

---


Electronic Theses and Dissertations, 2004-2019

---

2009

## Robust Estimation And Adaptive Guidance For Multiple Uavs' Cooperation

Randal Allen  
*University of Central Florida*

 Part of the [Mechanical Engineering Commons](#)  
Find similar works at: <https://stars.library.ucf.edu/etd>  
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Allen, Randal, "Robust Estimation And Adaptive Guidance For Multiple Uavs' Cooperation" (2009).  
*Electronic Theses and Dissertations, 2004-2019*. 3933.  
<https://stars.library.ucf.edu/etd/3933>

ROBUST ESTIMATION AND ADAPTIVE GUIDANCE FOR MULTIPLE UAVS'  
COOPERATION

by

RANDAL T. ALLEN  
B.S. University of Illinois, 1988  
M.S. University of Illinois, 1990  
E.A.A. Stanford University, 1994

A dissertation submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy  
in the department of Mechanical, Materials, and Aerospace Engineering  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Spring term  
2009

Major Professor: Chengying Xu

© 2009 Randal T. Allen

## ABSTRACT

In this paper, an innovative cooperative navigation method is proposed for multiple Unmanned Air Vehicles (UAVs) based on online target position measurements. These noisy position measurement signals are used to estimate the target's velocity for non-maneuvering targets or the target's velocity and acceleration for maneuvering targets. The estimator's tracking capability is physically constrained due to the target's kinematic limitations and therefore is potentially improvable by designing a higher performance estimator. An H-infinity filter is implemented to increase the robustness of the estimation accuracy. The performance of the robust estimator is compared to a Kalman filter and the results illustrate more precise estimation of the target's motion in compensating for surrounding noises and disturbances.

Furthermore, an adaptive guidance algorithm, based on the seeker's field-of-view and linear region, is used to deliver the pursuer to the maneuvering target. The initial guidance algorithm utilizes the velocity pursuit guidance law because of its insensitivity to target motion; while the terminal guidance algorithm leverages the acceleration estimates (from the H-infinity filter) to augment the proportional navigation guidance law for increased accuracy in engaging maneuvering targets.

The main objective of this work is to develop a robust estimator/tracker and an adaptive guidance algorithm which are directly applicable UAVs.

To the generations of my entire family, whose work ethic and sacrifices created the opportunity for me to strive for and achieve this goal; and to my mother and both of my grandmothers, who passed away within 15 months of one another while I was working on this dissertation; I thank you for your love which has created within me a strong sense of self-esteem. To the men and women of the armed services, including family members who fought in the American Revolution, I thank you for your honor and courage.

## **ACKNOWLEDGMENTS**

Firstly, I would like to thank my wife for her never-ending support. Second, I would like to thank my advisors: Dr. Chengying “Cheryl” Xu, Dr. Kurt Lin, and Dr. Alexander Leonessa for their subject matter expertise and guidance in my research. I would also like to thank Steve Roerman from Lone Star Aerospace for his mentoring and support. Next, I would like to thank Dr. Larry Chew from the Mechanical, Materials, and Aerospace Engineering (MMAE) department and Dr. Harold Klee and Dr. Michael Haralambous from the Electrical Engineering (EE) department for their valuable time in serving as thesis dissertation committee members. Furthermore, I would like to thank the staff of the MMAE department who have selflessly supported me. Finally, I would like to express special thanks to Dr. James McBrayer for inspiring me to pursue this endeavor.

# TABLE OF CONTENTS

LIST OF FIGURES .....	x
LIST OF TABLES .....	xiii
LIST OF ACRONYMS/ABBREVIATIONS .....	xiv
CHAPTER ONE: INTRODUCTION.....	1
1.1 Classical Guidance Algorithms.....	4
1.2 Modern Guidance Algorithms .....	6
1.3 Control Developments .....	8
1.4 Cooperative Control.....	10
1.5 Approach.....	12
1.6 Scope of Research.....	14
CHAPTER TWO: COOPERATIVE NAVIGATION .....	16
2.1 Discrete-Time Kalman Filter .....	17
2.1.1 Recursive Least Squares .....	17
2.2 Discrete-Time H-infinity Filter.....	24
2.3 Navigation Design .....	29
2.3.1 Target Modeling.....	29
2.3.2 Target Kinematic Limitations .....	32
2.3.3 Discrete-Time Kalman Filter Implementation.....	33

2.3.4 Discrete-Time H-infinity Filter Implementation .....	34
2.4 Simulation Results .....	35
2.4.1 Target, UAV, and Pursuer Modeling Parameters .....	35
2.4.2 Simulation Case One: Zero-Mean Noise Statistics.....	35
2.4.3 Simulation Case Two: Biased Noise Statistics .....	40
2.4.4 Simulation Case Three: Constrained Estimates.....	45
2.4.5 Variances, Covariances, and Gains.....	48
2.5 Cooperative Navigation Summary.....	53
<b>CHAPTER THREE: GUIDANCE AND CONTROL.....</b>	<b>56</b>
3.1 Airframe Dynamic Equations .....	58
3.1.1 Lateral Accelerations and Angular Rates .....	58
3.2 Flight Control System (Autopilot).....	61
3.2.1 Pitch Dynamics .....	61
3.2.2 Pitch Control .....	63
3.2.3 Gain Scheduling.....	66
3.3 Guidance Laws.....	68
3.3.1 Augmented Proportional Navigation Guidance Law.....	69
3.3.2 Velocity Pursuit Guidance Law.....	70
3.3.3 Guidance Logic.....	73
3.4 Simulation Results .....	75
3.4.1 Simulation of a Maneuvering Target.....	75
3.4.2 Simulation of a Fixed (Static) Target.....	83



3.4.3 Monte Carlo Simulation of 500 Engagements.....	91
3.5 Guidance and Control Summary.....	92
CHAPTER FOUR: CONCLUSIONS AND FUTURE WORK.....	95
4.1 Conclusions.....	95
4.2 Assumptions and Limitations .....	96
4.2.1 Time-Delay for UAV to Transmit Target’s Position.....	96
4.3 Recommendations and Improvements.....	97
4.3.1 Estimator Performance Improvements .....	97
4.3.2 Algorithm Execution Time .....	97
4.3.3 Including Velocity Measurement for the Estimator.....	97
4.3.4 Predictor versus Estimator .....	98
4.3.5 Designation Leading .....	98
APPENDIX A: AERODYNAMIC FORCES AND MOMENTS .....	99
A.1 Parameters.....	100
A.2 Coordinate Systems and Sign Conventions .....	100
A.3 Aerodynamic Forces .....	101
A.3.1 Axial Force.....	101
A.3.2 Normal Force .....	101
A.3.3 Side Force .....	102
A.4 Aerodynamic Moments.....	103
A.4.1 Rolling Moment .....	103

A.4.2 Pitching Moment.....	104
A.4.3 Yawing Moment .....	105
APPENDIX B: LYAPUNOV STABILITY PROOF FOR THE AUGMENTED PROPORTIONAL NAVIGATION GUIDANCE LAW .....	106
APPENDIX C: OPTIMAL GAIN FOR THE AUGMENTED PROPORTIONAL NAVIGATION GUIDANCE LAW .....	111
LIST OF REFERENCES .....	118

## LIST OF FIGURES

Figure 1	Time and Measurement Updates .....	20
Figure 2	Target Footprint .....	36
Figure 3	Relative Down Range Position Errors .....	37
Figure 4	Relative Cross Range Position Errors.....	37
Figure 5	Relative Down Range Velocity Errors .....	38
Figure 6	Relative Cross Range Velocity Errors .....	39
Figure 7	Relative Down Range Acceleration Errors.....	39
Figure 8	Relative Cross Range Acceleration Errors .....	40
Figure 9	Relative Down Range Position Errors .....	41
Figure 10	Relative Cross Range Position Errors.....	42
Figure 11	Relative Down Range Velocity Errors .....	43
Figure 12	Relative Cross Range Velocity Errors .....	43
Figure 13	Relative Down Range Acceleration Errors.....	44
Figure 14	Relative Cross Range Acceleration Errors .....	44
Figure 15	Relative Down Range Position Errors .....	45
Figure 16	Relative Cross Range Position Errors.....	46
Figure 17	Relative Down Range Velocity Errors .....	47
Figure 18	Relative Cross Range Velocity Errors .....	47
Figure 19	Relative Down Range Acceleration Errors.....	48
Figure 20	Relative Cross Range Acceleration Errors .....	48
Figure 21	Estimation Error Down Range Variances.....	49
Figure 22	Estimation Error Down Range Covariances.....	49

Figure 23	Estimation Error Cross Range Variances .....	50
Figure 24	Estimation Error Cross Range Covariances.....	50
Figure 25	Down Range Gains .....	52
Figure 26	Cross Range Gains .....	52
Figure 27	Kinematic Angles.....	59
Figure 28	Open-Loop Step-Response at Mach 0.33 .....	62
Figure 29	Open-Loop Step-Response at Mach 0.80 .....	63
Figure 30	Pitch Control Block Diagram.....	63
Figure 31	Closed-Loop Step-Response at Mach 0.33.....	65
Figure 32	Closed-Loop Step-Response at Mach 0.80.....	65
Figure 33	Pitch Control Block Diagram.....	66
Figure 34	Velocity Pursuit Guidance Geometry .....	71
Figure 35	Velocity Pursuit Guidance Block Diagram.....	71
Figure 36	State Transition Diagram .....	74
Figure 37	Relative Down Range Position Estimate Error.....	76
Figure 38	Relative Cross Range Position Estimate Error .....	76
Figure 39	Relative Down Range Velocity Estimate Error .....	77
Figure 40	Relative Cross Range Velocity Estimate Error.....	77
Figure 41	Relative Down Range Acceleration Estimate Error .....	78
Figure 42	Relative Cross Range Acceleration Estimate Error .....	78
Figure 43	Seeker Look Angles .....	79
Figure 44	Adaptive Guidance Logic .....	80
Figure 45	Side-View Trajectories .....	80

Figure 46	Top-View Trajectories .....	81
Figure 47	Down Range Trajectories versus Time .....	81
Figure 48	Cross Range Trajectories versus Time .....	82
Figure 49	Lateral Acceleration.....	83
Figure 50	Relative Down Range Position Estimate Error.....	84
Figure 51	Relative Cross Range Position Estimate Error .....	84
Figure 52	Relative Down Range Velocity Estimate Error .....	85
Figure 53	Relative Cross Range Velocity Estimate Error.....	85
Figure 54	Relative Down Range Acceleration Estimate Error .....	86
Figure 55	Relative Cross Range Acceleration Estimate Error .....	86
Figure 56	Seeker Look Angles .....	87
Figure 57	Adaptive Guidance Logic .....	88
Figure 58	Side-View Trajectories .....	89
Figure 59	Top-View Trajectories .....	89
Figure 60	Down Range Trajectories versus Time.....	90
Figure 61	Cross Range Trajectories versus Time .....	90
Figure 62	Lateral Acceleration.....	91

## LIST OF TABLES

Table 1	Relationships Between Estimates and Covariances.....	23
Table 2	RMS Position Errors for Zero-Mean Noise .....	38
Table 3	RMS Position Errors for Biased Noise .....	42
Table 4	RMS Position Errors for Constrained Estimates .....	46
Table 5	Monte Carlo Input Data .....	92
Table 6	Monte Carlo Results .....	92
Table 7	Effects of Time-Delay on Miss Distance.....	96

## LIST OF ACRONYMS/ABBREVIATIONS

6DOF	Six Degree-of-Freedom
APNG	Augmented Proportional Navigation Guidance
CEP	Circular Error Probable
DARPA	Defense Advanced Research Projects Agency
DoD	Department of Defense
ISR	Intelligence, Surveillance, and Reconnaissance
JDAM	Joint Direct Attack Munition
GDP	Gross Domestic Product
LAR	Launch Acceptability Region
PNG	Proportional Navigation Guidance
RoE	Rules of Engagement
RPV	Remotely Piloted Vehicle
SAL	Semi-Active Laser
SEAD	Suppression of Enemy Air Defense
T&E	Test & Evaluation
TAC	Target Acquisition Cooperative
TPS	Test Pilot School
UAV	Unmanned Air Vehicle
UCAV	Unmanned Combat Air Vehicle
USAF	United States Air Force
UVS	Unmanned Vehicle System
VPG	Velocity Pursuit Guidance

## CHAPTER ONE: INTRODUCTION

Recently, there has been a shift from *conventional* warfare where one nation attacks another nation, e.g. Japan attacking the United States at Pearl Harbor on 7 December 1941, followed by combat with clear lines of friendly and enemy troops, to *asymmetric* warfare where a few individuals attack a nation, e.g. the terrorist's attack of New York City's World Trade Center on 11 September 2001, followed by combat where enemy combatants are embedded in personnel and non-combatant areas.

Since the "Twin Towers" attack, an increasing need has arisen to intercept what are known as *time-critical* threats. As the name implies, time-critical threats reveal themselves for only brief moments in time or highly mobile. The challenge is that they are difficult to acquire and track by such assets as reconnaissance satellites which may only be available to view the area of interest for a limited period of time or may be optically impaired by atmospheric interferences. Also, if its attributes should happen to be compromised, it would be easy for the threat to relocate while the reconnaissance satellite is not overhead. This demonstrates a requirement for what is known as *persistence*. Persistence is the ability to gather intelligence, surveillance, and reconnaissance (ISR) over a continuous and extended period of time. Pierce et al. (2002) indicates that there is an increasing demand for *immediate* intelligence on the battlefield.

Along with the "Twin Towers" attack, the United States has been experiencing a contracting economy (some say a recession) while other nations are emerging/expanding, e.g. China and India. With a flattening or reduction in Gross Domestic Product (GDP), the United States is under fiscal pressure to make cuts to the Department of Defense



(DoD) budget. This has led to a reduction in military force where soldiers are already serving multiple tours of combat duty.

While time-critical threats have led to the need for persistence, persistence and the reduction in military forces (through budget cuts) have led to the improved development of Unmanned Air Vehicles (UAVs). Schulze and Buescher (2003) described how UAVs have evolved from remotely piloted vehicles (RPVs) to fully autonomous systems, which were capable of performing aerial objectives and full missions without assistance from a human operator. Christophersen et al. (2004) described future UAVs requiring enhanced capabilities, such as seeing and avoiding obstacles, tolerating unpredicted flight conditions, interfacing with payload sensors, tracking moving targets, and cooperating with manned and unmanned systems. Minor et al. (2005) reported the United States Air Force (USAF) Test Pilot School (TPS) recently developed and taught its first-ever Unmanned Air Vehicle (UAV)/Unmanned Combat Air Vehicle (UCAV) flight Test and Evaluation (T&E) course. With regard to performance, current light-weight UAVs are able to maintain a persistence of up to 48 hours, e.g. I-GNAT (General Atomics). With a persistent view of the area of interest, situational awareness is increased, and military forces are more responsive to time-critical threats. Still, in some cases, the threat manages to take evasive action and avoids interception. This is primarily due to the time lag between threat classification as the stimulus and weapon mobilization as the response.

Characteristically, these time-critical threats are “smart” surface targets (land or sea) with complex evasive dynamics, including anything above and beyond static fixed targets or surface targets with a constant velocity. In particular, the targets are intended to be maneuvering, i.e. accelerating and/or decelerating with random steering motion.

Furthermore, these “smart” surface targets have the capability of outmaneuvering their pursuers and potentially “know” they are being engaged. For example in the case of the Copperhead Semi-Active Laser (SAL) guided projectile program (Lundquist (2008)), the target could know it was being designated and therefore employ evasive counter actions.

The combination of asymmetric warfare, time-critical threats, persistence, and restrictive Rules of Engagement (RoE) are driving the need to arm the UAV platform with a precision, low collateral damage pursuer. The low collateral damage requirement stems from the public’s low tolerance for casualties (Pierce et al. (2002)). Of particular importance is the desire to take immediate action once a threat has been acquired, classified, and is being tracked.

Because the UAV platform is light-weight to begin with, the pursuer system itself must be light-weight. Even if the target cannot outmaneuver the pursuer or if the target doesn’t know it is being designated / engaged, the pursuer system needs to be *effective* because of the threat’s “high value,” e.g. a mobile scud launcher or a mobile command and control center. Other factors driving the pursuer system’s requirements are to carry multiple pursuers per sortie (*high-density carriage*) and to be low-cost due to the DoD’s budget cuts.

These *qualitative* requirements translate into a particular *quantitative* requirement: a highly effective guidance law is needed for a light-weight (less than 25lbs), low-cost, high-precision and low collateral damage (less than 3m Circular Error Probable, CEP) pursuer with limited acceleration capability (less than 5g), (Roemerman 2006). CEP is approximately equal to the  $1\sigma$  miss distance, (Fleeman 2006).

The remainder of this section includes a literature review of classical and modern *guidance* algorithms comparing their quantitative results with the quantitative requirements outlined in the previous paragraph. Subsequently, *control systems* literature will be reviewed for identical reasons. Note, however, that the research focus of this dissertation is on navigation and guidance algorithms, instead of control systems, i.e. autopilots. Next, *cooperative control* literature will be reviewed to examine its application in an attempt to meet the quantitative requirements. Finally, the author's philosophical approach will be discussed, including defined success criteria.

### 1.1 Classical Guidance Algorithms

By “classical,” the author is referring to guidance algorithms applied to a pursuer engaged with a static fixed target or a target moving with a constant velocity relative to the pursuer. Historically, these guidance algorithms were easily implemented using analog circuitry.

In general, guidance algorithms are characterized by their lateral acceleration commands. Pastrick et al. (1981), Lin (1991), and Song (2004) described the two most common types of classical guidance, pursuit guidance and Proportional Navigation Guidance (PNG), each of which has its own advantages and significant disadvantages.

With pursuit guidance, the angle between the longitudinal axis of the pursuer (attitude pursuit) and the line-of-sight to the target, called the “look angle” or the velocity vector of the pursuer (velocity pursuit) and the line-of-sight to the target, called the “lead angle” is driven to zero or some constant value (deviated pursuit). Even though this type of guidance algorithm is noise insensitive and easy to use with strapdown seekers, it is

impractical against moving targets owing to the high maneuver requirement. The high maneuverability requirement originates from the lack of line-of-sight *rate* information.

Proportional navigation guidance is where the heading rate is made proportional to the rate of the line-of-sight angle from the pursuer to the target. Mathematically, the acceleration command is given by:

$$a_c = AV_c\dot{\sigma} \quad (1.1)$$

where  $A$  is a proportional constant,  $V_c$  is the closing velocity, and  $\dot{\sigma}$  is the rate of the line-of-site angle. The constant of proportionality varies between 2 and 4 (Lin (1991)). Values less than 2 require infinite acceleration, while values greater than 4 tend to steer the pursuer in response to high frequency noise. Therefore, PNG is accurate against *constant* velocity targets, but is inaccurate against maneuvering targets, and as mentioned previously, stability is inevitably sensitive to noise.

Pursuit plus proportional computes guidance commands based on both algorithms and combines them with a time-varying weighting factor. At long ranges, the target motion appears noisy and the accuracy requirement is low, therefore the pursuit algorithm is weighted more heavily. At short ranges, the target's motion increases the acceleration requirement on the pursuer; therefore the proportional navigation algorithm is weighted more heavily. The combined performance of the pursuit and proportional navigation algorithm takes care of the disadvantages of both algorithms and improve the overall system performance. In fact, Takehira et al. (1998) combined pursuit guidance and proportional navigation guidance into a new guidance algorithm. However, the goal of their paper was to achieve an analytical solution without any consideration of the

acceleration requirements. No information was given with regard to the magnitude of the lateral acceleration command imposed on the pursuer.

With classical guidance algorithms, the common disadvantage is its incapability of intercepting maneuvering targets.

## 1.2 Modern Guidance Algorithms

As the pursuer's performance requirements increase, the demand for the associated hardware (e.g. seekers, gyroscopes, and accelerometers) and corresponding software also increases the need to modify guidance algorithms in order to intercept maneuvering targets.

Chadwick and Rose (1983) described a modification to the attitude pursuit guidance algorithm incorporating the flight heading. In terms of fin deflection ( $\delta$ ) for control, the modification may be expressed as:

$$\delta = k_p(\varepsilon + \mu\gamma) \quad (1.2)$$

where  $\varepsilon$  is the pursuit error,  $k_p$  and  $\mu$  are constants, and  $\gamma$  is the flight heading. However, the guidance algorithm was applied to a certain category of moving targets with a *constant* crossing velocity or a *constant* crossing acceleration. It was assumed that there were no variations in the target's velocity or acceleration, i.e. the target was not maneuvering; therefore, this algorithm is not suitable for a maneuvering target application.

Chatterji, G. and Pachter (1991) described a modification to the velocity pursuit guidance algorithm to account for crosswinds in the presence of stationary and moving

targets. In terms of fin deflections for yaw ( $\delta_y$ ) and pitch ( $\delta_p$ ) control, the modification may be expressed as:

$$\delta_y = K(\varepsilon_y + \mu\psi - \eta\dot{\psi}) - \psi_w \quad (1.3)$$

$$\delta_p = K(\varepsilon_p + \mu\theta - \eta\dot{\theta}) - \theta_w \quad (1.4)$$

where  $\varepsilon_y$  and  $\varepsilon_p$  are the yaw and pitch bore sight errors,  $\psi$  and  $\theta$  are the yaw and pitch attitude angles,  $\mu$  and  $K$  are proportional gains,  $\eta$  is the rate gain, and  $\psi_w$  and  $\theta_w$  are the yaw and pitch angle corrections for crosswind. However, there was no mention of the characteristics of the target's motion in the paper.

Rusnak (1996) developed explicit, closed-form solutions of advanced guidance algorithms for an acceleration-constrained missile and a randomly maneuvering target with noisy measurement positions. The derivation of the complicated, yet closed-form solution is rather complicated and will not be redeveloped here. However, the acceleration-constrained pursuer still needed 70g of acceleration to achieve a miss distance of 5 meters. This stringent acceleration requirement is difficult to implement with the current hardware capability.

Vergez and McClendon (1982) combined the proportional guidance algorithm with an extended Kalman filter to intercept a low-maneuverability target. Mathematically, the missile's acceleration command is expressed as follows:

$$A_{MC} = 3(S_R / t_{go}^2 + V_R / t_{go} + K_T A_T) \quad (1.5)$$

where  $S_R$  is the relative position referenced to the missile body,  $V_R$  is the relative velocity referenced to the missile body,  $A_T$  is the target acceleration referenced to the missile body, and  $K_T$  is the target acceleration gain. While the target performed a 9g

maneuver, no information was given about the pursuer's acceleration requirement. Lin (1991) stated that a pursuer's acceleration requirement is typically three times the target's maneuverability when proportional navigation guidance is employed. Thus, the pursuer would require approximately 27g of acceleration, if Vergez's and McClendon's algorithm is applied.

Deyst and Price (1973) developed an optimal stochastic guidance algorithm to intercept a maneuvering target with limited pursuer acceleration capability. While the derivation of the algorithm was rather complicated, the pursuer's acceleration requirements were reported to be between 10g and 20g of acceleration. This is approximately a 50% improvement over the previous results with regard to acceleration requirements.

From the aforementioned papers, it is clear that modern guidance algorithms reduce the pursuer's acceleration requirement for intercepting a maneuvering target. For the low-cost, light-weight pursuer system under consideration, it is desired to reduce the lateral acceleration requirement to 5g of acceleration or less. It is important to recognize the *lateral* acceleration requirement. This is because the proposed pursuer system has no longitudinal (thrust) acceleration capability. Because of this limiting acceleration requirement, alternative guidance algorithms or other strategies must be explored.

### 1.3 Control Developments

The previous two sub-sections focus on guidance algorithm development. Thereafter, the literature survey on the recent advancements in control will be presented. While guidance is the process of comparing the measured navigation state with a required navigation state and then computing acceleration commands to correct differences

between the two, control is the deflection of physical surfaces (fins) to bring about the required velocity change (Cochran et al. (1985)).

Since Ben-Asher and Yaesh (1998) authored advances in guidance algorithms, a number of papers have been written (Sharma and Calise (2000), Sharma and Lavretsky (2006), Stepanyan and Hovakimyan (2005), Wise et al. (2005), and Wise and Lavretsky (2006)) laying the foundation for adaptive and neural control as applied to autopilot design. In fact, Sharma and Lavretsky (2006) cited an adaptive autopilot application to the Joint Direct Attack (precision-guided) Munition (JDAM). DARPA report (2002), DARPA report (2003), and The Boeing Company report (2006) cited a successful intercept of an armored personnel carrier moving at 25 mph. However, no additional information was mentioned with regard to the target's maneuverability or the pursuer's acceleration requirement. Furthermore, because JDAMs range in size from 500lbs to 2000lbs, it is infeasible to utilize JDAM as a pursuer system for a UAV.

While adaptive and neural control autopilot developments have been implemented, their inherent complexity increases the cost for software development, hardware/software integration, and system testing. Therefore, this research focus is on developing a simple acceleration autopilot (Zipfel (2000)) which has been implemented and successfully tested over many years, thus meeting the need for a low-cost pursuer system.

