation state vector into the relative position vector, and the state covariance matrix into the relative position covariance matrix.

$$\hat{\mathbf{r}}_{rel_{2\times1}} = A_{rel}\hat{\mathbf{x}}_{\hat{n}\times1} \tag{6.56}$$

$$\hat{P}_{rel_{2\times2}} = A_{rel}\hat{P}_{\hat{n}\times\hat{n}}A_{rel}^{T} \tag{6.57}$$

where A_{rel} is:

$$A_{rel} = \begin{bmatrix} -I_{2\times2} & 0_{2\times2} & I_{2\times2} & 0_{2\times9} \end{bmatrix}_{2\times15} \tag{6.58}$$

and $\hat{P}_{\hat{n}\times\hat{n}}$ is the state error covariance matrix described in Section 4.1.1

6.3.7 Along-track Coordinate Frame

The along-track/cross-track coordinate frame informs the final engagement results more effectively than the inertial frame. The x-axis of the along-track frame is in the direction of the interceptor velocity vector (See Fig. 6.5). This is referred to as the along-track direction. The y-axis of the along-track frame is perpendicular to the interceptor velocity vector, and is called the cross-track direction. The relative position vector (Eq. 6.60) and the relative position covariance matrix (Eq. 6.61) are rotated to the along-track/cross-track frame with a direction cosine matrix $\hat{T}_i^a$. Note that θ represents the counter-clockwise angle from the inertial x-axis to the interceptor velocity vector.

$$\hat{T}_i^a = \begin{bmatrix} \cos\hat{\theta} & \sin\hat{\theta} \\ -\sin\hat{\theta} & \cos\hat{\theta} \end{bmatrix} \tag{6.59}$$

$$\hat{\mathbf{r}}_{rel,a} = \hat{T}_i^a\hat{\mathbf{r}}_{rel} \tag{6.60}$$

$$\hat{P}_{rel,a} = \hat{T}_i^a \hat{P}_{rel} (\hat{T}_i^a)^T = \hat{T}_i^a A \hat{P} A^T (\hat{T}_i^a)^T \tag{6.61}$$

6.3.8 Measurement Partial Derivatives

The $\hat{H}$ matrix for the measurements is the partial derivative of the measurement equation $\hat{h}$ with respect to the navigation state vector $\hat{\mathbf{x}}$. For example, $\hat{H}_{gps}$ is the partial derivative of a 2×1 measurement vector with respect to a 15×1 state, therefore the result is a 2×15 matrix.

$$\hat{H}_{gps} = \begin{bmatrix} I_{2\times2} & 0_{2\times8} & I_{2\times2} & 0_{2\times3} \end{bmatrix}_{2\times15} \tag{6.62}$$

Hrange and Hrange-rate are computed in a similar manner

$$\hat{H}_{range} = \frac{1}{||(\mathbf{r}_t - \mathbf{r})||} \begin{bmatrix} -(\mathbf{r}_t - \mathbf{r})^T & 0_{1\times2} & (\mathbf{r}_t - \mathbf{r})^T & 0_{1\times6} & 1 & 0_{1\times2} \end{bmatrix}_{1\times15} \tag{6.63}$$

$$\hat{H}_{rr} = \begin{bmatrix} \frac{\partial \tilde{z}_{rr}}{\partial \mathbf{r}} & \frac{\partial \tilde{z}_{rr}}{\partial \mathbf{v}} & \frac{\partial \tilde{z}_{rr}}{\partial \mathbf{r_t}} & \frac{\partial \tilde{z}_{rr}}{\partial \mathbf{v_t}} & 0_{1\times5} & 1 & 0 \end{bmatrix}_{1\times15} \tag{6.64}$$

where

$$\frac{\partial \tilde{z}_{rr}}{\partial \mathbf{r}} = \frac{-(\mathbf{v}_t - \mathbf{v})^T}{||(\mathbf{r}_t - \mathbf{r})||} \left[I_{2\times2} - \mathbf{i}_{rel} (\mathbf{i}_{rel})^T \right] \tag{6.65}$$

$$\frac{\partial \tilde{z}_{rr}}{\partial \mathbf{v}} = -\mathbf{i}_{rel}^T \tag{6.66}$$

$$\frac{\partial \tilde{z}_{rr}}{\partial \mathbf{r}_t} = \frac{(\mathbf{v}_t - \mathbf{v})^T}{||(\mathbf{r}_t - \mathbf{r})||} \left[I_{2\times2} - \mathbf{i}_{rel} (\mathbf{i}_{rel})^T \right] \tag{6.67}$$

$$\frac{\partial \tilde{z}_{rr}}{\partial \mathbf{v}_t} = \mathbf{i}_{rel}^T \tag{6.68}$$

$$\hat{H}_{los} = \frac{1}{||(\mathbf{r}_t - \mathbf{r})||^2} \begin{bmatrix} \frac{\partial \tilde{z}_{los}}{\partial \mathbf{r}} & 0_{1\times2} & \frac{\partial \tilde{z}_{los}}{\partial \mathbf{r}_t} & 0_{1\times8} & 1 \end{bmatrix}_{1\times15} \tag{6.69}$$

where

$$\frac{\partial \tilde{z}_{los}}{\partial \mathbf{r}} = \begin{bmatrix} B(\mathbf{r}_t - \mathbf{r}) & -C(\mathbf{r}_t - \mathbf{r}) \end{bmatrix} \tag{6.70}$$

$$\frac{\partial \tilde{z}_{los}}{\partial \mathbf{r}_t} = \begin{bmatrix} -B(\mathbf{r}_t - \mathbf{r}) & C(\mathbf{r}_t - \mathbf{r}) \end{bmatrix} \tag{6.71}$$

$$B = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{6.72}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{6.73}$$

These equations and partial derivatives are substituted into equations 4.13-4.15 to update the state and covariance in the EKF.

6.3.9 EKF Covariance Propagation Partial Derivatives

The partial derivative of the navigation state dynamics equation $\hat{\mathbf{f}}$ with respect to the navigation state $\hat{\mathbf{x}}$ is

$$\hat{F}_{\hat{\mathbf{x}}} = \begin{bmatrix} \frac{\partial \dot{\hat{r}}}{\partial \hat{\mathbf{x}}}_{2\times 15} \\ \frac{\partial \dot{\hat{\mathbf{v}}}}{\partial \hat{\mathbf{x}}}_{2\times 15} \\ \frac{\partial \dot{\hat{\mathbf{r}}}_t}{\partial \hat{\mathbf{x}}}_{2\times 15} \\ \frac{\partial \dot{\hat{\mathbf{v}}}_t}{\partial \hat{\mathbf{x}}}_{2\times 15} \\ \frac{\partial \dot{\hat{\mathbf{b}}}}{\partial \hat{\mathbf{x}}}_{7\times 15} \end{bmatrix} = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} & 0_{2\times 11} \\ 0_{2\times 8} & -I_{2\times 2} & 0_{2\times 5} \\ 0_{2\times 6} & I_{2\times 2} & 0_{2\times 7} \\ & 0_{2\times 15} & \\ 0_{7\times 8} & I_{7\times 7} A_{\hat{\tau}} & \end{bmatrix}_{15\times 15} \tag{6.74}$$

where

$$A_{\hat{\tau}} = diag(-\frac{1}{\hat{\tau}_{acc1}}, -\frac{1}{\hat{\tau}_{acc2}}, -\frac{1}{\hat{\tau}_{gps1}}, -\frac{1}{\hat{\tau}_{gps2}}, -\frac{1}{\hat{\tau}_r}, -\frac{1}{\hat{\tau}_{rr}}, -\frac{1}{\hat{\tau}_{los}})$$

The partial derivative of the navigation state dynamics equation $\hat{\mathbf{f}}$ with respect to the inertial measurement state $\tilde{\mathbf{y}}$ is

$$\hat{F}_{\tilde{\mathbf{y}}} = \begin{bmatrix} \frac{\partial \dot{\hat{\mathbf{r}}}}{\partial \tilde{\mathbf{y}}}_{2\times 2} \\ \frac{\partial \dot{\hat{\mathbf{v}}}}{\partial \tilde{\mathbf{y}}}_{2\times 2} \\ \frac{\partial \dot{\hat{\mathbf{v}}}_t}{\partial \tilde{\mathbf{y}}}_{2\times 2} \\ \frac{\partial \dot{\hat{\mathbf{v}}}_t}{\partial \tilde{\mathbf{y}}}_{2\times 2} \\ \frac{\partial \dot{\hat{\mathbf{b}}}}{\partial \hat{\mathbf{x}}}_{7\times 2} \end{bmatrix} = \begin{bmatrix} 0_{2\times 2} \\ I_{2\times 2} \\ 0_{2\times 2} \\ 0_{2\times 2} \\ 0_{7\times 2} \end{bmatrix}_{15\times 2} \tag{6.75}$$

The partial derivative of the design model inertial sensor with respect to the navigation state vector is $\hat{C}_{\hat{\mathbf{x}}}$:

$$\hat{C}_{\hat{\mathbf{x}}} = \left[\frac{\partial \hat{\mathbf{c}}}{\partial \hat{\mathbf{x}}}\right]_{2\times 15} = \begin{bmatrix} 0_{2\times 8} & I_{2\times 2} & 0_{2\times 5} \end{bmatrix} \tag{6.76}$$

The navigation noise assignment matrix $\hat{B}$ is

$$\hat{B} = \begin{bmatrix} 0_{6\times 9} \\ I_{9\times 9} \end{bmatrix}_{15\times 9} \tag{6.77}$$

The equations for $\hat{F}_{\hat{\mathbf{x}}}$, $\hat{F}_{\tilde{\mathbf{y}}}$,$\hat{C}_{\hat{\mathbf{x}}}$, and $\hat{B}$ are substituted into equation 4.13 to propagate the EKF covariance:

$$\dot{\hat{P}} = [\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}\hat{C}_{\hat{\mathbf{x}}}]\hat{P} + \hat{P}[\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}\hat{C}_{\hat{\mathbf{x}}}]^T + \hat{F}_{\tilde{\mathbf{y}}}\hat{S}_\eta\hat{F}_{\tilde{\mathbf{y}}}^T + \hat{B}\hat{S}_w\hat{B}^T \tag{6.78}$$

In the Monte Carlo simulation, the EKF covariance is used in the Kalman gain equation for the navigation state measurement update (see Eq. 4.14).

6.4 LinCov Simulation

Augmented state linear covariance analysis utilizes an augmented state vector to calculate statistics of truth dispersions and true estimation error. The augmented state vector $\mathbf{X}$ is written as:

$$\mathbf{X} = \begin{bmatrix} \delta\mathbf{x} \\ \delta\hat{\mathbf{x}} \end{bmatrix} \tag{6.79}$$

The linearized dynamics from Chapter 4, equations 4.74 and 4.74, are restated for convenience:

$$\delta\dot{\mathbf{x}} = F_{\mathbf{x}}\delta\mathbf{x} + F_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + B\mathbf{w} \tag{6.80}$$

$$\delta\dot{\hat{\mathbf{x}}} = \left[\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\right]\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}}C_{\mathbf{x}}\delta\mathbf{x} + \hat{F}_{\tilde{\mathbf{y}}}\boldsymbol{\eta} \tag{6.81}$$

In order for the simulation to calculate the statistics of the augmented state vector (see equations 4.79-4.86), the partial derivatives from equations 6.80 and 6.81 are calculated at each time step along the nominal trajectory. The equations for $\hat{F}_{\hat{\mathbf{x}}}$, $\hat{F}_{\tilde{\mathbf{y}}}$,$\hat{C}_{\hat{\mathbf{x}}}$, and B are the same as the Monte Carlo simulation equations 6.74-6.76 and 6.18. The remaining partial derivatives are shown below:

$$F_{\mathbf{x}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{r}}}{\partial \mathbf{x}}_{2\times 19} \\ \frac{\partial \dot{\mathbf{v}}}{\partial \mathbf{x}}_{2\times 19} \\ \frac{\partial \dot{\mathbf{r}}_t}{\partial \mathbf{x}}_{2\times 19} \\ \frac{\partial \dot{\mathbf{v}}_t}{\partial \mathbf{x}}_{2\times 19} \\ \frac{\partial \dot{\mathbf{a}}_t}{\partial \mathbf{x}}_{2\times 19} \\ \frac{\partial \dot{\mathbf{b}}}{\partial \mathbf{x}}_{7\times 19} \end{bmatrix} = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} & 0_{2\times 13} \\ & 0_{2\times 17} & \\ 0_{2\times 6} & I_{2\times 2} & 0_{2\times 9} \\ 0_{2\times 8} & I_{2\times 2} & 0_{2\times 7} \\ 0_{2\times 8} & -\alpha I_{2\times 2} & 0_{2\times 7} \\ 0_{7\times 10} & I_{7\times 7} A_\tau & \end{bmatrix}_{17\times 17} \tag{6.82}$$

where

$$A_\tau = diag(-\frac{1}{\tau_{acc1}}, -\frac{1}{\tau_{acc2}}, -\frac{1}{\tau_{gps1}}, -\frac{1}{\tau_{gps2}}, -\frac{1}{\tau_r}, -\frac{1}{\tau_{rr}}, -\frac{1}{\tau_{los}})$$

$$F_{\hat{\mathbf{u}}} = \begin{bmatrix} 0_{2\times 2} \\ I_{2\times 2} \\ 0_{13\times 2} \end{bmatrix}_{17\times 2} \tag{6.83}$$

$$C_{\mathbf{x}} = \left[\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right]_{2\times 17} = \begin{bmatrix} 0_{2\times 10} & I_{2\times 2} & 0_{2\times 5} \end{bmatrix} \tag{6.84}$$

$$C_{\hat{\mathbf{u}}} = \left[\frac{\partial \mathbf{c}}{\partial \mathbf{u}}\right]_{2\times 2} = [I_{2\times 2}] \tag{6.85}$$

$$M_{\mathbf{x}} = \begin{bmatrix} I_{8\times 8} & 0_{8\times 2} & 0_{8\times 7} \\ 0_{7\times 8} & 0_{7\times 2} & I_{7\times 7} \end{bmatrix}_{15\times 17} \tag{6.86}$$

The partial derivative of the guidance law $\hat{\mathbf{u}}$ with respect to the navigation state vector $\hat{\mathbf{x}}$ is $\hat{G}_{\hat{x}}$. The derivation is found in Appendix B.

$$\boxed{\hat{G}_{\hat{\mathbf{x}}} = \begin{bmatrix} -\sin(\hat{\lambda}) \\ \cos(\hat{\lambda}) \end{bmatrix} \frac{\partial \left[\hat{k_p}\hat{V}_c\dot{\hat{\lambda}}\right]}{\partial \hat{\mathbf{x}}} + \hat{k_p}\hat{V}_c\dot{\hat{\lambda}}\frac{\partial}{\partial \hat{\mathbf{x}}}\begin{bmatrix} -\sin(\hat{\lambda}) \\ \cos(\hat{\lambda}) \end{bmatrix}} \tag{6.87}$$

where

$$\boxed{\frac{\partial \left[\hat{k_p}\hat{V}_c\dot{\hat{\lambda}}\right]}{\partial \hat{\mathbf{x}}} = \hat{k_p}\hat{V}_c\frac{\partial \dot{\hat{\lambda}}}{\partial \hat{\mathbf{x}}} + \hat{k_p}\dot{\hat{\lambda}}\frac{\partial \hat{V}_c}{\partial \hat{\mathbf{x}}}} \tag{6.88}$$

and

$$\boxed{\frac{\partial}{\partial \hat{\mathbf{x}}}\begin{bmatrix} -\sin(\hat{\lambda}) \\ \cos(\hat{\lambda}) \end{bmatrix} = \begin{bmatrix} \frac{\partial \hat{g}_1}{\partial \hat{\mathbf{x}}} \\ \frac{\partial \hat{h}_1}{\partial \hat{\mathbf{x}}} \end{bmatrix}_{2\times 15}} \tag{6.89}$$

The following substitutions are utilized for equations 6.87-6.89:

$$\frac{\partial \dot{\hat{\lambda}}}{\partial \hat{\mathbf{x}}} = \frac{1}{||(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})||^2}\begin{bmatrix} -(\hat{\mathbf{v}}_t - \hat{\mathbf{v}})^T A^T & -(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T A & (\hat{\mathbf{v}}_t - \hat{\mathbf{v}})^T A^T & (\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T A & 0_{2\times 7} \end{bmatrix}_{1\times 15}$$

$$- \frac{(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T A(\hat{\mathbf{v}}_t - \hat{\mathbf{v}})}{||(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})||^4}\begin{bmatrix} -2(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T & 0_{1\times 2} & 2(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T & 0_{1\times 9} \end{bmatrix}_{1\times 15} \tag{6.90}$$

$$\frac{\partial \hat{V}_c}{\partial \hat{\mathbf{x}}} = \frac{1}{||(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})||}\begin{bmatrix} (\hat{\mathbf{v}}_t - \hat{\mathbf{v}})^T & (\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T & -(\hat{\mathbf{v}}_t - \hat{\mathbf{v}})^T & -(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T & 0_{1\times 7} \end{bmatrix}_{1\times 15}$$

$$+ \frac{(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T(\hat{\mathbf{v}}_t - \hat{\mathbf{v}})}{||(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})||^3}\begin{bmatrix} -(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T & 0_{1\times 2} & (\hat{\mathbf{r}}_t - \hat{\mathbf{r}})^T & 0_{1\times 9} \end{bmatrix}_{1\times 15} \tag{6.91}$$

$$\frac{\partial \hat{g}_1}{\partial \hat{\mathbf{x}}} = -\cos(\hat{\lambda})\frac{\partial \hat{\lambda}}{\partial \hat{f}_1}\frac{[C(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]\begin{bmatrix} -B & 0_{1\times 2} & B & 0_{1\times 9} \end{bmatrix}}{[C(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]^2}$$

$$+ \cos(\hat{\lambda})\frac{\partial \hat{\lambda}}{\partial \hat{f}_1}\frac{[B(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]\begin{bmatrix} -C & 0_{1\times 2} & C & 0_{1\times 9} \end{bmatrix}}{[C(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]^2} \tag{6.92}$$

$$\frac{\partial \hat{h}_1}{\partial \hat{\mathbf{x}}} = -\sin(\hat{\lambda})\frac{\partial \hat{\lambda}}{\partial \hat{f}_1}\frac{[C(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]\begin{bmatrix} -B & 0_{1\times 2} & B & 0_{1\times 9}\end{bmatrix}}{[C(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]^2}$$
$$+\sin(\hat{\lambda})\frac{\partial \hat{\lambda}}{\partial \hat{f}_1}\frac{[B(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]\begin{bmatrix} -C & 0_{1\times 2} & C & 0_{1\times 9}\end{bmatrix}}{[C(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]^2} \tag{6.93}$$

$$\frac{\partial \hat{\lambda}}{\partial \hat{f}_1} = \frac{[C(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]^2}{[C(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]^2 + [B(\hat{\mathbf{r}}_t - \hat{\mathbf{r}})]^2} \tag{6.94}$$

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{6.95}$$

$$B = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{6.96}$$

With the partial derivatives calculated, it is possible to run equations 4.79-4.86.

Sections 6.3-6.4 serve as a template for implementing LinCov in a missile engagement problem. These sections develop the equations for a missile engagement LinCov simulation, and the Monte Carlo simulation that is used to validate it. They also define the state vector, guidance law, sensor models, and partial derivatives needed for the simulation. The next step is to validate the LinCov simulation with the Monte Carlo simulation.

6.5 LinCov Simulation Validation

Sections 6.5 and 6.6 contain a LinCov analysis for a missile engagement. When developing a LinCov simulation, it is important to first validate the simulation by comparing the results to a trusted nonlinear simulation method, usually the Monte Carlo method. Once the LinCov is validated, it can replicate a full system analysis in a fraction of the time. The LinCov simulation can be utilized to quantify system behavior in a variety of ways: sensor trade studies, sensitivity analysis, guidance law analysis, etc. The computation speed of LinCov enables full-domain study of system inputs in a way that is not possible with sample based methods due to time constraints. This section proceeds as follows:

sections 6.5.1-6.5.2 detail the steps needed to validate the LinCov covariance propagation and update equations against a Monte Carlo simulation. The propagation and update results for the LinCov simulation are then shown. Next, Section 6.5.3 defines the complex-step derivative approximation method for validating partial derivatives, and employs the method to numerically validate the LinCov partial derivatives. In Section 6.5.5, the full LinCov simulation is run and compared to the Monte Carlo simulation. Finally, Section 6.6 presents an error budget study that investigates the sensitivity of the miss distance to sensor parameters and initial conditions.

6.5.1 State Vector and Covariance Propagation Validation

In this section, the implementation of the differential equations for the system are checked by plotting the nominal reference trajectories based on truth state dynamics and filter state dynamics with all sources of error turned off [39]. The implementation of the covariance propagation equations are validated by comparing them to Monte Carlo statistics. For the state vector propagation comparison, the guidance law and actuator are active, but the discrete measurement noise, accelerometer noise, and disturbance acceleration noise sources are all disabled. This setup is also called the nominal reference trajectory. Figure 6.6 shows the position nominal reference trajectory for the missile and the target. It shows that

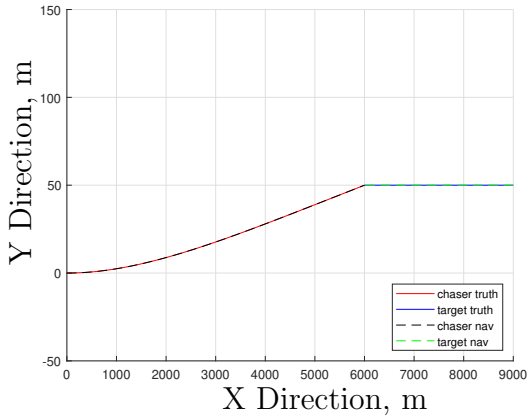

Fig. 6.6: LinCov Validation Nominal Reference Trajectory

the difference between the truth state and the navigation state is zero, as expected. This indicates that the Monte Carlo equations for propagating the truth state and the navigation state with all error sources turned off are working correctly.

Next, the target is initialized farther away, allowing the simulation to run longer. This makes it easier to analyze the covariance propagation equations. The navigation state covariance matrix is propagated using equation 4.13 with the specified $\hat{\mathbf{x}}_0$ and $\hat{P}_0$. In this validation setup, there are no measurements, and the initial conditions of the state for each run vary according to the initial covariance values. Figure 6.7 illustrates a few example plots of the covariance propagation and ensemble variance of the estimation error for each state. The Monte Carlo simulation results should show that the $\pm 3\sigma$ range contains the

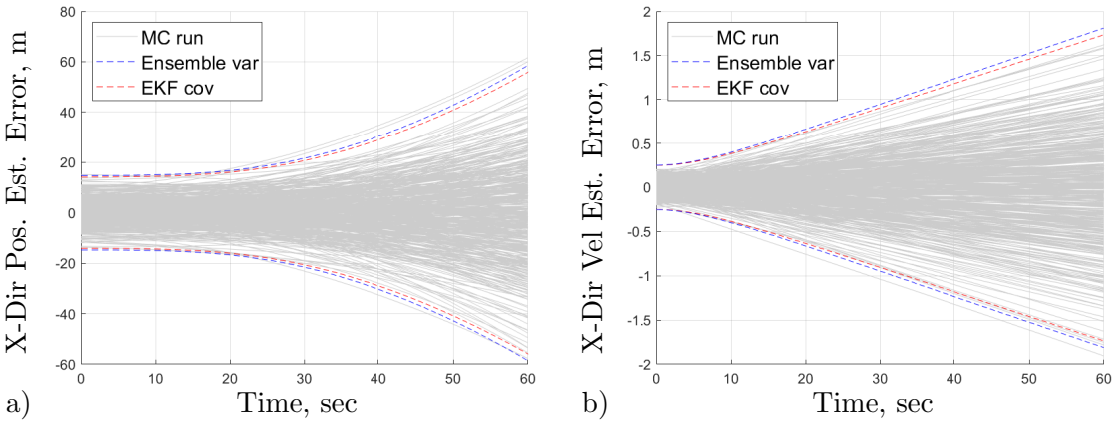

Fig. 6.7: Covariance Propagation Validation Plots

data 99.7% of the time. The covariance values are validated by comparing the ensemble variance statistics of all the runs with the linearized covariance propagation results. The plots in Figure 6.7 show that the implementation of the covariance propagation equation is correct, as the ensemble statistics match up well with the filter covariance.

6.5.2 Measurement Update Validation

The plots in this section show the results after implementing the measurement update equations. The range, range rate and line-of-sight measurements are taken every 0.1 seconds and the GPS measurement is taken every 1.0 seconds. The same checks used in the propagation step can also be used here. In Figure 6.8, the covariance update appears to be working correctly, as the ensemble statistics match up well with the filter covariance. The saw tooth appearance of the plots shows that the system is updating at the correct rate.

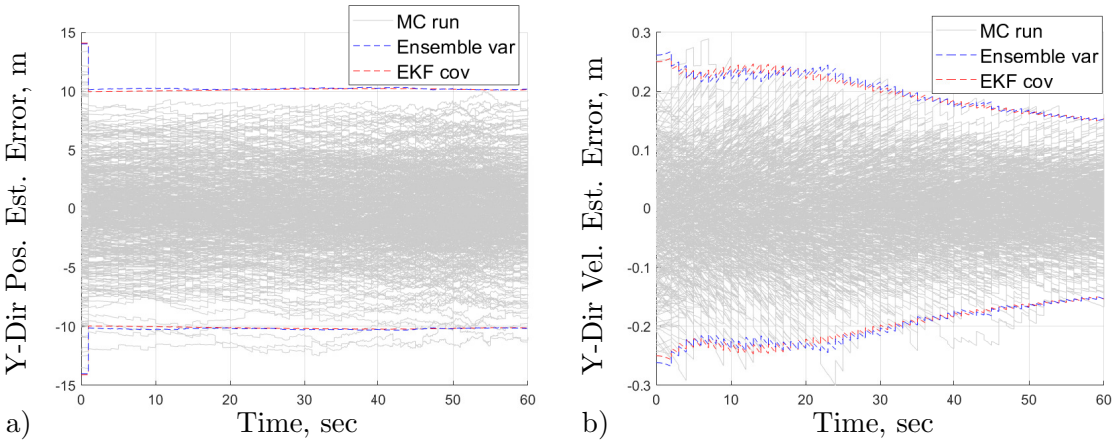

Fig. 6.8: Covariance Update Validation Plots

The residual $[\tilde{z}_i - \hat{H}(t_i)\hat{\mathbf{x}}(t_i^-)]$ is useful for validating the implementation of the discrete measurement equations. The first term $\tilde{z}_i$ is the measurement from the sensor and the $\hat{H}(t_i)\hat{\mathbf{x}}(t_i^-)$ term is the predicted measurement from the navigation state based on the first term Taylor Series approximation of the measurement from equation 4.36. The residual has a mean of zero and a covariance of $HPH^T + R$ (see section 5.4 of [22]). Figure 6.9 shows that the residuals behave as expected and stay within the 3 sigma bounds. This is a good check on the measurement implementation in the filter.

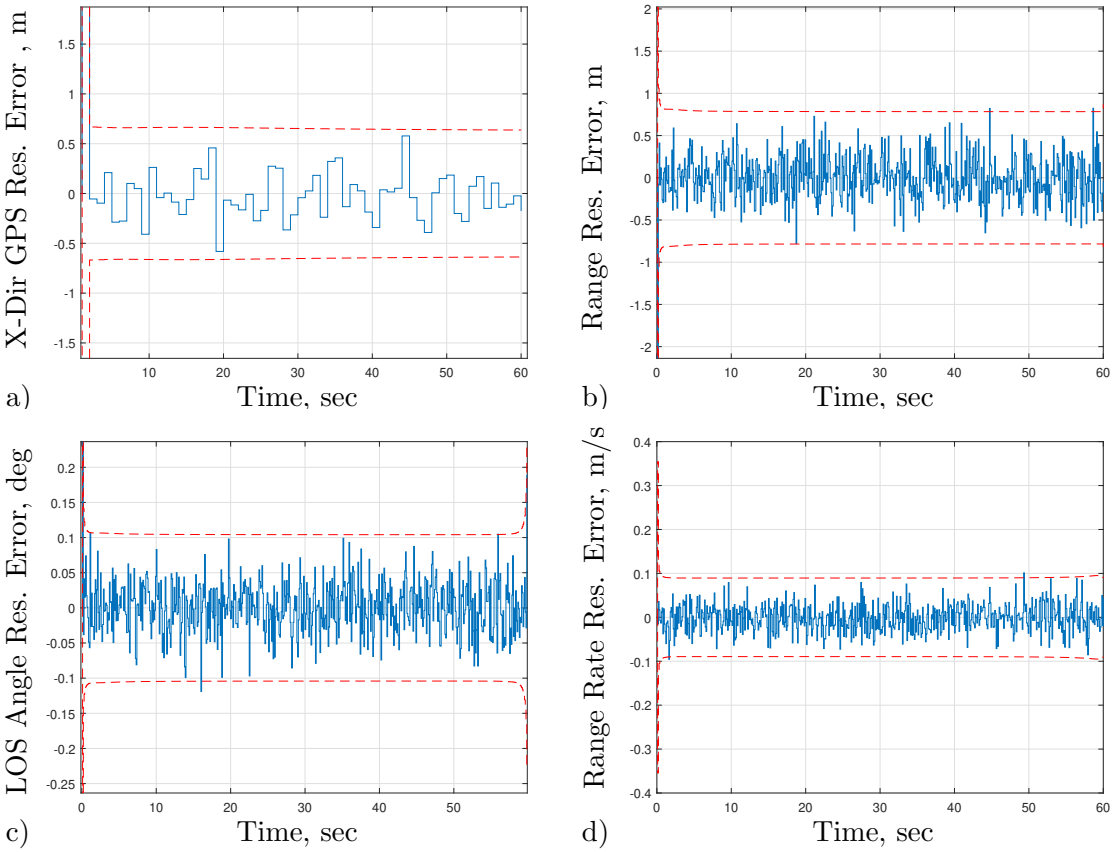

Fig. 6.9: Measurement Residual Validation Plots

6.5.3 Partial Derivative Validation

The LinCov propagation and update equations rely on the partial derivatives of the dynamics, control, and measurements. It is important to calculate them analytically where possible and validate them numerically. The derivatives are shown in section 6.3.8. The validation method that is used for the partial derivatives is the complex-step derivative approximation (CSDA) approach described in [91]. In this method, complex variables are used to calculate the first derivative using the following equation:

$$\frac{\partial f}{\partial x} \approx \frac{Im[f(x+ih)]}{h} \tag{6.97}$$

where f is the function, x is the independent variable, and h is the finite difference interval. The CSDA method is more robust than other numerical methods because it does not use

the subtration operation. This prevents subtractive cancellation errors and gives nearly exact results when the step size is below 10^{-16}. This method is also useful for equations where the analytic derivatives are difficult or impossible to find. In this case, an accurate derivative can be found numerically and used for the covariance equations. The following section shows an example of CSDA method.

Complex-Step Derivative Approximation Example: Range Rate

The measurement equation for the range rate measurement is as follows:

$$z_{rr} = \frac{\partial}{\partial t}||(\mathbf{r}_t - \mathbf{r})|| = \frac{(\mathbf{r}_t - \mathbf{r})}{||(\mathbf{r}_t - \mathbf{r})||} \cdot (\mathbf{v}_t - \mathbf{v}) \tag{6.98}$$

It is desirable to validate the partial derivative of the measurement with respect to the state vector x. The measurement is a scalar therefore the derivative is a 1×15 matrix:

$$H_{rr} = \begin{bmatrix} \frac{\partial z_{rr}}{\partial \mathbf{r}} & \frac{\partial z_{rr}}{\partial \mathbf{v}} & \frac{\partial z_{rr}}{\partial \mathbf{r}_t} & \frac{\partial z_{rr}}{\partial \mathbf{v}_t} & 0_{1\times 5} & 1 & 0 \end{bmatrix}_{1x15} \tag{6.99}$$

The individual partials in equation 6.99 are defined in section 6.3.8.

The steps to complete the approximation are as follows: set the measurement to be a complex variable, define functions to work with complex arguments, and finally, add a small complex increment step, h, to the desired state and substitute it into equation 6.97.

The coding for this example is done in Matlab, and accordingly, a few functions in equation 6.98 need to be defined or redefined. The default Matlab norm function calculates the Euclidean norm using the complex conjugate of the input. This creates a scalar output, which does not work for this problem. The 'cnorm' function from [36] is used instead. This function keeps the output in complex form. The same issue is found in the Matlab 'dot' function. The 'complex dot product' can be found by manually multiplying the individual elements together and adding them instead. With these two sub-function routines in place, the measurement equation is ready to handle complex numbers.

Substituting in different step sizes into the CSDA equation 6.97 and subtracting it from the analytic partial derivative yields the results from Figure 6.10. The results are consistent

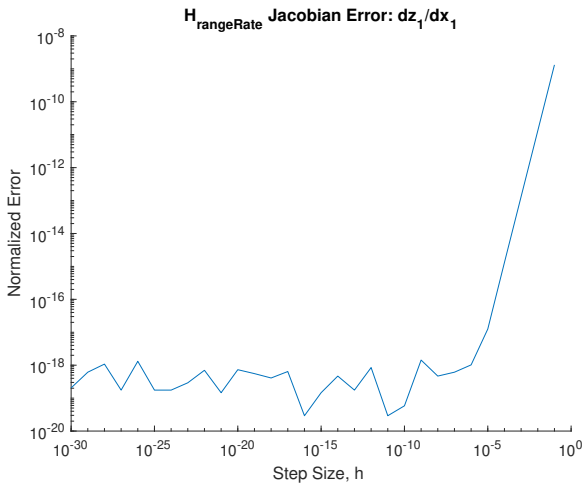

Fig. 6.10: CSDA Validation for Range Rate Measurement

with the results from [91]. There is a linear decrease in the error as the step size starts decreasing from 0.1 to 10^{-5}. Once the step size is below 10^{-5}, the normalized error levels off at about 10^{-18}. These results validate the analytic calculation of the first element of the Jacobian H_{rr}, which is the partial derivative of the range rate measurement with respect to the x-direction position of the interceptor. Each element of the Jacobian H_{rr} is validated using this same process.