Note that the research focus of this dissertation is on navigation and guidance instead of on control, i.e. autopilots. The autopilot content is included in order to complete the navigation, guidance, and control loop.



## 1.4 Cooperative Control

A cooperative control system is defined to be multiple dynamic entities that share information to accomplish a common objective. Childers (2007) reported that the United States Army is committed to a paradigm shift in the way future ground military operations will be conducted. With the deployment of unmanned systems, future forces will achieve increased mobility, lethality, and survivability. Clough (2003) reported on the United States Air Force's challenges related to autonomous control for unmanned air vehicles, including the need for situational awareness (which relates to persistence); the need for autonomous UAVs and their human supervisors to understand one another's intent and actions; and the need for artificial intelligence in the UAV's "brain" to plan and react as humans would, yet with some amount of unpredictability. There have been a number of papers written in response to these challenges. Reichard et al. (2003) described the use of a behavior-based, intelligent control architecture to integrate internal self-situational awareness, external self-situational awareness, and autonomous control. White (2004) claimed that a human operator will probably be required to supervise the Unmanned Combat Air Vehicle (UCAV) to satisfy some Rules of Engagement (RoE) and international legal obligations. Price (2007) introduced the "virtual UAV leader" as a new concept for semi-autonomous control of UAVs that bridges the gap between conventional manual control of a single UAV and fully autonomous, cooperative control of UAVs in large numbers. While the aforementioned papers address semi-autonomy, Johnson et al. (2004) proposed the design, development, and testing of unmanned aerial vehicles with highly automated search capabilities where all functions are automated and human operator assistance is not required.

These developments have led Murray (2006) to report on military applications of cooperative control. Examples of cooperative control include multiple UAV path planning with collision avoidance and rendezvous for use in surveillance and air support missions (Leonard and Fiorelli (2001), McLain et al. (2001), Arslan et al. (2002), Bellingham et al. (2002), Yang and Zhao (2002), Rysdyk et al. (2005), and Rysdyk (2006)); search and acquisition for rescue operations (Ablavsky and Snorrason (2000), Polycarpou et al. (2001), Flint et al. (2002), Yang et al. (2002), and Yang et al. (2004)); and Suppression of Enemy Air Defense (SEAD) (Atkinson (2003)). Recently, Clifford et al. (2007) reported on the success of the Target Acquisition Cooperative (TAC) Unmanned Vehicle System (UVS) in which a group of autonomous vehicles worked cooperatively to identify and classify targets in a predefined target area.

While path planning, collision avoidance, rendezvous, search, and acquisition are important aspects of cooperative control, *classification* and *tracking* are fundamental in the author's application. Chandler et al. (2001) discussed cooperative target classification where multiple views are statistically combined using "joint probabilities" to maximize the probability of correct target classification over various aspect angles. Rao et al. (1993), Mutambara (1999), Julier et al. (2000), Brunke and Campbell (2004), and Whitacre and Campbell (2007) discussed various ways of combining the statistical information as well. This is referred to as "cooperative information fusion." Once a target is correctly classified, it must be continually tracked. In the context of spacecraft tracking for rendezvous and docking, Thienel et al. (2006) introduced the notion of "non-cooperative" targets, i.e. the target (to be docked with) does not transmit knowledge of its attitude or position. Compare this with other "targets" that do cooperate, e.g. a ground

convoy transmits its kinematic information to UAVs for support. Brown and Carter (2005), Frew and Lawrence (2005), Wise and Rysdyk (2006), and Frew et al. (2008) discussed tracking of a non-cooperative target while UAVs avoided collision as they loitered overhead. The purpose was to maintain a stand-off distance to avoid surface-to-air attack while the UAV successfully tracked a maneuvering target.

While target classification and tracking are current capabilities of UAVs, a natural extension is to utilize this information for command and control of an on-board light-weight, low-cost pursuer system. This will be one contributor toward designing a simple guidance algorithm to intercept a maneuvering, non-cooperative, target.

### 1.5 Approach

In an effort to construct a pursuer system for UAVs, a system-of-systems approach will be taken. Stevenson et al. (2007) commented on how pursuer systems and Intelligence, Surveillance, and Reconnaissance (ISR) systems are becoming more network-enabled. State-of-the-art systems technology enables improved military mission effectiveness and situational awareness. Collaborative on-board adaptive mission controllers improve mission management for teams of semi-autonomous entities. These controllers automate path planning, perform pursuer-target pairing, and optimize task allocation in real-time; all with minimum human intervention. The intent is to reduce the resource management and mission planning burden on the battle commander by using adaptive on-board targeting maps to do dynamic planning in real-time.

Because UAVs are capable of individual target classification and tracking, they are being designed to participate in “cooperative navigation,” where one UAV shares information with one another vehicle through active message passing. In this application,

one UAV tracks the maneuvering (or non-maneuvering) target and shares the information with the pursuer aboard another UAV. The pursuer estimates the target's kinematics by implementing a 3-state, discrete-time, H-infinity filter with inequality constraints placed on the estimates. Note that a maneuvering target has position, velocity, and acceleration kinematics while a non-maneuvering target possesses only position and velocity kinematics. Therefore, the H-infinity filter will assume noisy acceleration for estimating position and velocity in the case of a non-maneuvering target; and will assume noisy jerk (acceleration rate) for estimating position, velocity, and acceleration in the case of a maneuvering target. Once the UAV with the pursuer has confirmed the target and is in position to release the pursuer, the tracking UAV continues to communicate target location information directly with the pursuer for command guidance purposes. Later in the pursuer's trajectory, the tracking UAV becomes the designating UAV. At this time, the target is illuminated with laser energy for the pursuer to switch from the command guidance mode to semi-active laser mode, thus reducing the error in the maneuvering target's location leading to improve performance by reducing the miss distance.

Recall that there is still a guidance issue related to the light-weight, acceleration-limited pursuer to be dealt with. Because classical and modern guidance algorithms and adaptive/neural autopilot control laws require at least 10g of acceleration capability, a new guidance algorithm needs to be developed to intercept maneuvering surface targets. The algorithm will incorporate pursuit guidance at long ranges (because of its noise insensitivity) and an augmented form of proportional navigation guidance at short ranges (because of its accuracy). Pursuit guidance plus augmented proportional navigation guidance computes acceleration commands based on both algorithms and combines them

based on the field-of-view of the pursuer's seeker. At long ranges with a wide field-of-view, the target motion appears noisy and the accuracy requirement is low, therefore the pursuit guidance algorithm is implemented. At short ranges with a more narrow field-of-view, the target's motion increases the acceleration requirement on the pursuer; therefore the augmented proportional navigation guidance algorithm is used. It is proposed that the combined performance of the pursuit and augmented proportional navigation algorithm will be within the 5g acceleration requirement. Note that the augmented form of the proportional navigation guidance algorithm includes a term representing the acceleration estimate of the target. The augmented proportional navigation guidance algorithm will be constructed such that the original form of the proportional navigation guidance algorithm will be implemented for case of a non-maneuvering target at short ranges.

For completeness, there is a simple acceleration autopilot (flight control system) proposed by (Zipfel 2000) which controls the pursuer's airframe.

### 1.6 Scope of Research

The main objective of this work is to develop a robust estimator and an adaptive guidance law and simulate the performance as applied to Unmanned Air Vehicles. The following tasks are to be fulfilled in this research.

- Implementation of cooperative navigation where one UAV measures a (non-cooperative) target's position and shares the information with another UAV containing the pursuer; where the pursuer estimates the target's kinematics using a constrained, 3-state, discrete-time, H-infinity filter
- Implementation of an adaptive guidance algorithm to intercept a maneuvering target where the lateral acceleration requirement is less than 5g.

- Development of a high-fidelity simulation to support feasibility studies.

If these criteria can be achieved, then a recommendation may be made to incorporate this guidance algorithm with cooperative estimation into low-cost, light-weight, precision pursuer systems, with limited acceleration capability, currently in development.

## CHAPTER TWO: COOPERATIVE NAVIGATION

Target estimation is a procedure to estimate the kinematics of a maneuvering or non-maneuvering target. It is an important problem in target tracking due to the uncertainty in maneuvers of the target. In a hostile environment, the target will try to avoid being tracked by maneuvering in such a way so that its motion is difficult to follow. While classical approaches to estimation include linear weighted and unweighted least squares, these algorithms require all the data available beforehand. In this application, the data is measured sequentially. One could consider the recursive least squares algorithm, but this does not take the state dynamics into consideration. However, it is the precursor to the Kalman filter.

The conventional approach to estimation is to use a Kalman filter [Kalman (1960)] which minimizes the variance of the estimation error. However, the innovation in this application is to use an H-infinity filter and compare its performance to that of the Kalman filter. Please see Simon (2006) for greater detail on the development of the H-infinity filter.

In this section, both the discrete-time Kalman and H-infinity filters are presented. This is followed by the development of the target model with its associated kinematic limitations. Next, the implementation details of the Kalman and H-infinity filters are discussed. Finally, the simulation results are presented to compare the performance of the two filters.

## 2.1 Discrete-Time Kalman Filter

### 2.1.1 Recursive Least Squares

Although recursive least squares does not account for the state dynamics, the algorithm aims to achieve the same result of the Kalman filter, i.e. it minimizes the variance of the estimation error. It also sets the stage for estimating the state when all the measurements are not available ahead of time. As each new measurement is made available, the estimate  $\hat{\mathbf{x}}$  is updated. Suppose  $\hat{\mathbf{x}}$  is the estimate after time  $k-1$  measurements are taken. Then, a new measurement  $\mathbf{y}_k$  is obtained at time  $k$ . It is desired to update the estimate using only the new information from the measurement  $\mathbf{y}_k$ . This is because the size of the data grows with increasing measurements which would lead to processor overflow.

The recursive least squares algorithm is written as

$$\begin{aligned}\mathbf{y}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1})\end{aligned}\tag{2.1}$$

where  $\mathbf{H}_k$  determines which states are being measured,  $\mathbf{v}_k$  is the measurement noise,  $\hat{\mathbf{x}}_k$  is the state estimate,  $\mathbf{K}_k$  is the gain and the quantity  $(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1})$  is the difference between the measurement and the state estimation, named as the measurement residual error.

In order to optimize  $\mathbf{K}_k$ , the sum of the variances of the estimation error  $\mathbf{x} - \hat{\mathbf{x}}$  is minimize where the cost function  $J_k$  is defined as



$$\begin{aligned}
J_k &= E[(\mathbf{x}_1 - \hat{\mathbf{x}}_1)^2] + \dots + E[(\mathbf{x}_n - \hat{\mathbf{x}}_n)^2] \\
&= E(\varepsilon_1^2 + \dots + \varepsilon_n^2) \\
&= E(\boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon}) \\
&= E[\text{Tr}(\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T)] \\
&= \text{Tr} \mathbf{P}_k
\end{aligned} \tag{2.2}$$

where  $n$  is the dimension of the vectors  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  and  $\mathbf{P}_k$  called the estimation error covariance matrix, is symmetric. At this point, a recursive formula for  $\mathbf{P}_k$  is developed to better define the cost function  $J_k$ .

$$\begin{aligned}
\mathbf{P}_k &= E(\boldsymbol{\varepsilon} \boldsymbol{\varepsilon}^T) \\
&= E\{[(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \boldsymbol{\varepsilon}_{k-1} - \mathbf{K}_k \mathbf{v}_k][(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \boldsymbol{\varepsilon}_{k-1} - \mathbf{K}_k \mathbf{v}_k]^T\} \\
&= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) E(\boldsymbol{\varepsilon}_{k-1} \boldsymbol{\varepsilon}_{k-1}^T) (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T - \\
&\quad \mathbf{K}_k E(\mathbf{v}_k \boldsymbol{\varepsilon}_{k-1}^T) (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T - \\
&\quad (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) E(\boldsymbol{\varepsilon}_{k-1} \mathbf{v}_k^T) \mathbf{K}_k^T + \mathbf{K}_k E(\mathbf{v}_k \mathbf{v}_k^T) \mathbf{K}_k^T
\end{aligned} \tag{2.3}$$

If  $\boldsymbol{\varepsilon}_{k-1}$  and  $\mathbf{v}_k$  are uncorrelated and  $\mathbf{v}_k$  is zero-mean

$$E(\mathbf{v}_k \boldsymbol{\varepsilon}_{k-1}^T) = E(\mathbf{v}_k) E(\boldsymbol{\varepsilon}_{k-1}) = 0 \tag{2.4}$$

Therefore, with  $\mathbf{P}_{k-1} = E(\boldsymbol{\varepsilon}_{k-1} \boldsymbol{\varepsilon}_{k-1}^T)$ , equation (2.3) becomes

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \tag{2.5}$$

where  $\mathbf{R}_k = E(\mathbf{v}_k \mathbf{v}_k^T)$  is the covariance of  $\mathbf{v}_k$ . This form of the covariance matrix is guaranteed to be positive definite, provided the initial estimation error covariance matrix  $\mathbf{P}_{k-1}$  and the measurement covariance matrix  $\mathbf{R}_k$  are both positive definite. This is the recursive relationship used to calculate the covariance of the estimation error.

Returning to the optimization of  $\mathbf{K}_k$ , the partial derivative of  $J_k$  is computed and set it equal to zero

$$\frac{\partial J_k}{\partial \mathbf{K}_k} = 2(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k-1} (-\mathbf{H}_k^T) + 2\mathbf{K}_k \mathbf{R}_k = 0 \quad (2.6)$$

Solving this equation for  $\mathbf{K}_k$  results in

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k^T \mathbf{R}_k^{-1} \quad (2.7)$$

Equations (2.1), (2.5), and (2.7) represent the recursive least squares algorithm. It is useful for estimating a constant vector  $\hat{\mathbf{x}}$ . However, it is challenging if the vector  $\hat{\mathbf{x}}$  is time-dependent, which leads to the development of the Kalman filter.

### 2.1.2 The Discrete-Time Kalman Filter

The Kalman filter (Kalman (1960)) is an enhancement to the recursive least squares filter because it allows for estimation of a time-dependent vector. As it will be explained in the section, the Kalman filter incorporates the propagation of the state mean and state covariance through time.

Suppose a linear discrete-time dynamic system is given as follows:

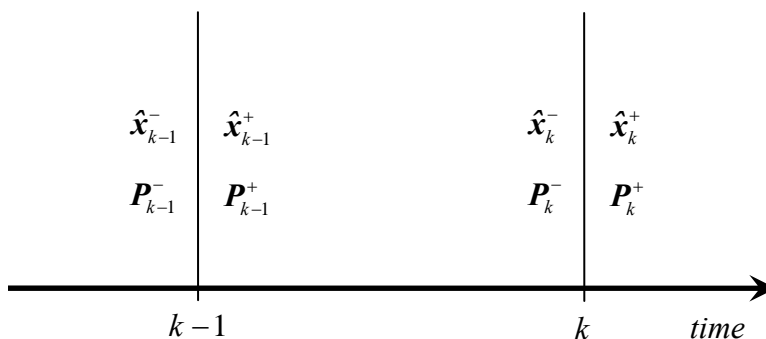
$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_{k-1} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{y}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \end{aligned} \quad (2.8)$$

where  $\mathbf{w}_{k-1}$  and  $\mathbf{v}_k$  are assumed to be zero-mean, uncorrelated, white noise with known covariance matrices  $\mathbf{Q}_k$  and  $\mathbf{R}_k$ , respectively

$$\begin{aligned} \mathbf{w}_k &\sim (0, \mathbf{Q}_k) \\ \mathbf{v}_k &\sim (0, \mathbf{R}_k) \\ E[\mathbf{w}_k \mathbf{w}_j^T] &= \mathbf{Q}_k \delta_{k-j} \\ E[\mathbf{v}_k \mathbf{v}_j^T] &= \mathbf{R}_k \delta_{k-j} \\ E[\mathbf{v}_k \mathbf{w}_j^T] &= 0 \end{aligned} \quad (2.9)$$

where  $\delta_{k-j}$  is the Kronecker delta function; that is  $\delta_{k-j} = 1$  if  $k = j$  and  $\delta_{k-j} = 0$  if  $k \neq j$ . As before, the task is to estimate  $\mathbf{x}_k$  based on the noisy measurements  $\mathbf{y}_k$ .

However, now the dynamic system will also be part of the estimation process. It is noted that if all of the measurements (including the measurement  $\mathbf{y}_k$  at time  $k$ ) are available to estimate  $\mathbf{x}_k$ , then an “*a posteriori*” estimate is computed, denoted by  $\hat{\mathbf{x}}_k^+$ . If all of the measurements (*not* including the measurement  $\mathbf{y}_k$  at time  $k$ ) are available to estimate  $\mathbf{x}_k$ , then an “*a priori*” estimate is computed, denoted by  $\hat{\mathbf{x}}_k^-$  (figure 1).



**Figure 1** Time and Measurement Updates

Furthermore, it is noted that both  $\hat{\mathbf{x}}_k^-$  and  $\hat{\mathbf{x}}_k^+$  are estimates of  $\mathbf{x}_k$ . However, the estimate  $\hat{\mathbf{x}}_k^+$  is better than the estimate  $\hat{\mathbf{x}}_k^-$  because  $\hat{\mathbf{x}}_k^+$  incorporates the most recent measurement  $\mathbf{y}_k$ .  $\hat{\mathbf{x}}_0^+$  represents the initial estimate or the expected value of the initial state of  $\mathbf{x}_0$

$$\hat{\mathbf{x}}_0^+ = E[\mathbf{x}_0] \quad (2.10)$$

In general,  $\mathbf{P}_k$  represents the covariance of the estimation error. In particular,  $\mathbf{P}_k^-$  represents the covariance of the estimation error of  $\hat{\mathbf{x}}_k^-$ , while  $\mathbf{P}_k^+$  represents the covariance of the estimation error of  $\hat{\mathbf{x}}_k^+$

$$\begin{aligned} \mathbf{P}_k^- &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \\ \mathbf{P}_k^+ &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)(\mathbf{x}_k - \hat{\mathbf{x}}_k^+)^T] \end{aligned} \quad (2.11)$$

Figure 1 shows that for the measurement at time  $k-1$ , the estimate  $\hat{\mathbf{x}}_{k-1}^+$  and the covariance of the estimate error  $\mathbf{P}_{k-1}^+$  are computed. This is known as the *measurement update at time  $k-1$* . Between time  $k-1$  and time  $k$ , the estimate  $\hat{\mathbf{x}}_{k-1}^+$  and the covariance of the estimate error  $\mathbf{P}_{k-1}^+$  are propagated to  $\hat{\mathbf{x}}_k^-$  and  $\mathbf{P}_k^-$ , respectively. This is known as the *time update* from  $k-1$  to  $k$ . Then, at time  $k$ , the estimate  $\hat{\mathbf{x}}_k^+$  and the covariance of the estimate error  $\mathbf{P}_k^+$  are computed. This is known as the *measurement update at time  $k$* .

The estimation process starts with the initial state estimate  $\hat{\mathbf{x}}_0^+$ . From equation (2.8) the mean of  $\mathbf{x}$  propagates with time as

$$\bar{\mathbf{x}}_k = \mathbf{F}_{k-1} \bar{\mathbf{x}}_{k-1} \quad (2.12)$$

since  $\mathbf{w}_k$  is assumed to be zero-mean. Therefore, the state estimate is propagated from  $k-1$  to  $k$  as

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1}^+ \quad (2.13)$$

This is the time update equation for the state estimate. Therefore, using the initial state estimate  $\hat{\mathbf{x}}_0^+$ , the time update  $\hat{\mathbf{x}}_1^-$  is obtained from equation (2.13) with  $k=1$ .

The time update for the covariance of the estimate error starts with the initial covariance  $\mathbf{P}_0^+$ . Incidentally, if  $\mathbf{x}_0$  is known perfectly, then  $\mathbf{P}_0^+ = \mathbf{0}$ ; if  $\mathbf{x}_0$  is unknown, then  $\mathbf{P}_0^+ = \infty \mathbf{I}$ , where  $\mathbf{P}_0^+$  is the uncertainty in the initial estimate

$$\mathbf{P}_0^+ = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T] \quad (2.14)$$

In general,

$$\begin{aligned}
\mathbf{P}_k &= E[(\mathbf{x}_k - \bar{\mathbf{x}}_k)(\mathbf{x}_k - \bar{\mathbf{x}}_k)^T] \\
&= E[(\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} - \bar{\mathbf{x}}_k)(\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} - \bar{\mathbf{x}}_k)^T]
\end{aligned} \tag{2.15}$$

Substituting equation (2.12) for  $\bar{\mathbf{x}}_k$  yields

$$\begin{aligned}
\mathbf{P}_k &= E[(\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} - \mathbf{F}_{k-1}\bar{\mathbf{x}}_{k-1})(\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} - \mathbf{F}_{k-1}\bar{\mathbf{x}}_{k-1})^T] \\
&= E[(\mathbf{F}_{k-1}(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1}) + \mathbf{w}_{k-1})(\mathbf{F}_{k-1}(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1}) + \mathbf{w}_{k-1})^T] \\
&= E[\mathbf{F}_{k-1}(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1})(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1})^T \mathbf{F}_{k-1}^T + \mathbf{F}_{k-1}(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1})\mathbf{w}_{k-1}^T \\
&\quad + \mathbf{w}_{k-1}(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1})^T \mathbf{F}_{k-1}^T + \mathbf{w}_{k-1}\mathbf{w}_{k-1}^T]
\end{aligned} \tag{2.16}$$

Because the estimation error  $\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1}$  and the process noise  $\mathbf{w}_{k-1}$  are uncorrelated, the covariance of the estimation error propagates with time as

$$\mathbf{P}_k = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{2.17}$$

where  $\mathbf{Q}_{k-1}$  and  $\mathbf{P}_{k-1}$  are defined as  $E[\mathbf{w}_{k-1}\mathbf{w}_{k-1}^T]$  and  $E[(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1})(\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1})^T]$ , respectively. Therefore, the estimation error covariance is propagated from  $k-1$  to  $k$  by the following equation

$$\mathbf{P}_k^- = \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{2.18}$$

This is the time update equation for the propagation of the estimate error. Therefore, using the initial state estimate  $\mathbf{P}_0^+$ , the time update  $\mathbf{P}_1^-$  is obtained from equation (2.13) with  $k=1$ .

Next, the *measurement update* equations are derived to take  $\hat{\mathbf{x}}_k^-$  to  $\hat{\mathbf{x}}_k^+$  and  $\mathbf{P}_k^-$  to  $\mathbf{P}_k^+$ . The measurement  $\mathbf{y}_k$  changes the estimation based on the recursive least squares algorithm: equations (2.7), (2.1), and (2.5)

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_k \mathbf{H}_k^T \mathbf{R}_k^{-1} \\
\hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k-1}) \\
\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T
\end{aligned} \tag{2.19}$$

where  $\hat{\mathbf{x}}_{k-1}$  and  $\mathbf{P}_{k-1}$  are the estimate and the covariance of the estimate error *before* the measurement  $\mathbf{y}_k$ , and  $\hat{\mathbf{x}}_k$  and  $\mathbf{P}_k$  are the estimate and the covariance of the estimate error *after* the measurement  $\mathbf{y}_k$ . Analogously,  $\hat{\mathbf{x}}_k^-$  and  $\mathbf{P}_k^-$  are the estimate and the covariance of the estimate error *before* the measurement  $\mathbf{y}_k$ , and  $\hat{\mathbf{x}}_k^+$  and  $\mathbf{P}_k^+$  are the estimate and the covariance of the estimate error *after* the measurement  $\mathbf{y}_k$ . These relationships are shown in Table 1 below.

<b>Recursive Least Squares Filtering</b>	<b>Kalman Filtering</b>
$\hat{\mathbf{x}}_{k-1}$ = estimate before $\mathbf{y}_k$ is processed	$\hat{\mathbf{x}}_k^-$ = a priori estimate
$\mathbf{P}_{k-1}$ = covariance before $\mathbf{y}_k$ is processed	$\mathbf{P}_k^-$ = a priori covariance
$\hat{\mathbf{x}}_k$ = estimate after $\mathbf{y}_k$ is processed	$\hat{\mathbf{x}}_k^+$ = a posteriori estimate
$\mathbf{P}_k$ = covariance after $\mathbf{y}_k$ is processed	$\mathbf{P}_k^+$ = a posteriori covariance

By replacing  $\hat{\mathbf{x}}_{k-1}^-$  with  $\hat{\mathbf{x}}_k^-$ ,  $\mathbf{P}_{k-1}$  with  $\mathbf{P}_k^-$ ,  $\hat{\mathbf{x}}_k$  with  $\hat{\mathbf{x}}_k^+$ , and  $\mathbf{P}_k$  with  $\mathbf{P}_k^+$ ,

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_k^+ \mathbf{H}_k^T \mathbf{R}_k^{-1} \\
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \\
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T
\end{aligned} \tag{2.20}$$

the *measurement update* equations for  $\hat{\mathbf{x}}_k$  and  $\mathbf{P}_k$  are obtained. The matrix  $\mathbf{K}_k$  is called the Kalman filter gain.

A summary of the Kalman filter includes both the time update equations and the measurement update equations from equations (2.13), (2.18), and (2.20) or

$$\begin{aligned}
\hat{\mathbf{x}}_k^- &= \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1}^+ \\
\mathbf{P}_k^- &= \mathbf{F}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \\
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{R}_k^{-1} \\
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (y_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \\
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T
\end{aligned} \tag{2.21}$$

Recall that there were assumptions associated with the derivation of the Kalman filter. In particular equation (2.5) was developed assuming  $\boldsymbol{\varepsilon}_{k-1}$  and  $\mathbf{v}_k$  were uncorrelated and  $\mathbf{v}_k$  was zero-mean and equation (2.13) was developed assuming  $\mathbf{w}_k$  was zero-mean. Additionally, if the statistics of the process noise  $\mathbf{w}_k$  and the measurement noise  $\mathbf{v}_k$  are unknown, then the performance of the Kalman filter will be suboptimal.

While the Kalman filter performs well if the noise statistics are known, it is desired to have a filter that is more robust to uncertainties in the noise statistics. Simon (2000) notes that the Kalman filter is also called the  $H_2$  filter because it minimizes the two-norm of the transfer function from the process noise  $\mathbf{w}_k$  to the estimation error  $\mathbf{x}_k - \hat{\mathbf{x}}_k^-$ .

## 2.2 Discrete-Time H-infinity Filter

The Kalman filter is effective for estimating the states of a system when the noise statistics are known and the assumptions associated with the Kalman filter are satisfied. However, modern applications need a filtering algorithm that is robust with regard to various process noises (dynamic system modeling errors and disturbances) and measurement noise. The superiority of the H-infinity filter over the Kalman filter is that it *does not make any assumptions* about process noise or measurement noise while minimizing the worst case estimation error.

Suppose a linear discrete-time dynamic system is given as follows:

$$\begin{aligned}\mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{y}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k\end{aligned}\tag{2.22}$$

where  $\mathbf{x}_k$  is the state vector,  $\mathbf{y}_k$  is the output vector,  $\mathbf{F}_{k-1}$  is called the system matrix, and  $\mathbf{H}_k$  is called the output matrix. The statistics of the noise processes  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are *unknown*.

The objective of the H-infinity filter is to find a state estimate  $\hat{\mathbf{x}}_k$  that will minimize the worst possible effect that  $\mathbf{w}_k$  and  $\mathbf{v}_k$  have on the estimation error  $\mathbf{x}_k - \hat{\mathbf{x}}_k$ . While the H-infinity filter is trying to minimize the estimation error,  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are conspiring against this objective. In mathematical notation, this is written as  $\min_{\hat{\mathbf{x}}} \max_{\mathbf{w}, \mathbf{v}} J$  where  $J$  is some measure of how well the estimator is performing. If  $\mathbf{w}_k$  and  $\mathbf{v}_k$  were large, then their mission would be accomplished. Moreover, in order to make this method more meaningful, limits are placed on  $\mathbf{w}_k$  and  $\mathbf{v}_k$ .

$$\begin{aligned}\sum_{k=0}^{N-1} \mathbf{w}_k^T \mathbf{Q}_k^{-1} \mathbf{w}_k &\leq W \\ \sum_{k=0}^{N-1} \mathbf{v}_k^T \mathbf{R}_k^{-1} \mathbf{v}_k &\leq V\end{aligned}\tag{2.23}$$

where  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  are positive definite matrices selected by the designer. For example, the designer could use the covariance of the process noise and measurement noise, respectively. The discrete-time performance index is given by

$$(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{S}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k)\tag{2.24}$$

where  $\mathbf{S}_k$  is also a positive definite matrix selected by the designer. In this application, the designer uses a diagonal matrix where the elements are weighted to reflect the



importance of each state to be estimated. Combining the constraints from equation (2.23) to the performance index from equation (2.24) yields the discrete-time cost function

$$J_0 = \frac{\frac{1}{2} \sum_{k=0}^{N-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{S}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k)}{\frac{1}{2} \sum_{k=0}^{N-1} (\mathbf{w}_k^T \mathbf{Q}_k^{-1} \mathbf{w}_k + \mathbf{v}_k^T \mathbf{R}_k^{-1} \mathbf{v}_k)} \quad (2.25)$$

While this problem poses difficulties in attempting a solution, a related problem can be solved. It is desired to find an estimator such that

$$J_0 < 1/\theta \quad (2.26)$$

where  $\theta$  is some constant selected by the designer. A state estimate can be found so that the maximum value of  $J_0$  is always less than  $1/\theta$  regardless of the process noise  $\mathbf{w}_k$  and measurement noise  $\mathbf{v}_k$ . Combining equation (2.25) with equation (2.26) yields

$$J_0 = \frac{\frac{1}{2} \sum_{k=0}^{N-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{S}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k)}{\frac{1}{2} \sum_{k=0}^{N-1} (\mathbf{w}_k^T \mathbf{Q}_k^{-1} \mathbf{w}_k + \mathbf{v}_k^T \mathbf{R}_k^{-1} \mathbf{v}_k)} < \frac{1}{\theta} \quad (2.27)$$

Cross multiplying and simplifying yields

$$\frac{1}{2} \sum_{k=0}^{N-1} (\mathbf{w}_k^T \mathbf{Q}_k^{-1} \mathbf{w}_k + \mathbf{v}_k^T \mathbf{R}_k^{-1} \mathbf{v}_k) - \frac{1}{2} \theta \sum_{k=0}^{N-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{S}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k) > 0 \quad (2.28)$$

The state estimate that solves this problem is found by letting the Hamiltonian,  $H$ , be defined as the adjoint of the cost function through the use of the Lagrange multiplier  $\boldsymbol{\lambda}_k$ , equation (2.28) with the state equation(2.22), that is

$$H = \frac{1}{2} \mathbf{w}_k^T \mathbf{Q}_k^{-1} \mathbf{w}_k + \frac{1}{2} \mathbf{v}_k^T \mathbf{R}_k^{-1} \mathbf{v}_k - \frac{1}{2} \theta (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{S}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k) + \boldsymbol{\lambda}_k^T (\mathbf{F}_{k-1} \mathbf{x}_{k-1} + \mathbf{w}_{k-1}) \quad (2.29)$$

The solution of this problem is based on optimal control, Bryson and Ho (1975).

Substituting  $\mathbf{v}_k$  from equation (2.22) into equation (2.29) yields

$$\begin{aligned}
H = & \frac{1}{2} \mathbf{w}_k^T \mathbf{Q}_k^{-1} \mathbf{w}_k + \frac{1}{2} (\mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k)^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathbf{H}_k \mathbf{x}_k) \\
& - \frac{1}{2} \theta (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{S}_k (\mathbf{x}_k - \hat{\mathbf{x}}_k) + \boldsymbol{\lambda}_k^T (\mathbf{F}_{k-1} \mathbf{x}_{k-1} + \mathbf{w}_{k-1})
\end{aligned} \tag{2.30}$$

Next, the partial derivative of the Hamiltonian  $H$  is computed with respect to  $\mathbf{x}_k$  to obtain the Lagrange multiplier  $\boldsymbol{\lambda}_k$ , that is

$$\boldsymbol{\lambda}_{k+1}^T = - \frac{\partial H}{\partial \mathbf{x}_k} \tag{2.31}$$

where the negative sign is introduced to compute the *minimum* of  $H$ . After computation and simplification, this yields

$$\boldsymbol{\lambda}_{k+1} = \theta \mathbf{S}_k \mathbf{x}_k - \theta \mathbf{S}_k \hat{\mathbf{x}}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k - \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{x}_k - \mathbf{F}_k^T \boldsymbol{\lambda}_k \tag{2.32}$$

Then, the partial derivative of the Hamiltonian  $H$  is computed with respect to  $\mathbf{w}_k$ , that is

$$0 = \frac{\partial H}{\partial \mathbf{w}_k} \tag{2.33}$$

Upon computation and simplification, this yields

$$\mathbf{w}_k = -\mathbf{Q}_k \boldsymbol{\lambda}_k \tag{2.34}$$

Substituting this result back into the state equation (2.22)

$$\mathbf{x}_k = \mathbf{F}_{k-1} \mathbf{x}_{k-1} - \mathbf{Q}_{k-1} \boldsymbol{\lambda}_{k-1} \tag{2.35}$$

Since this problem is linear, a linear solution is postulated as:

$$\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1} - \mathbf{P}_{k+1} \boldsymbol{\lambda}_{k+1} \tag{2.36}$$

where  $\hat{\mathbf{x}}_{k+1}$  and  $\mathbf{P}_{k+1}$  are to be determined.

Substituting equation (2.35) on the left side and equation (2.32) on the right side and substituting equation (2.36) for each occurrence of  $\mathbf{x}_k$ , yields the following equation in

$\hat{\mathbf{x}}_k$ ,  $\mathbf{y}_k$ , and  $\boldsymbol{\lambda}_k$

$$\begin{aligned}
F_k \hat{\mathbf{x}}_k - F_k \mathbf{P}_k \boldsymbol{\lambda}_k - \mathbf{Q}_k \boldsymbol{\lambda}_k &= \hat{\mathbf{x}}_{k+1} + \theta \mathbf{P}_{k+1} \mathbf{S}_k \mathbf{P}_k \boldsymbol{\lambda}_k - \mathbf{P}_{k+1} \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k \\
&+ \mathbf{P}_{k+1} \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \hat{\mathbf{x}}_k - \mathbf{P}_{k+1} \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k \boldsymbol{\lambda}_k + \mathbf{P}_k \mathbf{F}_k^T \boldsymbol{\lambda}_k - \mathbf{P}_{k+1} \boldsymbol{\lambda}_k
\end{aligned} \tag{2.37}$$

Collecting  $\boldsymbol{\lambda}_k$  terms and rearranging yields

$$\mathbf{P}_{k+1} = F_k \mathbf{P}_k [I - \theta \mathbf{S}_k \mathbf{P}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k]^{-1} F_k^T + \mathbf{Q}_k \tag{2.38}$$

Collecting  $\hat{\mathbf{x}}_k$  and  $\mathbf{y}_k$  terms and rearranging yields

$$\hat{\mathbf{x}}_{k+1} = F_k \hat{\mathbf{x}}_k + F_k \mathbf{P}_k (I - \theta \mathbf{S}_k \mathbf{P}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k)^{-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k) \tag{2.39}$$

Substituting equation (2.38) and equation (2.39) into equation (2.36) substantiates the initial postulate. Summarizing the H-infinity equations:

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_k [I - \theta \mathbf{S}_k \mathbf{P}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k]^{-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} \\
\hat{\mathbf{x}}_{k+1} &= F_k \hat{\mathbf{x}}_k + F_k \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k) \\
\mathbf{P}_{k+1} &= F_k \mathbf{P}_k [I - \theta \mathbf{S}_k \mathbf{P}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k]^{-1} F_k^T + \mathbf{Q}_k
\end{aligned} \tag{2.40}$$

Notice both  $\mathbf{K}_k$  and  $\mathbf{P}_{k+1}$  involve identical matrix inversions thus requiring that  $I - \theta \mathbf{S}_k \mathbf{P}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k > 0$ . However, this poses no problem because the calculations of  $\mathbf{K}_k$  and  $\mathbf{P}_{k+1}$  are independent of the measurements  $\mathbf{y}_k$ . Therefore, the possible existence of singularities may be checked at each time-step to make sure the matrix may be inverted. Furthermore, since both  $\mathbf{K}_k$  and  $\mathbf{P}_{k+1}$  reach steady-state values quickly, their calculations may even be taken off-line and hard-coded into the processing software ensuring there will be no matrix inversion problems.

By examining equation (2.40), one sees the term  $\theta \mathbf{S}_k \mathbf{P}_k$  is subtracted from both the  $\mathbf{K}_k$  and the  $\mathbf{P}_{k+1}$  calculation making the gain and the estimation error covariance matrix larger. This effectively places more emphasis on the measurements which is a way of making the H-infinity filter more robust to uncertainty in the process noise

(dynamic system modeling errors and disturbances) of the system model. In fact, the H-infinity filter was made to be robust by optimal design.

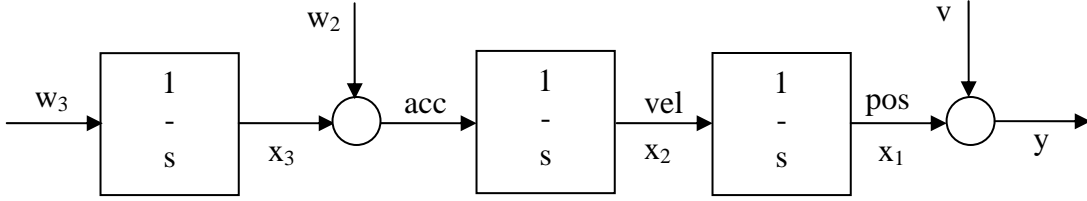
Having presented both the discrete-time Kalman filter and the discrete-time H-infinity filter, the next section will discuss how to apply these filters to target estimation.

### 2.3 Navigation Design

In this application, a UAV will estimate the maneuvering (or non-maneuvering) target's kinematics. A maneuvering target has position, velocity, and acceleration kinematics while a non-maneuvering target possesses only position and velocity kinematics. Therefore, a 3-state H-infinity filter will assume noisy acceleration for estimating position and velocity in the case of a non-maneuvering target; and will assume noisy jerk (acceleration rate) for estimating position, velocity, and acceleration in the case of a maneuvering target. Additionally, inequality constraints are placed on the H-infinity filter due to the limitations of the target's velocity and acceleration capabilities. In particular, the target's velocity is limited to 90 miles-per-hour and the target's acceleration is limited to  $\pm 0.27g$ . The performance of the H-infinity filter will be compared to the Kalman filter.

#### 2.3.1 Target Modeling

Consider a target whose acceleration and jerk are modeled by zero-mean process noise  $w_2$  and  $w_3$  with variances  $q_2$  and  $q_3$ , respectively. The target's position measurement  $y$  includes zero-mean noise  $v$  with variance  $r$  representing the sensor noise.



By letting  $x_1$ ,  $x_2$ , and  $x_3$  represent the position, velocity, and acceleration states, respectively, the three-state discrete-time representation of the system is

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= x_3 + w_2 \\
 \dot{x}_3 &= w_3 \\
 y &= x_1 + v
 \end{aligned} \tag{2.41}$$

In matrix form

$$\begin{aligned}
 \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ w_2 \\ w_3 \end{bmatrix} \\
 y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + v
 \end{aligned} \tag{2.42}$$

In general

$$\begin{aligned}
 \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{w} \\
 \mathbf{y} &= \mathbf{Cx} + \mathbf{v}
 \end{aligned} \tag{2.43}$$

In the discrete case,  $\mathbf{Q}_k$  was defined to be  $E[\mathbf{w}_k \mathbf{w}_k^T]$  after equation (2.17), therefore in the continuous case,  $\mathbf{Q}_c$  is defined to be  $E[\mathbf{w} \mathbf{w}^T]$ . The continuous-time process noise covariance matrix is represented by

$$\mathbf{Q}_c = \begin{bmatrix} 0 \\ q_2 \\ q_3 \end{bmatrix} \begin{bmatrix} 0 & q_2 & q_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & q_2^2 & q_2 q_3 \\ 0 & q_2 q_3 & q_3^2 \end{bmatrix} \tag{2.44}$$

Now, the discrete-time system is represented by

$$\begin{aligned}\mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{y}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k\end{aligned}\tag{2.45}$$

where

$$\mathbf{F}_k = \exp(\mathbf{A}\Delta t) = \mathbf{I} + \mathbf{A}\Delta t + \frac{1}{2}\mathbf{A}^2(\Delta t)^2 + \dots\tag{2.46}$$

$$\mathbf{w}_k = \int_0^{\Delta t} \exp[\mathbf{A}(\Delta t - \tau)]\mathbf{d}\tau\tag{2.47}$$

$$\mathbf{H}_k = \mathbf{C}\tag{2.48}$$

$$\mathbf{Q}_k = \int_0^{\Delta t} \mathbf{F}_{k-1}\mathbf{Q}_c\mathbf{F}_{k-1}^T\mathbf{d}\tau\tag{2.49}$$

Hence the three-state discrete-time system is represented by

$$\begin{aligned}\mathbf{x}_k &= \begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} \frac{1}{2}q_2(\Delta t)^2 + \frac{1}{6}q_3(\Delta t)^3 \\ q_2\Delta t + \frac{1}{2}q_3(\Delta t)^2 \\ q_3\Delta t \end{bmatrix} \\ \mathbf{y}_k &= [1 \ 0 \ 0]\mathbf{x}_{k-1} + \mathbf{v}_k\end{aligned}\tag{2.50}$$

such that the discrete-time system matrix

$$\mathbf{F}_k = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}\tag{2.51}$$

and

$$\mathbf{H}_k = [1 \ 0 \ 0]\tag{2.52}$$

The discrete-time process noise covariance matrix is represented by

$$\mathbf{Q}_k = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{bmatrix}\tag{2.53}$$

where

$$\begin{aligned}
Q_{11} &= \frac{1}{3}q_2^2(\Delta t)^3 + \frac{1}{4}q_{23}(\Delta t)^4 + \frac{1}{20}q_3^2(\Delta t)^5 \\
Q_{12} = Q_{21} &= \frac{1}{2}q_2^2(\Delta t)^2 + \frac{1}{2}q_{23}(\Delta t)^3 + \frac{1}{8}q_3^2(\Delta t)^4 \\
Q_{13} = Q_{31} &= \frac{1}{2}q_{23}(\Delta t)^2 + \frac{1}{6}q_3^2(\Delta t)^3 \\
Q_{22} &= q_2^2\Delta t + q_{23}(\Delta t)^2 + \frac{1}{3}q_3^2(\Delta t)^3 \\
Q_{23} = Q_{32} &= q_{23}\Delta t + \frac{1}{2}q_3^2(\Delta t)^2 \\
Q_{33} &= q_3^2\Delta t
\end{aligned} \tag{2.54}$$

and the discrete-time measurement noise covariance is represented by

$$R_k = r \tag{2.55}$$

### 2.3.2 Target Kinematic Limitations

Before implementing the filters, it is worthwhile to briefly discuss the characteristics of moving targets. Roemerman (2006) published information from a moving target study which placed limits on the velocity, acceleration, and Circular Error Probable (CEP) for moving targets. Recall Fleeman (2006) defined CEP as approximately equal to the  $1\sigma$  miss distance. In the Roemerman (2006) report, the maneuvering target engagement velocity was limited to 90 miles-per-hour or less and the total acceleration (longitudinal and lateral) was limited to less than 0.27g. Therefore, when it comes to designing an estimator for velocity and acceleration, this limiting information may be used to constrain the filter's estimate. Also to be noted from the Roemerman (2006) report is the requirement that the pursuer accuracy (CEP) must be less than 3m in miss distance. (This will be used later to measure the success of the guidance design.)

One may call into question whether limiting the kinematic estimates in this manner is viable. Consider another approach taken by Simon (2006) where the H-infinity problem is solved, subjected to inequality constraints associated with the estimate. This

algorithm locates the optimal estimate that meets the inequality constraint at each point along the trajectory. However, this requires an optimization routine, such as Matlab's `fmincon` function to compute the solution. The problem with this approach for a real-time system is that the computation of time of the optimal solution is indeterminate.

### 2.3.3 Discrete-Time Kalman Filter Implementation

The algorithm is implemented by first initializing the Kalman filter

$$\begin{aligned}\hat{\mathbf{x}}_0^+ &= E[\mathbf{x}_0] \\ \mathbf{P}_0^+ &= E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T]\end{aligned}\tag{2.56}$$

where  $E[\mathbf{x}_0]$  is the expectation of  $\mathbf{x}_0$  and  $E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T]$  is the expectation of  $(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T$ . If  $E[\mathbf{x}_0]$  is unknown, use  $\hat{\mathbf{x}}_0^+ = 0$  as the initial value and if  $E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)(\mathbf{x}_0 - \hat{\mathbf{x}}_0^+)^T]$  is unknown, use  $\mathbf{P}_0^+ \rightarrow \infty$  as the initial value. In this application, the initial target position will be known with some nominal amount of error.

Then, the following equations are computed at each time step  $k$ ,

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}^+ \\ \mathbf{P}_k^- &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \\ \mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{R}_k^{-1} \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(y_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T\end{aligned}\tag{2.57}$$

where the subscript  $k$  represents time at  $t = t_k$  and the subscript  $k - 1$  represents time at  $t = t_{k-1}$ . Also, note that if  $t_0 = t_{k-1}$  and  $t = t_k$  then  $\Delta t = t_k - t_{k-1}$ .

The Kalman filter of equation (2.57) is implemented with  $\mathbf{F}_k$  defined in equation (2.51),  $\mathbf{H}_k$  defined in equation (2.52),  $\mathbf{Q}_k$  defined in equation(2.53), and  $\mathbf{R}_k$  defined in equation (2.55).



### 2.3.4 Discrete-Time H-infinity Filter Implementation

The same as the case with the Kalman filter, the H-infinity filter is initialized with the target's position including some error, i.e.  $\mathbf{x}_0$  and  $\mathbf{P}_0$ .

At each time step  $k$ , the following equations are computed

$$\begin{aligned}\mathbf{P}_k &= \mathbf{F}_{k-1} \mathbf{P}_{k-1} [\mathbf{I} - \theta \mathbf{S}_{k-1} \mathbf{P}_{k-1} + \mathbf{H}_{k-1}^T \mathbf{R}_{k-1}^{-1} \mathbf{H}_{k-1} \mathbf{P}_{k-1}]^{-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \\ \mathbf{K}_k &= \mathbf{P}_k [\mathbf{I} - \theta \mathbf{S}_k \mathbf{P}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k]^{-1} \mathbf{H}_k^T \mathbf{R}_k^{-1} \\ \hat{\mathbf{x}}_k &= \mathbf{F}_{k-1} \hat{\mathbf{x}}_{k-1} + \mathbf{F}_{k-1} \mathbf{K}_{k-1} (\mathbf{y}_{k-1} - \mathbf{H}_{k-1} \hat{\mathbf{x}}_{k-1})\end{aligned}\quad (2.58)$$

where the subscript  $k+1$  represents time at  $t = t_{k+1}$  and the subscript  $k$  represents time at  $t = t_k$ . Also, note that if  $t_0 = t_k$  and  $t = t_{k+1}$  then  $\Delta t = t_{k+1} - t_k$ . The matrix  $\mathbf{S}_k$  determines which states of the system are to be estimated. Whether the target is maneuvering or not, estimations of both velocity and acceleration are needed. In state form, velocity and acceleration are represented by  $\mathbf{x}_2$  and  $\mathbf{x}_3$ , respectively. Therefore,

$$\mathbf{S}_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\quad (2.59)$$

The H-infinity filter of equation (2.58) is implemented with  $\mathbf{F}_k$  defined in equation (2.51),  $\mathbf{H}_k$  defined in equation (2.52),  $\mathbf{Q}_k$  defined in equation (2.53),  $\mathbf{R}_k$  defined in equation (2.55), and  $\mathbf{S}_k$  defined in equation (2.59).

The only remaining parameter is  $\theta$  which is the cost function bound specified by the designer. Recall the matrix inversion requirement for the H-infinity filter, i.e.  $\mathbf{I} - \theta \mathbf{S}_k \mathbf{P}_k + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k > 0$ . This equation can be used to place limitations on  $\theta$ . In fact, using  $\mathbf{S}_{k-1}$  defined in equation (2.59),  $\mathbf{H}_k$  defined in equation (2.52), and  $\mathbf{R}_k$  defined in equation (2.55),  $\theta$  must be less than  $P_{22}^{-1}$  and less than  $P_{33}^{-1}$ .

## 2.4 Simulation Results

### 2.4.1 Target, UAV, and Pursuer Modeling Parameters

The target is modeled as a surface vehicle with an initial position and velocity. Its acceleration and jerk are modeled as process noise with magnitudes  $1\text{m/sec}^2$  and  $0.1\text{m/sec}^3$ , respectively.

The UAVs are modeled to operate at an altitude of 20,000 feet with 200 knots true air speed. The UAV sensors are modeled to measure position within a resolution of  $\pm 5\text{m}$ . The time-delay used for communicating estimation information to either the UAV containing the pursuer or the pursuer itself is 500msec.

The pursuer's mass properties are company proprietary information due to the current competitive nature of the market. However, suffice it to say that its form factor would allow high-density carriage, i.e. multiple pursuers per UAV platform.

The position estimates are initialized to the target's *actual* position plus some normally distributed random error representing uncertainty; while the velocity and acceleration estimates are initialized to zero.