For the LinCov simulation, the CSDA method is applied to validate the analytical partial derivatives in sections 6.3.8, 6.3.9, and 6.4. The analytical derivatives are satisfactory for use in the LinCov simulation. See Appendix A for a code example of the CSDA process.

Table 6.1: Simulation Initial Conditions

State	Value	Units	3σ Uncertainty	Description
$\mathbf{r}(t_0)$	[0;0]	m	14.1	initial missile position vector
$\mathbf{v}(t_0)$	[1000;0]	m/s	0.25	initial missile velocity vector
$\mathbf{r}_t(t_0)$	[9000;50]	m	14.8	initial target position vector
$\mathbf{v}_t(t_0)$	[-500;0]	m/s	0.65	initial target velocity vector
$\mathbf{a}_t(t_0)$	[0;0]	g	5	initial target acceleration vector
$\mathbf{b}_{acc}(t_0)$	[0;0]	mg	1	initial accelerometer bias vector
$\mathbf{b}_{gps}(t_0)$	[0;0]	m	14.1	initial GPS bias vector
$b_r(t_0)$	0	m	3	initial range bias
$b_{rr}(t_0)$	0	m/s	0.4	initial range rate bias
$b_{los}(t_0)$	0	mrad	6	initial line-of-sight bias

6.5.4 Simulation Parameters

Figure 6.6 illustrates a single engagement run with the controller enabled, and the measurement noise and all sources of error and uncertainty disabled. This trajectory is called the nominal reference trajectory. For each step in the LinCov simulation, the measurements, control law, and dynamics are linearized about the corresponding point in the reference trajectory. Table 6.1 contains the state initial conditions and Table 6.2 contains the noise values of the simulation and other simulation parameters. The initial position and velocity conditions are modeled after examples in [57] and [8] with typical GPS sensor uncertainties. The target and missile maximum acceleration values can be found in [92]. The target vehicle does not have a GPS sensor, so its initial position and velocity uncertainties are found by combining GPS measurement error with range or range rate measurement error. The initial target acceleration is nominally zero with the uncertainty falling within the range of the maximum acceleration. The sensor parameters are from the following references: GPS [63], range and range rate [56], LOS angle measurement [93], accelerometer [94, 95]. Relative measurement sensor biases are not often considered in missile state estimation publications [96], therefore these biases are modeled as follows: the time-constants are long compared to the engagement time and the missile will need to accurately estimate the biases in order to have a successful engagement. These biases will have a nominal value of zero. The target Singer motion parameters are identical to the maneuvering target in [97].

Table 6.2: Simulation Parameters

Parameter	Value	Units	Description
S_η	0.03	(m/s)/sqrt(hr)	3σ accelerometer random walk
S_w	0.3	(m/s)/sqrt(hr)	3σ missile dynamics random walk
S_{w_t}	10	(m/s)/sqrt(sec)	3σ target design model random walk
ν_{range}	0.75	m	3σ range measurement noise std dev
ν_{rr}	0.1	m/s	3σ range rate measurement noise std dev
ν_{los}	1.5	mrad	3σ line-of-sight measurement noise std dev
$\sigma_{acc,ss}$	1	mg	3σ steady state accelerometer bias
$\sigma_{gps,ss}$	14.1	m	3σ steady state GPS bias
$\sigma_{r,ss}$	3	m	3σ steady state range bias
$\sigma_{rr,ss}$	0.4	m/s	3σ steady state range rate bias
$\sigma_{los,ss}$	6	mrad	3σ steady state line-of-sight bias
τ_{acc}	1000	s	accelerometer bias time-constant
τ_{gps}	1100	s	GPS bias time-constant
τ_r	500	s	range bias time-constant
τ_{rr}	500	s	range rate bias time-constant
τ_{los}	500	s	line-of-sight bias time-constant
α	5	s	target acceleration time-constant
a_{max}	10	g	target max acceleration
p_{max}	0.05		target probability of max acceleration
p_0	0.1		target probability of no acceleration

6.5.5 Simulation Validation Results

The models from this chapter have been implemented into a Monte Carlo simulation and a LinCov simulation. The Monte Carlo simulation is run with 1000 runs and provides truth data to validate the linear covariance results. If the system is implemented properly, the LinCov generated statistics are equal to the ensemble statistics of the Monte Carlo runs. The hair plots from the Monte Carlo should create a normal distribution inside the 3σ covariance envelope, and the LinCov results should lie within the confidence interval of the Monte Carlo results. The plots in this section contain the results of the Monte Carlo and LinCov simulation comparison. The simulation is run with the control law, measurements, and all sources of error and uncertainty enabled.

The estimation error plots in Figure 6.11 contain the $\pm 3\sigma$ Linear Covariance estimation error P_{true}, the individual Monte Carlo samples (hair plots), the $\pm 3\sigma$ EKF covariance, the $\pm 3\sigma$ Monte Carlo estimation error, and the 95% confidence interval for the 3σ Monte Carlo estimation error. In general, the Lincov statistics fall within the confidence interval of the Monte Carlo statistics and the hair plots stay within the 3σ bounds. This suggests that the LinCov simulation can accurately calculate the system estimation error. In summary, the LinCov simulation estimation error is validated for this system.

Figure 6.12 shows the relative position estimation error plots. These plots give further insight to the engagement, as the end goal of the engagement is to drive relative position to zero. The 3σ along track relative position error stays somewhat constant throughout the engagement, while the cross-track error is driven down as the engagement progresses. This is consistent with the geometry of the missile engagement problem. An error budget study may be performed to quantify the contribution of each error source to the overall relative position error [39]. Figure 6.13 illustrates the state vector truth dispersions. The truth dispersion covariance indicates how far the trajectories stray from the nominal path. The Figure 6.13 plots contain the individual Monte Carlo runs, the the $\pm 3\sigma$ Monte Carlo covariance with the 95% confidence interval, and the $\pm 3\sigma$ linear covariance. For all the truth

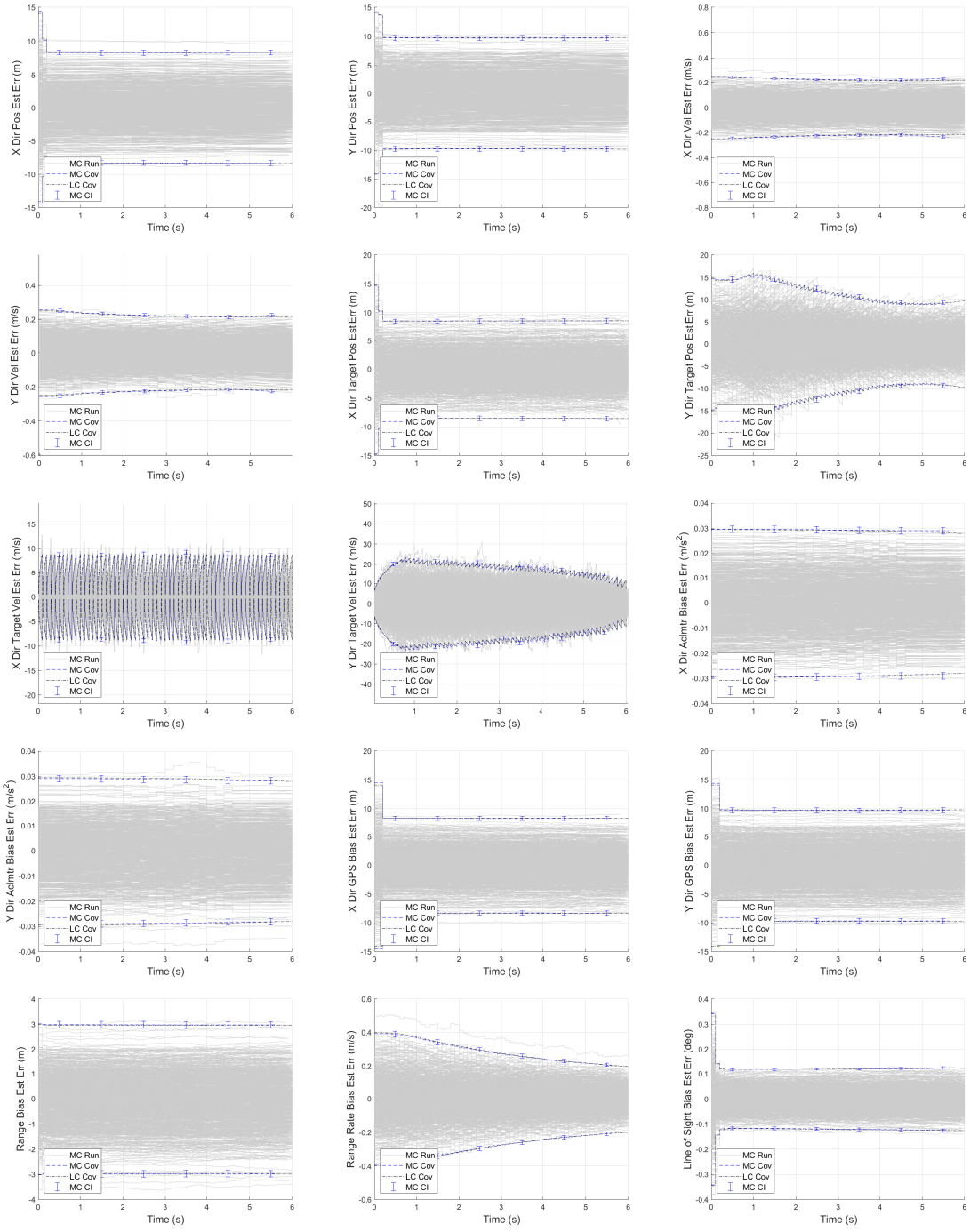

Fig. 6.11: LinCov Estimation Error Validation Plots

states, the LinCov statistics fall within the confidence interval of the Monte Carlo statistics and the hair plots stay within the 3σ bounds. This validates the LinCov truth dispersions.

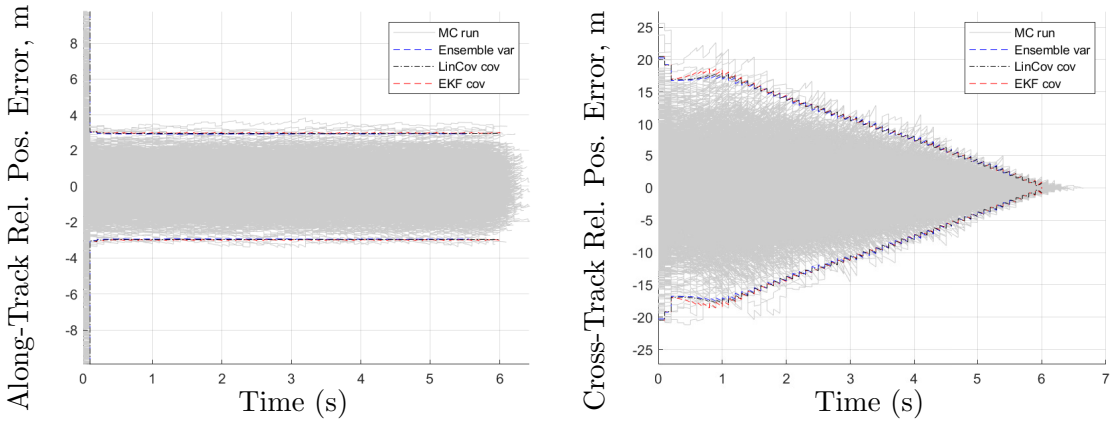

Fig. 6.12: Linear Covariance Simulation Relative Position Estimation Error

The relative position truth dispersions in the along track coordinate frame are presented in Figure 6.14. The relative position dispersions are important to consider because a missile engagement is a terminal constraint problem with the desired final relative position of zero. The covariance of the relative cross-track position at the final time is a direct measure of the engagement success. The cross-track dispersions go toward zero at the end of the flight as the missile maneuvers to ensure an intercept. Figure 6.14b shows that the LinCov accurately computes the statistics of the relative cross-track position at the final time, which is a good approximation of miss distance for crossing angles below forty-five degrees [41].

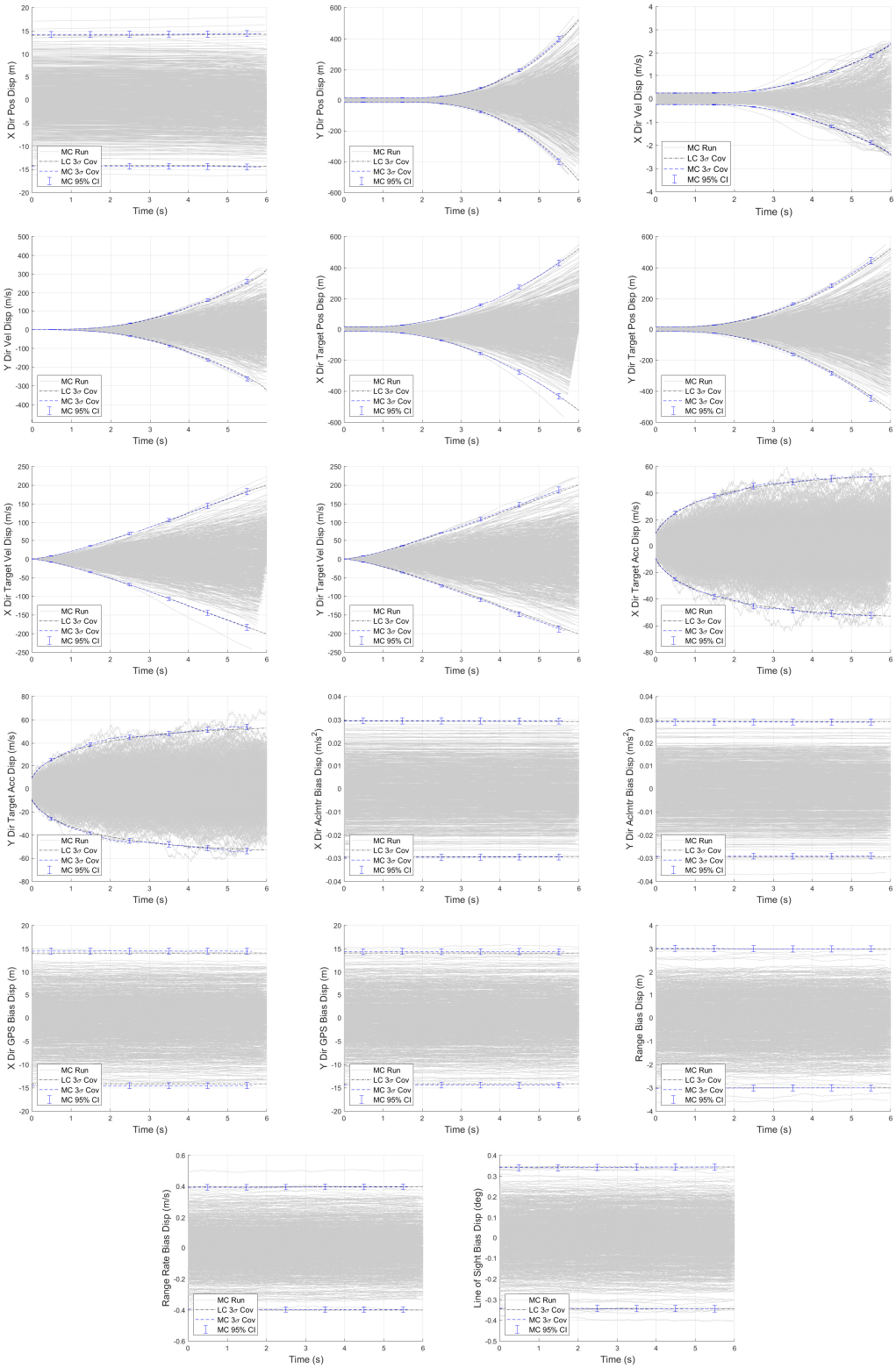

Fig. 6.13: LinCov Truth Dispersion Validation Plots

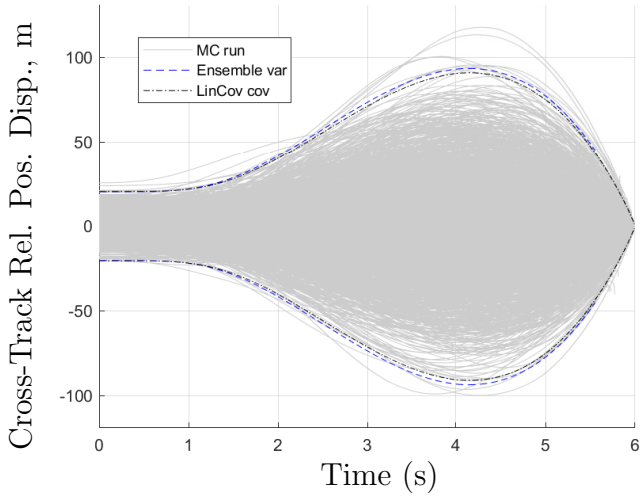

Fig. 6.14: Linear Covariance Simulation Relative Position Truth Dispersions

6.6 Miss Distance Error Budget Study

Linear covariance analysis can quickly run through trade studies of simulation parameters. This section contains an example miss distance error budget analysis. The error budget study quantifies how much each error source contributes to the relative position dispersions throughout the flight and at the final time. The simulation parameter values are listed in 6.5.4.

An error budget study is performed by running the LinCov simulation multiple times with only one error source active at a time [8]. At the end, the simulation is run with all error sources active. A combined plot illustrates how each error source contributes to the total error throughout the flight. The root sum squared of the errors for each run adds up to the total error for any time step. Relative cross-track position is the important metric in this simulation, therefore the following plots include the relative position covariance in the along-track frame.

Figure 6.15 contains an error budget for relative position dispersions. The main error

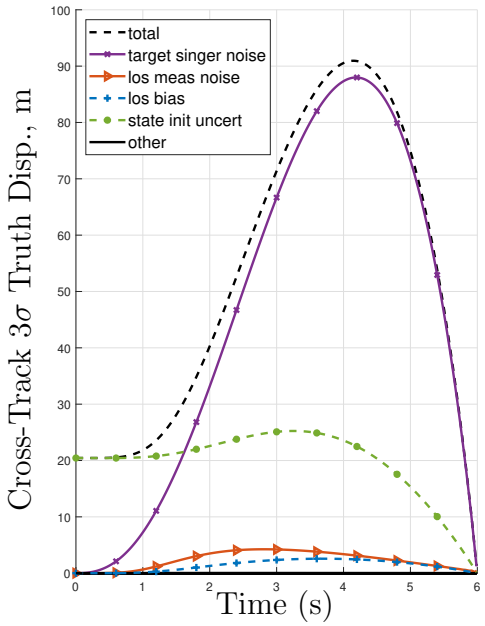

Fig. 6.15: Relative Position Truth Dispersion Error Budget

sources are shown in color, and include the target evasive Singer accelerations, the line-of-sight measurement noise, the line-of-sight sensor bias, and the initial state uncertainty. The solid black line near zero represents the sum of all the minor error sources. The minor error sources include the accelerometer sensor white noise , the range measurement bias and noise, the range rate measurement bias and noise, and the GPS measurement bias and noise. The contributions of the minor error sources to the relative position truth dispersions are so small that they barely register on the graph. On the other hand, the engagement dispersions are heavily influenced by the initial state uncertainty, the target evasive acceleration, and the line-of-sight sensor. It takes about 1.5 seconds for the target acceleration to become the largest source of dispersion, and this is not overcome until the final second of the flight. Figure 6.16 illustrates the dispersion covariance at the final time step. It shows how much

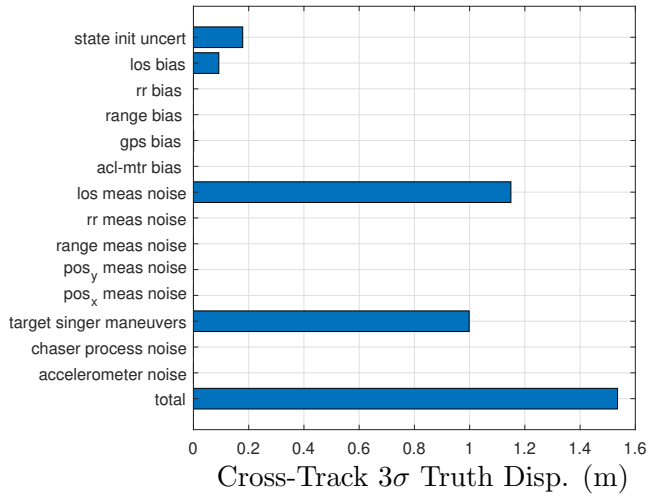

Fig. 6.16: Final Time Relative Position Truth Dispersion Error Budget

of the final miss distance can be attributed to each source of error, and which ones do not contribute much at all. The engagement miss distance is most sensitive to the line-of-sight measurement noise, as this contributes about 1 meter of the total 1.5 meter final miss distance. Other major contributors to the miss distance are target accelerations, initial state uncertainty, and the line-of-sight sensor bias.

Based on the results of this error budget study, a few modifications will be made to the engagement model going forward. Since the inertial GPS position measurement noise and bias do not contribute to the total relative position truth dispersions, the GPS measurement will be removed from the sensor suite and the GPS bias will no longer be estimated. This is not to say that a GPS sensor is not important for launch phase or mid-course phase of missile flight, just that it is not a major factor for the terminal homing phase of air-to-air targeting. (A missile with an inertially fixed target would certainly benefit from GPS measurements during terminal homing.)

Similarly, the measurement noise/biases from the range measurement and the range rate measurement are not significant contributors to the final miss distance. This is a result of the distinct geometry and dynamics of the terminal homing missile engagement problem. Many missiles steer with fins or divert thrusters, which initiates a lateral acceleration [16, 98]. During terminal homing, missiles do not undergo significant axial accelerations, and therefore it is common to assume constant closing velocity between missile and target (see page 3-3 of [73]). In fact, for many tactical IR missile applications of proportional navigation guidance, the closing velocity is not measured, but rather guessed (see chapter 2 of [8]). Consequently, the range and range rate sensors will be removed from the missile engagement model in subsequent chapters, and the along track dynamics of the engagement will be assumed to be deterministic.

This chapter defined a missile engagement LinCov simulation, and outlined various methods for validating elements of a LinCov simulation. The LinCov simulation was validated using the defined methods, and found to be an accurate approximation for the missile engagement system. An error budget study was performed, allowing insight into the significant error contributions to the engagement miss distance. In this chapter, the interceptor missile is assumed to have a significant acceleration advantage over the target, an assumption that will be removed in the upcoming chapters.

CHAPTER 7
STATISTICAL LINCOV AND TUNABLE LINCOV

The LinCov analysis presented in Chapter 6 demonstrated that augmented state linear covariance can be successfully applied to missile engagement problems for statistical system analysis. This implementation assumed that the missile had much larger (greater than 3x) maximum acceleration capability than the target. During many engagements, the missile does not have such a large acceleration advantage, therefore it is important to investigate the effect of sustained actuator saturation on system behavior. This chapter presents two methods for dealing with saturations in a missile engagement simulation. The first method is LinCov with statistical linearization and is henceforth referred to as Statistical LinCov. This method utilizes the augmented state linear covariance framework, while adding in statistical linearization for the nonlinear elements. This is a new feature for augmented state LinCov with many potential GN&C applications. The second method is called Tunable LinCov, and it involves switching modes to a saturated system once the 3σ acceleration dispersions have grown to a specified magnitude. Both simulations are developed and validated in this chapter.

Section 7.1 provides the simulation assumptions for this chapter, and Section 7.2 presents the nonlinear models used for the Monte Carlo analysis. Then, in Section 7.3, the linearized models are defined for the Statistical LinCov analysis. Following this, Section 7.4 provides the validation results of the Statistical LinCov. Section 7.5 defines the Tunable LinCov simulation, which is validated in Section 7.6.

7.1 Simulation Assumptions

The simulation assumptions in this chapter are similar to the LinCov assumptions in Section 6.2, with a few key differences. The problem is still assumed to be a 2D point mass problem that is trimmed to horizontal flight by the time it reaches terminal homing. In addition to the assumptions from LinCov simulation, this chapter assumes that the along track dynamics of the engagement are deterministic. This is because during terminal homing,

missiles do not undergo significant axial accelerations, and therefore it is common to assume constant closing velocity between missile and target [16, 73]. This assumption is frequently used for missile engagement analysis [8, 57], and aligns with the results from the study in Section 6.6. One effect of this assumption is that the guidance law is only applied in the cross-track plane. Finally, the pursuing missile acceleration advantage assumption is relaxed in this chapter, as the missile parameter a_{max} is now a varying system input.

7.2 Nonlinear Models for Monte Carlo Analysis

Statistical LinCov constitutes an augmented state linear covariance algorithm with statistical linearization implemented for the nonlinear elements. This section presents the Monte Carlo validation model for Statistical LinCov. It defines the truth and navigation states, truth dynamics, actuator and sensor design models, control and guidance laws, and navigation dynamics models.

7.2.1 Truth States

The truth state vector focuses on the lateral dynamics of the intercept. Figure 7.1 shows the engagement geometry of the state vector. The state vector contains a relative

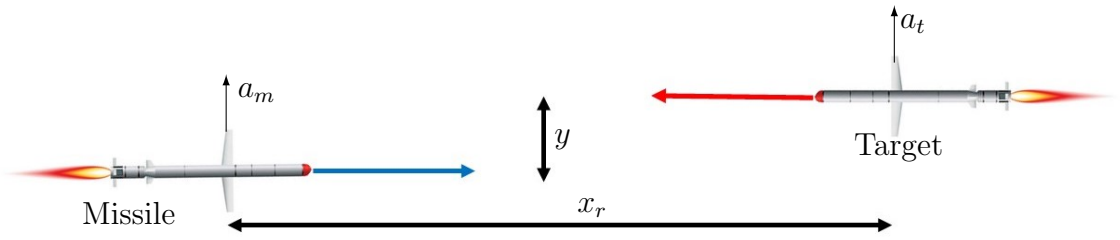

Fig. 7.1: Target Engagement Geometry

lateral position state y, a relative lateral velocity state $\dot{y}$, an unlimited missile acceleration state $a_{m,un}$, a target acceleration state a_t, and bias states for the sensors. A key element of nomenclature for this chapter is the unlimited missile acceleration $a_{m,un}$. This term is different than the actual realized acceleration of the missile a_m, which is not an element of the truth state vector. The two elements are related by the saturation function $a_m = sat(a_{m,un})$.

Note that the along-track range x_r is assumed to be deterministic, and is not an element of the state vector. The state vector has a total of 8 elements and is defined as:

$$\mathbf{x} = \begin{bmatrix} y \\ \dot{y} \\ a_{m,un} \\ a_t \\ b_a \\ s_a \\ \epsilon_a \\ b_{los} \end{bmatrix}_{8\times 1} \tag{7.1}$$

where b_a is an accelerometer bias, b_{los} is the line-of-sight measurement bias, s_a is the accelerometer scale factor, and ϵ_a is the accelerometer misalignment angle.

7.2.2 Truth State Dynamics

The state dynamics are generalized with the nonlinear function $\dot{\mathbf{x}} = f(\mathbf{x},\hat{\mathbf{u}}) + B\mathbf{w}$. The dynamics are a function of the state $\mathbf{x}$, the actuator commands $\hat{\mathbf{u}}$, and the process noise vector $\mathbf{w}$. The scale factor, misalignment angle, and biases are modeled as first-order Markov processes [22]. The missile acceleration is modeled as the commanded acceleration with a 1st-order lag and a saturation limiter. The truth state dynamics are:

$$\ddot{y} = a_t - sat(a_{m,un}) \tag{7.2}$$

$$\dot{a}_{m,un} = \frac{\hat{a}_c - a_{m,un}}{\tau_m} \tag{7.3}$$

$$\dot{a}_t = -\frac{a_t}{\tau_t} + w_t \tag{7.4}$$

$$\dot{b}_a = -\frac{b_a}{\tau_b} + w_b \tag{7.5}$$

$$\dot{s}_a = -\frac{s_a}{\tau_s} + w_s \tag{7.6}$$

$$\dot{\epsilon}_a = -\frac{\epsilon}{\tau_\epsilon} + w_\epsilon \tag{7.7}$$

$$\dot{b}_{los} = -\frac{b_{los}}{\tau_{los}} + w_{los} \tag{7.8}$$

where $\hat{a}_c$ is the commanded acceleration from the flight computer. The saturation function $sat(x)$ in Equation 7.2 is shown in Figure 7.2. The saturation equation is written as

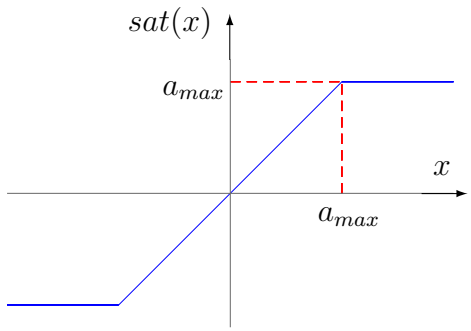

Fig. 7.2: Saturation Operator

$$sat(x) = \left\{ \begin{array}{ll} x, & \text{if } |x| \leq a_{max} \\ a_{max} sgn(x), & \text{if } |x| > a_{max} \end{array} \right\} \tag{7.9}$$

The process noise vector is:

$$\mathbf{w} = \begin{bmatrix} w_t \\ w_b \\ w_s \\ w_\epsilon \\ w_{los} \end{bmatrix}_{5\times 1} \tag{7.10}$$

The noise is modeled as zero-mean Gaussian white noise. The element w_t represents the random evasive acceleration maneuvers by the target and the remaining elements represent process noise in the bias dynamics. The PSD values of the noise vector are found in equations 7.12 and 7.11 below. The steady state covariance of each exponentially correlated random variable (ECRV) is written as σ_{ss}, with time-constant τ.

$$E\big[\mathbf{w}(t)\mathbf{w}(t')^T\big] = S_{\mathbf{w}}\delta(t-t') \tag{7.11}$$

$$S_{\mathbf{w}5\times5} = diag\Big(\frac{2\sigma_{t,ss}^2}{\tau_t}, \frac{2\sigma_{b,ss}^2}{\tau_b}, \frac{2\sigma_{s,ss}^2}{\tau_s}, \frac{2\sigma^2\epsilon, ss}{\tau_\epsilon}, \frac{2\sigma_{los,ss}^2}{\tau_{los}}\Big) \tag{7.12}$$

7.2.3 Guidance Law

The guidance law $\hat{\mathbf{u}}$ calculates the lateral acceleration that is needed for the missile to maintain a collision course with the target. The variable $\hat{\mathbf{u}}$ has an overhat, indicating that the guidance command is computed using the flight computers estimate of the states. The guidance law equation from Chapter 6 is:

$$\hat{\mathbf{u}} = \hat{k}_p\hat{V}_c\dot{\hat{\lambda}} \tag{7.13}$$

where $\hat{k}_p$ is the controller gain, $\hat{V}_c$ is the closing velocity, and $\dot{\hat{\lambda}}$ is the line-of-sight rate. In order to write this guidance law in terms of the state vector elements, an alternate form is derived.

Figure 7.3 shows a visualization of the time of closest approach. The missile and target are initially separated by a relative position of $\mathbf{r}_{rel}$, and the missile is approaching the target with a relative velocity $\mathbf{v}_{rel}$. The magnitude of the relative position at the final time is labeled as r_{tf}, and represents the closest approach distance (see also Figures 6.2 and 6.3). The angles λ and θ_l represent the line-of-sight angle and the lead angle of the engagement. The line-of-sight coordinate frame and the inertial frame are also pictured. Suppose the

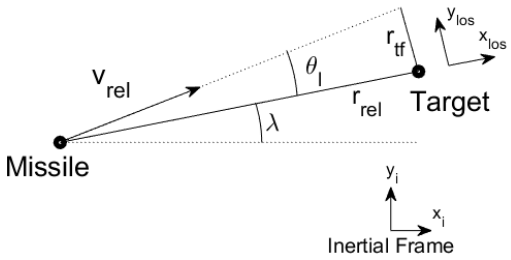

Fig. 7.3: Final Time Relative Position In LOS Frame

relative position and velocity vectors in the inertial frame are defined as follows:

$$\mathbf{r}_{rel} = \begin{bmatrix} r_1 \\ \\ r_2 \end{bmatrix} \tag{7.14}$$

$$\mathbf{v}_{rel} = \begin{bmatrix} v_1 \\ \\ v_2 \end{bmatrix} \tag{7.15}$$

Then, the relative position vector in the LOS frame is:

$$\mathbf{r}_{rel}^{los} = T_i^{los}\mathbf{r}_{rel} \tag{7.16}$$

where T_i^{los} is the DCM from equation 6.59

$$T_i^{los} = \begin{bmatrix} \cos\lambda & \sin\lambda \\ \\ -\sin\lambda & \cos\lambda \end{bmatrix} \tag{7.17}$$

and λ is the line-of-sight angle

$$\lambda = \arctan\frac{r_2}{r_1} \tag{7.18}$$

The line-of-sight rate equation is the time derivative of equation 7.18

$$\dot{\lambda} = \frac{\delta\lambda}{\delta t} = \frac{r_1 v_2 - r_2 v_1}{||\mathbf{r}_{rel}||^2} \tag{7.19}$$

Utilizing rectilinear dynamics, and assuming constant velocity, the final time relative position vector is written as

$$\mathbf{r}_{rel,tf} = \mathbf{r}_{rel} + \mathbf{v}_{rel} t_{go} = \begin{bmatrix} r_{tf_1} \\ \\ r_{tf_2} \end{bmatrix} \tag{7.20}$$

The final time relative position vector in the LOS frame is a function of the current relative position elements r_1 and r_2 and the current relative velocity elements v_1 and v_2, rotated by the direction cosine matrix T_i^{los}.

$$\mathbf{r}_{rel,tf}^{los} = T_i^{los} \mathbf{r}_{rel,tf} = \begin{bmatrix} (r_1 + v_1 t_{go}) \cos\lambda + (r_2 + v_2 t_{go}) \sin\lambda \\ -(r_1 + v_1 t_{go}) \sin\lambda + (r_2 + v_2 t_{go}) \cos\lambda \end{bmatrix} \tag{7.21}$$

The closest approach distance r_{tf} from Figure 7.3 is the second element of equation 7.21. Substituting in the values for $\sin\lambda$ and $\cos\lambda$:

$$r_{tf} = -(r_1 + v_1 t_{go}) \frac{r_2}{||\mathbf{r}_{rel}||} + (r_2 + v_2 t_{go}) \frac{r_2}{||\mathbf{r}_{rel}||} \tag{7.22}$$

Expanding and simplifying:

$$r_{tf} = t_{go} \frac{r_1 v_2 - r_2 v_1}{||\mathbf{r}_{rel}||} \tag{7.23}$$

Substituting equation 7.23 into equation 7.19

$$\dot{\lambda} = \frac{1}{t_{go}} \frac{r_{tf}}{||\mathbf{r}_{rel}||} \tag{7.24}$$

Assuming $V_c = \frac{x_r}{t_{go}}$, substituting equation 7.24 into the guidance law from equation 6.19, and substituting in the state vector values from Figure 7.1 yields the guidance law:

$$\hat{\mathbf{u}} = \hat{\mathbf{g}}(\hat{\mathbf{x}}) = \hat{a}_c = \hat{k} \hat{V}_c \dot{\hat{\lambda}} = \hat{k} \left(\frac{x_r}{t_{go}} \right) \left(\frac{1}{t_{go}} \frac{r_{tf}}{||\mathbf{r}_{rel}||} \right) \tag{7.25}$$

Substituting in Equation 7.23

$$\hat{a}_c = \hat{k} \left(\frac{x_r}{t_{go}} \right) \frac{1}{t_{go}} \frac{t_{go}}{||\mathbf{r}_{rel}||} \left(\frac{r_1 v_2 - r_2 v_1}{||\mathbf{r}_{rel}||} \right) \tag{7.26}$$

Assuming $r_1 = \hat{x}_r$, $r_2 = \hat{y}$, $v_1 = -\hat{V}_c = -\frac{\hat{x}_r}{\hat{t}_{go}}$, and $v_2 = \dot{\hat{y}}$, Equation 7.26 becomes:

$$\hat{a}_c = \hat{k}\left(\frac{\hat{x}_r}{\hat{t}_{go}}\right)\frac{1}{\hat{t}_{go}}\frac{\hat{t}_{go}}{||\mathbf{r}_{rel}||}\left(\frac{\hat{x}_r}{\hat{t}_{go}}\frac{\left(\dot{\hat{y}}t_{go} + \hat{y}\right)}{||\mathbf{r}_{rel}||}\right) \tag{7.27}$$

Substituting in $||\mathbf{r}_{rel}|| = \sqrt{\hat{x}_r^2 + \hat{y}^2}$ and simplifying yields the final form of the guidance law:

$$\hat{a}_c = \frac{\hat{k}\hat{x}_r^2}{\hat{t}_{go}^2}\frac{(\dot{\hat{y}}\hat{t}_{go} + \hat{y})}{(\hat{x}_r^2 + \hat{y}^2)} \tag{7.28}$$

This is the guidance law used by the Monte Carlo simulation. Note that the simplified form of this guidance law is used elsewhere, e.g. see [5, 73].