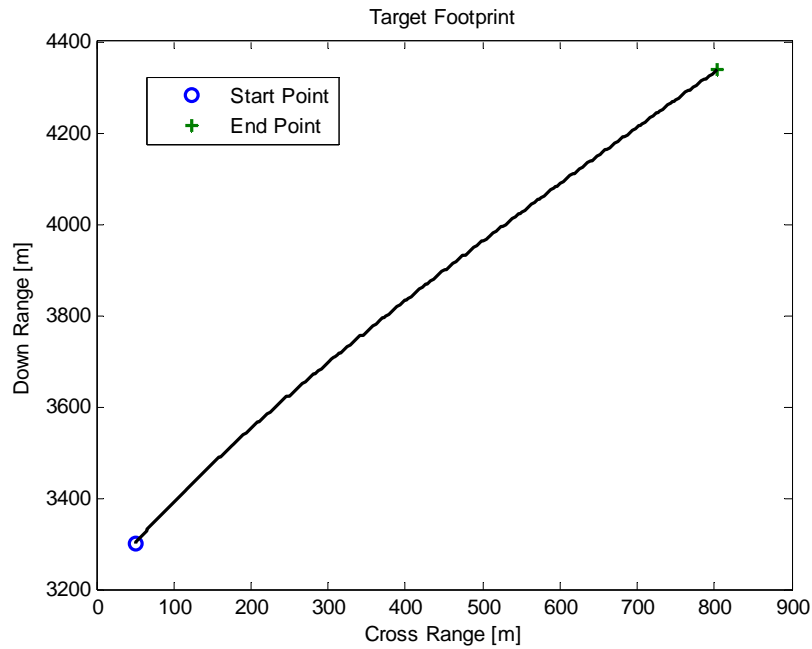
Finally, the tuning parameter  $\theta$  for the H-infinity filter is set to  $1\text{E}-2$ .

### 2.4.2 Simulation Case One: Zero-Mean Noise Statistics

The process noises (jerk and acceleration) are initialized to be unbiased, i.e. with zero-mean. The error covariance matrix,  $\mathbf{P}$ , is initialized with the uncertainties in the position (5m), velocity (3m/sec), and acceleration ( $1\text{m/sec}^2$ ); while the off-diagonal elements are set to zero.

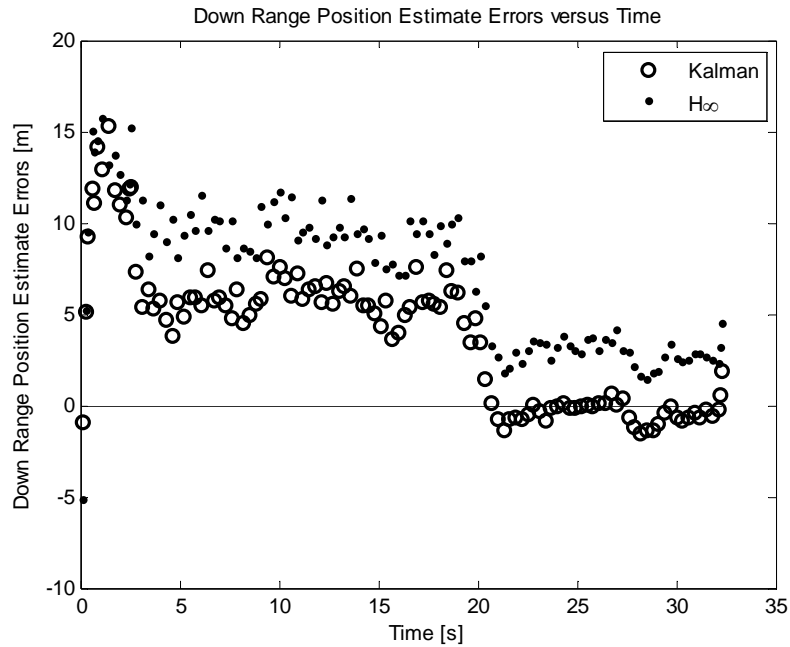
$$\mathbf{P} = \begin{bmatrix} 5^2 & 0 & 0 \\ 0 & 3^2 & 0 \\ 0 & 0 & 1^2 \end{bmatrix} = \begin{bmatrix} 25 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.60)$$

Figure 2 displays the target's motion when driven by noisy jerk and acceleration.

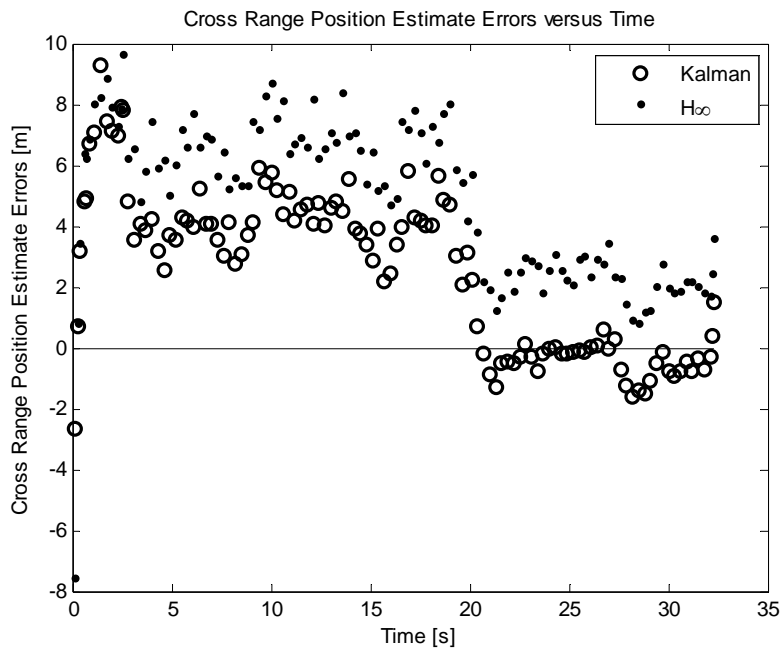


**Figure 2 Target Footprint**

Figure 3 and figure 4 display the relative error between the actual and the estimated down range and cross range position for both the Kalman and the H-infinity filters.



**Figure 3 Relative Down Range Position Errors**



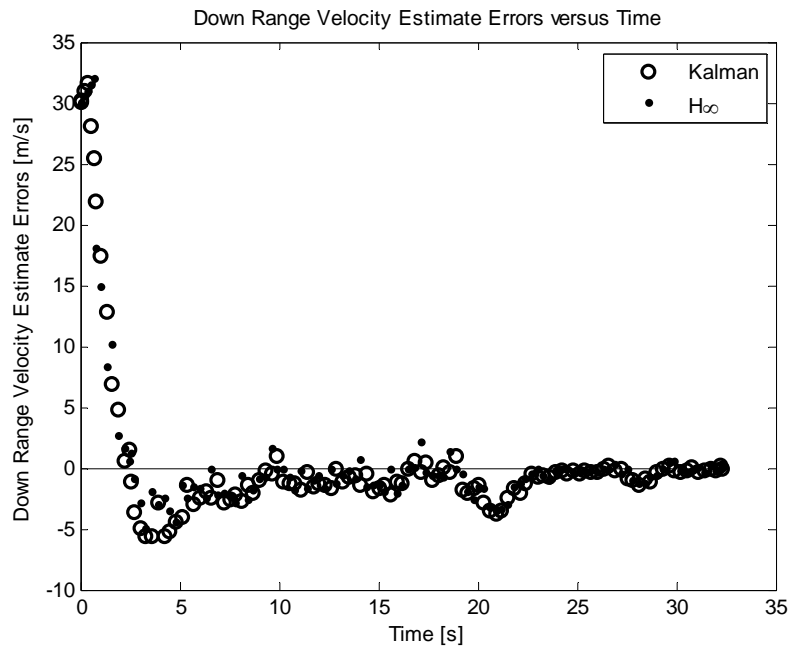
**Figure 4 Relative Cross Range Position Errors**

Table 2 below shows the Kalman filter outperforms the H-infinity filter. This is expected because the noise statistics were initialized without any biases which are precisely the assumptions associated with the *optimal* development of the Kalman filter algorithm.

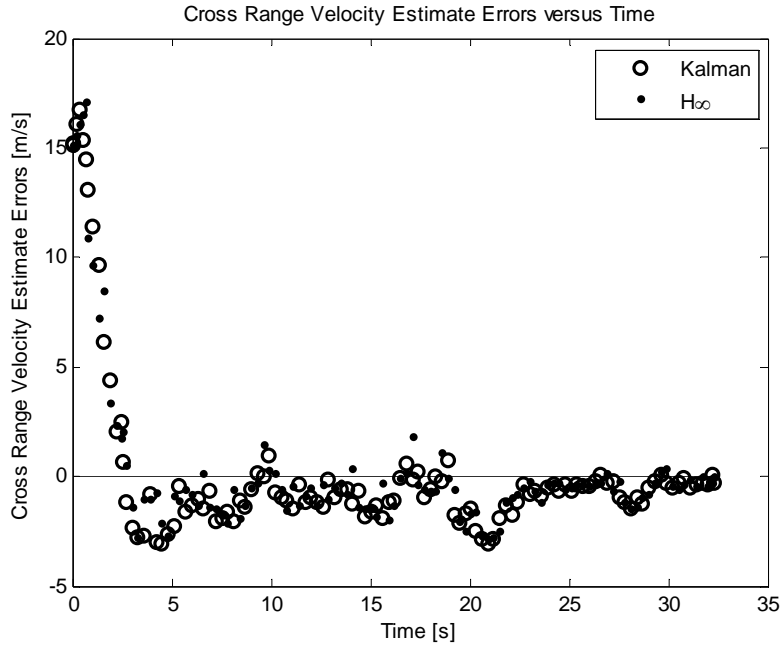
**Table 2 RMS Position Errors for Zero-Mean Noise**

	Down Range	Cross Range
	Position Error (m)	Position Error (m)
Kalman	0.4	0.3
H-infinity	1.9	1.5

Figure 5 and figure 6 display the relative error between the actual and the estimated down range and cross range velocity for both the Kalman and the H-infinity filters.

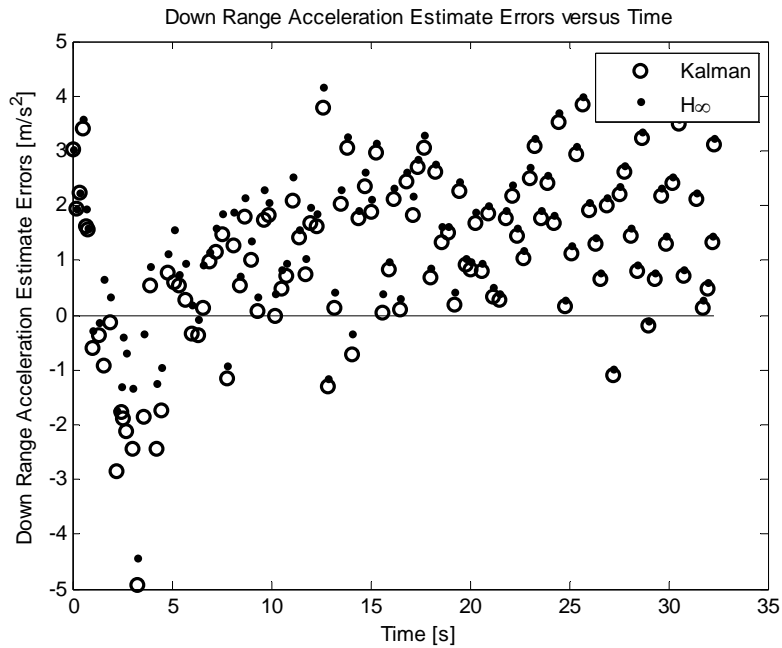


**Figure 5 Relative Down Range Velocity Errors**

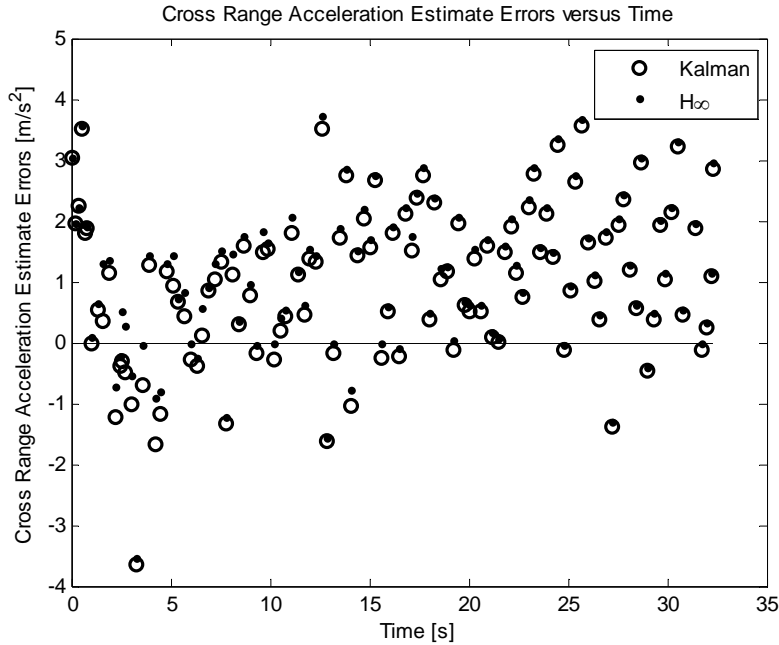


**Figure 6** Relative Cross Range Velocity Errors

Figure 7 and figure 8 display the relative error between the actual and the estimated down range and cross range acceleration for both the Kalman and the H-infinity filters.



**Figure 7** Relative Down Range Acceleration Errors



**Figure 8** Relative Cross Range Acceleration Errors

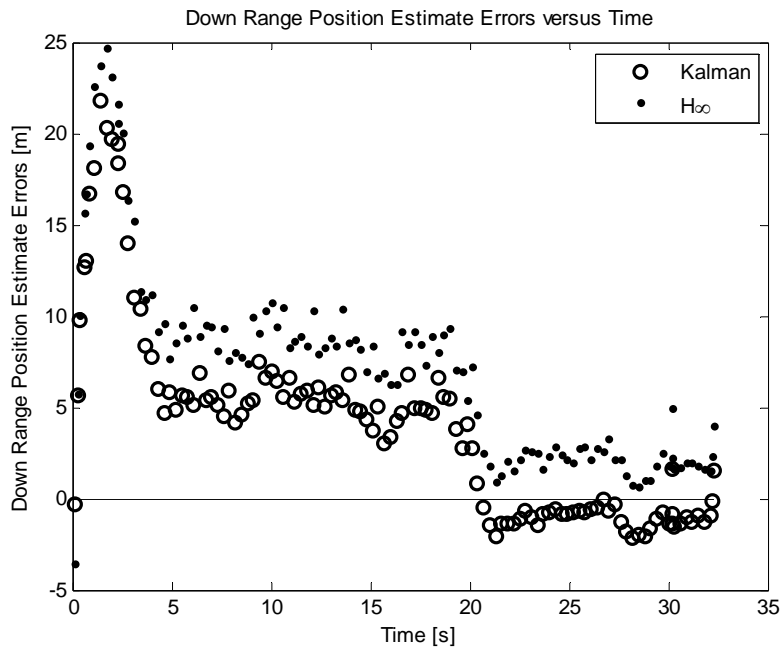
The significance of examining the estimated acceleration is because, in the next chapter, it will be shown that the terminal guidance system depends on an estimate of the target's acceleration.

#### 2.4.3 Simulation Case Two: Biased Noise Statistics

The process noises (jerk and acceleration) and the measurement noise are now initialized *with* biases, i.e. non-zero-mean statistics. The process noise biases are  $0.01\text{m/s}^3$  in jerk and  $1\text{m/s}^2$  in acceleration. Recall the error covariance matrix,  $\mathbf{P}$ , is initialized with the uncertainties in the position (5m), velocity (3m/sec), and acceleration ( $1\text{m/sec}^2$ ). In the previous case considered, the off-diagonal elements of the error covariance matrix,  $\mathbf{P}$ , were set to zero, implying the estimation errors are uncorrelated. If the error covariances of the states are not independent, then the estimation error in position and velocity, position and acceleration, and velocity and acceleration are correlated.

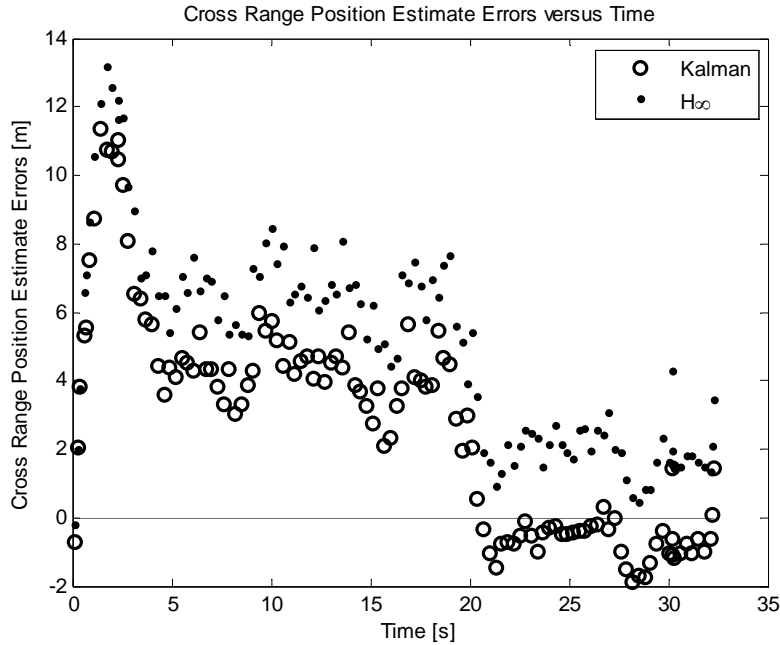
$$\mathbf{P} = \begin{bmatrix} 5*5 & 5*3 & 5*1 \\ 3*5 & 3*3 & 3*1 \\ 1*5 & 1*3 & 1*1 \end{bmatrix} = \begin{bmatrix} 25 & 15 & 5 \\ 15 & 9 & 3 \\ 5 & 3 & 1 \end{bmatrix} \quad (2.61)$$

Figure 9 and figure 10 display the relative error between the actual and the estimated down range and cross range position for both the Kalman and the H-infinity filters.



**Figure 9**                      **Relative Down Range Position Errors**





**Figure 10 Relative Cross Range Position Errors**

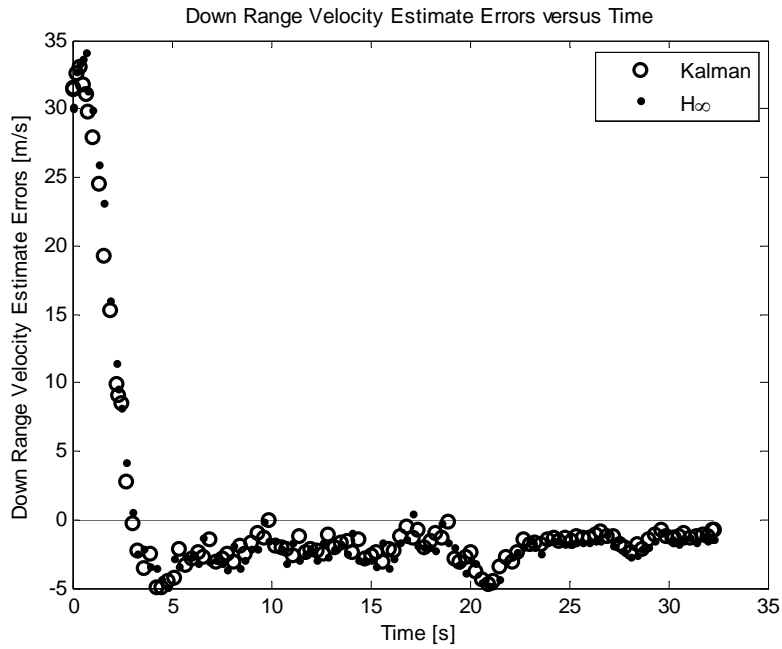
The Root-Mean-Square (RMS) errors in down range and cross range position are summarized in Table 3 below.

**Table 3 RMS Position Errors for Biased Noise**

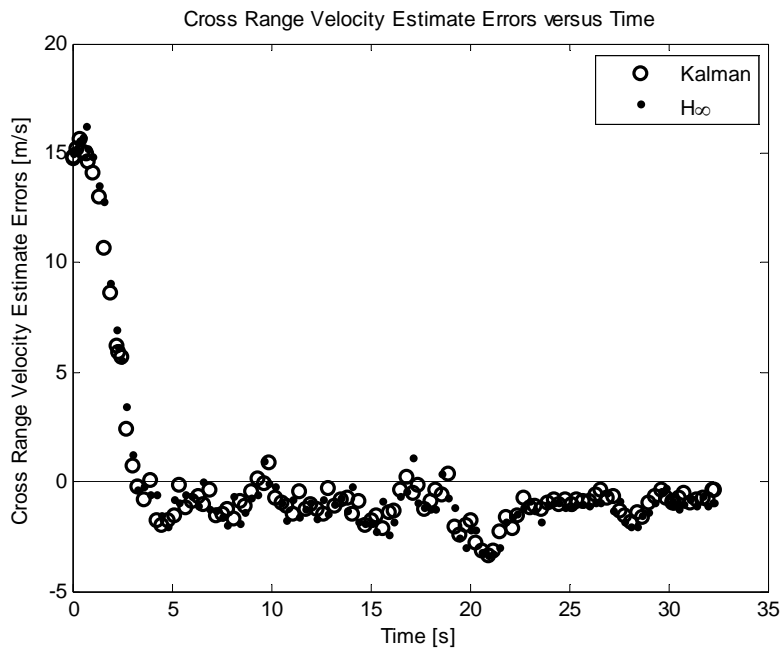
	Down Range Position Error (m)	Cross Range Position Error (m)
Kalman	1.3	3.1
H-infinity	0.7	0.5

Now that biases have been included in the noise statistics, the assumptions of the Kalman filter have been violated and the solution is no longer optimal. This is why now, *the H-infinity filter outperforms the Kalman filter.*

Figure 11 and figure 12 display the relative error between the actual and the estimated down range and cross range velocity for both the Kalman and the H-infinity filters.

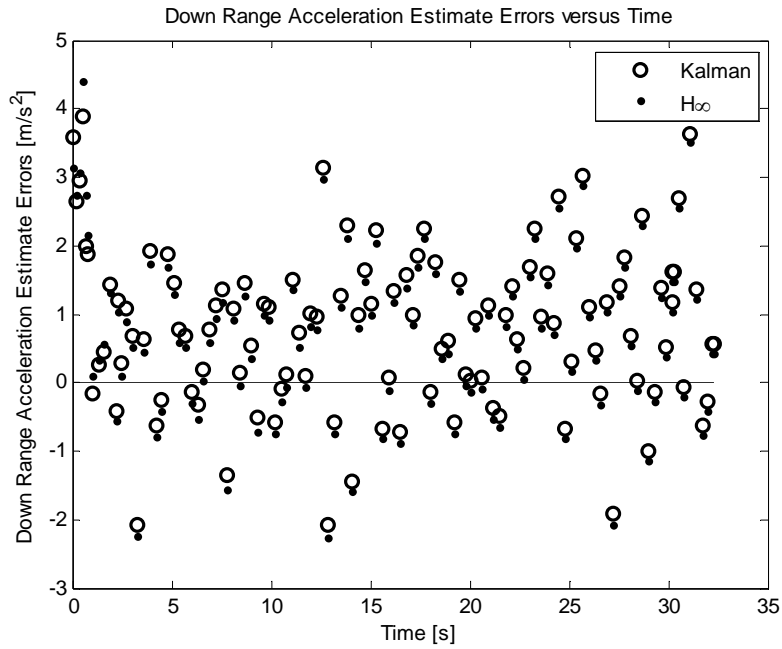


**Figure 11 Relative Down Range Velocity Errors**

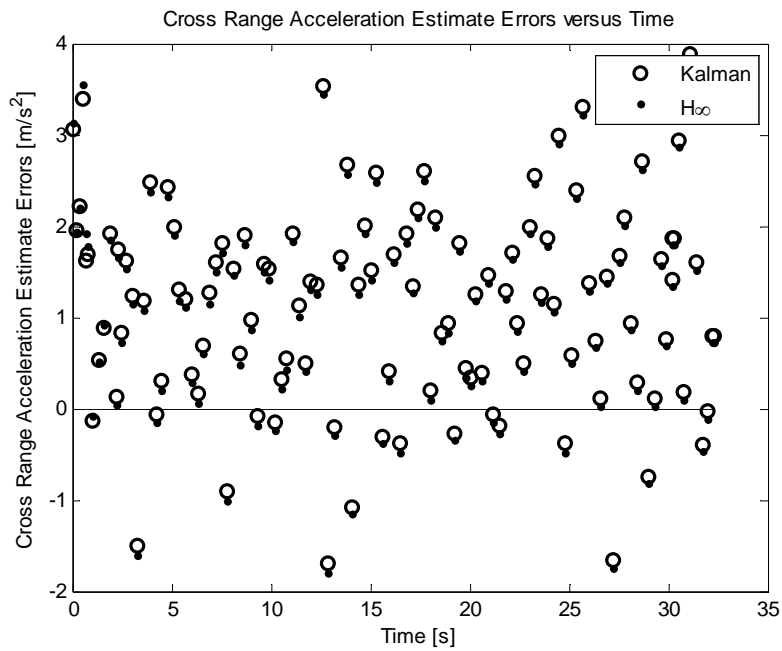


**Figure 12 Relative Cross Range Velocity Errors**

Figure 13 and figure 14 display the relative error between the actual and the estimated down range and cross range acceleration for both the Kalman and the H-infinity filters.



**Figure 13** Relative Down Range Acceleration Errors

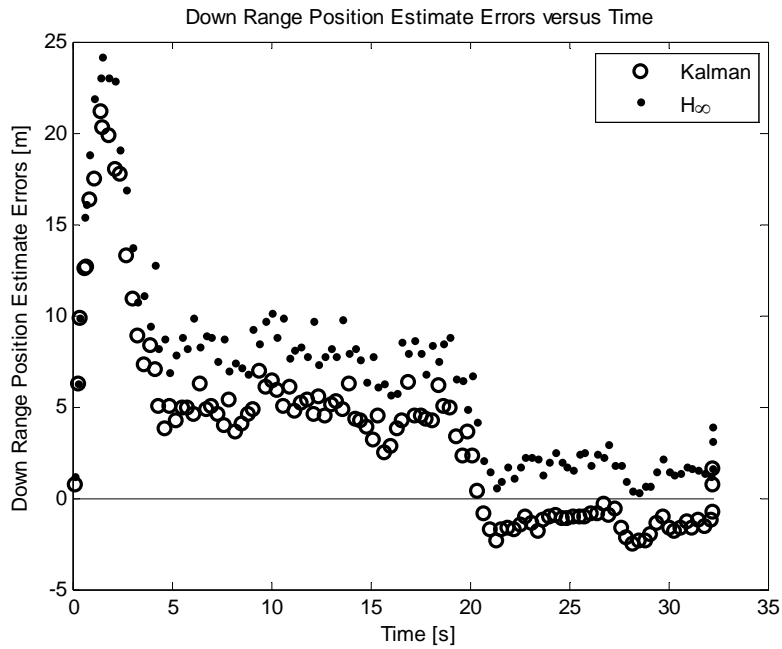


**Figure 14** Relative Cross Range Acceleration Errors

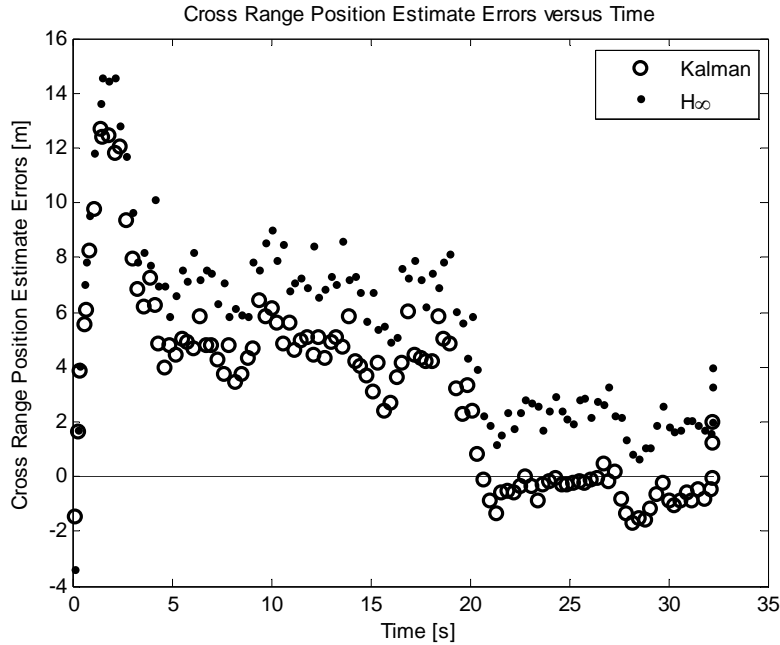
### 2.4.4 Simulation Case Three: Constrained Estimates

One final excursion is to examine the performance of the filters when the target velocity and acceleration state estimates are constrained to 90mph (40m/sec) and 0.27g (2.65m/sec<sup>2</sup>), respectively.

Figure 15 and figure 16 display the relative error between the actual and the estimated down range and cross range position for both the Kalman and the H-infinity filters.



**Figure 15** Relative Down Range Position Errors



**Figure 16 Relative Cross Range Position Errors**

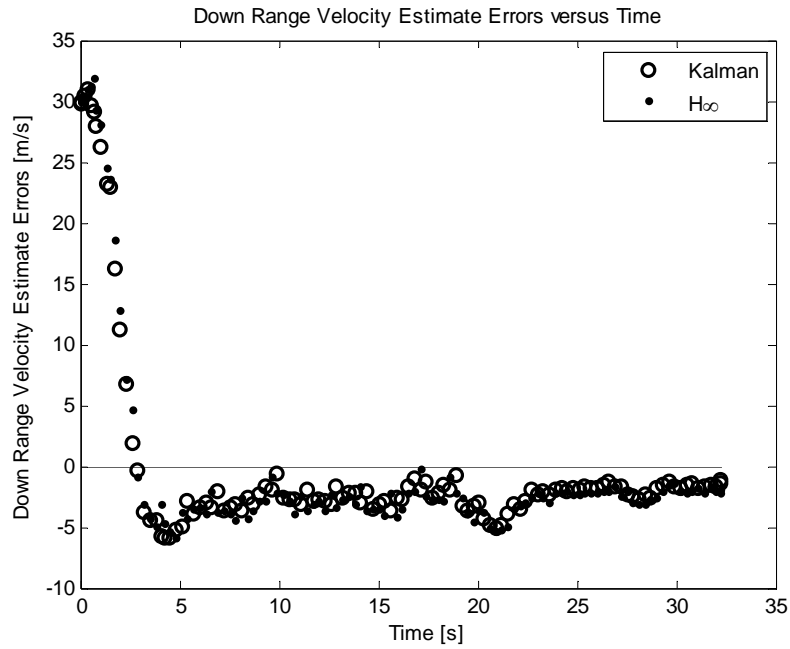
The Root-Mean-Square (RMS) errors in down range and cross range position are summarized in Table 4 below.

**Table 4 RMS Position Errors for Constrained Estimates**

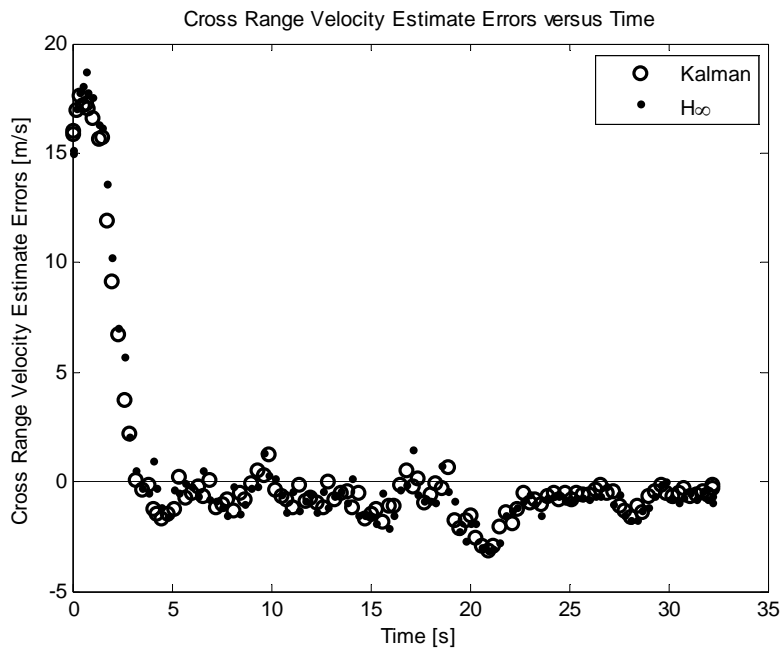
	Down Range Position Error (m)	Cross Range Position Error (m)
Kalman	1.3	1.3
H-infinity	0.7	0.4

Now that the state estimates have been constrained, there is a slight improvement in the cross range estimate by their H-infinity filter as compared with Table 3 of the previous section. Additionally, the H-infinity filter still outperforms the Kalman filter because the biases in the noise statistics are still in use.

Figure 17 and figure 18 display the relative error between the actual and the estimated down range and cross range velocity for both the Kalman and the H-infinity filters.

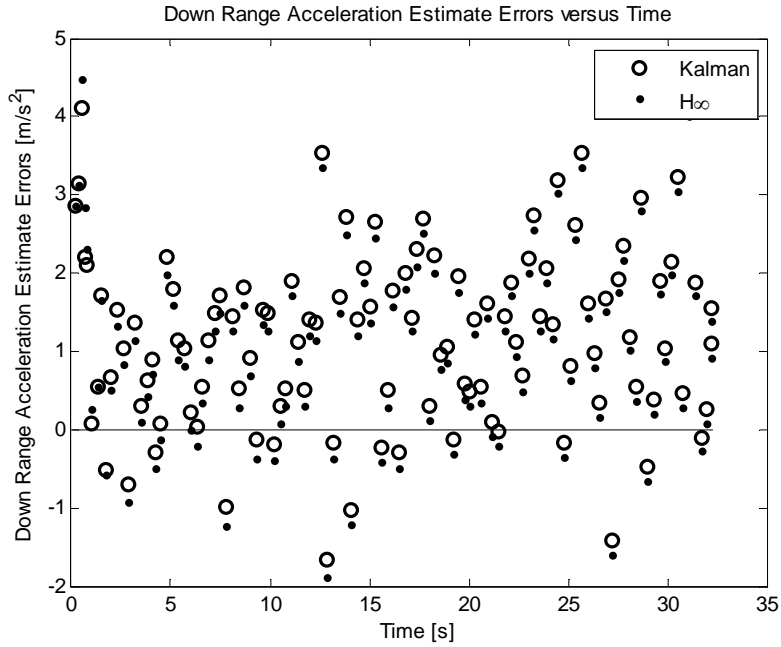


**Figure 17 Relative Down Range Velocity Errors**

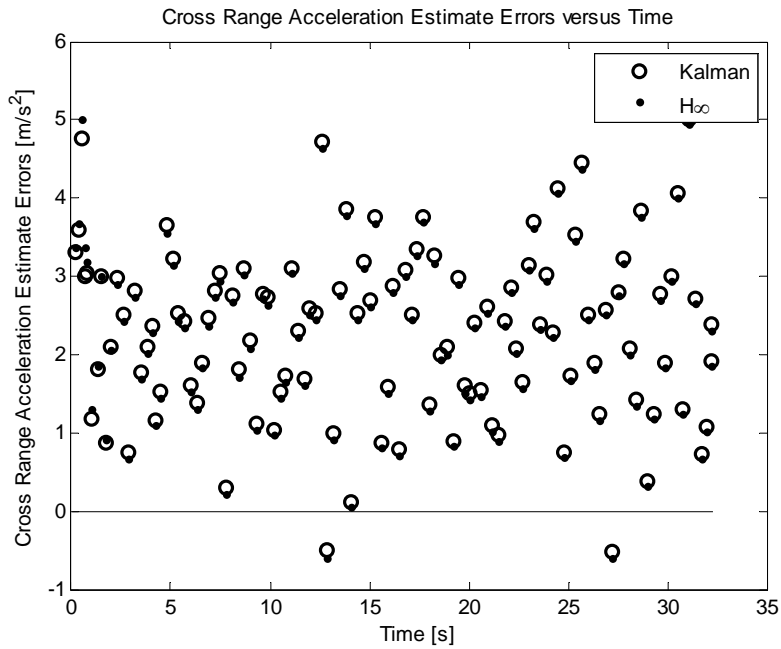


**Figure 18 Relative Cross Range Velocity Errors**

Figure 19 and figure 20 display the relative error between the actual and the estimated down range and cross range acceleration for both the Kalman and the H-infinity filters.



**Figure 19** Relative Down Range Acceleration Errors

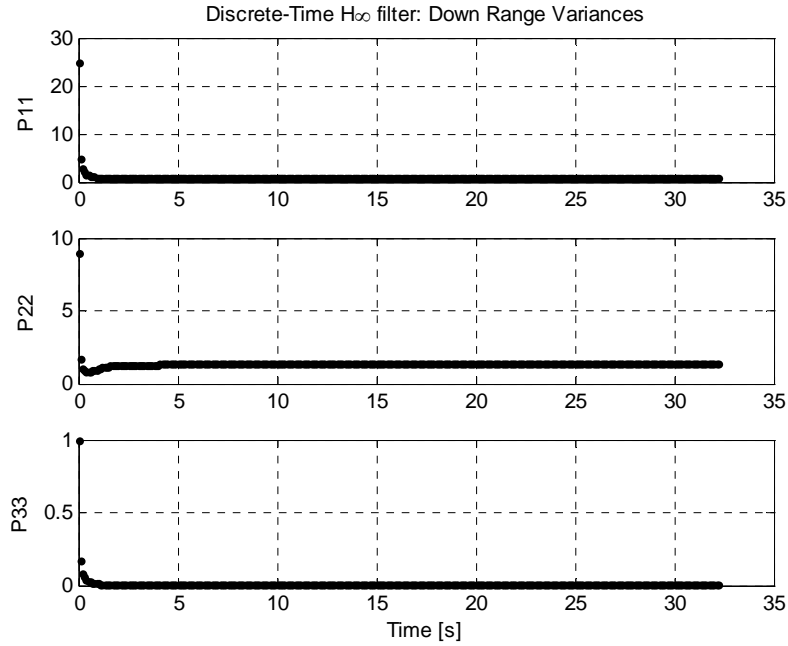


**Figure 20** Relative Cross Range Acceleration Errors

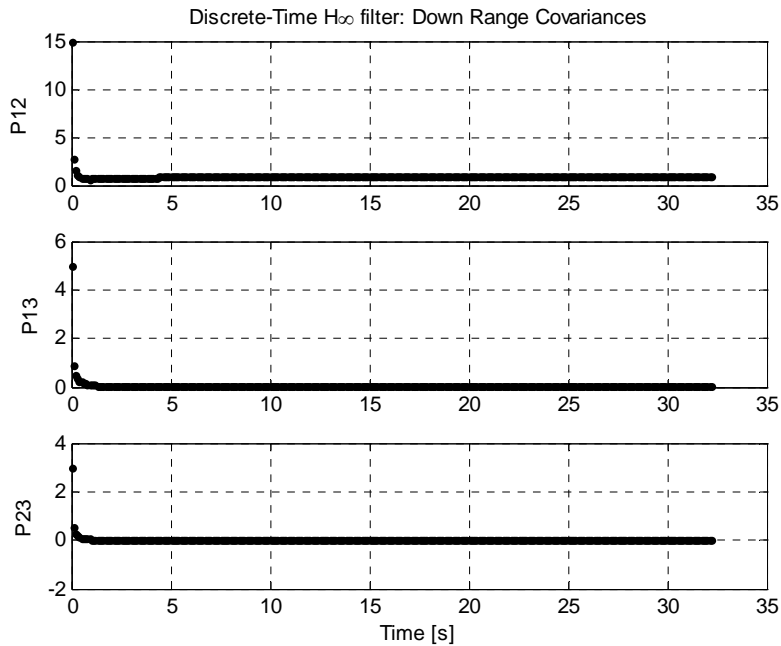
### 2.4.5 Variances, Covariances, and Gains

Continuing with the biased noise statistics with constrained state estimates from the previous section, figures 21-24 display the six unique elements of the (symmetric) H-

infinity estimation error covariance matrix for both down range and cross range estimates. Recall for a symmetric matrix  $P_{21} = P_{12}$ ,  $P_{31} = P_{13}$ , and  $P_{32} = P_{23}$ .

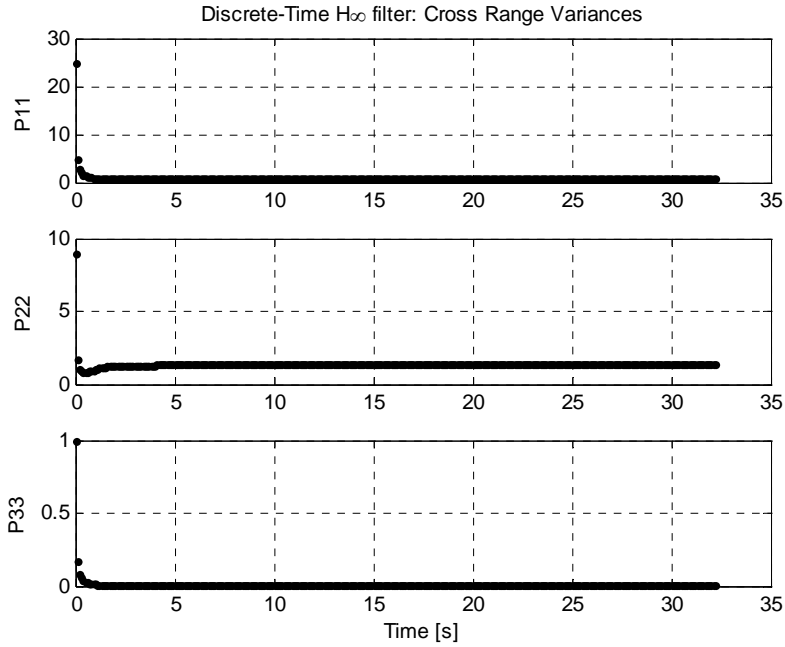


**Figure 21 Estimation Error Down Range Variances**

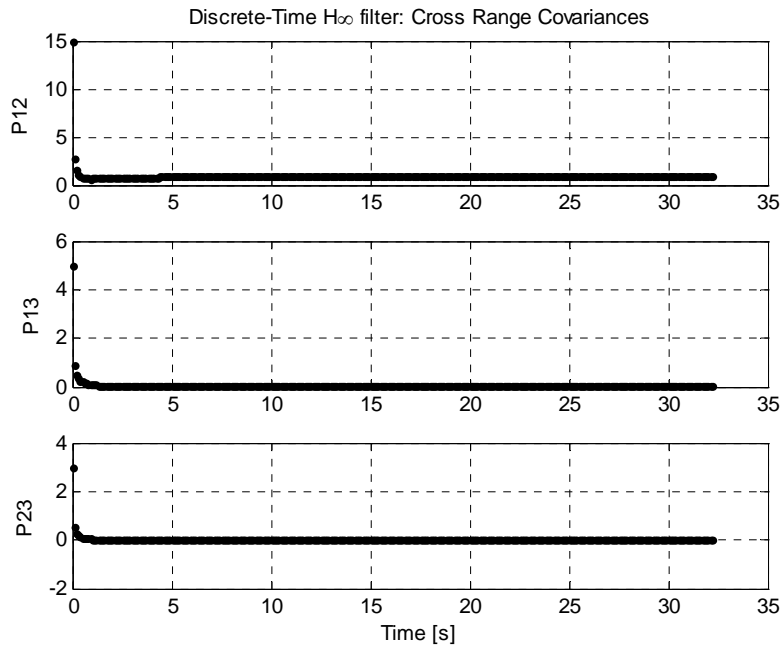


**Figure 22 Estimation Error Down Range Covariances**





**Figure 23 Estimation Error Cross Range Variances**



**Figure 24 Estimation Error Cross Range Covariances**

The numerical *steady-state* values of the estimation error covariance matrix are

$$\mathbf{P} = \begin{bmatrix} 0.979 & 0.841 & 0.011 \\ 0.841 & 1.332 & 0.018 \\ 0.011 & 0.018 & 0.004 \end{bmatrix} \quad (2.62)$$

for both down range and cross range.

The conclusion to be drawn from these figures is that each of the elements converges to a steady-state value within 2sec. This is significant if there is a processing time issue associated with the H-infinity filter. Rather than computing each element of the estimation error covariance matrix, the designer could simply hard-code the steady-state values into the embedded systems software. To determine the *stable*, steady-state values of  $\mathbf{P}$ , consider equation (2.58) and set  $\mathbf{P}_k = \mathbf{P}_{k-1}$ , which yields

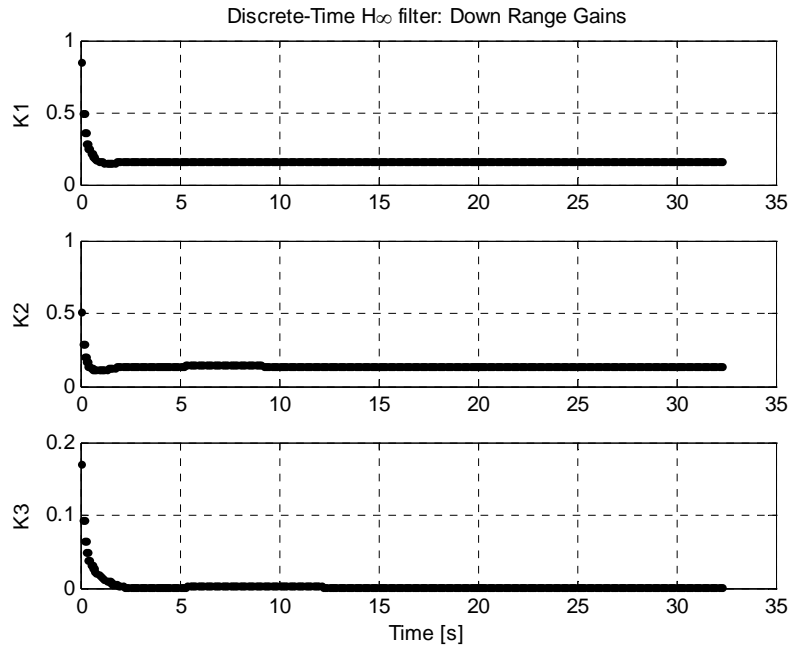
$$\mathbf{P} = \mathbf{F}\mathbf{P}[\mathbf{I} - \theta\mathbf{S}\mathbf{P} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}]^{-1}\mathbf{F}^T + \mathbf{Q} \quad (2.63)$$

where the subscript  $k$  has been dropped for legibility. This is an algebraic, discrete-time, Riccati equation which may be solved, off-line, by numerical iteration. The solution yields the steady-state values for  $\mathbf{P}$ .

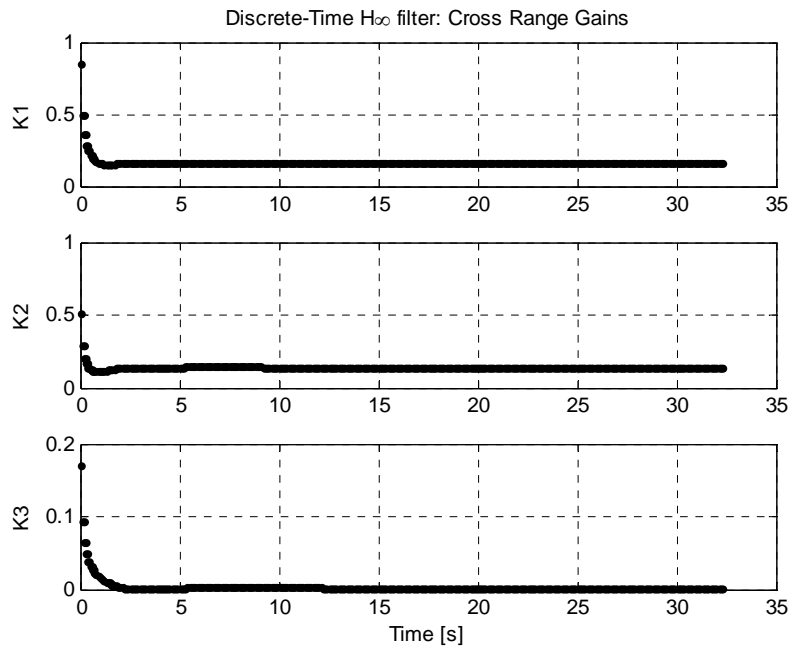
Figure 25 and figure 26 display the three H-infinity gains for both down range and cross range estimates. As explained in the previous case, each gain converges to a steady-state value within 2sec. The numerical steady-state values of the gains are

$$\mathbf{K} = [0.165 \quad 0.142 \quad 0.002] \quad (2.64)$$

for both down range and cross range.



**Figure 25** Down Range Gains



**Figure 26** Cross Range Gains

## 2.5 Cooperative Navigation Summary

In the case where the noise statistics are zero-mean, i.e. unbiased, the Kalman filter outperforms the H-infinity filter. This is expected because the Kalman filter is the optimal solution when the assumptions of zero-mean noise statistics are true. While figures 3-8 show the relative error of each filter in estimating the target's kinematics, table 2 quantifies the RMS performance of each filter. Recall the significance of examining the acceleration estimate is because the Augmented Proportional Navigation Guidance (APNG) law depends on the estimated target's acceleration as will be seen in the next chapter.