7.2.4 Sensor Truth Models

The accelerometer sensor measures the true specific force a_m on the vehicle corrupted by a first-order Markov process bias b_a and zero-mean Gaussian white noise η. The sensor also has a misalignment angle ϵ and a scale factor s.

$$\tilde{y} = (1 + s_a)\Big(a_m + b_a + \eta\Big)\cos\epsilon_a \tag{7.29}$$

The accelerometer measurement $\tilde{y}$ is modeled as a continuous inertial measurement (see Section 4.1). The symbol $\tilde{y}$ is used to be consistent with the material developed in Chapter 4 and is not to be confused with the lateral position y. The Gaussian white noise in the sensor model η has a PSD S_η.

The discrete line-of-sight measurement includes a bias and white measurement noise. The bias b_{los} is modeled as a first-order Markov process and the measurement noise is a zero-mean Gaussian white noise ν_k with variance R_{los}. The line-of-sight angle measurement equation is:

$$\tilde{z}_{los} = \tilde{\lambda} = \arctan\left(\frac{y}{x_r}\right) + b_{los} + \nu_{los} \tag{7.30}$$

where the components of the relative position vector from the missile to the target are x_r and y (see Figure 7.1). The variable y is the lateral relative position and is an element of the

state vector. Since the along track range of the target x_r is deterministic, it is proportional to t_{go} and the constant closing velocity V_c.

$$x_r = V_c t_{go} \tag{7.31}$$

7.2.5 Navigation Filter Design Model

The navigation filter design model is used to design the navigation filter presented in the next section. The design model provides the models used to develop the extended Kalman filter (EKF) state and state covariance propagation and update equations. In this problem, the design model uses model replacement [90] for the missile velocity dynamics, therefore the design model does not contain the $a_{m,un}$ state or the guidance law.

$$\mathbf{x}^{dm} = \begin{bmatrix} y^{dm} \\ \dot{y}^{dm} \\ a_t^{dm} \\ b_a^{dm} \\ s_a^{dm} \\ \epsilon_a^{dm} \\ b_{los}^{dm} \end{bmatrix}_{7\times 1} \tag{7.32}$$

The design model dynamics are generalized with the nonlinear function

$$\dot{\mathbf{x}}^{dm} = \mathbf{f}^{dm}(\mathbf{x}^{dm}, \tilde{\mathbf{y}}^{dm}) + B^{dm}\mathbf{w}^{dm} \tag{7.33}$$

The design model accelerometer measurement is modeled with the following equation:

$$\tilde{y}^{dm} = (1 + s^{dm})\left(a_m^{dm} + b_{acc}^{dm} + \eta^{dm}\right)\cos\epsilon^{dm} \tag{7.34}$$

The design model accelerometer noise η^{dm} has the following noise characteristics:

$$E[\eta^{dm}(t)] = 0 \tag{7.35}$$

$$E[\eta^{dm}(t)\eta^{dm}(t')] = S_{\eta}^{dm}\delta(t-t') \tag{7.36}$$

The design model dynamics are:

$$\ddot{y}^{dm} = a_t^{dm} - \left(\frac{\tilde{y}^{dm}}{(1+s_a^{dm})\cos\epsilon_a^{dm}} - b_a^{dm} - \eta^{dm}\right) \tag{7.37}$$

$$\dot{a}_t^{dm} = -\frac{a_t^{dm}}{\tau_t^{dm}} + w_t^{dm} \tag{7.38}$$

$$\dot{b}_a^{dm} = -\frac{b_a^{dm}}{\tau_b^{dm}} + w_b^{dm} \tag{7.39}$$

$$\dot{s}_a^{dm} = -\frac{s_a^{dm}}{\tau_s^{dm}} + w_s^{dm} \tag{7.40}$$

$$\dot{\epsilon}_a^{dm} = -\frac{\epsilon_a^{dm}}{\tau_\epsilon^{dm}} + w_\epsilon^{dm} \tag{7.41}$$

$$\dot{b}_{los}^{dm} = -\frac{b_{los}^{dm}}{\tau_{los}^{dm}} + w_{los}^{dm} \tag{7.42}$$

Note that second term in equation 7.37 represents a model for the "compensated" accelerometer measurements. Or in other words, the bias estimates are used to estimate the true vehicle acceleration based on the current accelerometer measurement reading.

The design model process noise vector is identical to the truth state process noise vector. The design model process noise vector is:

$$\mathbf{w}^{dm} = \begin{bmatrix} w_t^{dm} \\ w_b^{dm} \\ w_s^{dm} \\ w_\epsilon^{dm} \\ w_{los}^{dm} \end{bmatrix}_{5\times 1} \tag{7.43}$$

with noise characteristics:

$$E\big[\mathbf{w}^{dm}(t)\mathbf{w}^{dm}(t')^T\big] = S_{\mathbf{w}}^{dm}\delta(t-t') \tag{7.44}$$

$$S_{\mathbf{w}}^{dm}{}_{5\times5} = diag\Big(\frac{2\sigma_{t,ss}^{dm^2}}{\tau_a^{dm}}, \frac{2\sigma_{b,ss}^{dm^2}}{\tau_a^{dm}}, \frac{2\sigma_{s,ss}^{dm^2}}{\tau_s^{dm}}, \frac{2\sigma_{\epsilon,ss}^{dm^2}}{\tau_\epsilon^{dm}}, \frac{2\sigma_{los,ss}^{dm^2}}{\tau_{los}^{dm}}\Big) \tag{7.45}$$

7.2.6 EKF State Propagation

Based on the design model, the navigation state is propagated by integrating the state dynamics $\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\hat{\mathbf{x}}, \tilde{\mathbf{y}})$ where $\tilde{y}$ represents the accelerometer measurements in Equation 7.29. The navigation state vector includes estimates of the lateral position and velocity, target acceleration, and sensor parameters.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{y} \\ \dot{\hat{y}} \\ \hat{a}_t \\ \hat{b}_a \\ \hat{s}_a \\ \hat{\epsilon}_a \\ \hat{b}_{los} \end{bmatrix}_{7\times1} \tag{7.46}$$

The dynamics of the navigation state vector are written as:

$$\dot{\hat{y}} = \hat{a}_t - \left(\frac{\tilde{y}}{(1+\hat{s})\cos\hat{\epsilon}} - \hat{b}_a\right) \tag{7.47}$$

$$\dot{\hat{a}}_t = -\frac{\hat{a}_t}{\hat{\tau}_t} \tag{7.48}$$

$$\dot{\hat{b}}_a = -\frac{\hat{b}_a}{\hat{\tau}_b} \tag{7.49}$$

$$\dot{\hat{s}}_a = -\frac{\hat{s}_a}{\hat{\tau}_s} \tag{7.50}$$

$$\dot{\hat{\epsilon}}_a = -\frac{\hat{\epsilon}_a}{\hat{\tau}_\epsilon} \tag{7.51}$$

$$\dot{\hat{b}}_{los} = -\frac{\hat{b}_{los}}{\hat{\tau}_{los}} \tag{7.52}$$

7.2.7 EKF State Update

The partial derivative of the line-of-sight measurement equation with respect to the state vector is:

$$H_{los} = \left[\begin{array}{ccc} \frac{x_r}{\tilde{y}^2+x_r^2} & 0_{1\times5} & 1 \end{array} \right]_{1\times7} \tag{7.53}$$

Using this partial derivative, the state update equation is

$$\hat{x}_k^+ = \hat{x}_k^- + \hat{K}_k[\tilde{z}_k - \hat{h}(\hat{\mathbf{x}}_k)] \tag{7.54}$$

where the Kalman Gain is:

$$\hat{K}_k = \hat{P}_k^- \hat{H}_k^T [\hat{H}_k \hat{P}_k^- \hat{H}_k^T + \hat{R}_\nu]^{-1} \tag{7.55}$$

The equations for state covariance P are provided in the next two sections.

7.2.8 EKF Covariance Propagation

To setup the state covariance propagation equations, the partial derivatives of the design model state dynamics with respect to the state and the process noise is required. The partial derivative of the navigation state dynamics equation $\hat{\mathbf{f}}$ with respect to the navigation state $\hat{\mathbf{x}}$ is

$$\hat{F}_{\hat{\mathbf{x}}} = \left[\begin{array}{ccccccc} & & & 0 & 1 & 0_{1\times5} & \\ 0_{1\times2} & 1 & 1 & \frac{\tilde{y}}{(1+\hat{s}_a)^2\cos\hat{\epsilon}_a} & -\frac{\tilde{y}\tan\hat{\epsilon}_a}{(1+\hat{s}_a)\cos\hat{\epsilon}_a} & 0 & \\ & & & 0_{5\times2} & I_{5\times5}A_{\hat{\tau}} & & \end{array} \right]_{7\times7} \tag{7.56}$$

where

$$A_{\hat{\tau}} = diag(-\frac{1}{\hat{\tau}_t}, -\frac{1}{\hat{\tau}_b}, -\frac{1}{\hat{\tau}_s}, -\frac{1}{\hat{\tau}_\epsilon}, -\frac{1}{\hat{\tau}_{los}})$$

The accelerometer measurement $\tilde{y}$ is also a function of the state, so the partial derivative of the dynamics with respect to the state needs to be carried into the accelerometer measurements (via the chain rule). The partial derivative of the navigation state dynamics equation $\hat{\mathbf{f}}$ with respect to the accelerometer measurement $\tilde{\mathbf{y}}$ is

$$\hat{F}_{\tilde{\mathbf{y}}} = \begin{bmatrix} 0 \\ -\frac{1}{(1+\hat{s}_a)\cos\hat{\epsilon}_a} \\ 0_{5\times 1} \end{bmatrix}_{7\times 1} \tag{7.57}$$

Since the navigation model of the accelerometer measurement is:

$$\hat{\mathbf{c}}(\hat{\mathbf{x}}) = (1+\hat{s}_a)\left(a_m + \hat{b}_a\right)\cos\hat{\epsilon}_a \tag{7.58}$$

The partial derivative of $\hat{\mathbf{c}}$ with respect to $\hat{\mathbf{x}}$ is $\hat{C}_{\hat{\mathbf{x}}}$:

$$\hat{C}_{\hat{\mathbf{x}}} = \left[\frac{\partial\hat{\mathbf{c}}}{\partial\hat{\mathbf{x}}}\right]_{1\times 7} = \begin{bmatrix} 0 & 0 & 0 & \frac{\partial\hat{\mathbf{c}}}{\partial\hat{b}_a} & \frac{\partial\hat{\mathbf{c}}}{\partial\hat{s}} & \frac{\partial\hat{\mathbf{c}}}{\partial\hat{\epsilon}} & 0 \end{bmatrix}_{1\times 7} \tag{7.59}$$

where

$$\frac{\partial\hat{\mathbf{c}}}{\partial\hat{b}_a} = (1+\hat{s}_a)\cos\hat{\epsilon}_a$$

$$\frac{\partial\hat{\mathbf{c}}}{\partial\hat{s}_a} = \left(a_m + \hat{b}_a\right)\cos\hat{\epsilon}_a$$

$$\frac{\partial\hat{\mathbf{c}}}{\partial\hat{\epsilon}_a} = -(1+\hat{s}_a)\left(a_m + \hat{b}_a\right)\sin\hat{\epsilon}_a$$

The matrix $\hat{B}$ that maps the navigation process noise into the state dynamics is given by

$$\hat{B} = \begin{bmatrix} 0_{2\times 5} \\ I_{5\times 5} \end{bmatrix}_{7\times 5} \tag{7.60}$$

Using the generic covariance propagation equation in 4.13 and Equations 7.56-7.60, the differential equations for the EKF covariance propagation are given by

Equations 7.56-7.60 are substituted into equation 4.13 to propagate the EKF covariance:

$$\dot{\hat{P}} = (\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}\hat{C}_{\hat{\mathbf{x}}})\hat{P} + \hat{P}(\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}}\hat{C}_{\hat{\mathbf{x}}})^T + \hat{F}_{\tilde{\mathbf{y}}}\hat{S}_\eta\hat{F}_{\tilde{\mathbf{y}}}^T + \hat{B}\hat{S}_w\hat{B}^T \tag{7.61}$$

7.2.9 EKF Covariance Update

The state covariance is updated after a measurement using the equation

$$P_k^+ = [I - \hat{K}_k\hat{H}_k]\hat{P}_k^-[I - \hat{K}_k\hat{H}_k]^T + \hat{K}_k\hat{R}_\nu\hat{K}_k^T \tag{7.62}$$

Where K is the Kalman gain from Equation 7.55 and R represents the discrete measurement noise covariance matrix of the measurement being processed. The measurement partial derivative $\hat{H}$ is defined in the next section.

7.3 Statistical LinCov Model

This section presents the equations for the augmented state linear covariance simulation with statistical linearization. The general setup of the problem is the same; the states, measurements, and parameters are the same as the Monte Carlo setup. The difference however is that the Statistical LinCov simulation does not propagate individual trajectories, just the overall system statistics. In order to propagate only the statistics, the nonlinear models for the truth and navigation states need to be linearized. This section contains all the partial derivatives and statistical linearization equations needed to develop the Statistical LinCov simulation.

With a traditional linearization of a nonlinear system, the dynamics are generalized with the nonlinear function $\dot{\mathbf{x}} = f(\mathbf{x},\hat{\mathbf{u}}) + B\mathbf{w}$ (see Section 6.3.2). For statistical linearization, the nonlinear elements of the dynamics are separated from the linear elements with the following general equation:

$$\dot{\mathbf{x}} = F_{lin}\mathbf{x} + G_{lin}\hat{\mathbf{u}} + \mathbf{f}_{nonlin}(\mathbf{x}) + B\mathbf{w} \tag{7.63}$$

where F_{lin} is the partial derivative of the linear portion of the dynamics with respect to the state vector $\mathbf{x}$ and G_{lin} is the partial derivative of the linear portion of the dynamics with respect to the guidance law $\hat{\mathbf{u}}$. F_{lin} and G_{lin} are defined in Section 7.3.1. The nonlinear portion of the system is defined as:

$$\mathbf{f}_{nonlin}(\mathbf{x}) = A_n sat(a_{m,un}) \tag{7.64}$$

$$A_n = \begin{bmatrix} 0 & -1 & 0_{1\times 6} \end{bmatrix}^T_{1\times 8} \tag{7.65}$$

Equations 7.64 and 7.65 indicate that out of 8 state vector elements, the second element is the one with the saturation in the dynamics.

Since the truth dynamics are nonlinear, the statistical linearization techniques from Chapter 5 are utilized for the Statistical LinCov simulation. This involves propagating a mean value trajectory (otherwise called a nominal reference trajectory) and propagating/updating the system covariance matrix. Note that the mean value trajectory is not affected by measurement updates, so it does not need an update equation.

The mean value of the state vector $\dot{\overline{\mathbf{x}}}$, is propagated with equation 7.66, which is found by substituting equations 7.63 and 5.62 into equation 5.60:

$$\dot{\overline{\mathbf{x}}} = \check{\mathbf{f}} = F_{lin}\overline{\mathbf{x}} + G_{lin}\overline{\mathbf{u}} + A_n\check{f} \tag{7.66}$$

where $\check{f}$ is the random input describing function (RIDF) for the saturation function from equation 5.50:

$$\check{f} = \sigma_x\left[G\left(\frac{a_{max}+\mu_x}{\sigma_x}\right) - G\left(\frac{a_{max}-\ mu_x}{\sigma_x}\right)\right] - \mu_x \tag{7.67}$$

The statistical linearization augmented state covariance propagate and update equations are found by combining the augmented state covariance framework from Section 4.3.1 with the CADET method from Section 5.5.2. First, the linearized system is defined as:

$$\delta\dot{\mathbf{x}} = N\delta\mathbf{x} + F_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}}\delta\hat{\mathbf{x}} + B\mathbf{w} \tag{7.68}$$

$$\delta\dot{\hat{\mathbf{x}}} = \left[\hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}} C_{\hat{\mathbf{u}}} \hat{G}_{\hat{\mathbf{x}}}\right]\delta\hat{\mathbf{x}} + \hat{F}_{\tilde{\mathbf{y}}} C_{\mathbf{x}} \delta\mathbf{x} + \hat{F}_{\tilde{\mathbf{y}}}\boldsymbol{\eta} \tag{7.69}$$

where the partial derivatives in Equations 7.68 and 7.69 are fully defined in Section 7.3.1. Next, the development of the LinCov proceeds in a similar way to Chapter 4. The augmented state vector is defined as:

$$\mathbf{X} = \begin{bmatrix} \delta\mathbf{x} \\ \\ \delta\hat{\mathbf{x}} \end{bmatrix} \tag{7.70}$$

The augmented state vector X is zero mean with covariance C_A:

$$E[\mathbf{X}] = \mathbf{0} \quad E[\mathbf{X}\mathbf{X}^T] = C_A \tag{7.71}$$

The dynamics and update equations for the augmented state vector $\mathbf{X}$ are:

$$\dot{\mathbf{X}} = \mathcal{N}\mathbf{X} + \mathcal{G}\boldsymbol{\eta} + \mathcal{W}\mathbf{w} \tag{7.72}$$

$$\mathbf{X}_k{}^{+} = \mathcal{A}_k\mathbf{X}_k^{-} + \mathcal{B}_k\boldsymbol{\nu}_k \tag{7.73}$$

where

$$\mathcal{N} = \begin{bmatrix} N & F_{\hat{\mathbf{u}}}\hat{G}_{\hat{\mathbf{x}}} \\ \\ \hat{F}_{\tilde{\mathbf{y}}}C_{\mathbf{x}} & \hat{F}_{\hat{\mathbf{x}}} + \hat{F}_{\tilde{\mathbf{y}}} C_{\hat{\mathbf{u}}} \hat{G}_{\hat{\mathbf{x}}} \end{bmatrix}_{m\times m} \quad \mathcal{G} = \begin{bmatrix} 0_{n\times n_y} \\ \\ \hat{F}_{\tilde{\mathbf{y}}} \end{bmatrix}_{m\times n_y} \quad \mathcal{W} = \begin{bmatrix} B_{n\times n_w} \\ \\ 0_{\hat{n}\times n_w} \end{bmatrix}_{m\times n_w} \tag{7.74}$$

$$\mathcal{A}_k = \begin{bmatrix} I_{n\times n} & 0_{n\times\hat{n}} \\ \\ \hat{K}_k H_{\mathbf{x},k} & I_{\hat{n}\times\hat{n}} - \hat{K}_k\hat{H}_{\hat{\mathbf{x}},k} \end{bmatrix}_{m\times m} \quad \mathcal{B}_k = \begin{bmatrix} 0_{n\times n_z} \\ \\ \hat{K}_{k,\,\hat{n}\times n_z} \end{bmatrix}_{m\times n_z} \tag{7.75}$$

The Kalman gain $\hat{K}_k$ from Equation 7.75 is given by Equation 7.55, and the measurement partial $\hat{H}_{\hat{\mathbf{x}}}$ is defined in the next section.

The statistical augmented state covariance matrix C_A is propagated and updated with the following equations:

$$\dot{C}_A = \mathcal{N}C_A + C_A\mathcal{N}^T + \mathcal{G}S_\eta\mathcal{G}^T + \mathcal{W}S_w\mathcal{W}^T \tag{7.76}$$

$$C_A(t_i^+) = \mathcal{A}_i C_A(t_i^-)\mathcal{A}_i^T + \mathcal{B}_i R_\nu \mathcal{B}_i^T \tag{7.77}$$

The truth dispersion covariance equation is:

$$D_{\text{true}} = E[\delta\mathbf{x}\delta\mathbf{x}^T] = [I_{n\times n}\ \ 0_{n\times\hat{n}}]C_A[I_{n\times n}\ \ 0_{n\times\hat{n}}]^T \tag{7.78}$$

The covariance of the true estimation errors is given by the following equations:

$$P_{\text{true}} = [-M_{\mathbf{x}}\ \ I_{\hat{n}\times\hat{n}}]C_A[-M_{\mathbf{x}}\ \ I_{\hat{n}\times\hat{n}}]^T \tag{7.79}$$

where

$$M_{\mathbf{x}} = \begin{bmatrix} I_{2\times2} & 0 & 0_{2\times5} \\ 0_{5\times2} & 0 & I_{5\times5} \end{bmatrix}_{7\times8} \tag{7.80}$$

7.3.1 Statistical LinCov Partial Derivatives

This section contains the partial derivatives for the Statistical LinCov simulation. The partial derivatives in this section follow the partial derivative nomenclature from Chapter 6, with the addition of F_{lin} and G_{lin}, which are used to separate the linear elements of the model from the nonlinear elements of the model. The partial derivatives from equation 7.66 are defined as:

$$F_{lin} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0_{1\times4} \\ 0 & 0 & 0 & 1 & 0_{1\times4} \\ 0 & 0 & -\frac{1}{\tau_m} & 0 & 0_{1\times4} \\ 0 & 0 & 0 & -\frac{1}{\tau_t} & 0_{1\times4} \\ 0 & 0 & 0 & 0 & I_{4\times4}A_\tau \end{bmatrix}_{8\times8} \tag{7.81}$$

where

$$A_\tau = diag(-\frac{1}{\tau_b}, -\frac{1}{\tau_s}, -\frac{1}{\tau_\epsilon}, -\frac{1}{\tau_{los}})$$

and

$$G_{lin} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\hat{\tau}_m} \\ 0_{5\times 1} \end{bmatrix}_{8\times 1} \tag{7.82}$$

The equations for $\hat{F}_{\tilde{\mathbf{x}}}$, $\hat{F}_{\tilde{\mathbf{y}}}$, and B are the same the Monte Carlo simulation equations 7.56, 7.57, and 7.60. The matrix N represents the quasi-linear system dynamics matrix (QLSDM). N is similar to the partial derivative $F_{\mathbf{x}}$ from Equation 6.82, however, N also includes the statistical derivative of the saturation function. In other words, N represents a system dynamics matrix with the appropriate RIDF substituted in for the nonlinear saturation element. The remaining partial derivatives are defined below:

$$N = \begin{bmatrix} 0 & 1 & 0 & 0 & 0_{1\times 4} \\ 0 & 0 & -n & 1 & 0_{1\times 4} \\ 0 & 0 & -\frac{1}{\tau_m} & 0 & 0_{1\times 4} \\ 0 & 0 & 0 & -\frac{1}{\tau_t} & 0_{1\times 4} \\ 0 & 0 & 0 & 0 & I_{4\times 4}A_\tau \end{bmatrix}_{8\times 8} \tag{7.83}$$

where n is the RIDF from Equation 5.53:

$$n = \left[PI\left(\frac{a_{max} + \mu_x}{\sigma_x}\right) + PI\left(\frac{a_{max} - \mu_x}{\sigma_x}\right) \right] - 1 \tag{7.84}$$

Note that statistical linearization approximates the partial derivative of the saturation nonlinearity with respect to the acceleration input as the RIDF n:

$$\frac{\partial[sat(a_{m,un})]}{\partial a_{m,un}} \approx n \tag{7.85}$$

The RIDF n is also used in the $C_{\mathbf{x}}$ partial derivative:

$$C_{\mathbf{x}} = \left[\begin{array}{cccccccc} 0 & 0 & \frac{\partial \mathbf{c}}{\partial a_{m,un}}\Big|_{\overline{x}} & 0 & \frac{\partial \mathbf{c}}{\partial b_a}\Big|_{\overline{x}} & \frac{\partial \mathbf{c}}{\partial s}\Big|_{\overline{x}} & \frac{\partial \mathbf{c}}{\partial \epsilon}\Big|_{\overline{x}} & 0 \end{array} \right]_{1\times 8} \tag{7.86}$$

where

$$\frac{\partial \mathbf{c}}{\partial a_{m,un}}\Big|_{\overline{x}} = n(1+\overline{s}_a)\cos\overline{\epsilon}_a$$

$$\frac{\partial \mathbf{c}}{\partial b_a}\Big|_{\overline{x}} = (1+\overline{s}_a)\cos\overline{\epsilon}_a$$

$$\frac{\partial \mathbf{c}}{\partial s}\Big|_{\overline{x}} = \big(sat(\overline{a}_{m,un}) + \overline{b}_a + \overline{\eta}\big)\cos\overline{\epsilon}_a$$

$$\frac{\partial \mathbf{c}}{\partial \epsilon}\Big|_{\overline{x}} = -(1+\overline{s}_a)\big(sat(\overline{a}_{m,un}) + \overline{b}_a + \overline{\eta}\big)\sin\overline{\epsilon}_a$$

The guidance law partial derivatives are:

$$\hat{G}_{\hat{\mathbf{x}}} = \left[\begin{array}{ccc} \frac{\partial \hat{a}_c}{\partial \hat{y}}\Big|_{\overline{x}} & \frac{\partial \hat{a}_c}{\partial \dot{\hat{y}}}\Big|_{\overline{x}} & 0_{1\times 5} \end{array} \right]_{1\times 7} \tag{7.87}$$

where

$$\frac{\partial \hat{a}_c}{\partial \hat{y}}\Big|_{\overline{x}} = \frac{kx_r^2}{t_{go}^2}\left[\frac{x_r^2 - \overline{y}(2\dot{\overline{y}}t_{go} + \overline{y})}{(x_r^2+\overline{y}^2)^2}\right]$$

$$\frac{\partial \hat{a}_c}{\partial \dot{\hat{y}}}\Big|_{\overline{x}} = \frac{kx_r^2}{t_{go}(x_r^2+\overline{y}^2)}$$

$$F_{\hat{\mathbf{u}}} = \left[\begin{array}{c} 0 \\ 0 \\ \frac{1}{\hat{\tau}_m} \\ 0_{5\times 1} \end{array} \right]_{8\times 1} \tag{7.88}$$

$$C_{\hat{\mathbf{u}}} = 0 \tag{7.89}$$

The measurement equation used by the LinCov is:

$$\tilde{z}_{los} = \frac{y}{x_r} + b_{los} + \nu_{los} \tag{7.90}$$

The partial derivative with respect to the state vector is:

$$H_{los} = \begin{bmatrix} \frac{1}{x_r} & 0_{1\times 5} & 1 \end{bmatrix}_{1\times 7} \tag{7.91}$$

It is noteworthy that the statistical linearization gain n appears multiple times in the linearization, specifically in the $C_{\mathbf{x}}$ and $F_{\mathbf{x}}$ partial derivatives. Basic Statistical LinCov and Basic LinCov do not incorporate the correlation between navigation errors and truth state dispersions, and its effect on overall closed-loop performance. Now that statistical linearization has been added to augmented state linear covariance analysis, it is possible to calculate the effect of the nonlinearity on the full closed-loop system performance. This is a very useful implementation of LinCov which expands its application to a wider range of GN&C systems!

7.4 Statistical LinCov Validation

With the LinCov and Monte Carlo models defined, the next step is validating the LinCov simulation. Table 7.1 outlines the initial conditions and initial uncertainty of the state vector. The units for the accelerometer bias are milli-g's (mg), which represent the

Table 7.1: Statistical LinCov Initial Conditions

State	Mean Value	Units	3σ Uncertainty	Description
$y(t_0)$	0	m	10	initial relative lateral position
$\dot{y}(t_0)$	0	m/s	1	initial relative lateral velocity
$a_{m,un}(t_0)$	0	m/s^2	0.1	initial unlimited missile lat. accel.
$a_t(t_0)$	0	m/s^2	0.1	initial target acceleration
$b_a(t_0)$	0	mg	1	initial accelerometer bias
$s_a(t_0)$	0	ppm	300	initial accelerometer scale factor
$\epsilon_a(t_0)$	0	mrad	0.5	initial aclmtr. misalignment
$b_{los}(t_0)$	0	mrad	3	initial line-of-sight bias

acceleration due to gravity times 10^{-3}. The units for the accelerometer scale factor are parts per million (ppm), which multiplies the scale factor by 1×10^6. Table 7.2 contains the system process noise parameters and other parameters needed to run the simulation

Table 7.2: Statistical LinCov Parameters

Parameter	Value	Units	Description
S_η	0.03	(m/s)/sqrt(hr)	3σ accelerometer random walk
ν_{los}	1.5	mrad	3σ line-of-sight measurement noise std dev
$\sigma_{w_t,ss}$	15	g	3σ target steady state 3σ acceleration
$\sigma_{a,ss}$	1	mg	3σ steady state accelerometer bias
$\sigma_{s,ss}$	1000	ppm	3σ steady state scale factor
$\sigma_{\epsilon,ss}$	0.5	mrad	3σ steady state accelerometer misalignment
$\sigma_{los,ss}$	3	mrad	3σ steady state line-of-sight bias
τ_m	1	s	missile acceleration low pass filter time-constant
τ_t	1	s	target acceleration time-constant
τ_b	1000	s	accelerometer bias time-constant
τ_s	500	s	scale factor time-constant
τ_ϵ	500	s	misalignment bias time-constant
τ_{los}	500	s	line-of-sight bias time-constant
k	4		control law gain
V_c	1500	m/s	engagement closing velocity
a_{max}	10	g	maximum lateral acceleration of the missile
t_{meas}	10	Hz	discrete measurement rate

Statistical LinCov is validated using the same process as Section 6.5.5. The Monte Carlo simulates 1000 trajectories and provides truth data statistics to validate the linear covariance results. The LinCov generated statistics should be within the confidence interval of the Monte Carlo statistics, and the $\pm 3\sigma$ covariance bounds should contain 99.7% of the data from the Monte Carlo trajectories. The simulation is run with the guidance law, measurements, and all noise sources activated.

Figure 7.4 illustrates the true estimation error of the state variables. The plot lines are defined as follows: the gray lines represent individual Monte Carlo runs, the black lines represent the $\pm 3\sigma$ Monte Carlo estimation error covariance, the black error bars represent the 95% confidence interval for the Monte Carlo estimation error covariance based on the

number of simulated runs, the blue lines represent the Kalman Filter covariance from the Monte Carlo, and the red dashed lines represent the Statistical LinCov estimation error covariance. For clarity, the confidence interval error bars are shown in discrete intervals on the plots. The measurements drive down the lateral position estimation error down throughout the engagement, as expected. The error initialization fits the expected values from table 7.1, and the $\pm 3\sigma$ LinCov covariance bounds are a satisfactory approximation of the Monte Carlo ensemble statistics.