When biases are included in the noise statistics, the assumptions of the Kalman filter have been violated and the solution is no longer optimal. In addition, equation (2.12) described the propagation of the state during the time update and equations (2.3) and (2.16) described the propagation of the estimation error covariance matrix during the measurement and time updates, respectively. These equations were greatly simplified due to the assumption that there is no correlation between the estimation error, the process noise, and the measurement noise. In this case, it was assumed that the off-diagonal elements of  $\mathbf{P}_{k+1}$  are non-zero. This is another reason why the H-infinity filter outperforms the Kalman filter in this case. While figures 9-14 show the relative error of each filter in estimating the target's kinematics, table 3 quantifies the RMS performance of each filter. These are the arguments for the robustness of the H-infinity filter. The H-infinity filter operates without any knowledge of the noise statistics. Its goal is to find the minimum estimation error given the maximum noise variance. Recall from equation (2.40), the term  $\theta \mathbf{S}_k \mathbf{P}_k$  is subtracted from both the  $\mathbf{K}_k$  and the  $\mathbf{P}_{k+1}$  calculation making

the gain and the estimation error covariance matrix larger. This effectively places more emphasis on the measurements which is a way of making the H-infinity filter more robust to uncertainty in the process noise of the system model. In fact, the H-infinity filter is made to be robust by optimal design.

When the state estimates are constrained, the overall RMS error is reduced by approximately 20% as may be seen by comparing table 4 with table 3. Also, the H-infinity filter still outperforms the Kalman filter because the biases in the noise statistics are still applied in this case. Figures 15-20 show the relative error of each filter in estimating the target's kinematics.

Finally, the six unique estimation error covariance matrix elements (P11, P22, P33, P12, P13, and P23) and the three gains (K1, K2, and K3) were plotted in figures 21-26 for both down range and cross range estimates. Equation (2.62) showed the steady-state estimation error covariance matrix values and equation (2.64) showed the steady-state values of the gains. It is noted that the steady-state estimation error covariance matrix is symmetric. This indicates that there are no precision issues associated with the on-line calculations. Also, it is noted that if there are issues associated with real-time execution of the software, these steady-state values could be written into the software to take the calculations off-line.

In summary, when biases and correlations exist with the noise statistics, the H-infinity filter outperforms the Kalman filter, particularly in estimating the acceleration of the target. In addition, the overall estimation error was reduced when constraints were placed on the state estimates. Therefore, it is recommended to implement a constrained, H-infinity filter to track a maneuvering target. Furthermore, the H-infinity filter costs no

more to implement than the Kalman filter with regard to hardware since both filters are strictly software implementations.

This concludes Chapter Two: Cooperative Navigation. In the next chapter, the constrained H-infinity filter's estimate of the target's acceleration will be applied to the Augmented Proportional Navigation Guidance (APNG) law.

## CHAPTER THREE: GUIDANCE AND CONTROL

For this application, recall that one UAV tracks the maneuvering (or non-maneuvering) target and shares the information with the pursuer aboard another UAV. Even though the pursuer has not been released, the pursuer estimates the target's kinematics by implementing a three-state, discrete-time, H-infinity filter with inequality constraints placed on the estimates. Once the UAV containing the pursuer has confirmed the target and is in position to release the pursuer, the tracking UAV continues to communicate target location information directly with the pursuer for command guidance purposes. It is noted that the pursuer is released into what is known as the Launch Acceptability Region (LAR), where the pursuer will reach the target if it is released into this "basket" in space. Later in the pursuer's trajectory, the tracking UAV becomes the designating UAV. At this time, the target is illuminated with laser energy for the pursuer to switch from the command guidance mode to semi-active laser mode, thus reducing the error in the maneuvering target's location leading to improve performance by reducing the miss distance. *Command guidance* is used for midcourse guidance and *semi-active laser guidance* is used for terminal guidance. This is known as *dual-mode guidance*. Whether the guidance *mode* is command or semi-active laser depends on whether the UAV is in the tracking or designating *mode*, respectively.

Recall that there is still a guidance *law* issue related to light-weight, acceleration-limited pursuers. Because the literature review showed classical and modern guidance laws and adaptive/neural autopilot control laws require at least 10g of acceleration capability, a new guidance algorithm needs to be developed to intercept maneuvering surface targets. Therefore, an algorithm is developed that incorporates the Velocity

Pursuit Guidance (VPG) law at long ranges (because of its noise *insensitivity*) and an Augmented Proportional Navigation Guidance (APNG) law at short ranges (because of its *accuracy*). The augmented form of the proportional navigation guidance law includes the acceleration estimate of the target as computed from the previous chapter. The algorithm will be constructed such that the original form of the proportional navigation guidance (PNG) law will be implemented for a non-maneuvering target.

The adaptive logic for switching from the VPG law to the APNG law is based on the *field-of-view* and the *linear region* of the pursuer's seeker. The field-of-view is the angular region that is seen by the pursuer, measured in azimuth and elevation. The linear region is that portion of the seeker's optics where the target's motion is measured linearly. Outside this region, the target's motion does not appear well-behaved.

At long ranges, with a wide field-of-view and low accuracy requirements, the target motion appears noisy; therefore the VPG law is put to task. At short ranges, within the linear region of the seeker, the target's motion increases the acceleration requirement on the pursuer; therefore the APNG law is utilized. It is proposed that the combined performance of the pursuit and augmented proportional navigation algorithm will be within the 5g acceleration requirement. The other performance requirement from Roemerma (2006) is that the miss distance is within 3m.

In this section, the dynamic equations of the pursuer's airframe are developed followed by the design of the pursuer's autopilot also known as the flight control system. Then, each of the guidance laws (VPG and APNG) is designed. Finally, simulation results are presented for performance analysis.



### 3.1 Airframe Dynamic Equations

The purpose of the guidance algorithm is to direct the pursuer to the target. This is achieved by commanding changes to the acceleration of the pursuer. It is noted that the pursuer does not have any longitudinal thrust characteristics; therefore, only lateral accelerations can be commanded which are achieved by making changes to control surfaces by means of fin deflections. The goal of the flight control system (autopilot) is to ensure the pursuer's airframe responds to these control surface deflections in a stable manner. Hence, it is first necessary to understand how a change in fin deflection affects the dynamics of the pursuer's airframe.

#### 3.1.1 Lateral Accelerations and Angular Rates

The aerodynamic forces and moments acting on the pursuer's airframe are developed in Appendix 1. In particular, the lateral dynamic equations for pitch (yaw) are developed based on the normal (side) force and the pitching (yawing) moment. Then, a relationship between fin deflection and lateral acceleration and angular acceleration is made.

From Appendix 1, the lateral acceleration based on the normal force is

$$a_z = N_\alpha \alpha + N_{\delta q} \delta q \quad (3.1)$$

where  $N_\alpha = C_{N\alpha} QS/m$  and  $N_{\delta q} = C_{N\delta q} QS/m$  and the angular acceleration based on the pitching moment is

$$\dot{q} = M_\alpha \alpha + M_{\delta q} \delta q + M_q q \quad (3.2)$$

where  $M_\alpha = C_{m\alpha} QSd/I_{yy}$ ,  $M_{\delta q} = C_{m\delta q} QSd/I_{yy}$ , and  $M_q = C_{mq} QSd^2/2VI_{yy}$  and all quantities are defined as in Appendix 1. Differentiating equation (3.1)

$$\dot{a}_z = N_\alpha \dot{\alpha} + N_{\delta q} \delta \dot{q} \quad (3.3)$$

Consider figure 27 below which shows pitch angle  $\theta$ , flight path angle  $\gamma$ , and angle-of-attack  $\alpha$ .  $X_E$  and  $Z_E$  are Earth coordinates,  $X_B$  is the airframe's body longitudinal axis, and  $V$  is the velocity of the airframe.

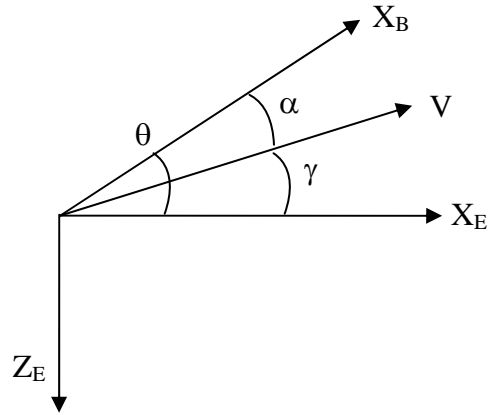


Figure 27

Kinematic Angles

From these relations,  $\theta = \alpha + \gamma$ . Differentiating,  $\dot{\theta} = \dot{\alpha} + \dot{\gamma}$ . It is customary to use  $q$  to represent pitch rate  $\dot{\theta}$ . Therefore,  $q = \dot{\alpha} + \dot{\gamma}$  or

$$\dot{\gamma} = q - \dot{\alpha} \quad (3.4)$$

Also from the definition of differentiation of a vector whose orientation varies,  $\dot{V} = \omega \times V$  where  $\omega$  is the angular velocity of the vector whose orientation is varying.

Therefore, from these relations,  $\dot{V} = \dot{\gamma} \hat{Y}_E \times V$  or

$$a_z = \dot{\gamma} V \quad (3.5)$$

which is perpendicular to the velocity vector,  $V$ . Combining equation (3.4) with equation (3.5) yields

$$\begin{aligned} a_z &= \dot{\gamma} V \\ &= (q - \dot{\alpha}) V \end{aligned} \quad (3.6)$$

Substituting  $\dot{\alpha}$  from equation (3.6) into equation (3.3) yields

$$\begin{aligned}\dot{a}_z &= N_\alpha \left( q - \frac{a_z}{V} \right) + N_{\delta q} \delta \dot{q} \\ &= N_\alpha q - \frac{N_\alpha}{V} a_z + N_{\delta q} \delta \dot{q}\end{aligned}\tag{3.7}$$

Eliminating  $\alpha$  from equation (3.1) and equation (3.2) yields

$$\dot{q} = \frac{M_\alpha}{N_\alpha} a_z + M_q q + \left( M_{\delta q} - M_\alpha \frac{N_{\delta q}}{N_\alpha} \right) \delta q\tag{3.8}$$

Ignoring the actuator dynamics for now,  $\delta \dot{q} = 0$  yields the following set of lateral

dynamic equations for pitch and assuming  $M_{\delta q} \gg M_\alpha \frac{N_{\delta q}}{N_\alpha}$

$$\begin{aligned}\dot{a}_z &= N_\alpha q - \frac{N_\alpha}{V} a_z \\ \dot{q} &= \frac{M_\alpha}{N_\alpha} a_z + M_q q + M_{\delta q} \delta q\end{aligned}\tag{3.9}$$

In matrix form

$$\begin{bmatrix} \dot{q} \\ \dot{a}_z \end{bmatrix} = \begin{bmatrix} M_q & \frac{M_\alpha}{N_\alpha} \\ N_\alpha & -\frac{N_\alpha}{V} \end{bmatrix} \begin{bmatrix} q \\ a_z \end{bmatrix} + \begin{bmatrix} M_{\delta q} \\ 0 \end{bmatrix} \delta q\tag{3.10}$$

In a similar fashion the set of lateral dynamic equations for yaw are

$$\begin{bmatrix} \dot{r} \\ \dot{a}_y \end{bmatrix} = \begin{bmatrix} LN_r & \frac{LN_\beta}{Y_\beta} \\ -Y_\beta & \frac{Y_\beta}{V} \end{bmatrix} \begin{bmatrix} r \\ a_y \end{bmatrix} + \begin{bmatrix} LN_{\delta r} \\ 0 \end{bmatrix} \delta r\tag{3.11}$$

where  $a_y$  is the lateral acceleration based on the side force,  $r$  is the angular acceleration based on yawing moment,  $Y_\beta = C_{Y\beta} QS/m$ ,  $LN_\beta = C_{n\beta} QSD/I_z$ ,  $LN_r = C_{nr} QSD^2/2I_zV$ , and  $LN_{\delta r} = C_{n\delta r} QSD/I_z$  and all other quantities are defined in Appendix 1.

With a relationship between fin deflection and lateral accelerations and angular accelerations now defined, the flight control system (autopilot) may be designed.

### 3.2 Flight Control System (Autopilot)

Recall that the goal of the flight control system (autopilot) is to ensure the pursuer's airframe responds to these control surface deflections in a stable manner. Because lateral accelerations are commanded by the guidance system, an autopilot that utilizes lateral and angular acceleration feedback will be used for stabilizing the pursuer's airframe when the control surfaces (fins) are deflected. This approach will also provide the timely response required during the terminal phase of the homing guidance system.

#### 3.2.1 Pitch Dynamics

The lateral dynamics in pitch were developed in the previous section.

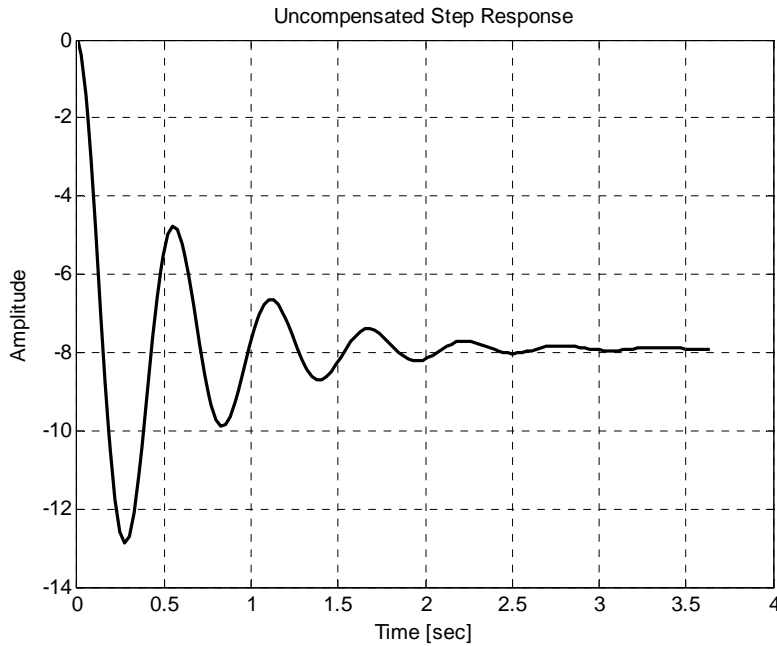
$$\begin{bmatrix} \dot{q} \\ \dot{a}_z \end{bmatrix} = \begin{bmatrix} M_q & \frac{M_\alpha}{N_\alpha} \\ N_\alpha & -\frac{N_\alpha}{V} \end{bmatrix} \begin{bmatrix} q \\ a_z \end{bmatrix} + \begin{bmatrix} M_{\delta q} \\ 0 \end{bmatrix} \delta q \quad (3.12)$$

where  $q$  is pitch rate,  $a_z$  is normal acceleration,  $\delta q$  is fin deflection in the pitch plane,

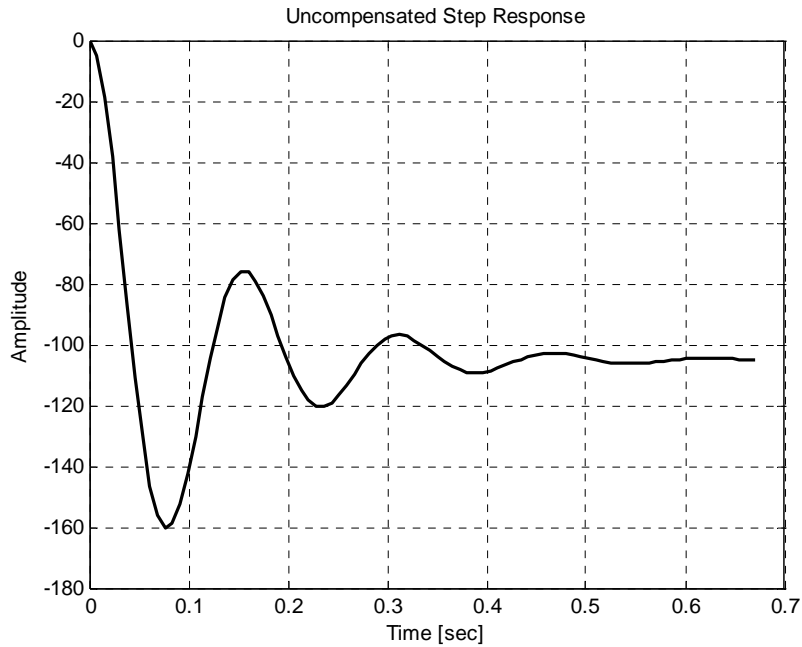
and  $V$  is velocity magnitude.  $N_\alpha = \frac{QS}{m} C_{N\alpha}$ ,  $M_\alpha = \frac{QSD}{I_y} C_{m\alpha}$ ,  $M_q = \frac{QSD^2}{2I_yV} C_{mq}$ , and

$M_{\delta q} = \frac{QSD}{I_y} C_{m\delta q}$  where  $Q$  is dynamic pressure,  $S$  is reference area,  $m$  is mass,  $d$  is

reference diameter,  $I_y$  is moment-of-inertia about the y-axis, and  $C_{N\alpha}$ ,  $C_{m\alpha}$ ,  $C_{mq}$ , and  $C_{m\delta q}$  are the aerodynamic coefficient parameters: normal force due to angle-of-attack, pitching moment due to angle-of-attack, pitching moment due to pitch rate, and pitching moment due to fin deflection, respectively. Figure 28 and figure 29 below show the open-loop step-response of the airframe's pitch dynamics at Mach numbers of 0.33 and 0.80, respectively, where Mach number is defined as the ratio of the air-flow velocity to the speed of sound. Both figures depict the need for a flight control system.



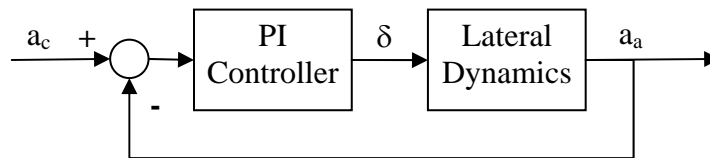
**Figure 28** Open-Loop Step-Response at Mach 0.33



**Figure 29** Open-Loop Step-Response at Mach 0.80

### 3.2.2 Pitch Control

The flight control system requirements are fast rise-time and zero steady-state errors. Therefore, proportional (P) control is used to achieve fast rise-time and integral (I) control is used to achieve zero steady-state error. A PI-controller is used because of its implementation ease. The block diagram for the flight control system is shown in Figure 30 below, where  $a_c$  represents commanded acceleration,  $\delta$  represents fin deflection, and  $a_a$  represents achieved acceleration.



**Figure 30** Pitch Control Block Diagram

The closed-loop system now consists of three states: two from the lateral pitch dynamics (the pitch rate  $q$  and the normal acceleration  $a_z$ ) plus the integrator from integral control. Hence three poles need to be placed.

The dominant closed-loop poles are determined from a second-order system

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (3.13)$$

where  $\zeta$  is the damping ratio and  $\omega_n$  is the natural frequency. For a second-order system, the percent overshoot is defined by

$$100 \exp\left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}\right) \quad (3.14)$$

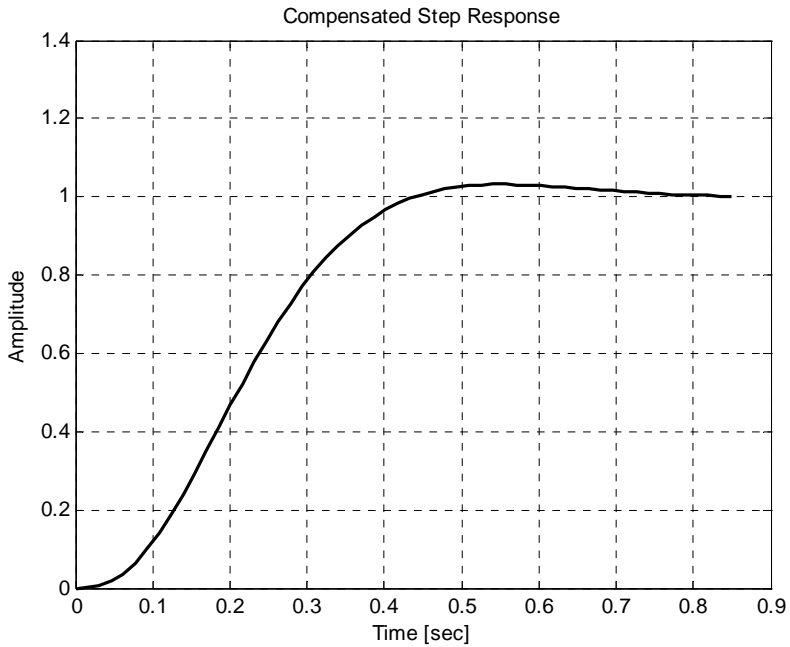
For 5% overshoot, the damping ratio is  $\zeta=0.7$ . The natural frequency of a second-order system is determined from the rise time  $t_r$  - which is the time needed for the response to attain 60% of its reference signal.

$$\omega_n = \frac{1+1.1\zeta+1.4\zeta^2}{t_r} \quad (3.15)$$

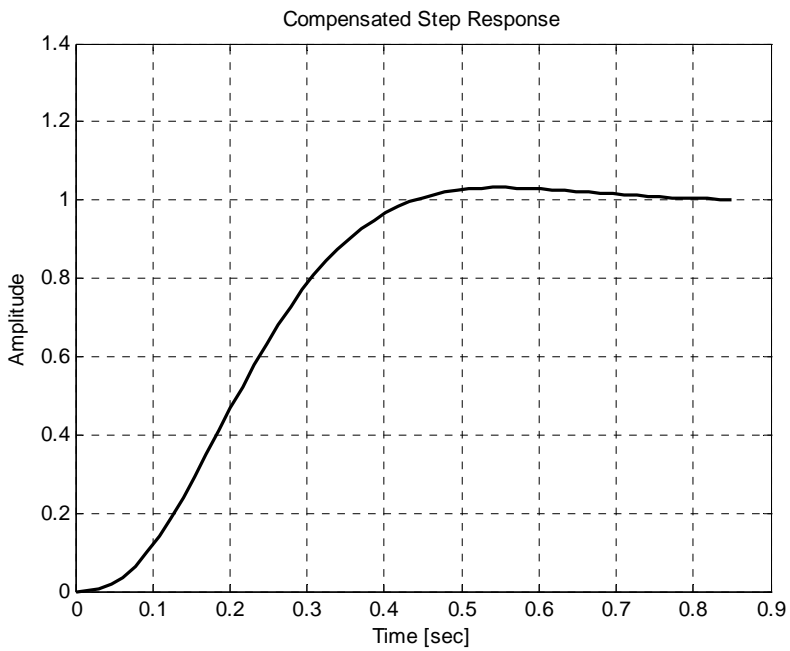
For  $\zeta=0.7$  and  $t_r=0.25\text{sec}$ ,  $\omega_n=9.8\text{rad/sec}$ . With  $\zeta=0.7$  and  $\omega_n=9.8\text{rad/sec}$ ,

$$\begin{aligned} s^2 + 2\zeta\omega_n s + \omega_n^2 &= (s + \zeta\omega_n + \omega_n\sqrt{1-\zeta^2}j)(s + \zeta\omega_n - \omega_n\sqrt{1-\zeta^2}j) \\ &= (s + 6.86 + 7.00j)(s + 6.86 - 7.00j) \end{aligned} \quad (3.16)$$

The third pole must be placed to the left, in this case at -15, to obtain the desired closed-loop step-response characteristics without interfering with the dominant closed-loop poles given in equation (3.16). Figure 31 and figure 32 below show the closed-loop step-response of the airframe's pitch dynamics to a step input at Mach numbers of 0.33 and 0.80, respectively.



**Figure 31 Closed-Loop Step-Response at Mach 0.33**



**Figure 32 Closed-Loop Step-Response at Mach 0.80**

Notice that both figures depict the same step-response (5% overshoot and a rise time of 0.25sec) regardless of flight conditions. In the next section, it will be shown how the gain scheduling is achieved for pitch control due to various flight conditions.



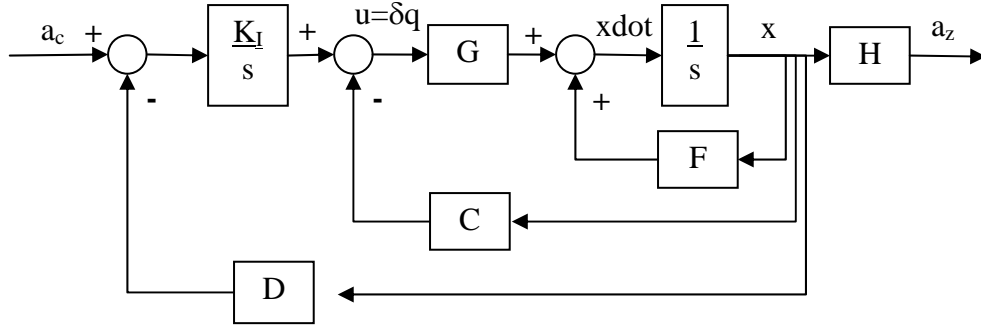
### 3.2.3 Gain Scheduling

Now that the appropriate poles have been placed, in order to achieve the desired closed-loop step-response characteristics, it remains to determine how these poles are placed for different flight conditions.