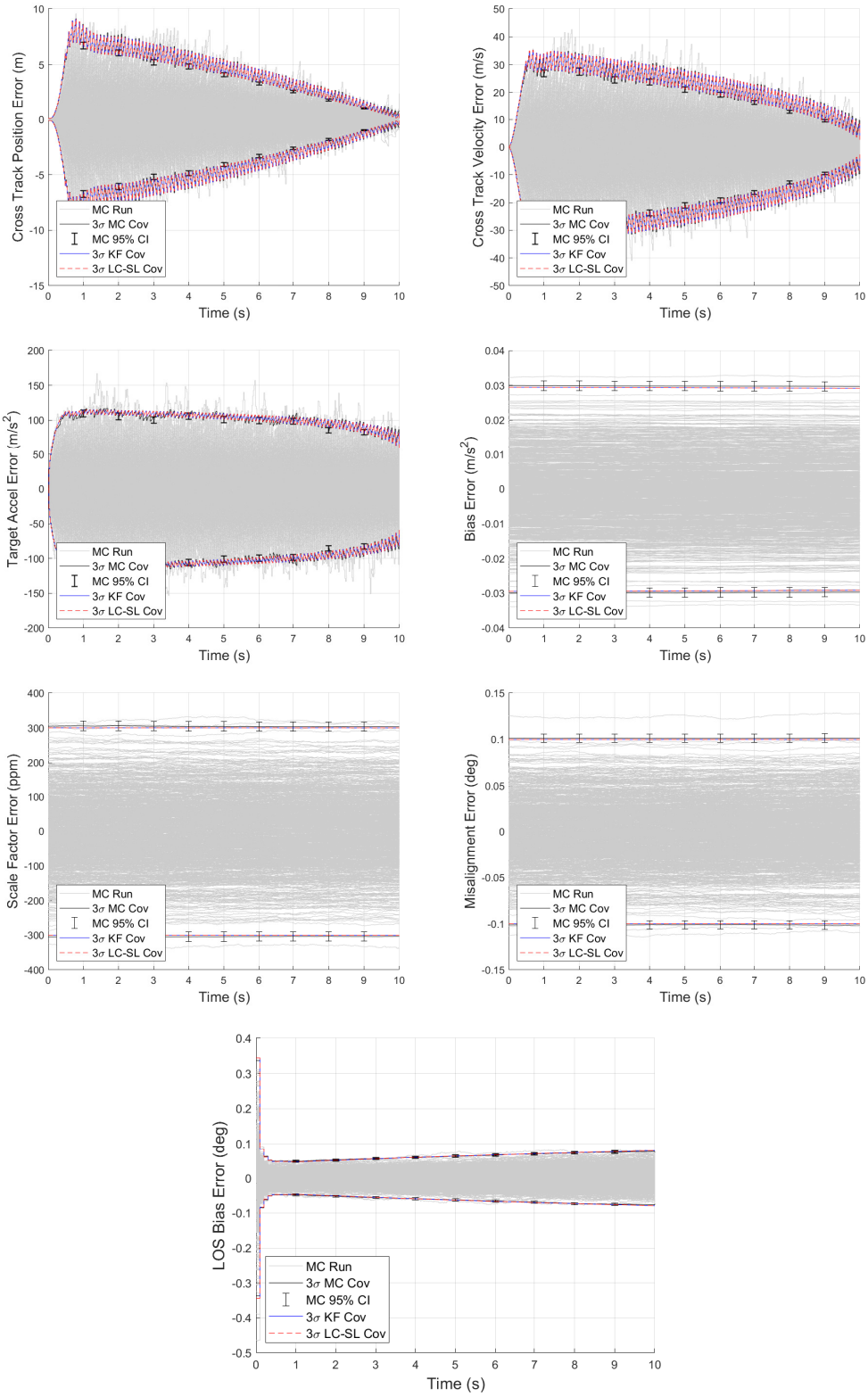

Fig. 7.4: Statistical LinCov Estimation Error Validation Plots

The truth dispersion validation plots are exhibited in Figure 7.5. Each plot contains the Monte Carlo trajectories and statistics, and two different LinCov bounds. The red lines represent the LinCov results with traditional linearization, while the blue lines represent the LinCov results with statistical linearization. For all the truth states, the statistical linearization LinCov accurately approximates the Monte Carlo results. States 1-3 in Figure 7.5 show that the traditional linearization does not capture the effect of the saturation, and overestimates the ability of the missile to drive down the miss distance on the target. The other state vector elements from Figure 7.5 are target states or sensor states, and therefore are not affected by the missile actuator saturation. The final cross-track position dispersions in plot 7.5.1 represents the 3σ miss distance of the engagement. With the acceleration bounds and sensors parameters from Table 7.2, the 3σ miss distance of the engagement is about 375 meters.

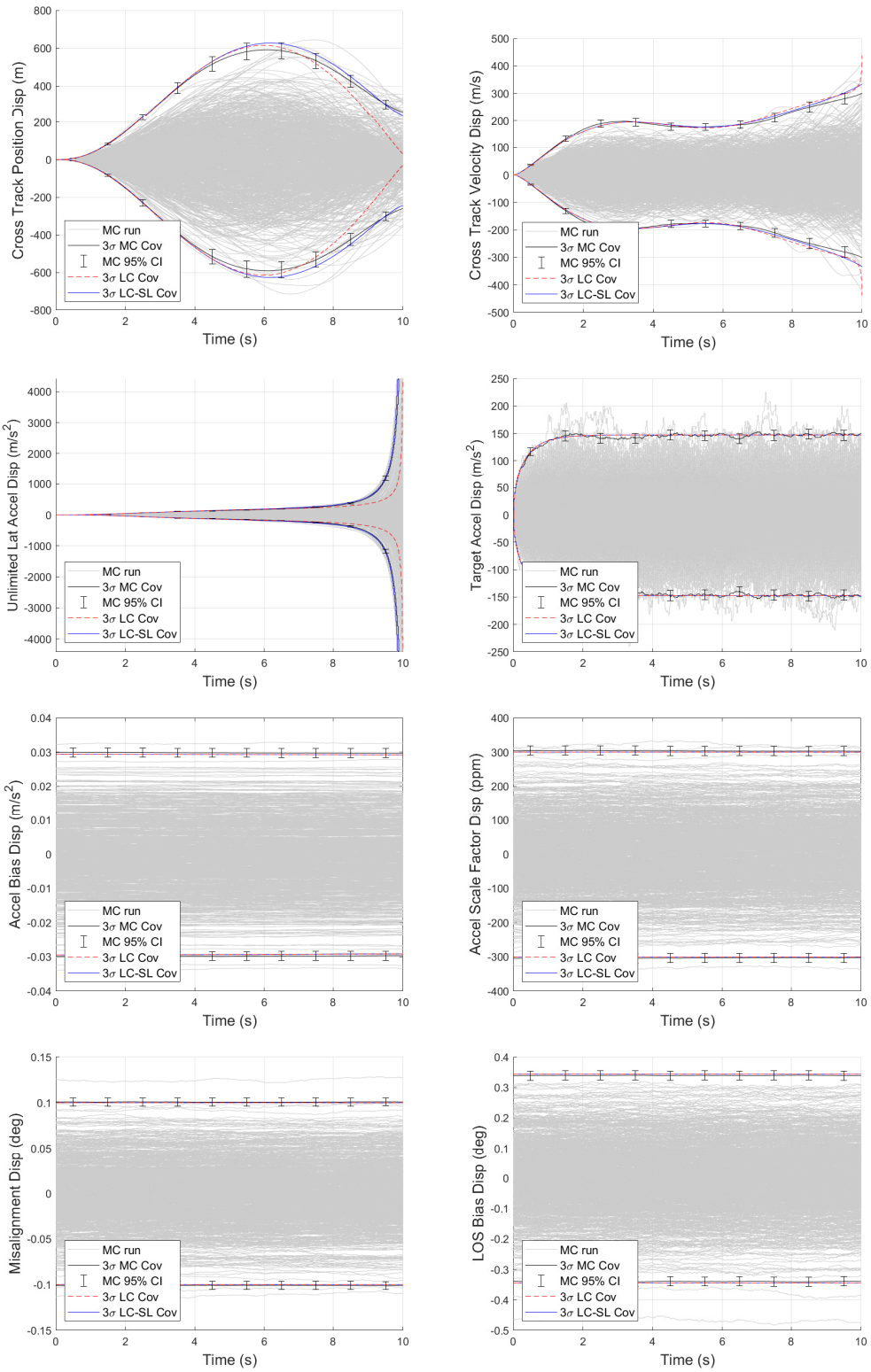

Fig. 7.5: Statistical LinCov Truth Dispersion Validation Plots

Figure 7.6 shows the Monte Carlo and LinCov results for a wide range of maximum missile acceleration values. The statistical linearization LinCov matches up well with the

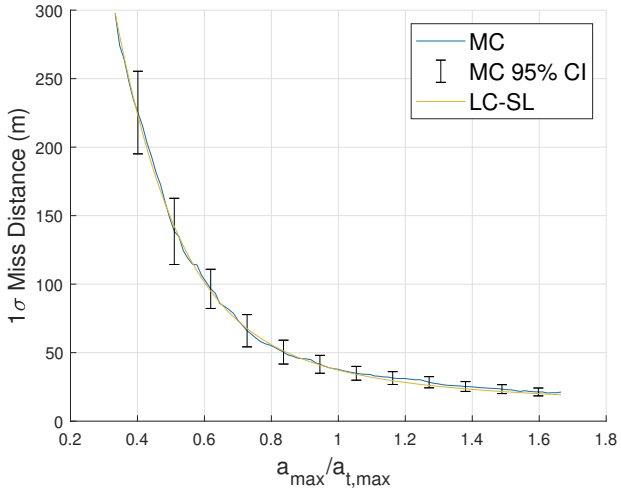

Fig. 7.6: Statistical LinCov Miss Distance With Varying Acceleration Limits

Monte Carlo for the full range of inputs. Different missile engagements operate with different saturation conditions, so it is beneficial to have an accurate linearization for all the operating conditions.

The computation time that goes into generating data for a parameter-varying Monte Carlo study is significant and highlights the benefits of statistical LinCov. Each data point on the blue line in Figure 7.6 represents 1000 Monte Carlo runs, while each data point on the yellow line represents 1 Statistical LinCov run. The Monte Carlo simulation took about 12 hours to run this study, while the Statistical LinCov simulation did the same analysis in less than 1 minute.

Statistical LinCov utilizes statistical linearization to approximate the saturation function in the state vector dynamics of a missile engagement problem. The statistical linearization is dependent on the probability density function (PDF) of the input. This method does not rely on mode switches in the GN&C algorithm, but rather a more robust linearization technique. The next approach, Tunable LinCov, will utilize GN&C mode switching to model actuator saturation system behavior.

7.5 Tunable LinCov

Another method for approximating nonlinear behavior in a LinCov simulation is called Tunable LinCov. This method is also new to augmented state linear covariance analysis. The basic idea of Tunable LinCov is to use traditional linearization techniques to create multiple system models, and switch between them using pre-defined mode switching criterion. It is called 'Tunable' LinCov because the mode switching parameters need to be tuned for a given problem. This section will demonstrate Tunable LinCov for the missile engagement problem.

A LinCov simulation does not simulate individual trajectories to determine when a saturation will occur, but rather a single nominal trajectory, along with the system performance statistics. For a missile engagement problem with a saturated actuator, the critical state element to monitor is the missile acceleration. For Tunable LinCov, the critical state element is the 1σ value of the missile acceleration.

The truth state vector and navigation state vectors for the Tunable Lincov are identical to the Statistical LinCov simulation. The Monte Carlo model used for validation of the Tunable LinCov simulation is the same as the model defined in Section 7.2. The LinCov simulation has two different modes, and switches modes according to the following condition:

$$\text{mode} = \left\{ \begin{array}{ll} \text{unsaturated}, & \text{if } \sigma_{disp,a_m} \leq a_{switch} \\ \text{saturated}, & \text{if } \sigma_{disp,a_m} > a_{switch} \end{array} \right\} \tag{7.92}$$

where a_{switch} is the tuned switching acceleration value, and σ_{disp,a_m} is defined as the standard deviation of the missile acceleration truth dispersion covariance:

The partial derivatives required for the Tunable LinCov are obtained relatively easily from the Statistical LinCov partial derivatives in Section 7.3. In the Unsaturated Mode, all occurrences of $sat(a_{m,un})$ and n in the Statistical LinCov partial derivatives are replaced by a_m and 1, respectively, to obtain the corresponding partial derivatives for the Tunable LinCov. It can be seen that only $F_{\mathbf{x}}$, $C_{\mathbf{x}}$ are affected. In the Saturated Mode all occurrences of $sat(a_{m,un})$ and n are replaced by a_{max} and 0, respectively.

7.6 Tunable LinCov Validation

This section contains the validation for Tunable LinCov, utilizing the same simulation parameters as Section 7.4. The first guess for the switching parameter is $a_{switch} = a_{max}$. This produces the dispersion plots in Figure 7.7. Note that Figure 7.7 does not contain the

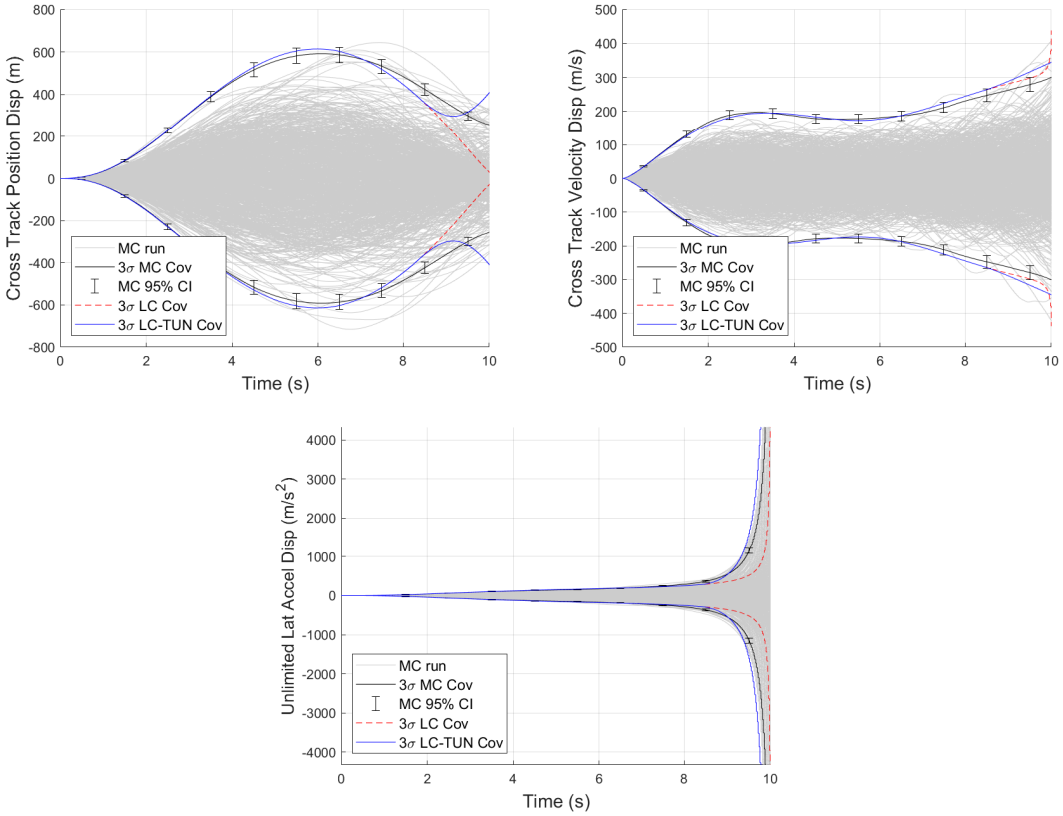

Fig. 7.7: Tunable LinCov- Step 1 Results

truth dispersion plots for the target acceleration or the sensor biases, as these elements are not significantly affected by the actuator saturation. The Tunable LinCov switches modes from unsaturated mode to saturated mode around 8.5 seconds into the simulation. The plots also show the result from traditional linearization LinCov. For this engagement example, the Tunable LinCov follows traditional linearization until the switching condition is met, and then behaves more like the statistical linearization thereafter.

The next step for tuning the LinCov is to vary switching parameter a_{switch} and find the value that best approximates the Monte Carlo results. The evaluation criteria for deciding the optimal value for a_{switch} is the 3σ Monte Carlo miss distance. Figure 7.8 shows the ratio between the LinCov 3σ miss distance and the Monte Carlo 3σ miss distance for varying values of a_{switch}. The Monte Carlo and LinCov miss distances match when a_{switch}/a_{max}

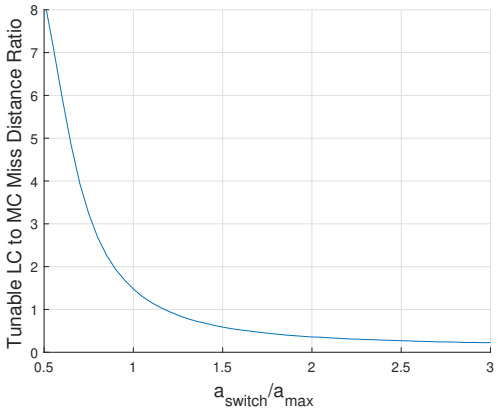

Fig. 7.8: Tunable LinCov Miss Distance Ratios Vs. a_{switch}/a_{max}

equals 1.2. Substituting in this value into the Tunable LinCov yields the best performance of the Tunable LinCov for these parameters.

Figure 7.9 shows how this value of a_{switch} affects the truth state dispersions. The LinCov changes modes at about 9 seconds into the simulation instead of at 8.5 seconds, and the final miss distance matches the Monte Carlo miss distance. The final value of the position dispersion ends up being correct, but the LinCov position dispersion statistics don't match the Monte Carlo statistics from $t = 6$ seconds to $t = 9.9$ seconds. Similarly, the cross-track velocity and unlimited acceleration statistics do not match the Monte Carlo statistics after approximately 6 seconds. These results suggest that the scope of Tunable LinCov is limited to the parameter it is optimized for. In the case of a terminal condition missile engagement, Tunable LinCov is accurate for the final miss distance statistics, but not necessarily for error budget analysis during the flight or velocity/acceleration analysis.

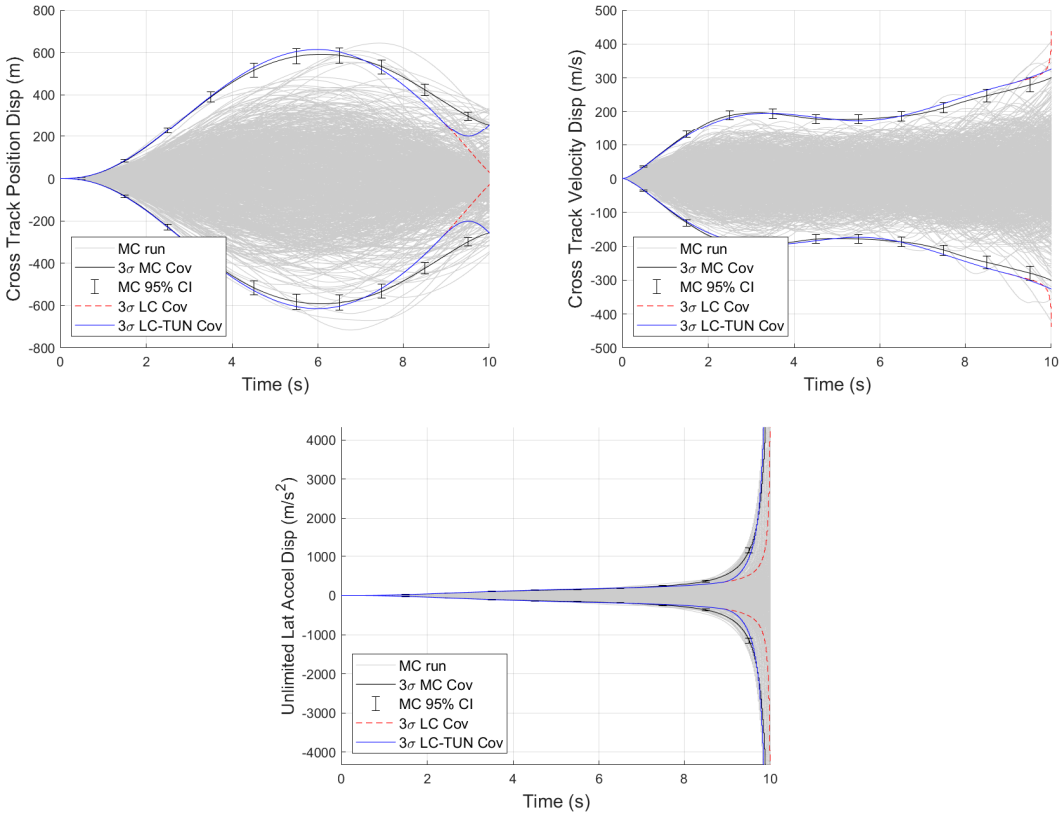

Fig. 7.9: Tunable LinCov- Step 2 Results

The remainder of the validation plots for the initial set of parameters are shown in Figures 7.10 and 7.11. Figure 7.10 shows the estimation error covariance for the Tunable LinCov. The estimation error for y, $\dot{y}$, and $a_{m,un}$ still match the Monte Carlo results, as the mode switching affects the truth dispersion statistics estimate more so than the estimation error statistics. Figure 7.11 contains the truth dispersion statistics for the sensor bias states. These states are not affected by the actuator saturation, and the Tunable LinCov matches the Monte Carlo truth dispersion statistics.

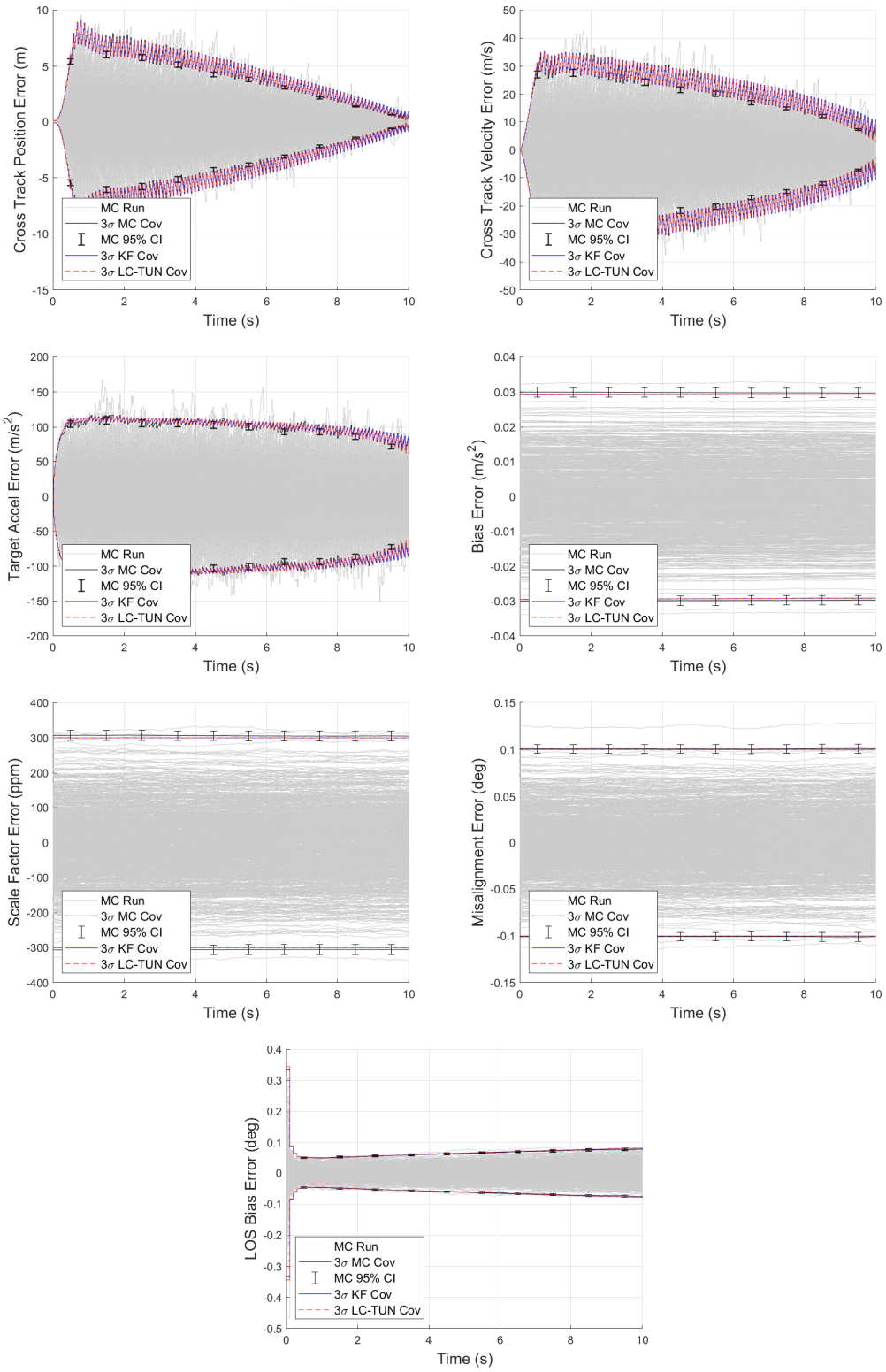

Fig. 7.10: Tunable LinCov Estimation Error Validation Plots

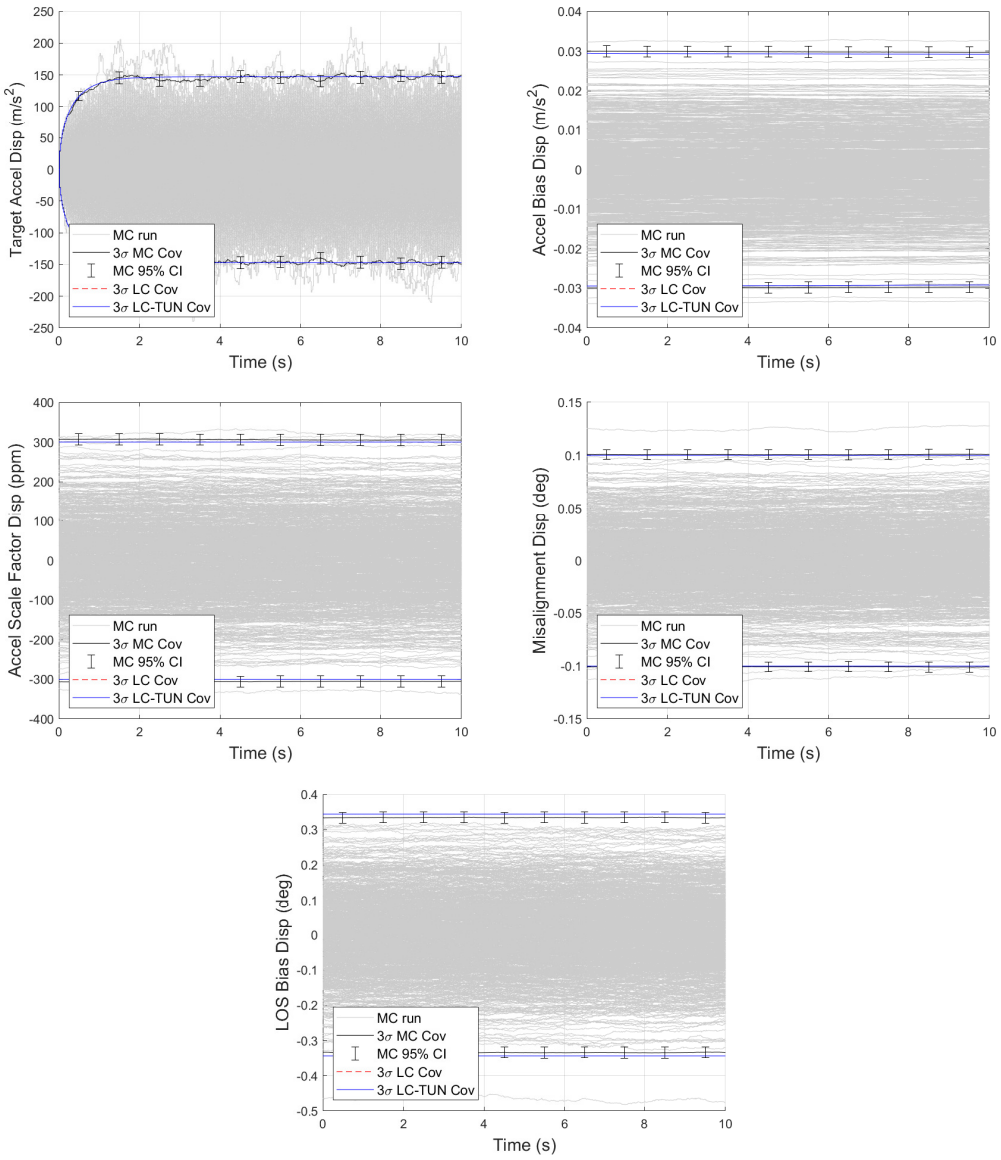

Fig. 7.11: Tunable LinCov Truth Dispersion Validation Plots

The final step for tuning the LinCov is to find the best value of a_{switch} for each value of a_{max}. The results of this step are found in Figure 7.12 for control gain values of 3,4, and 5. The data is then fitted to a curve to use when running the simulation.

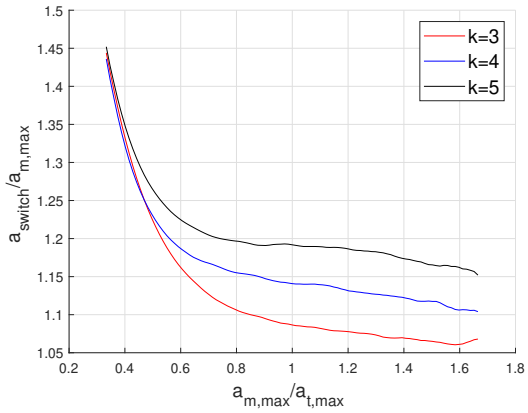

Fig. 7.12: Tunable LinCov Tuning Curve

For example, if it is desired to investigate a heavy saturation engagement scenario, a maximum missile acceleration value of 7.5g is chosen. Since the maximum target acceleration is 15g, the $a_{m,max}/a_{t,max}$ ratio to use for Figure 7.12 is 0.5. According to the tuning curve, the LinCov should use the switching parameter $a_{switch} = 9.2$g. The results from this example are shown in Figure 7.13. The Tunable LinCov miss distance values match the Monte Carlo values.

Figure 7.14 compares the miss distance results for Monte Carlo and Tunable LinCov for varying maximum acceleration values. The Tunable LinCov produces accurate miss distance statistics for the full range or acceleration inputs.

Tunable LinCov provides a viable solution for problems with GN&C mode switching. However, there are some limitations for Tunable LinCov. It gives accurate results for the states it is tuned for, but not necessarily all the states. For the missile engagement problem, Tunable LinCov provides accurate miss distances, and true estimation error statistics, but does not provide accurate position dispersions statistics for the entire flight. Another drawback to Tunable LinCov is that it requires extra testing to create a tuning curve.

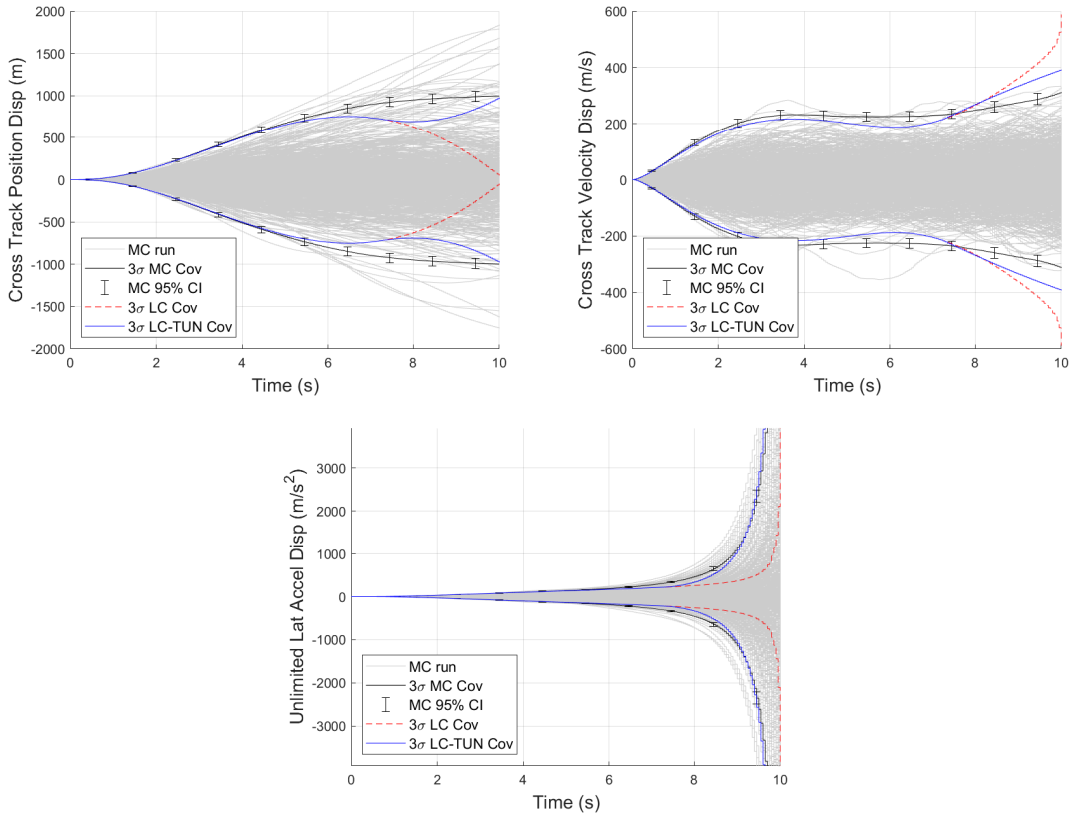

Fig. 7.13: Tunable LinCov Example Problem Results

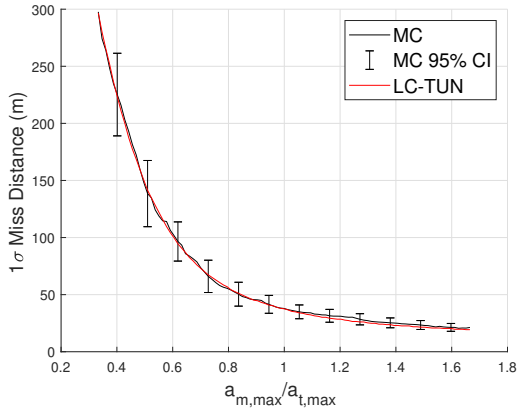

Fig. 7.14: Tunable LinCov Miss Distance With Varying Acceleration Limits

CHAPTER 8
LINCOV WITH EVENT TRIGGERS

Chapter 7 focused on building and validating LinCov tools that incorporate the effect of saturations in the actuator on system performance. Since the saturation of the missile actuator represents a change in the truth dynamics of the missile, it can also be modeled using event triggers. Historically, event triggers have been used in LinCov theory to model navigation based mode changes. Examples of this include terminal phase initiation maneuvers for orbital rendezvous based on relative angle estimates [71], or lunar ascent guidance mode changes based on estimated "time-to-go" until main engine cutoff [36]. The missile engagement saturation "event" is different than these other examples because it is does not represent a guidance mode change, but rather, a truth dynamics change. This chapter will review the navigation based event trigger equations and methods, introduce the new truth based event trigger equations, and incorporate them into a missile engagement LinCov simulation. This simulation will be referred to as Trigger LinCov. The simulation will be validated using the same Monte Carlo methods as Chapter 7.

8.1 Review of Navigation Based Event Triggers

The purpose of adding navigation based event triggers to augmented state linear covariance analysis is to incorporate the effect of a navigation based discrete event into the system performance statistics. A navigation event trigger utilizes the truth state dynamics equations, the navigation state dynamics equations, and a constraint equation Ψ that is a function of the navigation state vector $\hat{\mathbf{x}}$ at event time t_e. The constraint equation is written as:

$$\Psi[\hat{\mathbf{x}}(t_e)] = 0 \tag{8.1}$$

The constraint equation, truth state dynamics, and navigation state dynamics are linearized to quantify the effect of the discrete event on the augmented state covariance matrix C_A.

The three important metrics of event trigger theory are the time dispersion δt_e of the event, the variance of the event time σ_{te}^2, and the covariance shaping matrix Φ. Since the dynamics of the problem are stochastic in nature, the discrete event happens at a different time in each trajectory. The time dispersion δt_e is the difference in the nominal event time $\bar{t}_e$ and the event time for a specific trajectory t_e. The variance of the time dispersion σ_{te}^2 reveals the time window during which the event can be expected to happen. Typically, the noisier the environment, the wider the discrete event window is. The discrete event often changes the covariance of the system. This is incorporated into the LinCov by pre- and post- multiplying the augmented state covariance matrix C_A by the shaping matrix Φ. The equations for δt_e, σ_{te}^2, and Φ are derived in Section 4.4. The equations are summarized as follows:

$$\delta t_e = -\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) M_{\mathbf{x}}(\bar{t}_e) \dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1} \Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \delta\hat{\mathbf{x}}(\bar{t}_e) \tag{8.2}$$

$$\Omega(\bar{t}_e) = \begin{bmatrix} \mathbf{0}_{1\times n} & -\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) M_{\mathbf{x}}(\bar{t}_e) \dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1} \Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \end{bmatrix} \tag{8.3}$$

$$\sigma_{\delta t_e}^2 = \Omega(\bar{t}_e) C_A(\bar{t}_e) \Omega(\bar{t}_e)^T \tag{8.4}$$

$$\mathbf{\Phi}(\bar{t}_e) = \begin{bmatrix} I_{n\times n} & -\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) M_{\mathbf{x}}(\bar{t}_e) \dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1} \Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \\ 0_{\hat{n}\times n} & I_{\hat{n}\times\hat{n}} - M_{\mathbf{x}}(\bar{t}_e) \dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\hat{\mathbf{x}}}(\bar{t}_e) M_{\mathbf{x}}(\bar{t}_e) \dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1} \Psi_{\hat{\mathbf{x}}}(\bar{t}_e) \end{bmatrix} \tag{8.5}$$

$$C_a^+(\bar{t}_e) = \Phi(\bar{t}_e) C_a^-(\bar{t}_e) \Phi(\bar{t}_e)^T \tag{8.6}$$

The following example problem shows a missile engagement problem with navigation based event trigger used to model a guidance mode change. Suppose a missile is attempting to intercept a non-maneuvering target, and it is desired for the guidance law to be activated with a specified "time-to-go" before impact. The missile flies at constant velocity toward the target until the $\hat{t}_{go}$ threshold is reached, at which point the intercept guidance law is activated. This example is illustrated in Fig. 8.1. As is the case with any LinCov analysis, it is important to validate the simulation with Monte Carlo analysis. For the Monte Carlo simulation, the guidance law for each trajectory is enabled when the $\hat{t}_{go}$ threshold is reached. The Monte Carlo state vector truth dispersion statistics are then calculated as normal. For

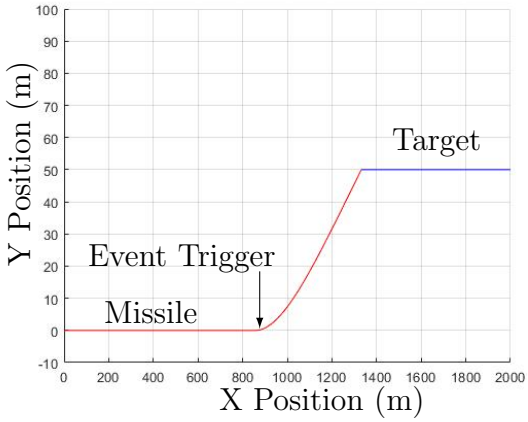

Fig. 8.1: Time-To-Go Missile Engagement Example Trajectory

the LinCov simulation, the augmented state covariance matrix is shaped by the shaping matrix Φ at the nominal time-to-go $\bar{t}_{go}$. Figure 8.2 illustrates the results of the mode change on the Y component of the velocity truth dispersion. The gray lines represent individual Monte Carlo trajectories, the blue dotted lines represent the Monte Carlo 3σ truth dispersion covariance and the black dotted lines represent the event trigger LinCov 3σ truth dispersions. The Event Trigger LinCov results align with the Monte Carlo results, indicating that the event trigger equations accurately model the mode change, even though it happens at a different time in each trajectory. Note that if the simulation time is not a fixed value, there is a period of time near the mode change where the Monte Carlo statistics are lost, as each trajectory runs for a different length of time.

8.2 Trigger LinCov: LinCov With Truth Based Event Triggers

For the missile engagement with actuator saturation problem, the discrete event constraint equation Ψ is a function of the truth state vector $\mathbf{x}$ at event time t_e. This is different than traditional applications of LinCov event triggers. In this case, the navigation based guidance law stays the same whether or not the actuator is saturated, but the truth dynamics are changed at the mode switch. The constraint equation for this type of problem is written as:

$$\Psi[\mathbf{x}(t_e)] = 0 \tag{8.7}$$

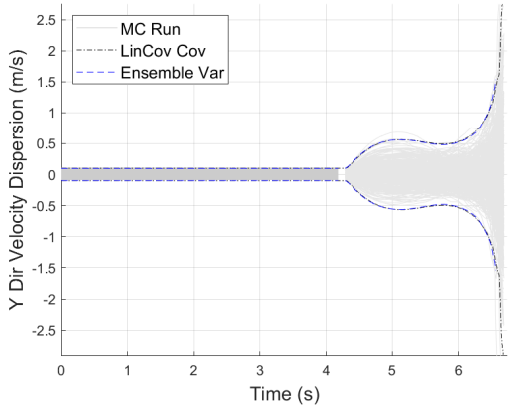

Fig. 8.2: Missile Engagement Event Trigger Example: Y Component Of The Velocity Truth Dispersion

The event trigger equations for a truth based discrete event are derived using a similar process as Section 4.4. The equations are shown below:

$$\delta t_e = -\left[\Psi_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\mathbf{x}}(\bar{t}_e)\delta\mathbf{x}(\bar{t}_e) \tag{8.8}$$

$$\Omega(\bar{t}_e) = \begin{bmatrix} -\left[\Psi_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\mathbf{x}}(\bar{t}_e) & 0_{1\times\hat{n}} \end{bmatrix} \tag{8.9}$$

$$\sigma^2_{\delta t_e} = \Omega(\bar{t}_e)C_A(\bar{t}_e)\Omega(\bar{t}_e)^T \tag{8.10}$$

$$\mathbf{\Phi}(\bar{t}_e) = \begin{bmatrix} I_{n\times n} - \dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\mathbf{x}}(\bar{t}_e) & 0_{n\times\hat{n}} \\ -M_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\left[\Psi_{\mathbf{x}}(\bar{t}_e)\dot{\bar{\mathbf{x}}}(\bar{t}_e)\right]^{-1}\Psi_{\mathbf{x}}(\bar{t}_e) & I_{\hat{n}\times\hat{n}} \end{bmatrix} \tag{8.11}$$

$$C_A^+(\bar{t}_e) = \Phi(\bar{t}_e)C_A^-(\bar{t}_e)\Phi(\bar{t}_e)^T \tag{8.12}$$

Equation 8.11 defines the covariance shaping matrix Φ that is used to incorporate the effect of the mode change into the augmented covariance matrix C_A. This equation has partial derivatives $\Psi_{\mathbf{x}}$ and $\dot{\bar{\mathbf{x}}}$ that are evaluated at the nominal event time $\bar{t}_e$. If the nominal trajectory dynamics equations $\dot{\bar{\mathbf{x}}}(t)$ are zero, as was the case in Chapter 7 with a nominal head-on intercept, then equation 8.11 reduces to the identity matrix. This means that the shaping matrix will not change the augmented state covariance matrix C_A at all, and any mode changes will not be modeled in the LinCov. Conceptually, this indicates that for

the missile engagement problem, event triggers are not suited to model a nominal head-on engagement with zero-mean random target evasive accelerations. Accordingly, this chapter will introduce a step input evasive maneuver similar to example problems from Chapter 4 of [8]. The simulation will begin with the defined simulation parameters, and at time $t = t_{evasive}$, the target will begin a constant acceleration evasive maneuver.

With the event trigger equations defined, the Trigger LinCov simulation model and Monte Carlo validation model can now also be defined. The Monte Carlo model for validation is the same as the model from Section 7.2, except for the following changes: the state vector will now have a "true missile acceleration" element a_m rather than an "unlimited missile acceleration" element $a_{m,un}$, an additional constant acceleration evasive maneuver is added, and the random evasive maneuvers by the target are smaller. The dynamics of the state vector are the same as equations 7.2-7.8 with the following differences:

$$\ddot{y} = a_t + a^*_{ref} - sat(a_{m,un}) \tag{8.13}$$

$$\dot{a}_m = \left\{ \begin{array}{ll} \frac{\hat{a}_c - a_m}{\tau_m}, & \text{if mode = unsaturated} \\ sgn(a_m) * a_{max}, & \text{if mode = saturated} \end{array} \right\} \tag{8.14}$$

where $a^*_{ref} = 0$ when $t < t_{evasive}$, and $a^*_{ref} = a_{t,evasive} = \text{constant}$ when $t >= t_{evasive}$. The LinCov switches modes from unsaturated mode to saturated mode when the event trigger constraint is met.

The target evasive maneuver from Chapter 7 was modeled as a zero mean exponentially correlated random variable (ECRV) with a steady state 3σ value of 15g. For this chapter, the target realizes a large step input acceleration that is deterministic, as well as a small ECRV acceleration. Conceptually, this is saying that for $t < t_{evasive}$ the target acceleration is somewhat constant and close to zero, and when $t > t_{evasive}$, the target acceleration is somewhat constant and close to $a_t = a_{t,evasive}$. The initial conditions and simulation parameters are defined in the next section. A visual representation of this maneuver is shown in Figure 8.3.

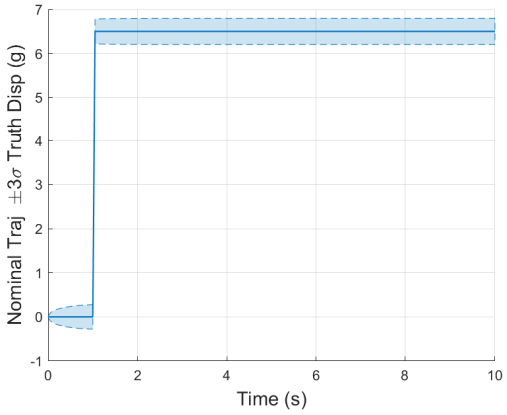

Fig. 8.3: Target Step Evasive Maneuver

The LinCov partial derivatives for the unsaturated mode are identical to the unsaturated mode partial derivatives from Section 7.5. The LinCov partial derivatives for the saturated mode are identical to the saturated mode partial derivatives from Section 7.5. The event trigger constraint equation is:

$$\Psi[\mathbf{x}(t_e)] = sat(a_{m,un}) - a_{max} \tag{8.15}$$

When this constraint equation is zero, the event triggers the mode change, and the augmented state covariance matrix C_A is shaped by the covariance shaping matrix Φ from equations 8.11 and 8.12. The LinCov utilizes the nominal reference trajectory (NRT) to calculate the nominal event time $\bar{t}_e$.

The partial derivative of the constraint equation Ψ with respect to the truth state vector $\mathbf{x}$ is:

$$\frac{\partial \Psi}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0_{1\times 5} \end{bmatrix}_{1\times 8} \tag{8.16}$$

8.3 Trigger LinCov Validation

This section contains the validation of the Trigger LinCov model. The NRT for the validation is shown in Figure 8.4. The blue lines represent the NRT and the red dotted

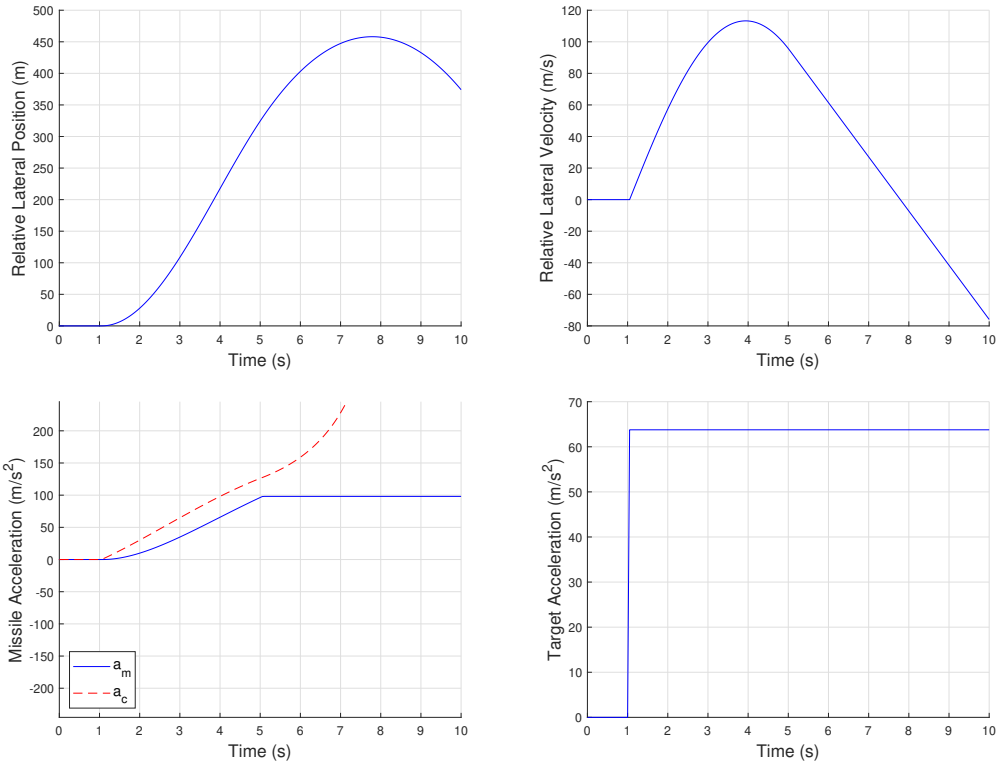

Fig. 8.4: Event Trigger LinCov Nominal Reference Trajectory

line in the missile acceleration plot represents the nominal trajectory guidance command. The target evasive step maneuver acceleration is initialized to $a_{t,evasive} = 6.5g$, and the maneuver time is $t_{evasive} = 1s$. This scenario represents a fairly heavy saturation, as the missile actuator saturates at $10g$ starting at about $t = 5s$. The nominal relative position plot shows that missile does not begin to reduce the lateral relative position until about $t = 7.5s$, and the miss distance from the engagement is about 375 meters. The initial conditions of the simulation are the same as Table 7.1. The simulation parameters are identical to Table 7.2, with the following difference:

$$\sigma_{w_t,ss} = 0.1g \tag{8.17}$$

The Monte Carlo runs 1000 trajectories and computes truth data statistics to validate the Trigger LinCov results. The Trigger LinCov statistics should be within the confidence interval of the Monte Carlo statistics, and the $\pm 3\sigma$ covariance bounds should contain 99.7% of the data from the Monte Carlo trajectories.

The estimation error validation plots are shown in Figure 8.5. The plot lines are defined as follows: the gray lines represent individual Monte Carlo runs, the black lines represent the $\pm 3\sigma$ Monte Carlo estimation error covariance, the black error bars represent the 95% confidence interval for the Monte Carlo estimation error covariance based on the number of simulated runs, the blue lines represent the Kalman Filter covariance from the Monte Carlo, and the red dashed lines represent the Trigger LinCov estimation error covariance. The confidence interval error bars are shown in discrete intervals on the plots. There is no noticeable jump in the true estimation error plots at $t = 5s$ when the actuator saturation occurs. This is as expected, as the even trigger equations do not affect the true estimation error [36]. Since the event trigger equations do not affect the true estimation error, there is not gap in the Monte Carlo statistics. The error initialization fits the expected values from table 7.1, and the $\pm 3\sigma$ LinCov covariance bounds are a satisfactory approximation of the Monte Carlo ensemble statistics.

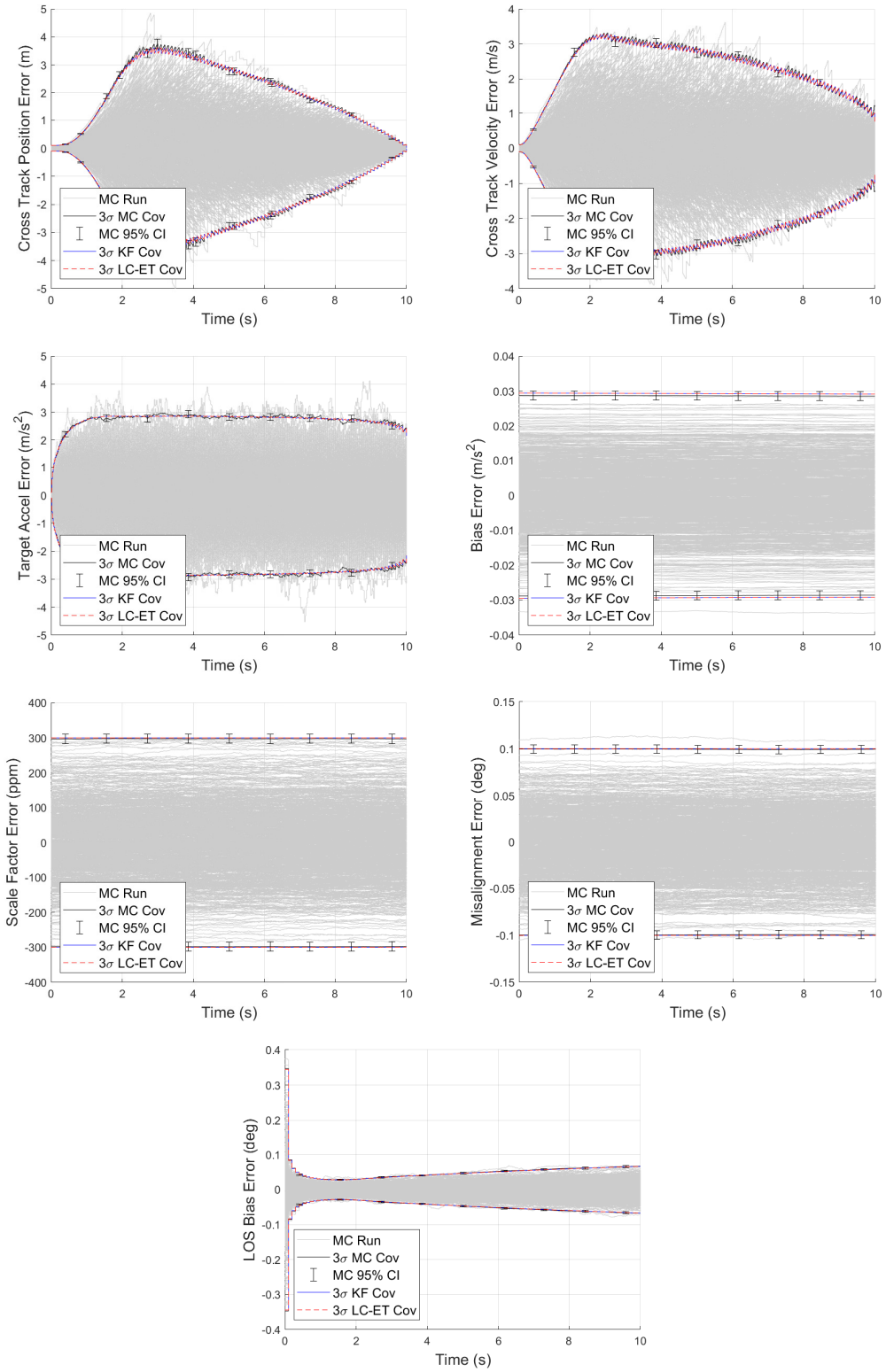

Fig. 8.5: Event Trigger LinCov Estimation Error Validation Plots

Figure 8.6 shows the truth dispersion validation plots. Each plot has the Monte Carlo trajectories and statistics, and two different LinCov results. The green lines represent the Statistical LinCov results, which are shown for reference, while the blue lines represent the Trigger LinCov results. Note that the lateral acceleration dispersion plot does not have Statistical LinCov results. The black lines represent the Monte Carlo truth dispersions, with 95% confidence interval error bars. States 1-3 are the states that are affected by the event trigger, and have small gaps in Monte Carlo statistics near the event time. This is because the saturation happens at a slightly different time for each run. The other truth states are not affected by the event trigger, and therefore do not have any gaps in Monte Carlo statistics.

At the event time, about $t = 5s$, the position and velocity dispersions start to grow a little faster since the actuator is saturated. Every trajectory has a saturation, so the lateral acceleration dispersions jump to zero at the event time. These changes are predicted by the Trigger LinCov, and match up with the Monte Carlo results. Keep in mind that since the NRT no longer has a miss distance of zero, the final time position dispersions do not represent the engagement miss distance. The 3σ miss distance in this case is equal to the nominal miss distance 375m $\pm$45m, (45 m is the final 3σ position truth dispersion). The final five states represent ECRV dispersion plots, and are accurately modeled by the Trigger LinCov. Overall, the event trigger equations have enabled the Trigger LinCov to switch modes, and the results are all within 5% of the Monte Carlo results. The Statistical LinCov results from Figure 8.6 all match the Monte Carlo results as well. This is another validation for Statistical LinCov using a different type of evasive maneuver.

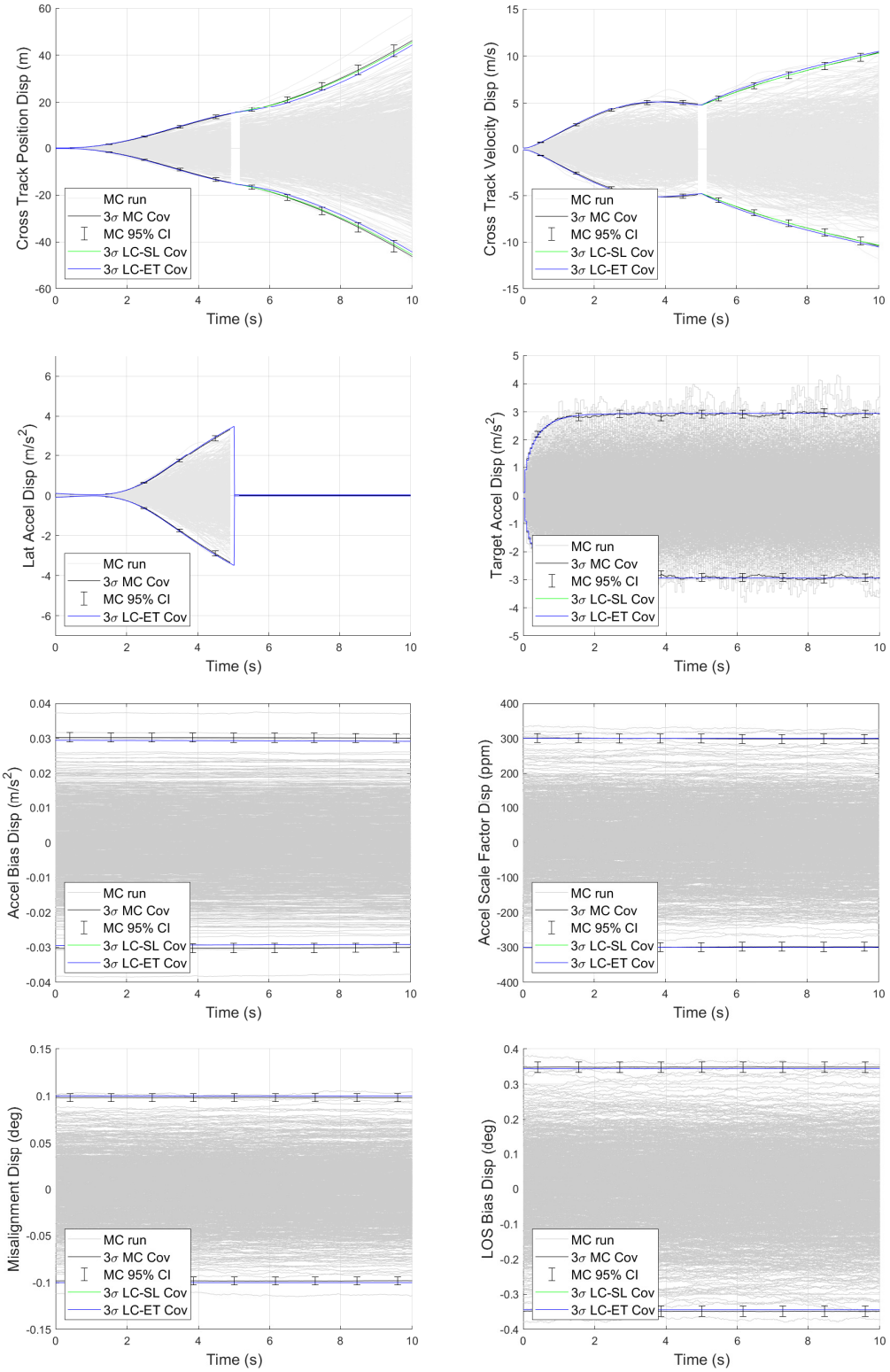

Fig. 8.6: Event Trigger LinCov Truth Dispersion Validation Plots

Now that a heavy saturation scenario has been validated, further insight can be gained by varying the levels of actuator saturation. The values for $t_{evasive}$ and $a_{t,evasive}$ will remain the same, but the value for a_{max} will vary from $a_{max} = 1g$ to $a_{max} = 20g$. This represents the full range from a heavy saturation to no saturation at all. For reference, Figure 8.7 shows the nominal miss distance for these values of a_{max}. The heavy saturation scenarios have high miss distances in the hundreds of meters, while the light saturation scenarios have miss distances close to zero.

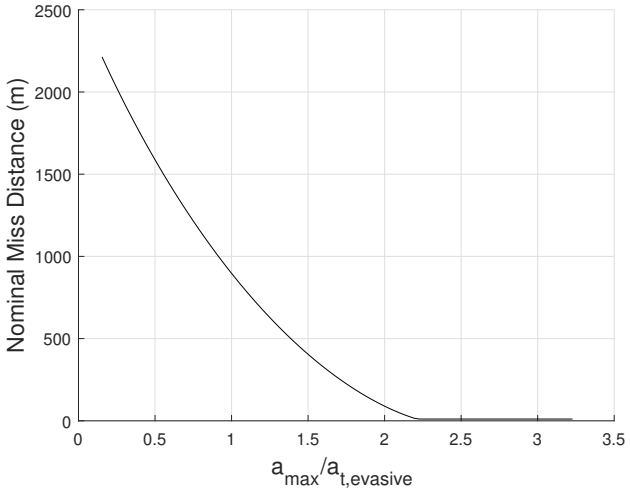

Fig. 8.7: Event Trigger LinCov Nominal Miss Distance With Varying Missile Acceleration Limits

Figure 8.8 shows the final position dispersions for varying values of a_{max}. The black represents the Monte Carlo results and 95% confidence intervals. The blue represents the Trigger LinCov results and the green represents the Statistical LinCov results. The Statistical LinCov and the Trigger LinCov accurately model the final position dispersions for varying levels of saturation.

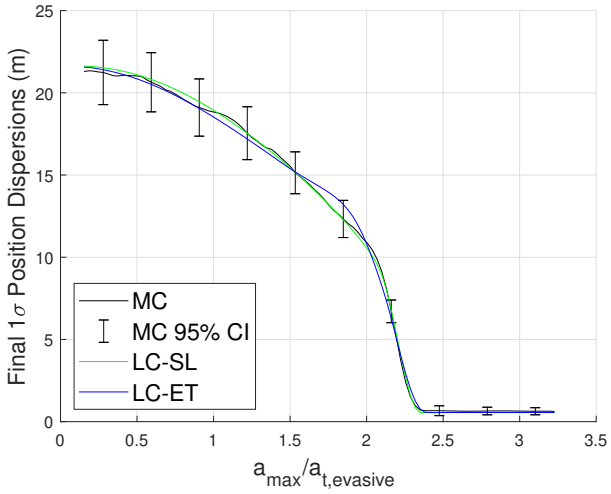

Fig. 8.8: Event Trigger LinCov Final Truth Dispersions With Varying Missile Acceleration Limits

8.4 Conclusion

This chapter defined the equations for truth based event triggers and the equations for Trigger LinCov. The equations were applied to the missile engagement problem, and compared to a Monte Carlo simulation. The Trigger LinCov simulation was able to accurately model the saturation nonlinearity in the missile engagement problem. This chapter also validated Statistical LinCov for evasive target step maneuvers.

CHAPTER 9

STATISTICAL LINCOV ENGAGEMENT ANALYSIS

Chapters 7 and 8 of this dissertation presented three different methods for incorporating a saturated actuator into a missile engagement problem: Statistical LinCov, Tunable LinCov, and Trigger LinCov. Each of these methods was incorporated into the augmented state linear covariance framework and tested against varying levels of actuator saturation. In general, the Statistical LinCov was shown to be the more robust of the three methods. It proved effective for engagements with random target maneuvers (modeled as ECRVs) and for engagements with step input target maneuvers. Although Tunable LinCov provided accurate miss distance statistics, it did not compute accurate truth dispersions for the entire length of the engagement. Trigger LinCov performed acceptably for target evasive step maneuvers, but not for random evasive maneuvers. Since Statistical LinCov does not have the limitations of Tunable LinCov and Trigger LinCov, this chapter will focus on Statistical LinCov analysis. The goal of this chapter is to show different analyses that can be done with Statistical LinCov. The analyses will cover varying control laws, sensor accuracy, target evasive maneuvers, and actuator responsiveness.

9.1 Guidance Laws: Proportional, Augmented, and Optimal Guidance

This section defines three different guidance laws for missile engagements: proportional navigation (PN) guidance, augmented proportional navigation (APN) guidance, and optimal navigation (ON) guidance.

The PN guidance law is the standard guidance law that is the basis for many missile engagement simulations. It utilizes the flight computer estimate of the relative lateral position and velocity between the missile and the target. This guidance law is derived using linear quadratic terminal control theory in Chapter 5 of [15]. PN guidance is optimized for a non-maneuvering target, no actuator delay, and a constant control gain. The equation for

the PN guidance law is:

$$\hat{a}_c = \frac{\hat{k}}{\hat{t}_{go}^2}\left(\hat{y} + \dot{\hat{y}}\hat{t}_{go}\right) \tag{9.1}$$

where the control gain $\hat{k}$ for the PN guidance is generally $3 <= \hat{k} <= 5$. Note that this is similar to, but not exactly a traditional PD controller, due to the $\hat{t}_{go}^{-1}$ terms.

The APN guidance law adds a feed-forward target acceleration term to the PN guidance law. This law is derived in Chapter 8 of [8] (see also page 116 of [99]). APN guidance is optimized for a target with constant acceleration, no actuator delay, and a constant control gain. The APN guidance law equation is:

$$\hat{a}_c = \frac{\hat{k}}{\hat{t}_{go}^2}\left(\hat{y} + \dot{\hat{y}}\hat{t}_{go} + 0.5\hat{a}_t\hat{t}_{go}^2\right) \tag{9.2}$$

where the values for the constant $\hat{k}$ are the same as for PN guidance, $3 <= \hat{k} <= 5$.

The ON guidance law appends a time-varying gain and a feed-forward missile acceleration term to the APN guidance law [57]. It also accounts for a missile acceleration first-order lag by including a missile acceleration time-constant. ON guidance is optimized for a constant target acceleration. The equation for ON guidance is:

$$\hat{a}_c = \frac{\hat{k}'}{\hat{t}_{go}^2}\left[\hat{y} + \dot{\hat{y}}\hat{t}_{go} + .5\hat{a}_t\hat{t}_{go}^2 - \hat{a}_m\hat{\tau}_m^2(e^{-\hat{\zeta}} + \hat{\zeta} - 1)\right] \tag{9.3}$$

where $\hat{k}'$ is the time-varying gain

$$\hat{k}' = \frac{6\hat{\zeta}^2(e^{-\hat{\zeta}} + \hat{\zeta} - 1)}{2\hat{\zeta}^3 + 3 + 6\hat{\zeta} - 6\hat{\zeta}^2 - 12\hat{\zeta}e^{-\hat{\zeta}} - 3e^{-2\hat{\zeta}}} \tag{9.4}$$

and $\hat{\zeta}$ is the time ratio

$$\hat{\zeta} = \frac{\hat{t}_{go}}{\hat{\tau}_m} \tag{9.5}$$

In this chapter, there are total of 7 guidance schemes that are considered: PN with $k = 3, 4, 5$, APN with $k = 3, 4, 5$, and ON. In general, a higher gain value yields a faster response of the guidance command, but is also more susceptible to overshoot or chatter. It

is important to remember that the missile engagement problem has various disturbances and navigation errors, which means that when the PN, APN, and ON guidance laws are implemented, they are not completely optimal, but rather "near optimal". However, if the trajectories remain in the immediate vicinity of the nominal reference trajectory (NRT), then "near optimal" guidance remains an effective solution [11, 15].