If the lateral pitch dynamics of equation (3.12) are represented, generally, by

$$\begin{aligned}\dot{x} &= Fx + Gu \\ y &= Hx\end{aligned}\tag{3.17}$$

then figure 33 becomes figure 35 below



**Figure 33** Pitch Control Block Diagram

where  $x = [q \ a_z]$ ,  $F = \begin{bmatrix} M_q & \frac{M_\alpha}{N_\alpha} \\ N_\alpha & -\frac{N_\alpha}{V} \end{bmatrix}$ ,  $G = \begin{bmatrix} M_{\delta q} \\ 0 \end{bmatrix}$ ,  $u = \delta q$ ,  $H = [0 \ 1]$ ,  $C = [k_2 \ k_1]$

( $k_2$  and  $k_1$  are gains to be determined),  $D = [0 \ 1]$ , and  $K_I$  represents the integral control gain to be determined.

The closed-loop system is determined from figure 33, where the control input  $u$  is written as

$$u = -Cx + K_I \int (a_c - a_z) dt\tag{3.18}$$

By introducing  $\eta = \int (a_c - a_z) dt$ , and noting  $\dot{\eta} = a_c - a_z$

$$u = -\mathbf{C}\mathbf{x} + K_I \eta \quad (3.19)$$

So that

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{F}\mathbf{x} + \mathbf{G}u \\ &= \mathbf{F}\mathbf{x} + \mathbf{G}(-\mathbf{C}\mathbf{x} + K_I \eta) \\ &= \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{C}\mathbf{x} + \mathbf{G}K_I \eta \end{aligned} \quad (3.20)$$

Combining both states into one closed-loop system

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{C}\mathbf{x} + \mathbf{G}K_I \eta \\ \dot{\eta} &= a_c - a_z \end{aligned} \quad (3.21)$$

In matrix form

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{F} + \mathbf{G}\mathbf{C} & \mathbf{G}K_I \\ [0 & -1] & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \eta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_c \quad (3.22)$$

The gains  $k_2, k_1$ , and  $K_I$  are determined by placing the poles equal to the eigenvalues of this system, where the eigenvalues are determined from

$$|s\mathbf{I} - \mathbf{A}| = (s - p_1)(s - p_2)(s - p_3) \quad (3.23)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{F} + \mathbf{G}\mathbf{C} & \mathbf{G}K_I \\ [0 & -1] & 0 \end{bmatrix} \quad (3.24)$$

Expanding  $\mathbf{F} + \mathbf{G}\mathbf{C}$  and  $\mathbf{G}K_I$  leads to

$$\begin{vmatrix} s - M_q - k_2 M_{\delta q} & \frac{M_\alpha}{N_\alpha} + k_1 M_{\delta q} & K_I M_{\delta q} \\ N_\alpha & s + \frac{N_\alpha}{V} & 0 \\ 0 & -1 & s \end{vmatrix} = (s - p_1)(s - p_2)(s - p_3) \quad (3.25)$$

Recall from equation (3.16) that the dominant closed-loop poles may be represented by

$$\begin{aligned} p_1 &= -\zeta\omega_n - \omega_n\sqrt{1-\zeta^2}j \\ p_2 &= -\zeta\omega_n + \omega_n\sqrt{1-\zeta^2}j \end{aligned} \quad (3.26)$$

The pitch gains are determined from the following equations:

$$K_I = \frac{\omega_n^2 p_3}{N_\alpha M_{\delta q}} \quad (3.27)$$

$$k_2 = \frac{1}{M_{\delta q}} \left( 2\zeta\omega_n + p_3 + M_q - \frac{N_\alpha}{V} \right) \quad (3.28)$$

$$k_1 = \frac{1}{N_\alpha M_{\delta q}} \left( \omega_n^2 + 2\zeta\omega_n p_3 + M_\alpha + \frac{M_q N_\alpha}{V} - k_2 \frac{M_{\delta q} N_\alpha}{V} \right) \quad (3.29)$$

where  $p_3$  is the third pole location (distant from the dominant closed-loop poles) and the other parameters have previously been defined.

Therefore, given any flight condition, the parameters  $N_\alpha$ ,  $M_{\delta q}$ ,  $M_q$ ,  $V$ , and  $M_\alpha$  are uniquely determined from the corresponding dynamic pressure  $Q$  and the gains  $k_2$ ,  $k_1$ , and  $K_I$  are scheduled accordingly. It is noted that since there is symmetry between the pitch plane and the yaw plane, the pitch controller may be used for yaw control as well.

### 3.3 Guidance Laws

A guidance algorithm is developed that incorporates Velocity Pursuit Guidance (VPG) at long ranges (because of its noise *insensitivity*) and Augmented Proportional Navigation Guidance (APNG) at short ranges (because of its *accuracy*). The adaptive logic for switching from the VPG law to the APNG law is based on the field-of-view and the linear region of the pursuer's seeker. At long ranges, with a wide field-of-view and

low accuracy requirements, the target motion appears noisy; therefore the VPG law is put to task. At short ranges, within the linear region of the seeker, the target's motion increases the acceleration requirement on the pursuer; therefore the APNG law is utilized.

The design of the guidance algorithm begins with APNG where the control law is determined by Lyapunov's stability analysis (including gain determination by optimal methods) followed by VPG gain determination by matching acceleration commands for smooth transitioning from one guidance law (VPG) to the other (APNG). This section concludes with a brief discussion on the logic associated with switching between the guidance laws.

### 3.3.1 Augmented Proportional Navigation Guidance Law

With proportional navigation guidance (PNG), the flight path angle rate  $\dot{\gamma}$  is made proportional to the line-of-sight rate  $\dot{\sigma}$  between the pursuer and the target.

$$\dot{\gamma} = \Lambda \dot{\sigma} \quad (3.30)$$

where  $\Lambda$  is the proportional constant. Recall from equation (3.5)

$$a_c = \dot{\gamma} V \quad (3.31)$$

Combining equation (3.30) and equation (3.31) yields

$$a_c = \Lambda V \dot{\sigma} \quad (3.32)$$

This is the acceleration command for PNG. Because the PNG law is based on the flight path angle rate and the line-of-sight rate, it performs well in the case of constant velocities but is impractical against maneuvering targets.

With augmented proportional navigation guidance (APNG), the lateral acceleration command  $a_c$  is given by the following equation

$$a_c = \Lambda V \dot{\sigma} + \frac{1}{2} \Lambda \hat{a}_T \quad (3.33)$$

where  $\Lambda$  is the proportional constant,  $V$  is velocity,  $\dot{\sigma}$  is the line-of-sight rate, and  $\hat{a}_T$  is the acceleration estimate of the target. This guidance law is more suited for maneuvering targets because it includes the target's acceleration estimate. Notice that if  $\hat{a}_T = 0$ , equation (3.32) is obtained. Please see Appendix 2 for the complete development of this guidance law based on Lyapunov's "direct method" of stability analysis as referenced in Narendra and Annaswamy (1989).

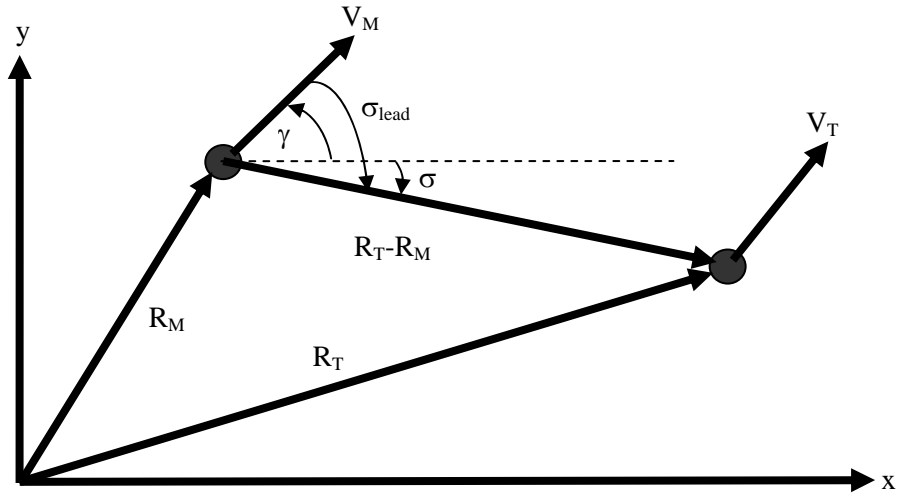
To complete this subsection, please see Appendix 3 for the determination of the proportional constant's value based on optimization methods of Bryson and Ho (1975). From the appendix, it is noted that  $\Lambda = 3$  so that the APNG guidance law is

$$a_c = 3V \dot{\sigma} + \frac{3}{2} \hat{a}_T \quad (3.34)$$

Furthermore, note that this guidance law is fully determined based on the parameters associated with the pursuer/target engagement.

### 3.3.2 Velocity Pursuit Guidance Law

With velocity pursuit guidance, the angle between the velocity vector of the pursuer and the line-of-sight to the target, called the "lead angle" is driven to zero.

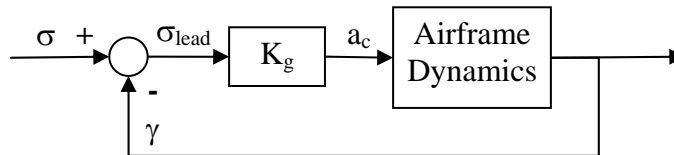


**Figure 34** Velocity Pursuit Guidance Geometry

This type of guidance algorithm is noise insensitive which makes it good for initial guidance because it avoids any unnecessary corrections due to target maneuvering and thus minimizing wasted energy. The lateral acceleration command  $a_c$  for velocity pursuit guidance is given by the following equation:

$$a_c = K_g \sigma_{lead} = K_g (\sigma - \gamma) \quad (3.35)$$

where  $K_g$  is a guidance gain and  $\sigma_{lead}$  is the lead angle (the difference between  $\sigma$  the line-of-sight angle and  $\gamma$  is the flight path angle). A block diagram of the VPG law is shown in figure 35 below.



**Figure 35** Velocity Pursuit Guidance Block Diagram

Because the VPG law is a proportional control system, the gain  $K_g$  may be determined from the steady-state error requirements. However, since it is desired to

minimize pursuer response to target maneuvering during the *initial* engagement, large steady-state errors are allowed because the guidance algorithm will eventually transition to APNG where accuracy is required. The steady-state error for proportional control is determined from figure 35:

$$\sigma_{lead}(t) = \sigma(t) - \gamma(t) \quad (3.36)$$

$$\gamma(t) = K_g G(t) \sigma_{lead}(t) \quad (3.37)$$

where  $G(t)$  represents the airframe dynamics. Substituting equation (3.37) into equation (3.36) yields

$$\begin{aligned} \sigma_{lead}(t) &= \sigma(t) - K_g G(t) \sigma_{lead}(t) \\ \sigma_{lead}(t) + K_g G(t) \sigma_{lead}(t) &= \sigma(t) \\ (1 + K_g G(t)) \sigma_{lead}(t) &= \sigma(t) \\ \sigma_{lead}(t) &= \frac{1}{1 + K_g G(t)} \sigma(t) \end{aligned} \quad (3.38)$$

Taking the Laplace transform, equation (3.38) becomes

$$\Sigma_{lead}(s) = \frac{1}{1 + K_g G(s)} \Sigma(s) \quad (3.39)$$

The steady-state error is determined from the Final Value Theorem which states

$$\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s) \quad (3.40)$$

Therefore, the steady-state error, due to a step input  $\Sigma(s) = 1/s$ , is determined as

$$\begin{aligned} \Sigma_{lead,ss} &= \lim_{s \rightarrow 0} s \Sigma_{lead}(s) \\ &= \lim_{s \rightarrow 0} s \frac{1}{1 + K_g G(s)} \frac{1}{s} \\ &= \frac{1}{1 + K_g G(0)} \end{aligned} \quad (3.41)$$

Assuming the DC gain of the airframe dynamics  $G(0)$  is unity, the steady-state error due to a unit step input is given by

$$\Sigma_{lead,ss} = \frac{1}{1 + K_g} \quad (3.42)$$

Rewriting equation (3.42) to determine  $K_g$

$$K_g = \frac{1}{\Sigma_{lead,ss}} - 1 \quad (3.43)$$

in units of  $(m/s^2)/rad$ . For zero steady-state error  $\Sigma_{lead,ss} = 0$ ,  $K_g \rightarrow \infty$ . However, this implies that the pursuer responds to every maneuver of the target. In application, it is desired that the guidance system waits until the target is within the linear region of the seeker before switching to the APNG law where the target is followed more accurately. Therefore, it is assumed that 10% steady-state error is sufficient to reduce the initial pursuer maneuvering to what appears as noisy target motion during the initial engagement. In this case,  $K_g = 9$ . Reducing the initial maneuvering decreases the aerodynamic drag on the pursuer which increases its available kinetic energy at impact.

### 3.3.3 Guidance Logic

Initially, the control surfaces (fins) are not allowed to move for the first 2sec of flight. This is to insure safe separation from the UAV. Once safe separation occurs, the guidance mode is based on pursuer/target engagement parameters. If the target is being tracked, but not designated, command guidance is implemented, i.e. the tracking UAV continues to communicate the target location to the pursuer whether it has been released from the UAV containing the pursuer or not. Later in the engagement, the tracking UAV designates the target and the guidance mode switches from command guidance to semi-



active laser (SAL) guidance. This section addresses the *logic* of the guidance algorithm, i.e. how the guidance law switches from VPG to APNG.

After release and the  $\Delta t$  of fin lock, if the initial azimuth and elevation of the line-of-sight  $\sigma$  are outside the Field-Of-View (FOV) of the seeker, the pursuer falls freely, i.e. ballistically. Once the azimuth and elevation of the target are within the FOV, the initial guidance is based on the VPG law. Then, once the azimuth and elevation of the target are within the *Linear Region* of the seeker, the guidance switches to the APNG law. Figure 36 below shows a diagram depicting transitions from the Release/Fin Lock state, through Search and VPG, to the final state APNG. The FOV is set to  $180^\circ$  to simulate *command mode* guidance by the tracking UAV. Therefore, the pursuer always “sees” the target. It is simply a matter of whether the target is within the *linear region* of the seeker whether VPG or APNG is implemented.

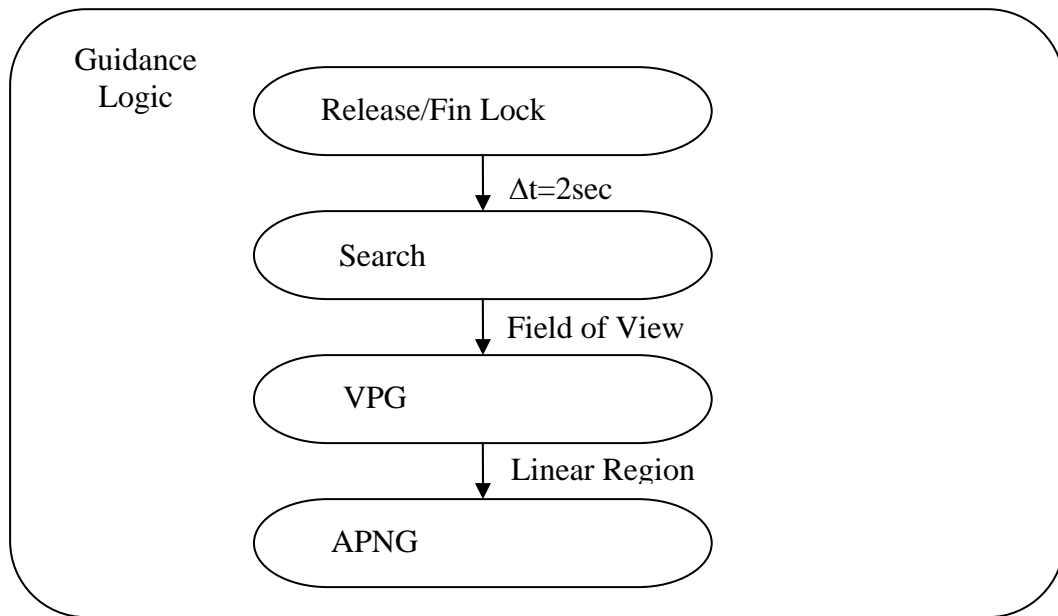


Figure 36 State Transition Diagram

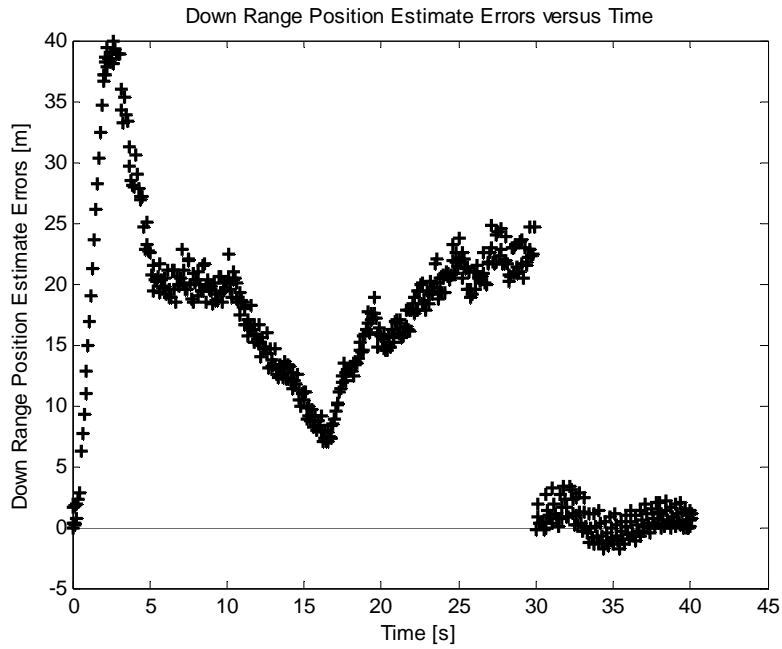
## 3.4 Simulation Results

### 3.4.1 Simulation of a Maneuvering Target

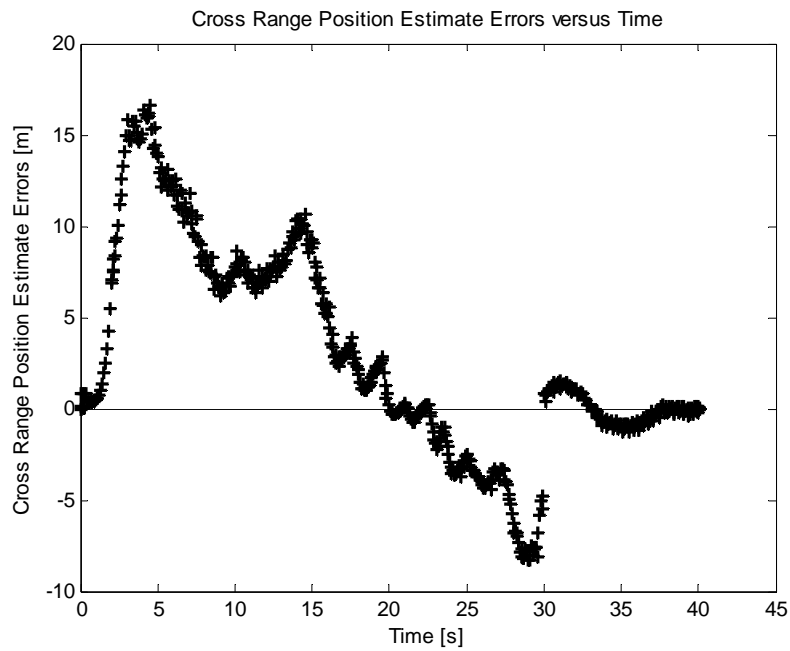
The UAV and pursuer parameters modeled as described in section 2.4.1. The target is modeled as a surface vehicle with an initial position and velocity. Its acceleration and jerk are modeled as process noise with magnitudes  $1\text{m/sec}^2$  and  $0.01\text{m/sec}^3$ , respectively. However, now the down range acceleration is allowed to change every 8 seconds and the cross range acceleration is allowed to change every 2 seconds. This is to represent forward acceleration and braking and lateral changes in velocity and position. Recall that the velocity and acceleration magnitudes are limited to 90mph (40m/sec) and  $0.27g$  ( $2.65\text{m/sec}^2$ ), respectively. Also, a steering model with a rate of  $45^\circ/\text{sec}$  is included to reflect realistic (smooth) changes in the target's motion.

The constrained, discrete-time H-infinity filter is implemented in a full six degree-of-freedom (6DOF) simulation to estimate the target's acceleration and subsequently used for guidance purposes.

Figure 37 and figure 38 display the relative error of the H-infinity filter in estimating both down range position and cross range position of the target. The H-infinity filter performs well in estimating the position of the target, particularly in the last 10sec of the engagement. At 30sec into the engagement, the guidance mode switches from command mode to SAL mode where the target's position measurement is performed by the pursuer and subsequently, the system delay is reduced from 500msec to 100msec.

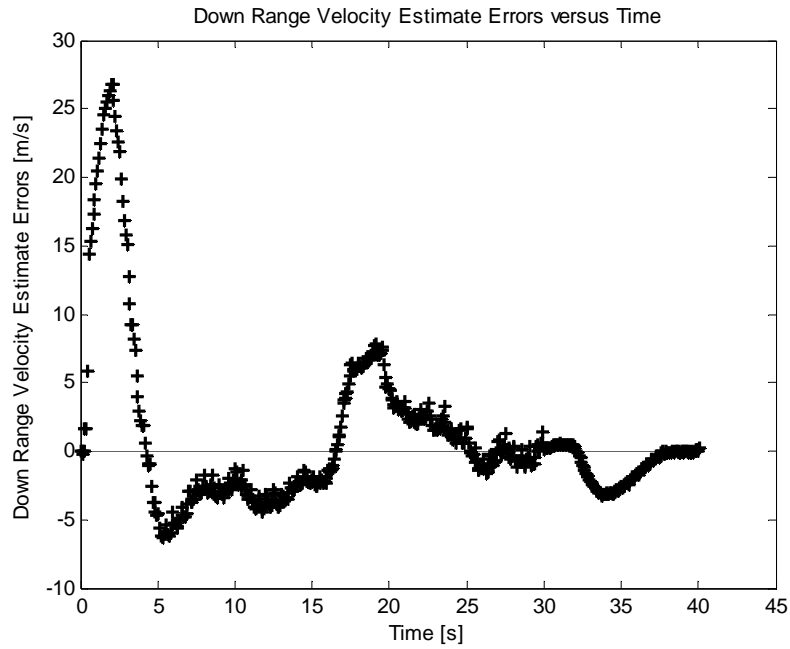


**Figure 37** Relative Down Range Position Estimate Error

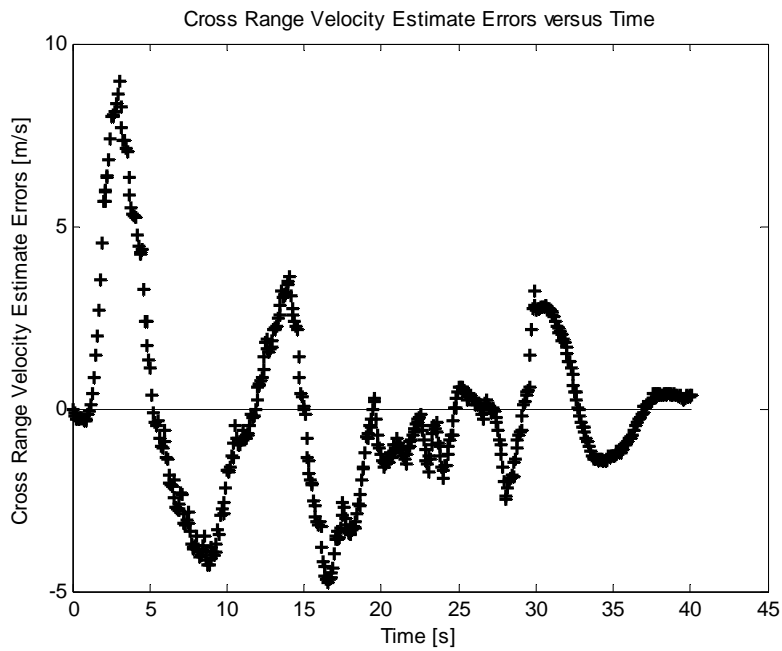


**Figure 38** Relative Cross Range Position Estimate Error

Figure 39 and figure 40 display the relative error of the H-infinity filter in estimating both down range velocity and cross range velocity of the target.