9.2 Guidance Law Performance Study

The first study introduces the behavior of the guidance laws against the random target maneuver from Chapter 7. This scenario is meant to represent a moderate saturation of the missile, with accurate sensors, and a moderately fast actuator response. The simulation parameters match Tables 7.1 and 7.2. Figure 9.1 shows the relative position truth dispersions of each guidance law. Each color in the plot represents a different guidance law, with different

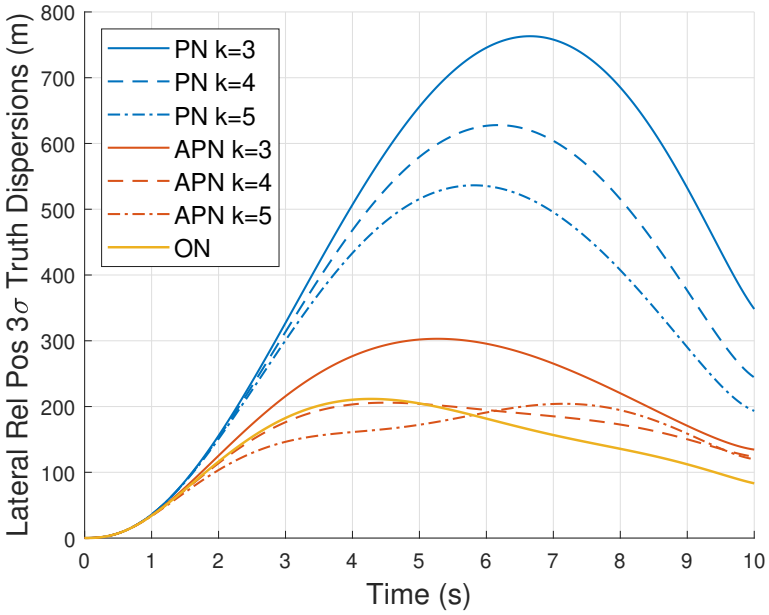

Fig. 9.1: Lateral Relative 3σ Dispersion for 7 Different Guidance Schemes

gains represented by different line tyeps. For example, the PN guidance law with a gain of 4 is shown with the blue dashed line. For reference, the PN guidance with $k_{gain} = 4$ is the

same result as the position dispersion plot from Figure 7.5. Note that because the random target maneuver is zero mean, the final time relative position truth dispersion is equal to the 3σ miss distance. In general, the APN and ON guidance laws perform much better than the PN guidance. Of all the guidance laws, the ON achieves the lowest 3σ miss distance. For the PN and APN guidance, the higher the gain, the faster the missile responds. It is noteworthy that the higher gain APN guidance laws respond a little faster to the target maneuvers than the ON. This makes sense, as the variable gain $\hat{k}'$ from the ON guidance law starts around 3.5 for this scenario, and eventually rises up above 10.

The next example shows how each guidance law performs against a target step maneuver, similar to the scenario from Section 8.3. The simulation parameters are identical to that engagement: a 6.5g target step evasive maneuver with a random component of $\sigma_{t,ss} = 0.1g$. The maneuver start time for this study is $t_{evasive} = 3s$. The $\pm 3\sigma$ target evasive maneuver trajectory envelope is illustrated in Figure 9.2. The solid line represents the NRT, and the shaded area represents the $\pm 3\sigma$ truth dispersions, or the area where 99.7% of the trajectories are expected to be. This scenario represents a light saturation for some guidance laws, and a heavy saturation for others.

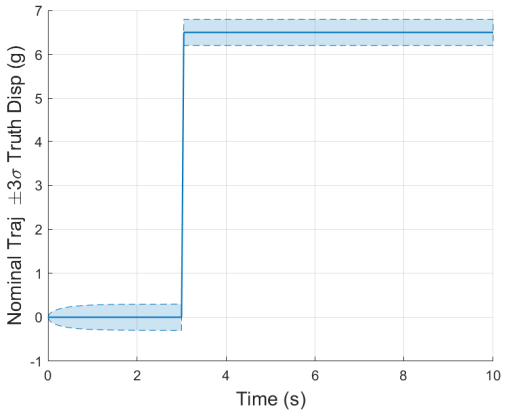

Fig. 9.2: Target Step Evasive Maneuver

The results for this example problem are shown in Figures 9.3 and 9.4. In Figure 9.3, the PN guidance law performance is shown with the color blue indicating a gain of 3, orange

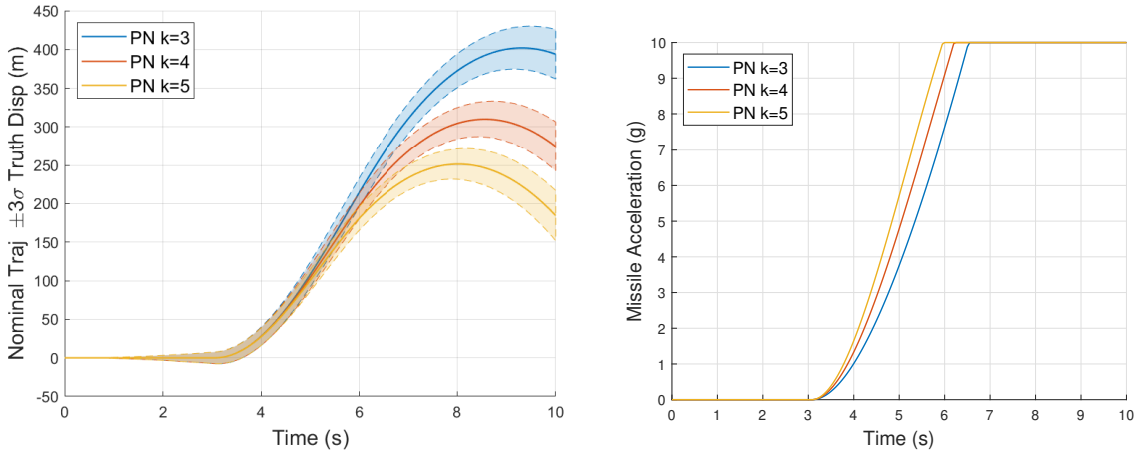

Fig. 9.3: PN Guidance Law Trajectories For A Target Step Evasive Maneuver: (left) Nominal Trajectory $\pm 3\sigma$ Position Dispersion, (right) Nominal Missile Acceleration

a gain of 4, and yellow a gain of 5. For the left side of Figure 9.3, the solid line represents the NRT, and the shaded area represents the $\pm 3\sigma$ truth dispersions for the corresponding line. For example, with PN $k = 3$ guidance, the relative lateral position at $t = 10s$ has a mean value of 394m $\pm$ 32m. The right side of Figure 9.3 contains the corresponding nominal acceleration profiles for each guidance law. In general, all three PN guidance laws saturate around $t = 6s$. The $k = 5$ trajectory is a little more responsive early in the engagement, and is only able to drive the final miss distance down to about 175m. The other two PN guidance laws have even higher miss distances. The miss distance results of this example, along with the final time 3σ truth dispersions, and the maximum realized missile acceleration values are summarized in Table 9.1.

Table 9.1: Guidance Law Miss Distances For A Target Step Evasive Maneuver

Guidance Law	Miss Distance (m)	3σ Truth Dispersion (m)	Max Missile Accel. (g)
PN $k = 3$	394	32	10
PN $k = 4$	274	32	10
PN $k = 5$	184	33	10
APN $k = 3$	15.0	2.6	9.8
APN $k = 4$	3.3	1.2	10
APN $k = 5$	0.4	1.8	10
ON	3	0.25	8.9

The APN and ON guidance law performances are shown in Figure 9.4 with each color representing a different guidance scheme. These guidance laws perform much better than

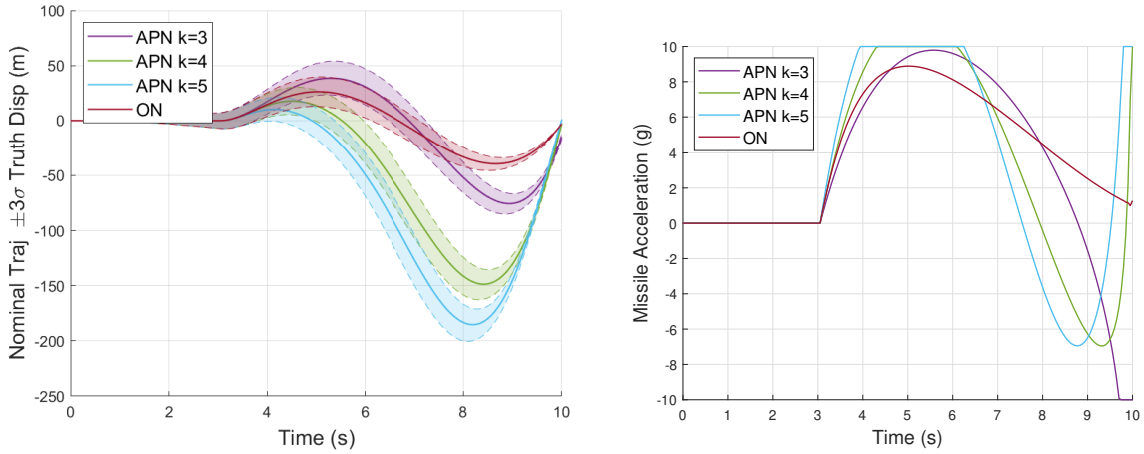

Fig. 9.4: APN/ON Guidance Law Trajectories For A Target Step Evasive Maneuver: (left) Nominal Trajectory $\pm 3\sigma$ Position Dispersion, (right) Nominal Missile Acceleration

the PN guidance. The miss distance is under 20m for all cases, and under 5m for APN $k = 4, 5$ and ON guidance (see Table 9.1). The truth dispersions are very small for the ON guidance. The ON also requires the least amount of actuator acceleration, having a maximum acceleration of 8.9g for this intercept. Figure 9.4 also shows the nominal acceleration profile for these guidance laws. The APN $k = 4, 5$ are the most aggressive to respond to the evasive maneuver, but they are also more prone to overshoot. In general, the higher the gain, the more responsive the missile is, and the lower the miss distance. Additionally, the guidance laws with more feed-forward terms have significantly better general performance.

9.3 Evasive Maneuver Study

Now that the guidance laws have been defined, the next study of interest is to the evasive maneuver times. For the random evasive maneuver, there is no specific start time, so the total engagement length is varied. For example, in Chapter 7, the total engagement time (initial t_{go} value) was 10 seconds, but in this study, the total engagement time varies from 1-10s. The initial conditions are held constant for this study. For the target step evasive

maneuver, step evasive maneuver start time varies from 0-10s. All the other parameters of the study stay the same as the previous section.

Figure 9.5 contains the 3σ miss distances for the random maneuvers. Each color

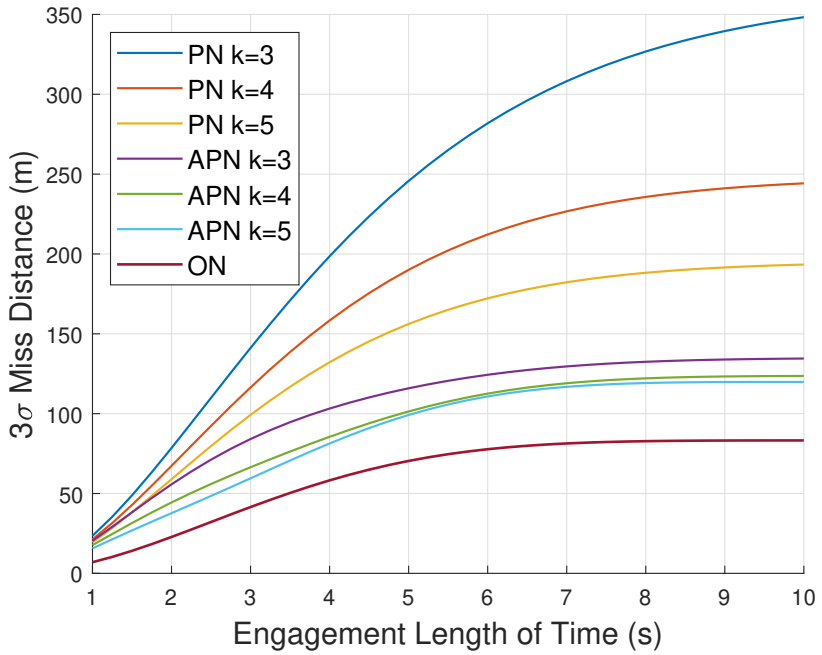

Fig. 9.5: Guidance Law Miss Distances Vs. Various Engagement Times For A Random Target Maneuver

represents a different guidance law tested. The x-axis of the plot indicates the total engagement time, and the y-axis the 3σ miss distance. The ON guidance performs best, followed by the APN guidance, and then the PN guidance. The higher gain guidance laws perform better than the lower gain ones. The miss distances in this scenario are somewhat high due to the high target acceleration.

The target step evasive maneuver scenario from the Section 9.2 is not quite as stressing as the random maneuver scenario. The evasive maneuver is 6.5g $\pm$0.3g. Using the PN guidance law, the missile almost always saturates. Using the APN guidance the missile saturate some of the time, and using the ON guidance law, the missile never saturates. Figure 9.6 shows the results for the PN guidance law. The color schemes are the same as for the previous section. The 3σ miss distances are all over 100m if the step maneuver

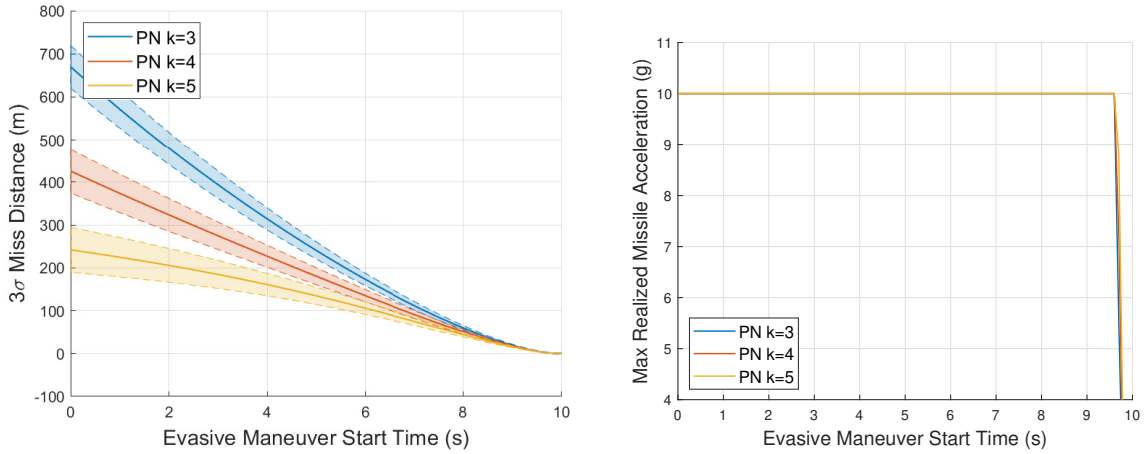

Fig. 9.6: PN Guidance Law Miss Distance as A Function Of Target Step Evasive Maneuver

begins with more than 4 seconds left in the engagement, and grow larger from there. This shows that the PN guidance laws can't keep up with this evasive maneuver. In general, PN guidance laws do not perform well against step maneuvers, as the PN guidance law is derived for a constant target velocity [57].

Figure 9.7 shows the results for the APN and ON guidance laws. The 3σ miss distances

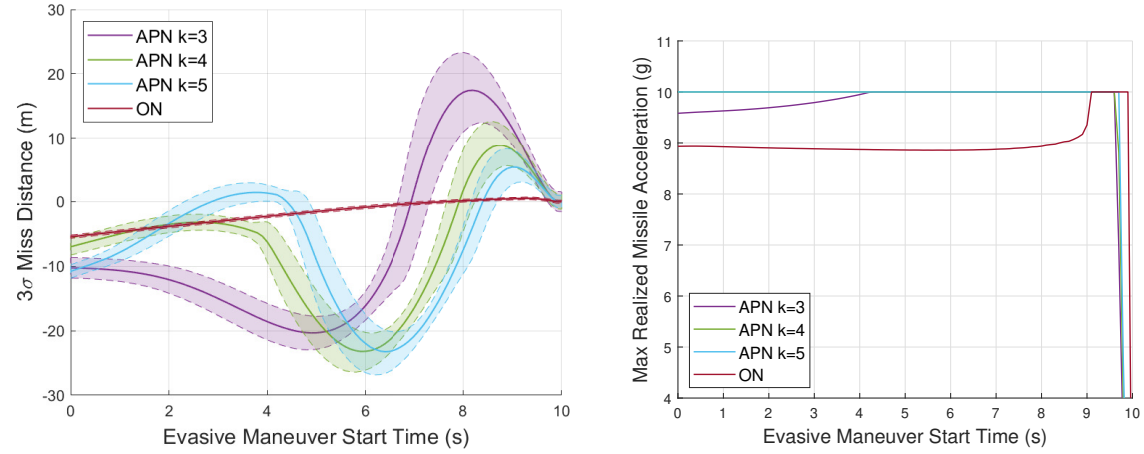

Fig. 9.7: APN/ON Guidance Law Miss Distance For A Time-Varying Target Step Evasive Maneuver

are under 25m for APN guidance, and under 5m for ON guidance. The largest miss distances

happen when the evasive maneuver begins about 8s into the scenario. The APN $k = 4, 5$ laws are very responsive and they saturate no matter when the maneuver begins. The ON guidance only saturates if the evasive maneuver begins after 9s, and even in this case, the miss distance remains small.

Since the ON guidance law doesn't saturate the actuator with $a_{t,evasive} = 6.5g$, a study is now conducted to investigate the performance of the guidance law as the target acceleration increases. Figure 9.8 shows the results of this study. Each color of the plot

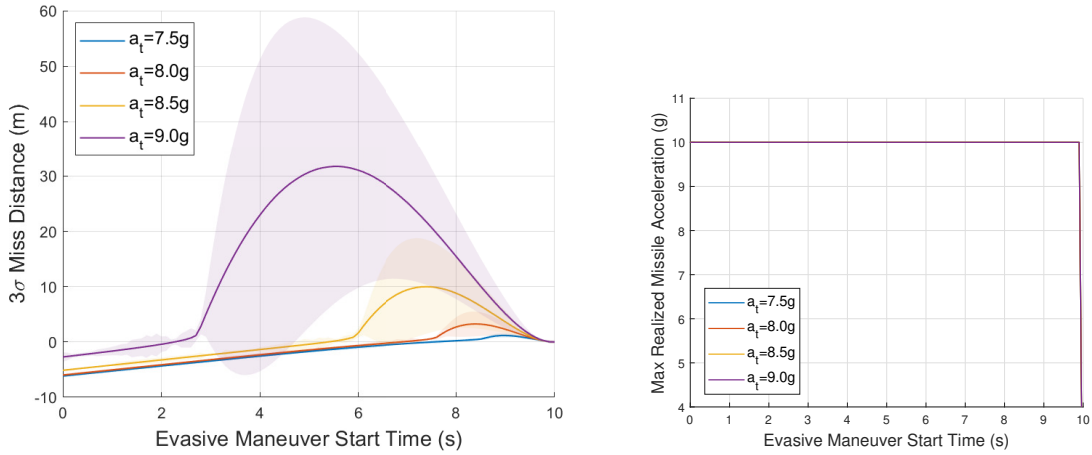

Fig. 9.8: ON Guidance Law Miss Distance Vs. Maneuver Start Time For Various Target Accelerations

represents a different target acceleration magnitude. The solid lines represent the nominal trajectory miss distance, while the shaded areas represent the 3σ envelope of miss distances. The target acceleration values vary from $a_{t,evasive} = 7.5g$ to $a_{t,evasive} = 9g$. Note that the missile maximum acceleration is 10g, so if the target evasive maneuver exceeds 10g, the missile will not be able to close any distance. The right side of Figure 9.8 shows that the ON guidance reaches saturation for any target maneuvers above 7.5g. For target accelerations of 8g and 8.5g, the ON guidance law starts seeing stronger saturations with maneuvers beginning after $t = 5s$. For target accelerations of 9g, the stronger saturations occur with maneuvers beginning after $t = 3s$.

The performance of the ON guidance law is very strong for all the cases studied so far. It is important to note however, that the sensors used so far have been very accurate. Not every system has accurate sensors, therefore a study to investigate the effect of sensor quality on system performance will be done in Section 9.5.

9.4 Actuator Responsiveness Study

Another variable that can significantly affect the outcome of an engagement is the responsiveness of the missile actuator. If there is a lag in implementing the acceleration commands, it can decrease the performance of the missile. This will be tested by varying the actuator time-constant parameter τ_m. So far in the chapter, this parameter has been constant at $\tau_m = 1s$. For this study, the time-constant τ_m will vary from a fast response of 0.25 seconds, to a slow response of 4.0 seconds. This actuator response will be tested against the random target maneuver and the step evasive maneuver from Section 9.3.

The results of the actuator responsiveness study for a random target maneuver are illustrated in Figure 9.9. The line colors are associated with the same guidance laws as the

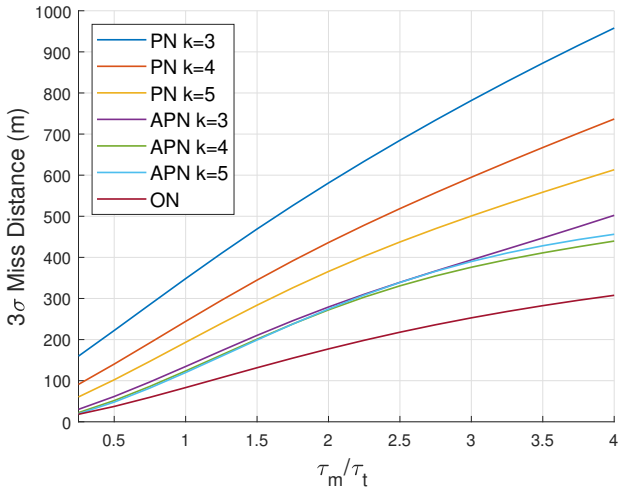

Fig. 9.9: Miss Distance Vs. Time-Constant Ratio For Random Target Maneuver

previous section. The results along the x-axis are normalized against the target maneuver time-constant τ_t. The y-axis shows the 3σ miss distance of the engagement for each time-

constant ratio. For a very responsive actuator, the APN guidance and the ON guidance all perform well, with 3σ miss distances under 25m. This is significant, as the steady state 3σ the target acceleration of 15g is higher than the maximum acceleration capability of missile, 10g. This shows that a missile with an acceleration disadvantage can still perform well against a target maneuver, as long as it is has a faster actuator time-constant and an advanced guidance law. In general, the 3σ miss distance decreases by approximately a factor of 3 as the $\frac{\tau_m}{\tau_t}$ ratio decreases from 1 to 0.25.

For part two of the actuator responsiveness study, the target evasive maneuver from Section 9.3 is considered, with the maneuver starting time ranging from 0-10s. The time-constant ratios vary from $\frac{\tau_m}{\tau_t} = 0.25 - 4$. The results of this study are illustrated in Figures 9.10-9.12. The left-side plots show the miss distances for the evasive maneuver start times. Each color represents a different time-constant ratio $\frac{\tau_m}{\tau_t}$, with the solid line designating the nominal miss distance, and the shaded area designating the 3σ truth dispersion trajectory envelope. The right-side plots show the maximum missile acceleration required for each trajectory. For Figures 9.10-9.12, there are multiple rows of plots, with each row indicating a different guidance law. The guidance law is written in bold at the top of each plot.

Figure 9.10 shows the PN guidance law performance. All the trajectories saturate the actuator, no matter when they start, and the earlier the maneuver, the larger the miss distance. The faster response actuators with ratios $\frac{\tau_m}{\tau_t} = 0.25$ and $\frac{\tau_m}{\tau_t} = 0.5$ show significant improvement for the PN guidance laws.

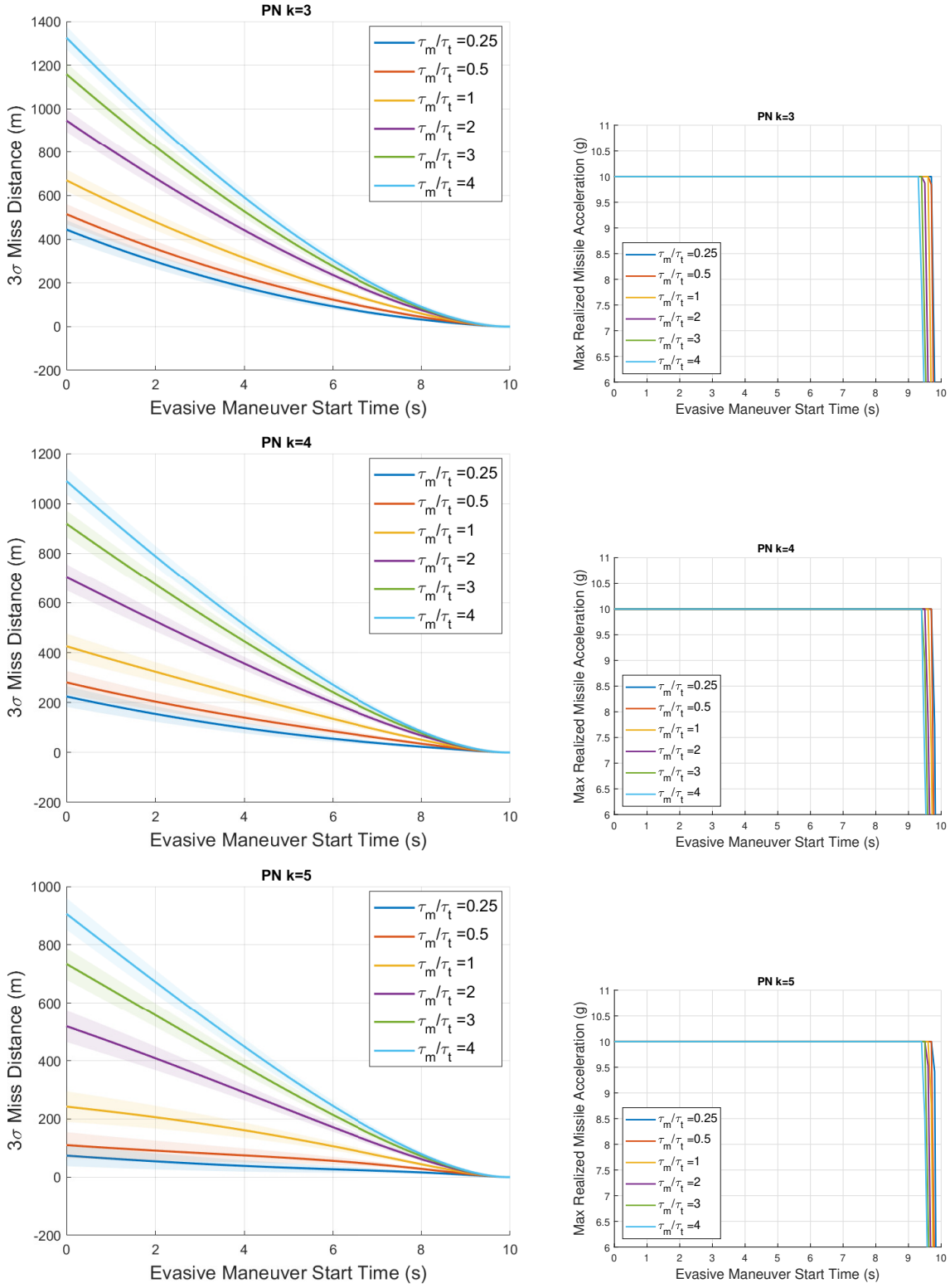

Fig. 9.10: PN Guidance Law Performance For Various Actuator Time-Constants

Figure 9.11 shows the APN guidance law performance. The higher gain APN guidance laws saturate more easily than the $k = 3$ guidance does. For long engagements, the $k = 3, 4$ guidance laws start to significantly decrease in performance for $\frac{\tau_m}{\tau_t}$ ratios above 1.5. For $\frac{\tau_m}{\tau_t}$ ratios of 0.5 and 0.25, the APN guidance law still has miss distances below 25m. This shows that a missile with an acceleration disadvantage can still perform well against a target step maneuver, as long as it is has a faster actuator time-constant and an advanced guidance law. For $k = 3$, the largest miss distances happen for early maneuver times and slower actuators. In general, with the slower actuators, there seems to be an oscillating behavior, almost like an actuator windup/overshoot cycle, due to the acceleration lag.

The ON guidance law performance is shown in Figure 9.12. This law performs exceedingly well, even with slow actuators. The acceleration plot in Figure 9.12 shows that the missile actuator is not saturating much for this guidance law. All the 3σ miss distances with $\frac{\tau_m}{\tau_t} = 0.25 - 4$ stay below 10m.

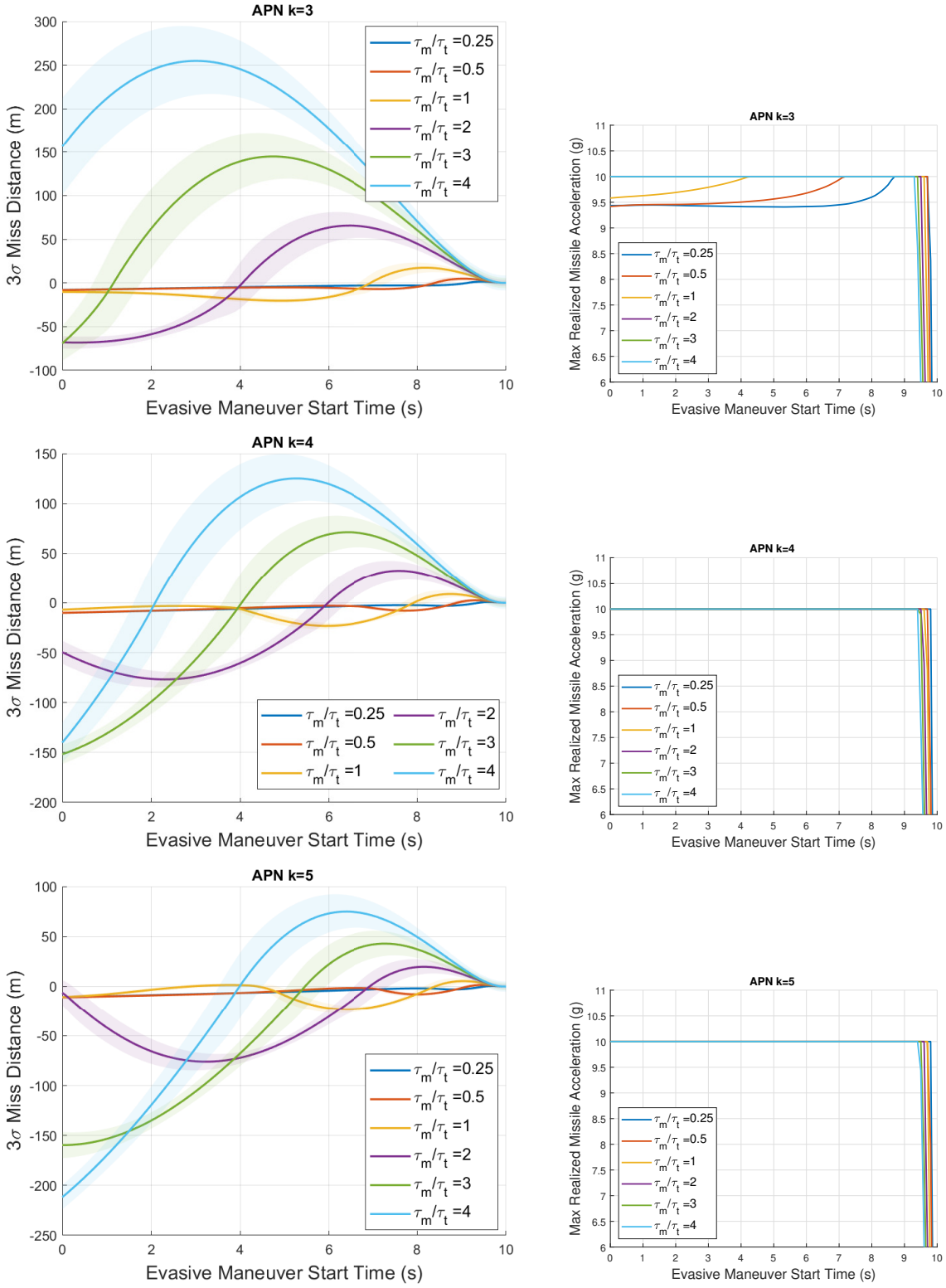

Fig. 9.11: APN Guidance Law Performance For Various Actuator Time-Constants

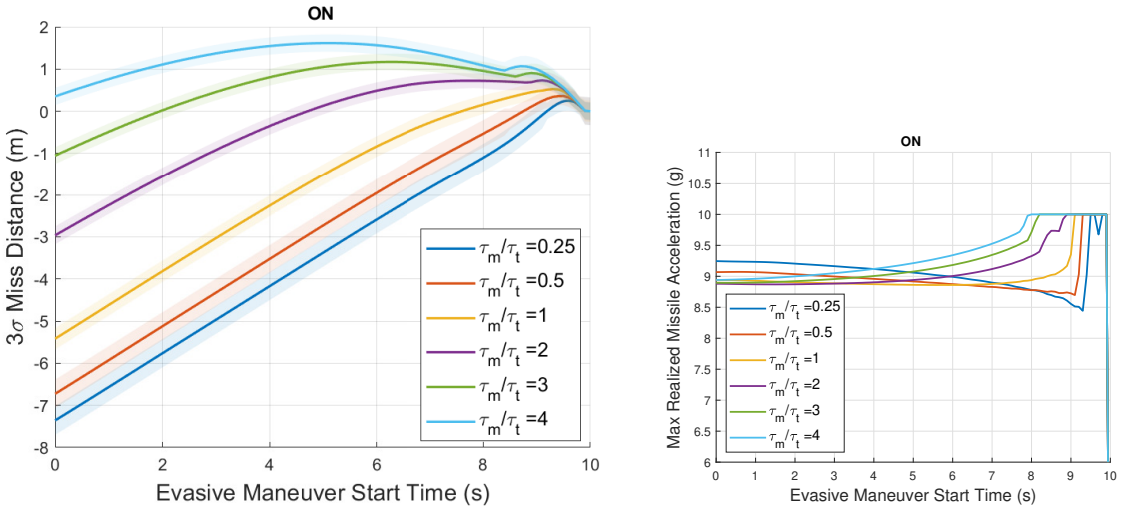

Fig. 9.12: ON Guidance Law Performance For Various Actuator Time-Constants

9.5 Sensor Quality Study

The studies so far in this chapter have focused on engagements with very accurate sensors. In general, missiles use very accurate tactical grade sensors, as performance is more important than cost. However, Statistical LinCov is the perfect tool run a trade study and investigate what type of performance drop off there is with lower grade sensors. For this section, there are be 3 levels of sensors considered. The "high" grade sensors are tactical grade sensors for military applications, the "low" grade sensors are the cheaper "common off the shelf" (COTS) sensors, and the "medium" grade sensors are somewhere in between. The sensor specifications for each level are found in Table 9.2. The accelerometer

Table 9.2: Sensor Specifications: High, Medium, And Low Accuracy

Symbol	High	Medium	Low	Units	Description
S_η	0.015	0.17	0.60	(m/s)/sqrt(hr)	3σ aclmtr. random walk
ν_{los}	0.2	2.0	10	mrad	3σ line-of-sight meas. noise std dev
$\sigma_{a,ss}$	1.2	2.1	42	mg	3σ steady state aclmtr. bias
$\sigma_{s,ss}$	300	750	5400	ppm	3σ steady state aclmtr. scale factor
$\sigma_{\epsilon,ss}$	0.4	4	40	mrad	3σ steady st. aclmtr. misalignment
$\sigma_{los,ss}$	0.05	0.5	2.5	mrad	3σ steady state line-of-sight bias

specifications for random walk, bias repeatability, and scale factor are from an Inertial Labs IMU spec sheet [100], and from two Collins Aerospace IMU spec sheets [101, 102]. The angle measurement specifications are from [103], and the accelerometer misalignment specifications are from [36].

This study is similar to the random maneuver study from Section 9.2. The target undergoes a zero mean random evasive maneuver with time-constant τ_t, and $\sigma_{t,ss} = 5g$. This maneuver creates a moderate actuator saturation for the missile. The simulation parameters for this study are found in Tables 7.1 and 7.2, with the specified sensor changes, and with a varying maximum missile acceleration capability.

Figure 9.13 shows the results of the sensor study for each of the guidance laws. The plots show the 3σ miss distances for varying levels of $\frac{a_{max}}{a_{t,max}}$. Each plot represents a different

guidance law, and each color represents a different sensor configuration. The lower $\frac{a_{max}}{a_{t,max}}$ ratios represent higher saturations. PN guidance is not affected much by the lower grade sensors, but for APN and ON guidance, the sensor grade has a pronounced effect. For high saturation APN/ON guidance, when $\frac{a_{max}}{a_{t,max}} < 1$, the 3σ miss distance jumps from about 100m with high quality sensors to 600m with low quality sensors. For lower saturation scenarios, the ON guidance benefits the most from high accuracy sensors.

Another way of arranging the data is to sort the results by sensor quality instead of by guidance law. This is shown in Figure 9.14. The x-axes and y-axes for these plots are the same as Figure 9.13, with one plot for each sensor grade. Each color represents a different guidance law, with the PN guidance laws shown as dashed lines. In these plots, it is clear that for higher actuator saturation and low accuracy sensors, the performance of the more advanced guidance laws is worse than PN guidance. For lower saturation scenarios, when $\frac{a_{max}}{a_{t,max}} > 1$, ON guidance still performs the best, no matter what the sensor grade is.

The Figure 9.14 plots are combined into one plot in Figure 9.15 to show more detail on the lower saturation region. This plot shows PN $k = 4$ in yellow, APN $k = 4$ in orange, and ON guidance in blue, and three sensor grades for each guidance law. This plot shows that as $\frac{a_{max}}{a_{t,max}}$ approaches 2, PN guidance approaches the performance of APN guidance, with ON guidance outperforming them both.

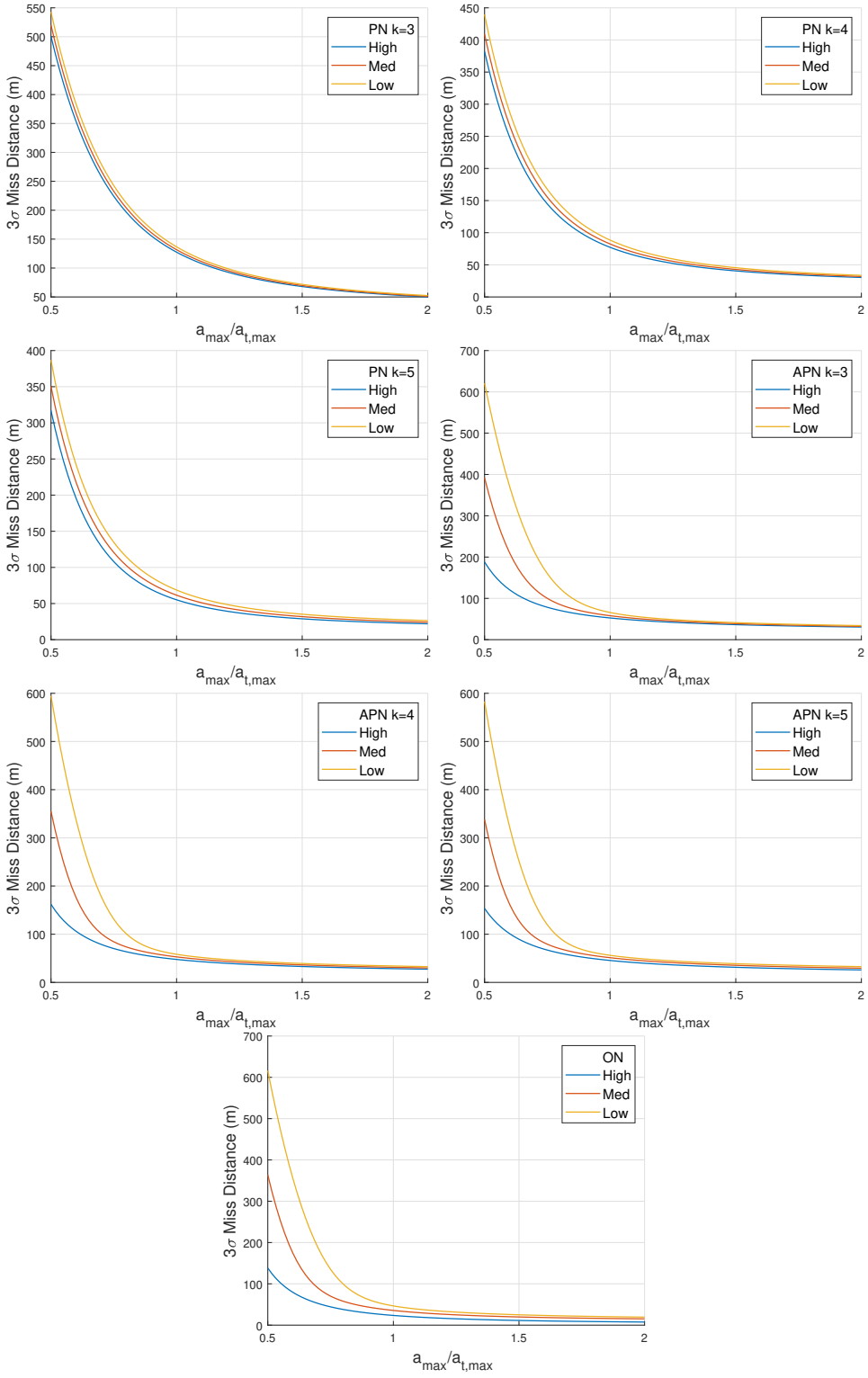

Fig. 9.13: Guidance Law Performance As A Function Of Sensor Quality And Missile/Target Acceleration

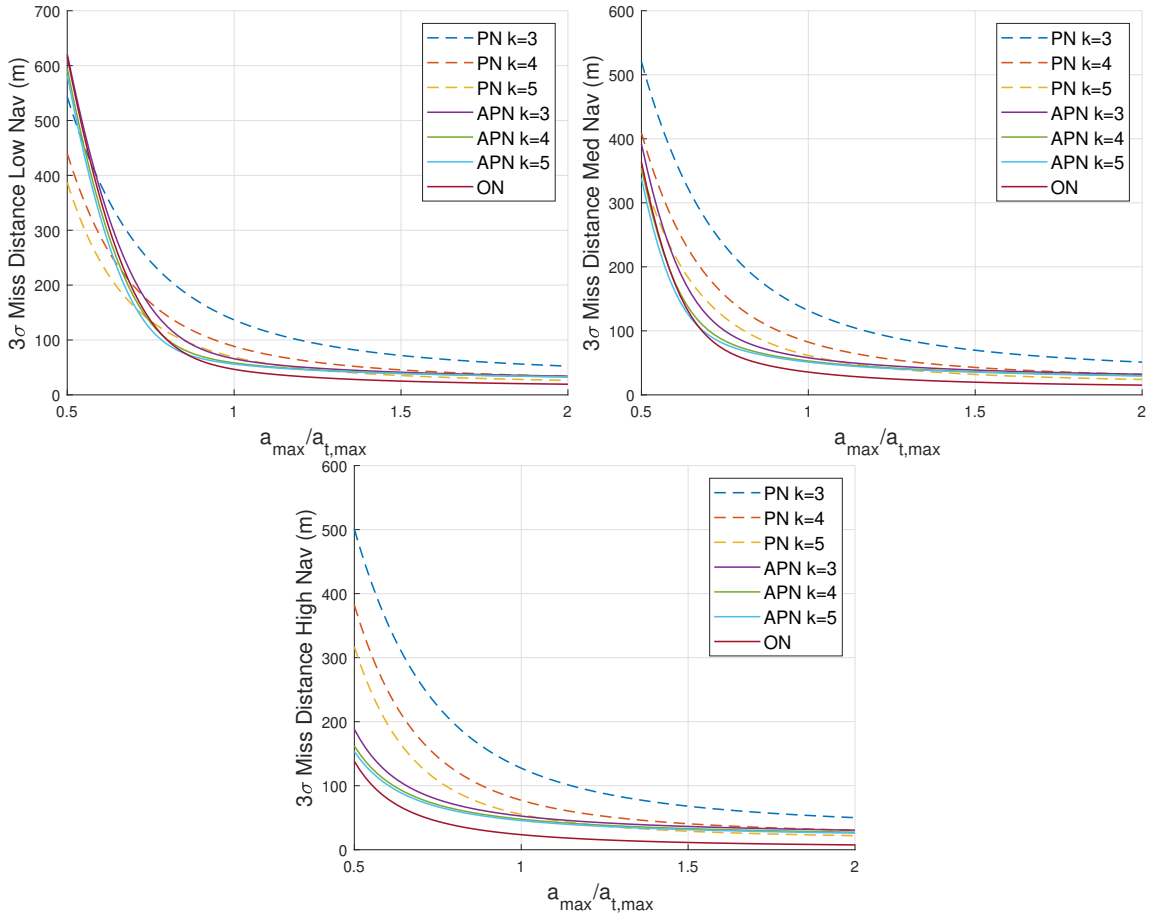

Fig. 9.14: Guidance Law Performance As A Function Of Sensor Quality And Missile/Target Acceleration Part 2

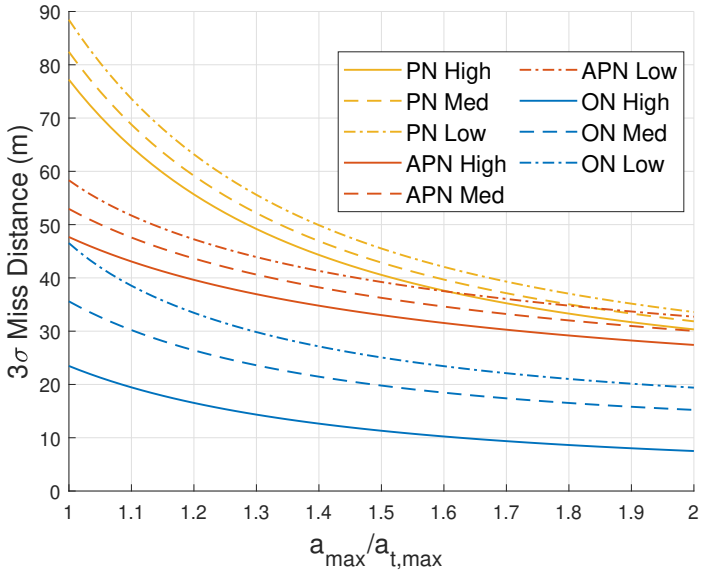

Fig. 9.15: Guidance Law Performance As A Function Of Sensor Quality And Missile/Target Acceleration Part 3

9.6 Trade Study Summary

In this chapter, Statistical LinCov was utilized to test a few common guidance laws against random target maneuvers and target evasive step maneuvers. The trade study investigated many parameter variations, such as engagement time, saturation levels, actuator responsiveness, and sensor grade. The Statistical LinCov simulation made the testing very straightforward and fast. The simulations from this chapter take about 40 minutes to run, while a Monte Carlo simulation running the same studies would take 200+ hours. This is a significant upgrade in computation time.

The trade studies found that PN guidance is inferior in most cases to APN and ON guidance. The exception to this is in high saturation engagements with low accuracy sensors. PN guidance performs better against random target maneuvers than step maneuvers, and if the actuator has a faster response time than the target maneuvers, can still maintain small miss distances.

APN guidance was shown to perform well against target step maneuvers and random maneuvers, and outperformed PN guidance in most cases. For low saturation engagements, the APN and PN guidance laws have similar performance. APN guidance has a significant drop off in performance if the actuator time-constant is slower than the target maneuver time-constant.

ON guidance proved to be the best performing guidance law in most cases. The most striking improvement over PN and APN guidance occurs in engagements with target step maneuvers and slow actuator time-constants. The ON guidance was able to keep miss distances under 10m for time-constant ratios up to $\frac{\tau_m}{\tau_t} = 4$, whereas APN guidance for the same case had miss distances over 150m, and PN guidance had miss distances over 200m. In general, ON does not saturate as easily as PN and APN guidance. For many of the scenarios is this chapter, PN and APN guidance laws showed high levels of saturation, while the ON guidance hardly saturated at all. The study also showed that in most cases ON guidance with low accuracy sensors outperforms APN and PN guidance with high accuracy sensors.

CHAPTER 10
SUMMARY AND CONCLUSIONS

This dissertation examined the customary methods of GN&C linear covariance analysis for the missile engagement problem. Missile engagements have traditionally been addressed by lower fidelity covariance analysis techniques such as basic linear covariance methods or basic statistical linear covariance methods. In order to bring a higher fidelity linear analysis to missile engagements, this dissertation investigated the application of augmented state linear covariance analysis to missile engagements. Augmented state linear covariance theory historically has relied on a 1st-order Taylor series approximation linearization technique, which is not very accurate for certain nonlinearities, such as saturation limiters. Missile flight dynamics frequently involve actuator saturation, thus eliminating the usefulness of traditional linearization for this problem. Monte Carlo analysis is also often used in missile engagement problems, but it is prohibitively expensive computationally for many studies.

Chapter 1 provided an introduction to the missile engagement phases of flight, and current analysis methods. Terminal homing, the final phase of flight for a missile engagement, was the main focus of this dissertation. The second chapter gives an overview of the research and defines the thesis statement. The literature survey in Chapter 3 discussed the different types of covariance analysis and an alternative linearization technique called statistical linearization. Statistical linearization is better suited to problems with sustained actuator saturation than traditional linearization methods. Chapter 3 categorized the most common linear covariance analysis methods into the following categories: Basic LinCov, Basic Statistical LinCov, and Augmented State LinCov (referred to as just "LinCov").

Chapters 4 and 5, developed in detail all the LinCov theory and statistical linearization theory needed to incorporate LinCov into missile engagements. This included defining the augmented state linear covariance framework, deriving the event trigger equations, and deriving the statistical linearization equations for the saturation nonlinearity. Chapter 6 applied LinCov to an unsaturated missile engagement problem, a new GN&C field for

LinCov. This chapter was recently published by the author in a journal paper [41], and laid the groundwork for Chapters 7 and 8.

Chapter 7 developed and validated a new application of LinCov called Statistical LinCov, which can incorporate nonlinearities. Basic Statistical LinCov and Basic LinCov do not incorporate the effect of navigation errors on truth state dispersions, and are not able to evaluate the overall GN&C closed-loop performance. Augmented State LinCov does incorporate this effect, but is limited by traditional linearization. Statistical LinCov bridges the gap by including the correlation effects of navigation errors on truth dispersions, while also utilizing statistical linearization. This is a very useful application of LinCov which expands its use to a wider range of GN&C systems.

Part 2 of Chapter 7 defined and validated another application of LinCov called Tunable LinCov. Tunable LinCov uses Augmented State LinCov theory and traditional linearization, but incorporates a nonlinearity by tuning the timing of GN&C mode changes. Between Statistical LinCov, and Tunable LinCov, Statistical LinCov was more accurate for modeling missile actuator saturation.

A third type of LinCov analysis was defined and validated in Chapter 8, called Trigger LinCov. This is Augmented State LinCov with an event trigger to incorporate a mode change or nonlinearity. The initial investigation into Trigger LinCov showed that it is capable of modeling step input evasive target maneuvers, but not a random evasive maneuver.

In order to demonstrate the effectiveness of Statistical LinCov, Chapter 9 presented a series of missile engagement trade studies. The studies investigated various target maneuvers, missile guidance laws, saturation levels, actuator responsiveness levels, and sensor suites. The trade studies found that proportional navigation (PN) guidance is inferior in most cases to augmented proportional navigation (APN) guidance and optimal navigation (ON) guidance, with an exception being a high saturation engagement with low accuracy sensors. APN guidance was shown to perform well against target step maneuvers and target ECRV maneuvers, but had a significant drop off in performance for any sort of actuator lag. This is the situation where the ON guidance shined the most. With ON guidance, the missile

engagement miss distances were mostly unaffected by actuator lag, even significant ones. In general, the ON guidance is the best performing of the three guidance laws in just about every scenario. The Statistical LinCov simulation made the testing very straightforward and fast. The simulations from this chapter took about 40 minutes to run, while a Monte Carlo simulation running the same studies would take 200+ hours.

CHAPTER 11
FUTURE WORK

This dissertation has laid out the theory and equations for Statistical LinCov, and tested it out in a few example problems. In general, it can be just as useful to know the limitations of a research tool as it is to know the good use cases. The limitations of Statistical LinCov could be found by relaxing some of the assumptions of the problem, such as the Gaussian input assumption. The Gaussian assumption used in this dissertation is not particularly limiting (see Chapter 1 of [73]), but one way to validate this assertion would be to test out non-Gaussian inputs to the nonlinear functions. A further validation of Statistical LinCov could also be done by implementing it in a full 6DOF missile engagement simulation.

Another way to test out the limitations of Statistical LinCov is to apply it to other GN&C problems. There are numerous applications of statistical linearization that have future research potential. Utilizing Statistical LinCov, any nonlinearity from Appendix E of [76] can be applied to LinCov, including limiters, relays, and deadbands. If it is desired to use Statistical LinCov for a nonlinear function that has not been previously tabulated, then the random input describing function (RIDF) would have to be derived.

One potential research application is attitude deadband control for satellites. Instead of continuously executing attitude corrections, it is common for satellites to wait until the attitude error meets the error threshold before correcting. This prevents the momentum wheels from running more than they need to. This type of attitude maneuver would be easily modeled by Statistical Linearization. Another deadband application would involve position control deadbands for satellite rendezvous and proximity operations (RPO). If a satellite wanted to stay on the v-bar within a certain distance of a resident space object (RSO), a deadband controller could activate once the estimated range reaches the threshold. Another nonlinearity that could be addressed by Statistical LinCov is minimum impulse control. For satellites flying in formation, it is important to run frequent station keeping/orbit maintenance maneuvers. However, satellite thrusters have a minimum impulse acceleration

they can impart, preventing the execution of very small accelerations. Statistical LinCov would be able to model this minimum impulse control effect.

Statistical LinCov also has potential application for nonlinear measurement equations. It could be applied to quantized measurements, or to measurements with nonlinear trigonometric functions. For example, with angles-only navigation between two satellites, the angle measurement equation is nonlinear and traditional linearization methods are not very accurate. This would require a derivation of the RIDFs for the measurement, which can be quite complex. Developing an Extended Kalman Filter that is capable of using statistical linearization for this measurement could dramatically improve the performance of angles-only navigation for satellites.

REFERENCES

[1] M. Weiss, "Adjoint Method for Missile Performance Analysis on State-Space Models," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 2, pp. 236–248, Mar. 2005.

[2] A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, and A. A. Sutherland, *Applied optimal estimation.* Cambridge, Mass.: MIT Press, 1974, oCLC: 254169501.

[3] J. Crassidis and J. Junkins, *Optimal Estimation of Dynamic Systems*, 2nd ed. New York: CRC Press, 2012.

[4] A. Gelb and R. S. Warren, "Direct Statistical Analysis of Nonlinear Systems: CADET," *AIAA Journal*, vol. 11, no. 5, pp. 689–693, May 1973.

[5] P. Zarchan, "Comparison of Statistical Digital Simulation Methods," AGARD, Tech. Rep. 273, Jul. 1988.

[6] ——, "Complete Statistical Analysis of Nonlinear Missile Guidance Systems - SLAM," *Journal of Guidance, Control, and Dynamics*, vol. 2, no. 1, pp. 71–78, Jan. 1979. [Online]. Available: http://arc.aiaa.org/doi/10.2514/3.55834

[7] R. S. Christensen and D. K. Geller, "Linear covariance techniques for closed-loop guidance navigation and control system design and analysis," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 228, no. 1, pp. 44–65, Jan. 2014. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0954410012467717

[8] P. Zarchan, *Tactical and Strategic Missile Guidance*, 6th ed. American Institute of Aeronautics and Astronautics, 2012.

[9] C. Price and R. Warren, "PERFORMANCE EVALUATION OF HOMING GUIDANCE LAWS FOR TACTICAL MISSILES," Analytic Sciences Corporation, Tech. Rep. TR-170-4, Jan. 1973.

[10] D. K. Geller, "Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, p. 11, Nov. 2006.

[11] D. Stallard, "Near-Optimal Stochastic Terminal Controllers," Ph.D. dissertation, Massachussets Institute of Technology, 1971.

[12] S. Wang, W. Wang, and S. Xiong, "Impact angle constrained three-dimensional integrated guidance and control for STT missile in the presence of input saturation," *ISA Transactions*, vol. 64, pp. 151–160, Sep. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0019057816300726

[13] A. Arrow, "Status and concerns for bank-to-turn control of tactical missiles," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 2, pp. 267–274, Mar. 1985. [Online]. Available: http://arc.aiaa.org/doi/10.2514/3.19970

[14] Y. Ho, A. Bryson, and S. Baron, "Differential games and optimal pursuit-evasion strategies," *IEEE Transactions on Automatic Control*, vol. 10, no. 4, pp. 385–389, Oct. 1965. [Online]. Available: http://ieeexplore.ieee.org/document/1098197/

[15] A. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. New York, NY: Taylor & Grancis Groub, 1975.

[16] N. A. Shneydor, *Missile guidance and pursuit: kinematics, dynamics and control*. Cambridge, UK: Woodhead Publishing, 1998.

[17] D. Geller, "Linear Covariance Techniques for Orbital Rendezvous Analysis and Autonomous Onboard Mission Planning," in *AIAA Guidance, Navigation, and Control Conference*. San Francisco, California: AIAA, Aug. 2005.

[18] M. S. Grewal and A. P. Andrews, "Applications of Kalman Filtering in Aerospace 1960 to the Present [Historical Perspectives]," *IEEE Control Systems*, vol. 30, no. 3, pp. 69–78, Jun. 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5466132/

[19] G. L. Smith, S. F. Schmidt, and L. A. McGee, "APPLICATION OF STATISTICAL FILTER THEORY TO THE OPTIMAL ESTIMATION OF POSITION AND VELOCITY ON BOARD A CIRCUMLUNAR VEHICLE," NASA Ames Research Center, Moffett Field, California, Technical Report R-135, 1962. [Online]. Available: https://ntrs.nasa.gov/citations/20190002215

[20] N. Wiener, *The Extrapolation, Interpretation and Smoothing of Stationery Time Series*. Cambridge, Mass: MIT Press, 1964, vol. 8.

[21] E. Purcell and T. Barbery, "Dispersions resulting from atmospheric variations," *Journal of Spacecraft and Rockets*, vol. 5, no. 8, Aug. 1968.

[22] P. Maybeck, *Stochastic Models, Estimation, and Control Volume 1*. Academic Press, 1979.

[23] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[24] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, Mar. 1961. [Online]. Available: https://asmedigitalcollection.asme.org/fluidsengineering/article/83/1/95/426820/New-Results-in-Linear-Filtering-and-Prediction

[25] S. F. Schmidt, "The Kalman filter - Its recognition and development for aerospace applications," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, 1981.

[26] J. R. Carpenter and C. N. D'Souza, "Navigation Filter Best Practices," *NASA Technical Report*, p. 149, 2018.

[27] J. Stauch and M. Jah, "Unscented Schmidt–Kalman Filter Algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 1, pp. 117–123, Jan. 2015. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.G000467

[28] B. D. Tapley, B. E. Schutz, and G. H. Born, *Statistical orbit determination.* Amsterdam ; Boston: Elsevier Academic Press, 2004.

[29] I. Y. Bar-Itzhack and R. R. Harmann, "Covariance Simulation of the EUVE Update Filter," in *Flight Mechanics/Estimation Theory Symposium*, NASA Goddard Space Flight Center, Greenbelt, MD, 1989, pp. 225–237.

[30] R. H. Battin, *An Introduction to the Mathmatics and Methods of Astrodynamics.* New York: AIAA, 1999.

[31] D. C. Woffinden, "Angles-Only Navigation for Autonomous Orbital Rendezvous," PhD Dissertation, Utah State University, Logan, Utah, 2008.

[32] R. Zanetti, "Autonomous Midcourse Navigation for Lunar Return," *Journal of Spacecraft and Rockets*, vol. 46, no. 4, pp. 865–873, Jul. 2009.

[33] N. B. Stastny and D. K. Geller, "Autonomous Optical Navigation at Jupiter: A Linear Covariance Analysis," *Journal of Spacecraft and Rockets*, vol. 45, no. 2, pp. 290–298, Mar. 2008.

[34] R. Christensen, D. K. Geller, and M. Hansen, "Linear Covariance Navigation Analysis of Range and Image Measurement Processing for Autonomous Lunar Lander Missions," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS).* Portland, OR, USA: IEEE, Apr. 2020, pp. 1524–1535. [Online]. Available: https://ieeexplore.ieee.org/document/9109838/

[35] D. K. Geller and D. P. Christensen, "Linear Covariance Analysis for Powered Lunar Descent and Landing," *Journal of Spacecraft and Rockets*, vol. 46, no. 6, pp. 1231–1247, Nov. 2009.

[36] M. B. Rose, "Statistical Methods for Launch Vehicle Guidance, Navigation, and Control (GN&C) System Design and Analysis," PhD Dissertation, Utah State University, Logan, Utah, Apr. 2012. [Online]. Available: https://digitalcommons.usu.edu/etd/1278

[37] R. Christensen, D. Geller, and M. Hansen, "LINEAR COVARIANCE NAVIGATION ANALYSIS FOR AUTONOMOUS LUNAR LANDER MISSIONS," *Rocky Mountain AAS Guidance, Navigation, and Control Conference Proceedings*, p. 13, 2020.

[38] K. Jin, D. K. Geller, and J. Luo, "Development and Validation of Linear Covariance Analysis Tool for Atmospheric Entry," *Journal of Spacecraft and Rockets*, vol. 56, no. 3, pp. 854–864, May 2019. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.A34297

[39] R. S. Christensen, "Linear Covariance Analysis For Gimbaled Pointing Systems," PhD Dissertation, Utah State University, Logan, Utah, Jul. 2013. [Online]. Available: https://digitalcommons.usu.edu/etd/1766

[40] J. S. Clawson and R. Christensen, "Linear Covariance Model Development for Target Engagements," in *AIAA Scitech 2021 Forum*. VIRTUAL EVENT: American Institute of Aeronautics and Astronautics, Jan. 2021. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2021-1464

[41] ——, "Linear Covariance Analysis for Head-On Target Engagements," *Journal of Spacecraft and Rockets*, pp. 1–12, Jan. 2022. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.A35178

[42] HAEUSSERMANN, W. (Bendix Corp., Huntsville, Ala.), "Developments in the field of automatic guidance and control of rockets," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 3, pp. 225–239, May 1981. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/3.19735

[43] M. W. Fossier, "The development of radar homing missiles," *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 6, pp. 641–651, Nov. 1984. [Online]. Available: https://arc.aiaa.org/doi/10.2514/3.19908

[44] J. A. Kaplan, A. R. Chappell, and J. W. McManus, "The Analysis of a Generic Air-to-Air Missile Simulation Model," NASA, Langley Research Center, Technical Memorandum 109057, 1994. [Online]. Available: https://www.cs.odu.edu/~mln/ltrs-pdfs/NASA-94-tm109057.pdf

[45] S. Bezick, I. Rusnak, and S. Gray, "Guidance of a Homing Missile via Nonlinear Geometric Control Methods," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 3, pp. 441–448, 1995.

[46] A. Von Moll, D. W. Casbeer, E. Garcia, and D. Milutinovic, "Pursuit-evasion of an Evader by Multiple Pursuers," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. Dallas, TX: IEEE, Jun. 2018, pp. 133–142. [Online]. Available: https://ieeexplore.ieee.org/document/8453470/

[47] V. R. Makkapati, W. Sun, and P. Tsiotras, "Optimal Evading Strategies for Two-Pursuer/One-Evader Problems," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 4, pp. 851–862, Apr. 2018. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.G003070

[48] E. Garcia, D. W. Casbeer, A. Von Moll, and M. Pachter, "Cooperative Two-Pursuer One-Evader Blocking Differential Game," in *2019 American Control Conference (ACC)*. Philadelphia, PA, USA: IEEE, Jul. 2019, pp. 2702–2709. [Online]. Available: https://ieeexplore.ieee.org/document/8814294/

[49] A. Y. Husodo, G. Jati, A. Octavian, and W. Jatmiko, "Enhanced Social Spider Optimization Algorithm for Increasing Performance of Multiple Pursuer Drones in Neutralizing Attacks From Multiple Evader Drones," *IEEE Access*, vol. 8, pp. 22 145–22 161, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8967069/

[50] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, "Basic Principles of Homing Guidance," *JOHNS HOPKINS APL TECHNICAL DIGEST*, vol. 29, no. 1, p. 17, 2010.

[51] V. H. L. Cheng and N. K. Gupta, "Advanced Midcourse Guidance for Air-to-Air Missiles," *Journal of Guidance, Control, and Dynamics*, vol. 9, no. 2, pp. 135–142, Mar. 1986.

[52] V. Turetsky and J. Shinar, "Missile guidance laws based on pursuit–evasion game formulations," *Automatica*, vol. 39, no. 4, pp. 607–618, Apr. 2003. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S000510980200273X

[53] S. Josef, "On the feasibility of "hit-to-kill" in the interception of maneuvering targets," in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*. Arlington, VA, USA: IEEE, 2001, pp. 3358–3363 vol.5. [Online]. Available: http://ieeexplore.ieee.org/document/946147/

[54] K. Mehrotra and P. Mahapatra, "A jerk model for tracking highly maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1094–1105, Oct. 1997. [Online]. Available: http://ieeexplore.ieee.org/document/624345/

[55] A. Sarkar and D. Ghose, "Realistic Pursuer Evader Engagement with Feedback Linearization Based Nonlinear Guidance Law," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Keystone, Colorado: American Institute of Aeronautics and Astronautics, Aug. 2006, pp. 1–12. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2006-6096

[56] M. E. Hough, "Reentry Maneuver Estimation Using Nonlinear Markov Acceleration Models," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1693–1710, Jul. 2017. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.G002577

[57] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd, "Modern Homing Missile Guidance Theory and Techniques," *JOHNS HOPKINS APL TECHNICAL DIGEST*, vol. 29, no. 1, p. 18, 2010.

[58] P. Zarchan, "A New Look at Classical vs Modern Homing Missile Guidance," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, pp. 78–85, 1981.

[59] V. Turetsky, "Capture zones of linear feedback pursuer strategies," *Automatica*, vol. 44, no. 2, pp. 560–566, Feb. 2008. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109807003391

[60] T. Shima and J. Shinar, "Time-Varying Linear Pursuit-Evasion Game Models with Bounded Controls," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 3, pp. 425–432, 2002.

[61] A. Sarkar, P. Tiwari, S. Srinivasan, R. Bhattacharjee, and D. Ghose, "Generalized PN Guidance Law for a Practical Pursuer Evader Engagement," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Austin, Texas: American Institute of Aeronautics and Astronautics, Aug. 2003, pp. 1–10. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2003-5651

[62] S. Ghosh, D. T. Davis, T. H. Chung, and O. A. Yakimenko, "Development and testing of the intercept primitives for planar UAV engagement," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. Miami, FL, USA: IEEE, Jun. 2017, pp. 286–293. [Online]. Available: http://ieeexplore.ieee.org/document/7991385/

[63] R. Beard, T. McLain, M. Goodrich, and E. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 911–922, Dec. 2002. [Online]. Available: http://ieeexplore.ieee.org/document/1159009/

[64] G. E. Collins, "An automatic UAV search, intercept, and follow algorithm for persistent surveillance," in *Proc. SPIE 7694, Ground/Air Multi-Sensor Interoperability, Integration, and Networking for Persistent ISR*, M. A. Kolodny, Ed., Orlando, Florida, Apr. 2010, p. 76940D. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.852236

[65] W. Chadwick, "Miss Distance of Proportional Navigation with Varying Velocity," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 662–666, 1985.

[66] J. Shinar and V. Turetsky, "Improved estimation is a prerequisite for successful terminal guidance," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 219, no. 2, pp. 145–156, Feb. 2005. [Online]. Available: http://journals.sagepub.com/doi/10.1243/095441005X9058

[67] A. Ratnoo and D. Ghose, "Impact Angle Constrained Guidance Against Nonstationary Nonmaneuvering Targets," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 1, pp. 269–275, Jan. 2010. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.45026

[68] S. Mondal and R. Padhi, "Selection of optimal time-to-go in generalized vector explicit guidance," in *2015 IEEE Conference on Control Applications (CCA)*. Sydney, Australia: IEEE, Sep. 2015, pp. 756–761. [Online]. Available: http://ieeexplore.ieee.org/document/7320708/

[69] I. Rusnak, H. Weiss, R. Eliav, and T. Shima, "Missile guidance with constrained terminal body angle," in *2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel*. Eilat, Israel: IEEE, Nov. 2010, pp. 000 045–000 049. [Online]. Available: http://ieeexplore.ieee.org/document/5662094/

[70] B. S. Kim, J. G. Lee, and H. S. Han, "Biased PNG law for impact with angular constraint," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 277–288, Jan. 1998. [Online]. Available: http://ieeexplore.ieee.org/document/640285/

[71] D. K. Geller, M. B. Rose, and D. C. Woffinden, "Event Triggers in Linear Covariance Analysis with Applications to Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 102–111, Jan. 2009. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.36834

[72] R. Zanetti, D. C. Woffinden, and A. Sievers, "Multiple Event Triggers in Linear Covariance Analysis for Spacecraft Rendezvous," *Journal of Guidance, Control,*

and Dynamics, vol. 35, no. 2, pp. 353–366, Mar. 2012. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.54965

[73] J. H. Taylor, "Handbook for the Direct Statistical Analysis of Missile Guidance Systems via CADET (Trademark) (Covariance Analysis Describing Function Technique)," ANALYTIC SCIENCES CORP READING MA, Tech. Rep., May 1975, section: Technical Reports. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA013397

[74] R. C. Booton, "Nonlinear control systems with random inputs," *IRE Transactions on Circuit Theory*, vol. 1, no. 1, pp. 9–18, Mar. 1954. [Online]. Available: https://ieeexplore.ieee.org/document/6373354/

[75] J. H. TAYLOR, "Random-input describing functions for multi-input non-linearities," *International Journal of Control*, vol. 23, no. 2, pp. 277–281, Feb. 1976, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207177608922159. [Online]. Available: https://doi.org/10.1080/00207177608922159

[76] A. Gelb and W. Vander Velde, *MULTIPLE-INPUT DESCRIBING FUNCTIONS AND NONLINEAR SYSTEM DESIGN*. McGraw-Hill, 1968.

[77] J. H. TAYLOR, "Direct Statistical Analysis of Missile Guidance Systems via CADET (Covariance Analysis Describing Function Technique)," Analytic Sciences Corporation, Tech. Rep. AD0783098, 1974.

[78] X. Wang and Hui Xie, "Simulation of covariance analysis describing equation technique (CADET) in missile hit probability calculation," in *2010 Sixth International Conference on Natural Computation*. Yantai, China: IEEE, Aug. 2010, pp. 4282–4285. [Online]. Available: http://ieeexplore.ieee.org/document/5583635/

[79] Y.-z. Luo and Z. Yang, "A review of uncertainty propagation in orbital mechanics," *Progress in Aerospace Sciences*, vol. 89, pp. 23–39, Feb. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0376042116301051

[80] B. G. Marchand, K. C. Howell, and R. S. Wilson, "Improved Corrections Process for Constrained Trajectory Design in the n-Body Problem," *Journal of Spacecraft and Rockets*, vol. 44, no. 4, pp. 884–897, Jul. 2007. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.27205

[81] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. Lake Louise, Alta., Canada: IEEE, 2000, pp. 153–158. [Online]. Available: http://ieeexplore.ieee.org/document/882463/

[82] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, Jul. 2010. [Online]. Available: https://ieeexplore.ieee.org/document/5546308/

[83] A. J. Bernard and D. K. Geller, "A Comparison of the Bootstrap Particle Filter and the Extended Kalman Filter and the Effects of Non-linear Measurements in Spacecraft Attitude Determination Using Light Curves," Utah State University, Tech. Rep., 2018.

[84] A. Papoulis, *Probability, random variables, and stochastic processes*. New York: McGraw-Hill, 1965.

[85] A. Gelb and R. Warren, "Direct statistical analysis of nonlinear systems - CADET," in *Guidance and Control Conference*. Stanford,CA,U.S.A.: American Institute of Aeronautics and Astronautics, Aug. 1972. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.1972-875

[86] J. Roberts and P. Spanos, *Random Vibration and Statistical Linearization*. Courier Corporation, 2003.

[87] A. Jazwinski, *Stochastic Theory and Filtering Theory*. New York: Academic Press, 1970.

[88] R. Singer, "Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970. [Online]. Available: http://ieeexplore.ieee.org/document/4103555/

[89] E. Lefferts, F. Markley, and M. Shuster, "Kalman Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 5, no. 5, pp. 417–429, Sep. 1982. [Online]. Available: http://arc.aiaa.org/doi/10.2514/3.56190

[90] M. E. Pittelkau, "Rotation Vector in Attitude Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 6, pp. 855–860, Nov. 2003. [Online]. Available: http://arc.aiaa.org/doi/10.2514/2.6929

[91] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso, "The complex-step derivative approximation," *ACM Transactions on Mathematical Software*, vol. 29, no. 3, pp. 245–262, Sep. 2003. [Online]. Available: http://portal.acm.org/citation.cfm?doid=838250.838251

[92] Y. Oshman, "An information fusion approach to missile guidance," in *2007 International Conference on Control, Automation and Systems*. Seoul, South Korea: IEEE, 2007, pp. xxxix–xliv. [Online]. Available: http://ieeexplore.ieee.org/document/4406500/

[93] E. H. Song, "A LOS Estimation Method for Bank-to-Turn Missiles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 4, pp. 1599–1608, Oct. 2008.

[94] M. Petovello, "Real-Time Integration of a Tactical-Grade IMU and GPS for High-Accuracy Positioning and Navigation," Dissertation, University of Calgary, Calgary, Alberta, Apr. 2003.

[95] "TITAN® Tactical Grade Inertial Measurement Unit (IMU) Spec Sheet," 2017. [Online]. Available: https://utcaerospacesystems.com/wp-content/uploads/2018/04/TITANR-Tactical-Grade-Inertial-Measurement-Unit-IMU.pdf

[96] M. E. Hough, "Acceleration Characterization for Reentry Orbit Determination with Unmodeled Maneuvers," *Journal of Guidance, Control, and Dynamics*, pp. 1–13, Mar. 2018. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.G003359

[97] H. Zhang, Y. Chang, D. Liu, and K. Ma, "An improved target tracking singer filter algorithm," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*. Dalian, China: IEEE, Oct. 2013, pp. 1070–1073. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6967288

[98] D. E. Williams, B. Friedland, and A. N. Madiwale, "Modern control theory for design of autopilots for bank-to-turn missiles," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 378–386, Jul. 1987. [Online]. Available: http://arc.aiaa.org/doi/10.2514/3.20228

[99] F. B. Hildebrand, *Methods of Applied Mathematics*, 2nd ed. New Jersey: Prentice-Hall, 1965.

[100] "Inertial Labs High Performance Fiber-Optic Gyroscope IMU Datasheet," 2023. [Online]. Available: https://www.inertiallabs.com/wp-content/uploads/2022/11/IMU-FI-200C_Datasheet.rev1_.1_Nov16_2022.pdf

[101] "SiIMU02 MEMS inertial measurement unit (IMU) Spec Sheet," 2023. [Online]. Available: https://prd-sc101-cdn.rtx.com/-/media/ca/product-assets/marketing/s/silmu02/siimu02_mems_jet_mis_datasheet_a4_web.pdf?rev=6fad15ae93d24a92831391f280bc2a82&hash=16EA6B1DC4147EC220597E6F3BDC7FE1

[102] "MINIM MS Spec Sheet," 2023. [Online]. Available: https://prd-sc101-cdn.rtx.com/-/media/ca/product-assets/marketing/m/minim/minim_ms_datasheet_a4.pdf?rev=5906069ce92841a58a70ed6f861958be&hash=E1FB2EBD8FC2C5F6FB8BEF5B67D75EE2

[103] M. E. Hough, "Optimal guidance and nonlinear estimation for interception of decelerating targets," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 2, pp. 316–324, Mar. 1995. [Online]. Available: http://arc.aiaa.org/doi/10.2514/3.21386

APPENDICES

APPENDIX A

Complex Step Derivative Approximation Code Example

For reference to this code, see the Martins et al. paper [91] and the Rose dissertation [36].

```
%% Complex Step Derivative Approximation Script
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This code will evaluate the complex step derivative approximation
% with varying time steps for a given function. It will plot the
% results against the analytic derivative results.
% See Martins et al. paper [91] and Rose dissertation [36].
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%719%%%%%%
%% Clear Command Window, Figures, and Variables
clc; close all; clearvars
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initialize variables
hVals=zeros(30,1);
for ind=1:30
   hVals(ind,1)=10^-ind;
end
h=10^-10;  n_states=8;  n_timeSteps=1;  n_funcStates=1;
df=zeros(n_funcStates,n_states,n_timeSteps);
x=[1;1;1;10;100;100;-10;0];
z0=CSDA_rangeRate(x);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculate the derivative at the given state with varying time steps
for j=1:length(hVals)
for i=1:n_states
    y=zeros(n_states,1);
```

```
    h=hVals(j);
    y(i,1)=h;
    z=complex(x,y);
    z1=CSDA_rangeRate(z);
    df(1:n_funcStates,i,j)=imag(z1)/h;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculate the analytic partial derivative at x0
r =   [ x(1);x(2) ];
v=    [ x(3);x(4) ];
rt =   [ x(5);x(6) ];
vt=    [ x(7);x(8) ];

rtm=rt-r;
i_rtm=rtm/norm(rtm);
vtm=vt-v;
dz_dr= -vtm'/norm(rtm)*( eye(2)-i_rtm*i_rtm' );
dz_drt= vtm'/norm(rtm)*( eye(2)-i_rtm*i_rtm' );

        H = [ dz_dr, -i_rtm',dz_drt, i_rtm'];

an_df=zeros(n_funcStates,n_states,length(hVals));
diff_abs=zeros(n_funcStates,n_states,length(hVals));

for i=1:length(hVals)
an_df(:,:,i)= H;
end
```

```
for i=1:length(an_df)
diff_abs(:,:,i)=abs(df(:,:,i)-an_df(:,:,i));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Plot the difference between the two different methods
plotIndex=1;
for i=1:n_states
   for j=1:n_funcStates
    figure;hold on;
    loglog(hVals,squeeze(diff_abs(j,i,:))/abs(z0(j)));
    str=sprintf('H_{rangeRate} Jacobian Error: dz_%i/dx_%i',j,i);
    title(str)
    set(gca, 'XScale', 'log')
    if abs(mean(diff_abs(j,i,:))) > 1e-12
    set(gca, 'YScale', 'log')
    end
    ylabel('Normalized Error')
    xlabel('Step Size, h ')
    plotIndex=plotIndex+1;
   end
end
function f = CSDA_rangeRate(x)
r =   [ x(1);x(2) ];
v=    [ x(3);x(4) ];

rt =   [ x(5);x(6) ];
vt=    [ x(7);x(8) ];
```

```
rtm=rt-r;

vtm=vt-v;


f=(rtm(1)*vtm(1)+rtm(2)*vtm(2))/cnorm(rtm);

end

function xmag = cnorm(x)

% CNORM(X) is the Euclidean norm (i.e., length or magnitude)

% of a vector X, where X is comprised of real or complex elements.


 % Compute Euclidean norm

 xmag = sqrt(x.'*x);

end
```

APPENDIX B

Derivation Of Prop. Nav. Guidance Law Partial Derivatives

B.1 System Model

State Vector

The state vector is written as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}_{2\times1} \\ \mathbf{v}_{2\times1} \\ \mathbf{r}_{t_{2\times1}} \\ \mathbf{v}_{t_{2\times1}} \end{bmatrix}_{8\times1} \tag{B.1}$$

Dynamics

The dynamics of the state vector are:

$$\dot{\mathbf{r}} = \mathbf{v} \tag{B.2}$$

$$\dot{\mathbf{v}} = \mathbf{u}_{2\times1} \tag{B.3}$$

$$\dot{\mathbf{r}}_t = \mathbf{v}_t \tag{B.4}$$

$$\dot{\mathbf{v}}_t = \mathbf{0}_{2\times1} \tag{B.5}$$

Guidance Law

The proportional navigation guidance law in a 2D frame is written as:

$$\mathbf{u} = \mathbf{g}(\mathbf{x}) = n_c \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} \tag{B.6}$$

where

$$n_c = nV_c\dot{\lambda} \tag{B.7}$$

$$\lambda = atan(\left[\frac{B(\mathbf{r}_t - \mathbf{r})}{C(\mathbf{r}_t - \mathbf{r})}\right]) \tag{B.8}$$

$$B = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{B.9}$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{B.10}$$

$$V_c = -\frac{(\mathbf{r}_t - \mathbf{r})^T(\mathbf{v}_t - \mathbf{v})}{||(\mathbf{r}_t - \mathbf{r})||} \tag{B.11}$$

$$\dot{\lambda} = \frac{(\mathbf{r}_t - \mathbf{r})^T A(\mathbf{v}_t - \mathbf{v})}{||(\mathbf{r}_t - \mathbf{r})||^2} \tag{B.12}$$

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{B.13}$$

Partial Derivative

The derivative of the guidance law with respect to the state vector is:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} = G_{\mathbf{x}} = \frac{\partial\left[nV_c\dot{\lambda}\right]}{\partial \mathbf{x}} \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} + nV_c\dot{\lambda} \begin{bmatrix} \partial \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} \\ \hline \partial\mathbf{x} \\ \\ \end{bmatrix} \tag{B.14}$$

This will be derived in three parts: the closing velocity partial derivative, the line-of-sight rate partial derivative, and the inertial frame rotation partial derivative.

B.2 Guidance Law Partial Derivative

$$\frac{\partial\left[nV_c\dot{\lambda}\right]}{\partial\mathbf{x}} = nV_c\frac{\partial\dot{\lambda}}{\partial\mathbf{x}} + n\dot{\lambda}\frac{\partial V_c}{\partial\mathbf{x}} \tag{B.15}$$

where

$$\frac{\partial V_c}{\partial\mathbf{x}} = \left[\begin{array}{cccc} \frac{\partial V_c}{\partial\mathbf{r}} & \frac{\partial V_c}{\partial\mathbf{v}} & \frac{\partial V_c}{\partial\mathbf{r}_t} & \frac{\partial V_c}{\partial\mathbf{v}_t} \end{array}\right]_{1\times 8} \tag{B.16}$$

$$\frac{\partial\dot{\lambda}}{\partial\mathbf{x}} = \left[\begin{array}{cccc} \frac{\partial\dot{\lambda}}{\partial\mathbf{r}} & \frac{\partial\dot{\lambda}}{\partial\mathbf{v}} & \frac{\partial\dot{\lambda}}{\partial\mathbf{r}_t} & \frac{\partial\dot{\lambda}}{\partial\mathbf{v}_t} \end{array}\right]_{1\times 8} \tag{B.17}$$

B.3 Closing Velocity Partial Derivative

The derivative of the closing velocity V_c with respect to the state vector $\mathbf{x}$ is:

$$\frac{\partial V_c}{\partial\mathbf{x}} = \frac{\partial\left[-\frac{(\mathbf{r}_t-\mathbf{r})^T(\mathbf{v}_t-\mathbf{v})}{||(\mathbf{r}_t-\mathbf{r})||}\right]}{\partial\mathbf{x}} \tag{B.18}$$

Using the Quotient Rule:

$$\frac{\partial\left[\frac{a}{b}\right]}{\partial\mathbf{x}} = \frac{\frac{\partial a}{\partial x}b - a\frac{\partial b}{\partial x}}{b^2} \tag{B.19}$$

with substitutions

$$a = -(\mathbf{r}_t-\mathbf{r})^T(\mathbf{v}_t-\mathbf{v}) \tag{B.20}$$

$$b = ||(\mathbf{r}_t-\mathbf{r})|| = \sqrt{(\mathbf{r}_t-\mathbf{r})^T(\mathbf{r}_t-\mathbf{r})} = \sqrt{c} \tag{B.21}$$

$$c = (\mathbf{r}_t-\mathbf{r})^T(\mathbf{r}_t-\mathbf{r}) \tag{B.22}$$

Simplifying and rearranging:

$$\frac{\partial a}{\partial\mathbf{x}} = \frac{\partial\left[-(\mathbf{r}_t-\mathbf{r})^T(\mathbf{v}_t-\mathbf{v})\right]}{\partial\mathbf{x}} = -(\mathbf{r}_t-\mathbf{r})^T\frac{\partial[(\mathbf{v}_t-\mathbf{v})]}{\partial\mathbf{x}} - (v_t-\mathbf{v})^T\frac{\partial[(\mathbf{r}_t-\mathbf{r})]}{\partial\mathbf{x}} \tag{B.23}$$

$$\frac{\partial[(\mathbf{v}_t - \mathbf{v})]}{\partial \mathbf{x}} = \begin{bmatrix} 0_{2\times2} & -I_{2\times2} & 0_{2\times2} & I_{2\times2} \end{bmatrix}_{2\times8} \tag{B.24}$$

$$\frac{\partial[(\mathbf{r}_t - \mathbf{r})]}{\partial \mathbf{x}} = \begin{bmatrix} -I_{2\times2} & 0_{2\times2} & I_{2\times2} & 0_{2\times2} \end{bmatrix}_{2\times8} \tag{B.25}$$

$$\begin{aligned} \frac{\partial a}{\partial \mathbf{x}} =& -(\mathbf{r}_t - \mathbf{r})^T \begin{bmatrix} 0_{2\times2} & -I_{2\times2} & 0_{2\times2} & I_{2\times2} \end{bmatrix}_{2\times8} \\ & -(\mathbf{v}_t - \mathbf{v})^T \begin{bmatrix} -I_{2\times2} & 0_{2\times2} & I_{2\times2} & 0_{2\times2} \end{bmatrix}_{2\times8} \end{aligned} \tag{B.26}$$

$$\frac{\partial a}{\partial \mathbf{x}} = \begin{bmatrix} (\mathbf{v}_t - \mathbf{v})^T & (\mathbf{r}_t - \mathbf{r})^T & -(\mathbf{v}_t - \mathbf{v})^T & -(\mathbf{r}_t - \mathbf{r})^T \end{bmatrix}_{1\times8} \tag{B.27}$$

$$\frac{\partial b}{\partial \mathbf{x}} = \frac{\partial b}{\partial c}\frac{\partial c}{\partial \mathbf{x}} \tag{B.28}$$

$$\frac{\partial b}{\partial c} = \frac{1}{2\sqrt{c}} = \frac{1}{2||(\mathbf{r}_t - \mathbf{r})||} \tag{B.29}$$

$$\frac{\partial c}{\partial \mathbf{x}} = \frac{\partial\big[(\mathbf{r}_t - \mathbf{r})^T(\mathbf{r}_t - \mathbf{r})\big]}{\partial \mathbf{x}} = (\mathbf{r}_t - \mathbf{r})^T\frac{\partial[(\mathbf{r}_t - \mathbf{r})]}{\partial \mathbf{x}} + (r_t - \mathbf{r})^T\frac{\partial[(\mathbf{r}_t - \mathbf{r})]}{\partial \mathbf{x}} \tag{B.30}$$

$$\frac{\partial c}{\partial \mathbf{x}} = \begin{bmatrix} -2(\mathbf{r}_t - \mathbf{r})^T & 0_{1\times2} & 2(\mathbf{r}_t - \mathbf{r})^T & 0_{1\times2} \end{bmatrix}_{1\times8} \tag{B.31}$$

$$\frac{\partial b}{\partial \mathbf{x}} = \frac{\partial b}{\partial c}\frac{\partial c}{\partial \mathbf{x}} \tag{B.32}$$

$$\frac{\partial b}{\partial \mathbf{x}} = \frac{1}{||(\mathbf{r}_t - \mathbf{r})||}\begin{bmatrix} -(\mathbf{r}_t - \mathbf{r})^T & 0_{1\times2} & (\mathbf{r}_t - \mathbf{r})^T & 0_{1\times2} \end{bmatrix}_{1\times8} \tag{B.33}$$

Substituting (B.27) and (B.33) into (B.19)

$$\begin{aligned} \frac{\partial V_c}{\partial \mathbf{x}} =& \frac{1}{||(\mathbf{r}_t - \mathbf{r})||}\begin{bmatrix} (\mathbf{v}_t - \mathbf{v})^T & (\mathbf{r}_t - \mathbf{r})^T & -(\mathbf{v}_t - \mathbf{v})^T & -(\mathbf{r}_t - \mathbf{r})^T \end{bmatrix}_{1\times8} \\ & + \frac{(\mathbf{r}_t - \mathbf{r})^T(\mathbf{v}_t - \mathbf{v})}{||(\mathbf{r}_t - \mathbf{r})||^3}\begin{bmatrix} -(\mathbf{r}_t - \mathbf{r})^T & 0_{1\times2} & (\mathbf{r}_t - \mathbf{r})^T & 0_{1\times2} \end{bmatrix}_{1\times8} \end{aligned} \tag{B.34}$$

B.4 Line-Of-Sight Rate Partial Derivative

The derivative of the line-of-sight angle rate $\dot{\lambda}$ with respect to the state vector is:

$$\frac{\partial \dot{\lambda}}{\partial \mathbf{x}} = \frac{\partial \left[\frac{(\mathbf{r}_t - \mathbf{r})^T A (\mathbf{v}_t - \mathbf{v})}{||(\mathbf{r}_t - \mathbf{r})||^2} \right]}{\partial \mathbf{x}} \tag{B.35}$$

Use the Quotient Rule (Eq. B.19) with the following substitutions:

$$a = (\mathbf{r}_t - \mathbf{r})^T A (\mathbf{v}_t - \mathbf{v}) \tag{B.36}$$

$$b = ||(\mathbf{r}_t - \mathbf{r})||^2 = (\mathbf{r}_t - \mathbf{r})^T (\mathbf{r}_t - \mathbf{r}) \tag{B.37}$$

$$\frac{\partial a}{\partial \mathbf{x}} = \frac{\partial \big[(\mathbf{r}_t - \mathbf{r})^T A (\mathbf{v}_t - \mathbf{v}) \big]}{\partial \mathbf{x}} = (\mathbf{r}_t - \mathbf{r})^T A \frac{\partial [(\mathbf{v}_t - \mathbf{v})]}{\partial \mathbf{x}} + (\mathbf{v}_t - \mathbf{v})^T A^T \frac{\partial [(\mathbf{r}_t - \mathbf{r})]}{\partial \mathbf{x}} \tag{B.38}$$

Substitute Equation B.24 and B.25 into Equation B.38:

$$\begin{aligned} \frac{\partial a}{\partial \mathbf{x}} = & (\mathbf{r}_t - \mathbf{r})^T \begin{bmatrix} 0_{2\times 2} & -A_{2\times 2} & 0_{2\times 2} & A_{2\times 2} \end{bmatrix}_{2\times 8} \\ & + (\mathbf{v}_t - \mathbf{v})^T \begin{bmatrix} -A^T{}_{2\times 2} & 0_{2\times 2} & A^T_{2\times 2} & 0_{2\times 2} \end{bmatrix}_{2\times 8} \end{aligned} \tag{B.39}$$

$$\frac{\partial a}{\partial \mathbf{x}} = \begin{bmatrix} -(\mathbf{v}_t - \mathbf{v})^T A^T & -(\mathbf{r}_t - \mathbf{r})^T A & (\mathbf{v}_t - \mathbf{v})^T A^T & (\mathbf{r}_t - \mathbf{r})^T A \end{bmatrix}_{1\times 8} \tag{B.40}$$

$$\frac{\partial b}{\partial \mathbf{x}} = \frac{\partial \big[(\mathbf{r}_t - \mathbf{r})^T (\mathbf{r}_t - \mathbf{r}) \big]}{\partial \mathbf{x}} = 2(\mathbf{r}_t - \mathbf{r})^T \frac{\partial [(\mathbf{r}_t - \mathbf{r})]}{\partial \mathbf{x}} \tag{B.41}$$

$$\frac{\partial b}{\partial \mathbf{x}} = \begin{bmatrix} -2(\mathbf{r}_t - \mathbf{r})^T & 0_{1\times 2} & 2(\mathbf{r}_t - \mathbf{r})^T & 0_{1\times 2} \end{bmatrix}_{1\times 8} \tag{B.42}$$

Applying the quotient rule

$$\frac{\partial \dot{\lambda}}{\partial \mathbf{x}} = \frac{1}{||(\mathbf{r}_t - \mathbf{r})||^2} \begin{bmatrix} -(\mathbf{v}_t - \mathbf{v})^T A^T & -(\mathbf{r}_t - \mathbf{r})^T A & (\mathbf{v}_t - \mathbf{v})^T A^T & (\mathbf{r}_t - \mathbf{r})^T A \end{bmatrix}_{1\times 8} \tag{B.43}$$

$$-\frac{(\mathbf{r}_t-\mathbf{r})^T A(\mathbf{v}_t-\mathbf{v})}{||(\mathbf{r}_t-\mathbf{r})||^4}\begin{bmatrix} -2(\mathbf{r}_t-\mathbf{r})^T & 0_{1\times 2} & 2(\mathbf{r}_t-\mathbf{r})^T & 0_{1\times 2}\end{bmatrix}_{1\times 8}$$

Combine the results from Equation B.34 and Equation B.43 into Equation B.15

B.5 Frame Rotation Partial Derivative

Find

$$\frac{\partial\begin{bmatrix}-\sin\lambda\\ \cos\lambda\end{bmatrix}}{\partial\mathbf{x}} \tag{B.44}$$

where λ is the line-of-sight angle given by:

$$\lambda = atan(\left[\frac{B(\mathbf{r}_t-\mathbf{r})}{C(\mathbf{r}_t-\mathbf{r})}\right]) \tag{B.45}$$

The following substitutions are utilized:

$$B = [\ 0 \quad 1\ ] \tag{B.46}$$

$$C = [\ 1 \quad 0\ ] \tag{B.47}$$

$$d = B(\mathbf{r}_t-\mathbf{r}) \tag{B.48}$$

$$e = C(\mathbf{r}_t-\mathbf{r}) \tag{B.49}$$

$$f = \frac{d}{e} \tag{B.50}$$

$$g = -\sin\lambda \tag{B.51}$$

$$h = \cos\lambda \tag{B.52}$$

The expanded partial derivative is:

$$\frac{\partial \begin{bmatrix} g \\ h \end{bmatrix}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial g}{\partial \mathbf{x}} \\ \\ \frac{\partial h}{\partial \mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial g}{\partial \mathbf{r}} & \frac{\partial g}{\partial \mathbf{v}} & \frac{\partial g}{\partial \mathbf{r}_t} & \frac{\partial g}{\partial \mathbf{v}_t} \\ \\ \frac{\partial h}{\partial \mathbf{r}} & \frac{\partial h}{\partial \mathbf{v}} & \frac{\partial h}{\partial \mathbf{r}_t} & \frac{\partial h}{\partial \mathbf{v}_t} \end{bmatrix}_{2\times 8} \tag{B.53}$$

where

$$\frac{\partial g}{\partial \mathbf{x}} = \frac{\partial g}{\partial \lambda}\frac{\partial \lambda}{\partial f}\frac{\partial f}{\partial \mathbf{x}} \tag{B.54}$$

$$\frac{\partial h}{\partial \mathbf{x}} = \frac{\partial h}{\partial \lambda}\frac{\partial \lambda}{\partial f}\frac{\partial f}{\partial \mathbf{x}} \tag{B.55}$$

Developing Equations B.54 and B.55:

$$\frac{\partial g}{\partial \lambda} = -\cos\lambda \tag{B.56}$$

$$\frac{\partial h}{\partial \lambda} = -\sin\lambda \tag{B.57}$$

$$\frac{\partial \lambda}{\partial f} = \frac{1}{1+f^2} = \frac{1}{1+\left[\frac{d}{e}\right]^2} = \frac{1}{1+\left[\frac{B(\mathbf{r}_t-\mathbf{r})}{C(\mathbf{r}_t-\mathbf{r})}\right]^2} = \frac{1}{1+\frac{[B(\mathbf{r}_t-\mathbf{r})]^2}{[C(\mathbf{r}_t-\mathbf{r})]^2}} \tag{B.58}$$

$$\frac{\partial \lambda}{\partial f} = \frac{[C(\mathbf{r}_t-\mathbf{r})]^2}{[C(\mathbf{r}_t-\mathbf{r})]^2 + [B(\mathbf{r}_t-\mathbf{r})]^2} \tag{B.59}$$

$$\frac{\partial f}{\partial \mathbf{x}} = \frac{\partial \left[\frac{d}{e}\right]}{\partial \mathbf{x}} = \frac{\frac{\partial d}{\partial \mathbf{x}}e - d\frac{\partial e}{\partial \mathbf{x}}}{e^2} \tag{B.60}$$

$$\frac{\partial d}{\partial \mathbf{x}} = \begin{bmatrix} -B_{1\times 2} & 0_{1\times 2} & B_{1\times 2} & 0_{1\times 2} \end{bmatrix}_{1\times 8} \tag{B.61}$$

$$\frac{\partial e}{\partial \mathbf{x}} = \begin{bmatrix} -C_{1\times 2} & 0_{1\times 2} & C_{1\times 2} & 0_{1\times 2} \end{bmatrix}_{1\times 8} \tag{B.62}$$

Substituting Equations B.61 and B.62 into Equation B.60:

$$\frac{\partial f}{\partial \mathbf{x}} = \frac{[C(\mathbf{r}_t-\mathbf{r})]\begin{bmatrix} -B_{1\times 2} & 0_{1\times 2} & B_{1\times 2} & 0_{1\times 2} \end{bmatrix}}{[C(\mathbf{r}_t-\mathbf{r})]^2}$$

$$\frac{-[B(\mathbf{r}_t - \mathbf{r})] \begin{bmatrix} -C_{1\times 2} & 0_{1\times 2} & C_{1\times 2} & 0_{1\times 2} \end{bmatrix}}{[C(\mathbf{r}_t - \mathbf{r})]^2} \tag{B.63}$$

Substituting Equations B.56, B.57, B.59, and B.63 into Equations B.54 and B.55.

$$\frac{\partial g}{\partial \mathbf{x}} = -\cos\lambda \frac{\partial \lambda}{\partial f} \frac{[C(\mathbf{r}_t - \mathbf{r})] \begin{bmatrix} -B_{1\times 2} & 0_{1\times 2} & B_{1\times 2} & 0_{1\times 2} \end{bmatrix}}{[C(\mathbf{r}_t - \mathbf{r})]^2}$$

$$+\cos\lambda \frac{\partial \lambda}{\partial f} \frac{[B(\mathbf{r}_t - \mathbf{r})] \begin{bmatrix} -C_{1\times 2} & 0_{1\times 2} & C_{1\times 2} & 0_{1\times 2} \end{bmatrix}}{[C(\mathbf{r}_t - \mathbf{r})]^2} \tag{B.64}$$

$$\frac{\partial h}{\partial \mathbf{x}} = -\sin\lambda \frac{\partial \lambda}{\partial f} \frac{[C(\mathbf{r}_t - \mathbf{r})] \begin{bmatrix} -B_{1\times 2} & 0_{1\times 2} & B_{1\times 2} & 0_{1\times 2} \end{bmatrix}}{[C(\mathbf{r}_t - \mathbf{r})]^2}$$

$$+\sin\lambda \frac{\partial \lambda}{\partial f} \frac{[B(\mathbf{r}_t - \mathbf{r})] \begin{bmatrix} -C_{1\times 2} & 0_{1\times 2} & C_{1\times 2} & 0_{1\times 2} \end{bmatrix}}{[C(\mathbf{r}_t - \mathbf{r})]^2} \tag{B.65}$$

Substituting Equations B.64 and B.65 into B.53:

$$\frac{\partial \begin{bmatrix} -\sin\lambda \\ \\ \cos\lambda \end{bmatrix}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial g}{\partial \mathbf{x}} \\ \\ \\ \frac{\partial h}{\partial \mathbf{x}} \end{bmatrix}_{2\times 8} \tag{B.66}$$

B.6 Combined Partial Derivative Results

Substitute the results from Equations B.15, B.34, B.43, B.59, and B.64-B.66 into Equation B.14:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} = G_{\mathbf{x}} = \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix} \frac{\partial \left[nV_c\dot{\lambda} \right]}{\partial \mathbf{x}} + nV_c\dot{\lambda} \begin{bmatrix} \dfrac{\partial \begin{bmatrix} -\sin\lambda \\ \cos\lambda \end{bmatrix}}{\partial \mathbf{x}} \\ \\ \end{bmatrix} \tag{B.67}$$

CURRICULUM VITAE

Jeffrey Scott Clawson

Education

Ph.D., Aerospace Engineering

Utah State University, Logan, Utah, 2023

M.S., Aerospace Engineering

Utah State University, Logan, Utah, 2016

B.S., Mechanical Engineering

Colorado School of Mines, Golden, Colorado, 2012

Professional Experience

System Integration Engineer

Northrop Grumman, 2012-2016

GN&C Engineer

Space Dynamics Laboratory, 2017-Present

Publications

Clawson, J. S., Christensen, R., "Linear Covariance Analysis for Head-on Target Engagements," AIAA Journal of Spacecraft and Rockets, 2022. doi: 10.2514/1.A35178

Clawson, J. S., Christensen, R., "Linear Covariance Model Development for Target Engagements,"AIAA SciTech Forum, Nashville, TN, USA, 2021, pp. 1-16, doi: 10.2514/6.2021-1464.

Awards

Colorado School of Mines Presidential Scholarship

State Farm National Merit Scholarship

Space Dynamics Lab Tomorrow Ph.D. Fellowship