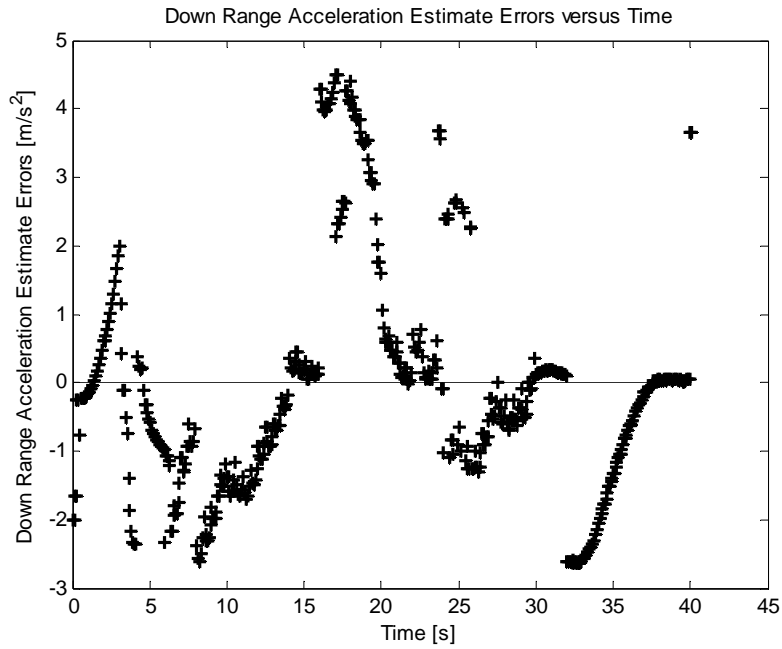


**Figure 39** Relative Down Range Velocity Estimate Error

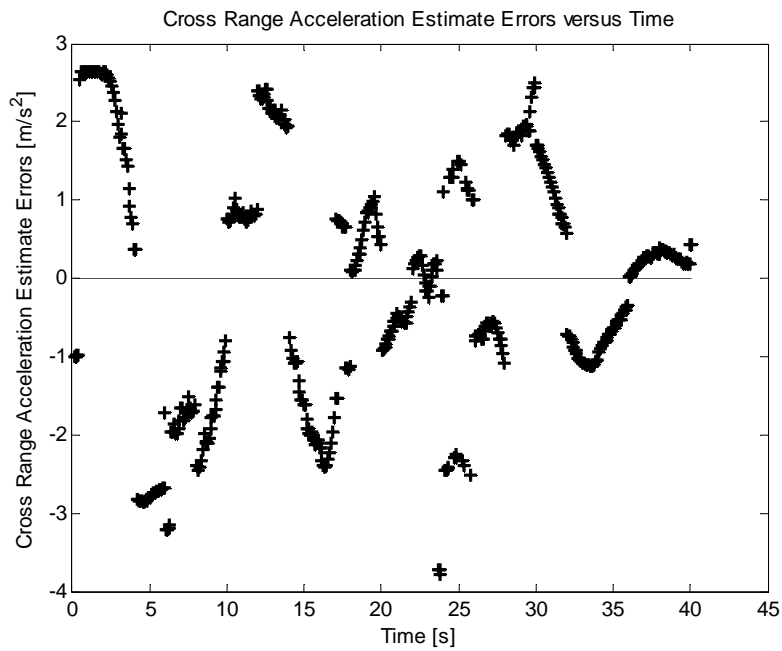


**Figure 40** Relative Cross Range Velocity Estimate Error

Figure 41 and figure 42 display the relative error of the H-infinity filter in estimating both down range acceleration and cross range acceleration of the target.



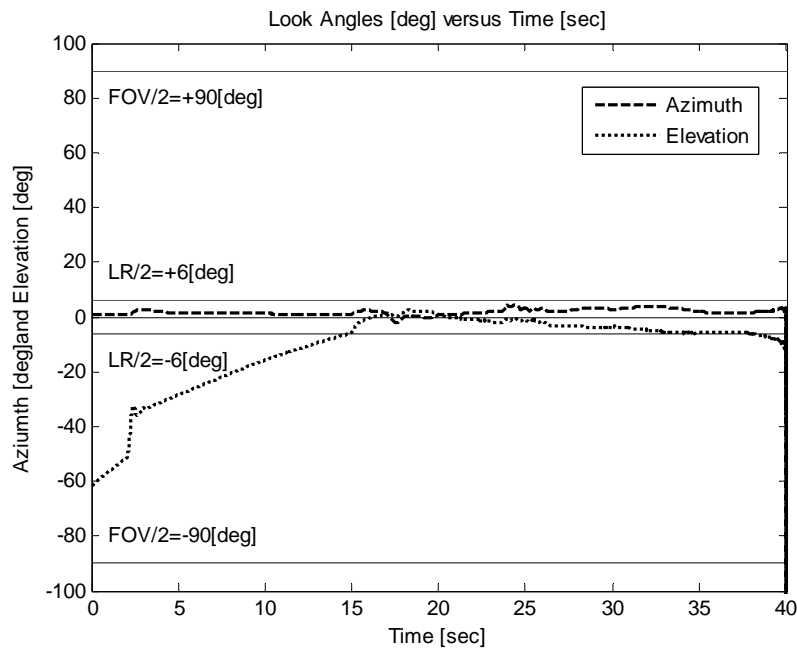
**Figure 41** Relative Down Range Acceleration Estimate Error



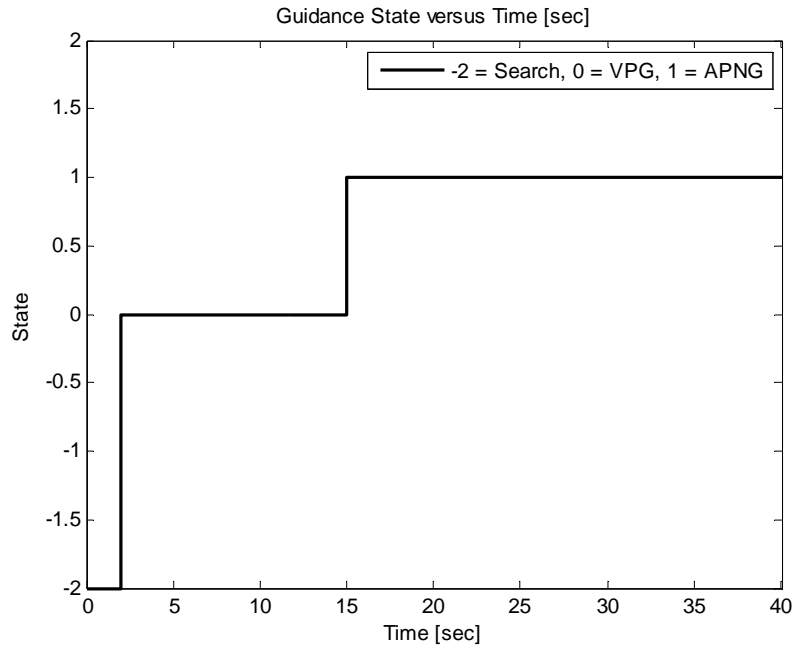
**Figure 42** Relative Cross Range Acceleration Estimate Error

The acceleration estimate provides a better estimate in the final 10sec of the engagement which is critical for the terminal APNG law.

Figure 43 displays the seeker look angles and how they come into the field-of-view (FOV =  $\pm 90^\circ$ ) and linear region (LR =  $\pm 6^\circ$ ) of the seeker. Since the seeker's FOV is  $180^\circ$ , it always "sees" the target. Notice at 2sec into the engagement, which is the fin lock time, the elevation rate changes from an intercept of  $-60^\circ$  to a rate with an intercept of  $-40^\circ$ . This is the transition from the Search state to the VPG state. VPG drives the elevation look angle toward zero. Notice the azimuth is already near zero.

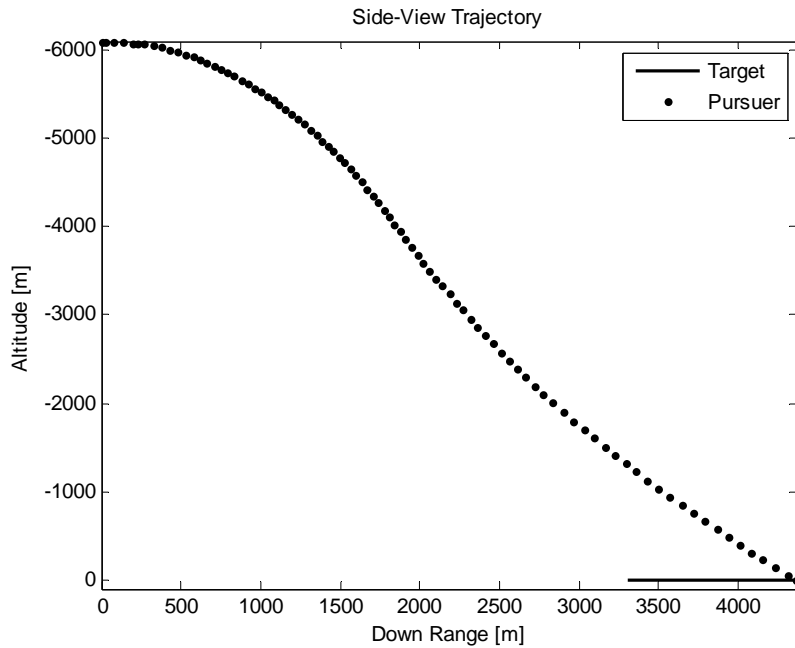


**Figure 43** Seeker Look Angles

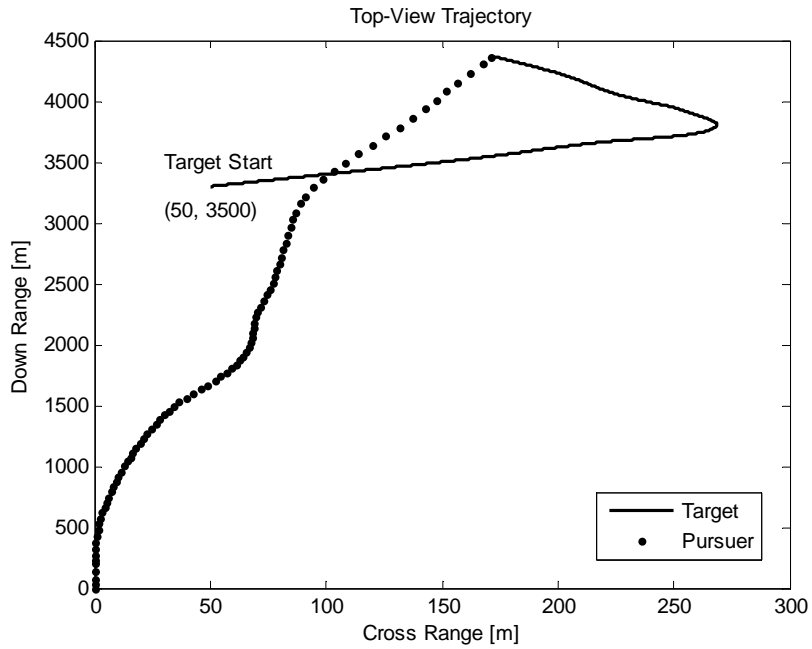


**Figure 44 Adaptive Guidance Logic**

Eventually, the elevation comes to be within the linear region of the seeker. At this time, the guidance law switches from VPG to APNG. Figure 44 (above) displays the transitions from the Search state through the VPG state to the APNG state.

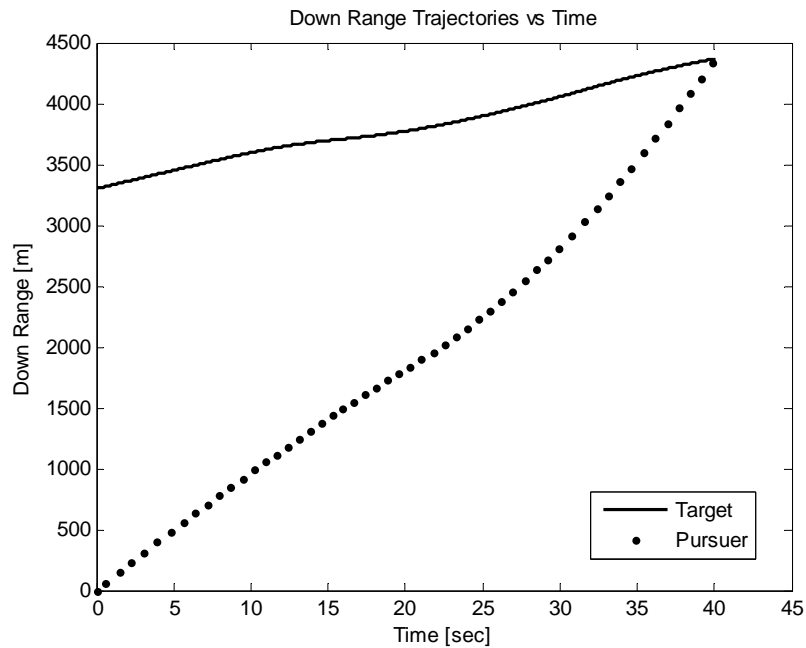


**Figure 45 Side-View Trajectories**



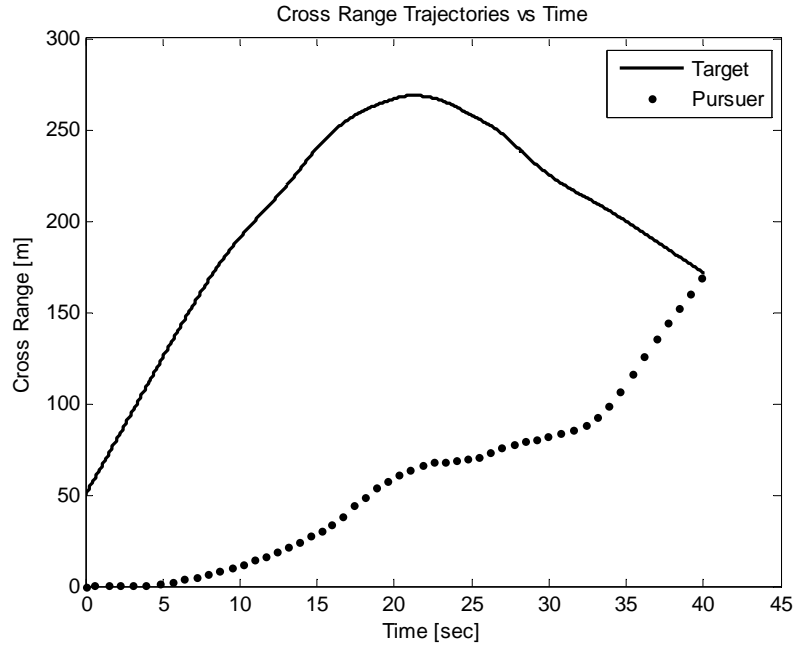
**Figure 46** Top-View Trajectories

Figure 45 and figure 46 (above) display the side-view and top-view trajectories of the pursuer as it intercepts the target.



**Figure 47** Down Range Trajectories versus Time

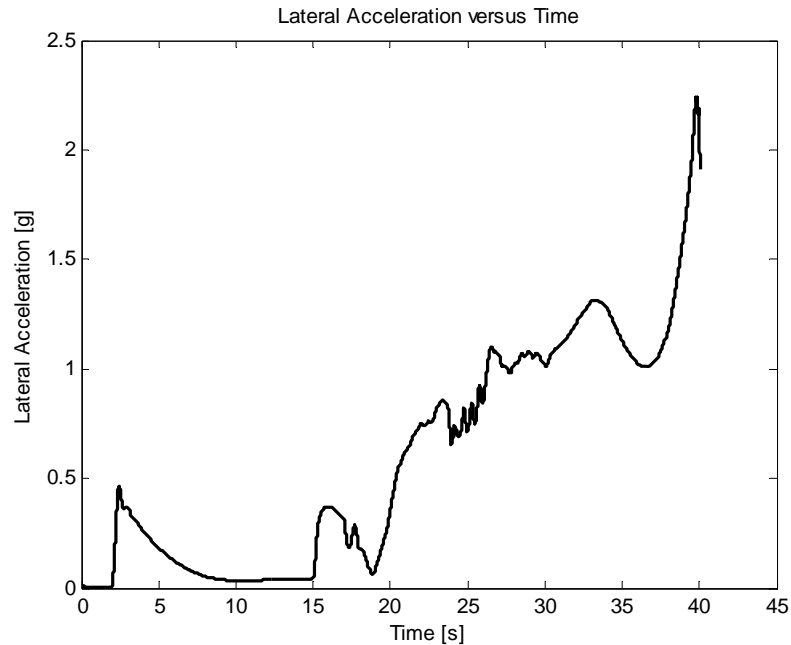




**Figure 48 Cross Range Trajectories versus Time**

In particular, figure 46 shows the initial insensitivity of the VPG law to the target's motion – where the figure shows a smooth, curved trajectory toward the target's general location. Then, at 15sec (see figure 47 and figure 48 above), which corresponds to 50m cross range and 1500m down range, the guidance law switches from VPG to APNG. The response of the pursuer to the target's motion is observed for the remainder of the trajectory.

Figure 49 (below) displays the total lateral acceleration of the pursuer. The spike at the end of the engagement corresponds to the pursuer performing its final maneuver in an attempt to intercept the target with *zero* miss distance. Theoretically, the lateral acceleration would be infinite for zero miss distance. However, the finite lateral acceleration corresponds to a non-zero miss distance. Ignoring the spike, it is observed that the total lateral acceleration was just below 1.5g which is well within the 5g requirement.



**Figure 49** Lateral Acceleration

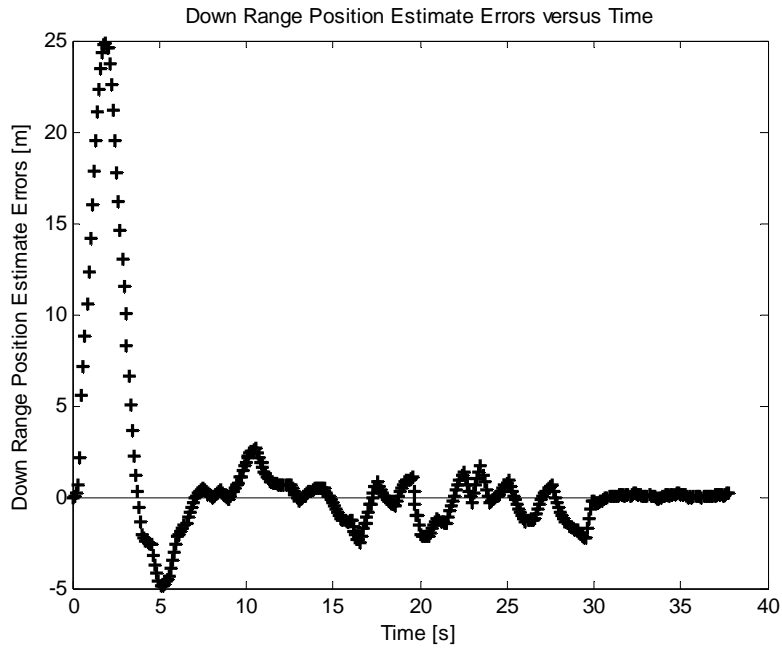
For this particular engagement, the miss distance was recorded to be 0.09m which is well within the 3m CEP (Circular Error Probable) required to be considered a successful intercept.

### 3.4.2 Simulation of a Fixed (Static) Target

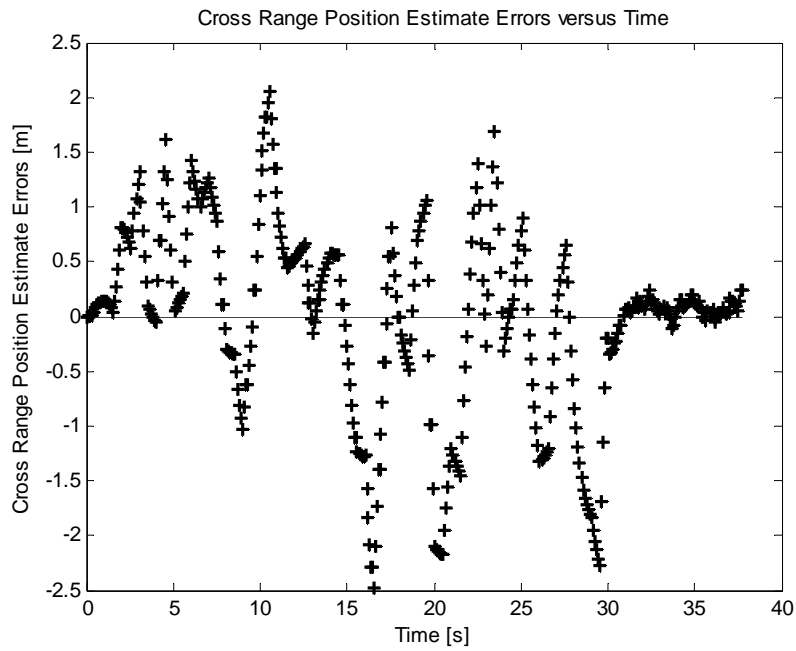
Another case of interest is to simulate the case of a fixed (static) target to be sure the system isn't tuned with any biases toward intercepting maneuvering targets. After all, remaining still could be a tactical countermeasure.

Figure 50 and figure 51 display the relative error of the H-infinity filter in estimating both down range position and cross range position of the target. The H-infinity filter performs well in estimating the position of the target, particularly in the last 10sec of the engagement. Recall that at 30sec into the engagement, the guidance mode switches from command mode to SAL mode where the target's position measurement is

performed by the pursuer and subsequently, the system delay is reduced from 500msec to 100msec.

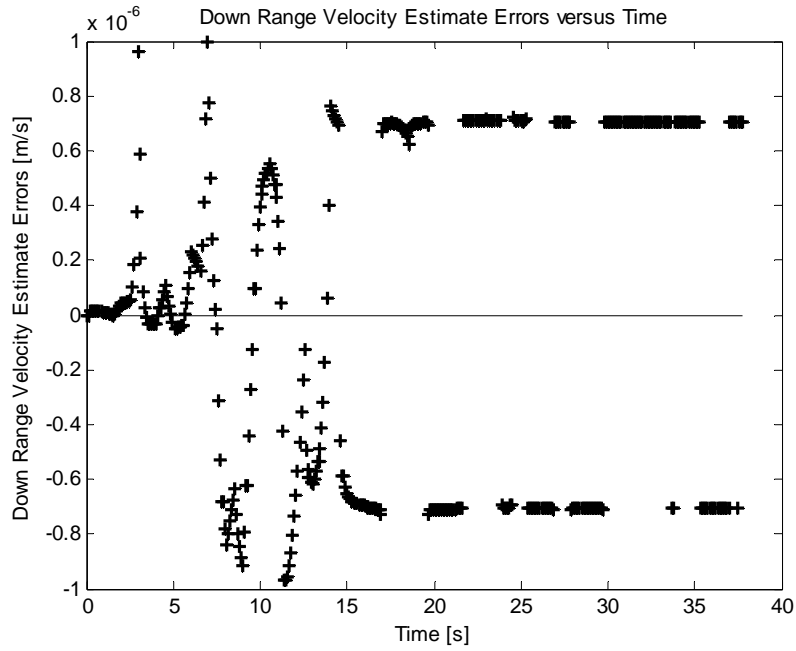


**Figure 50** Relative Down Range Position Estimate Error

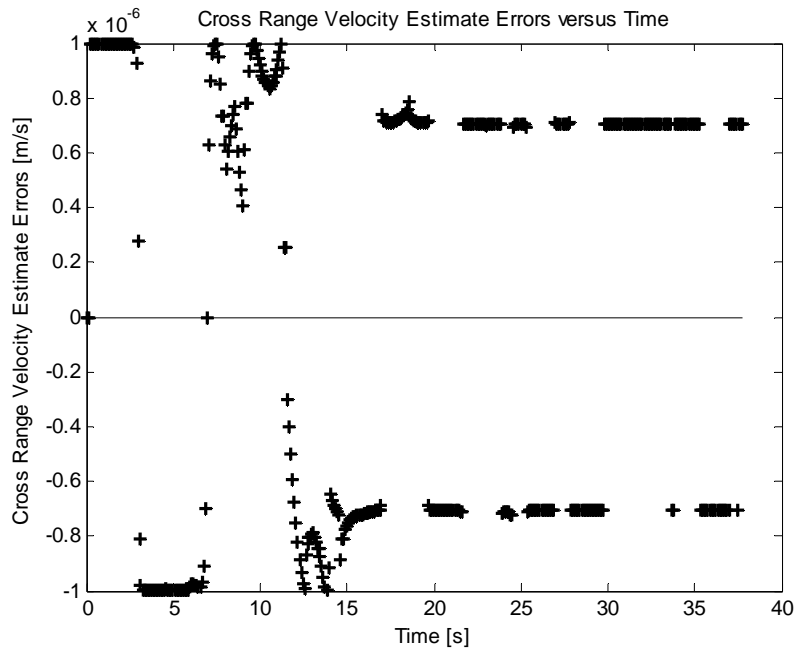


**Figure 51** Relative Cross Range Position Estimate Error

Figure 52 and figure 53 display the relative error of the H-infinity filter in estimating both down range velocity and cross range velocity of the target. Note that the target is static so the velocity is zero.

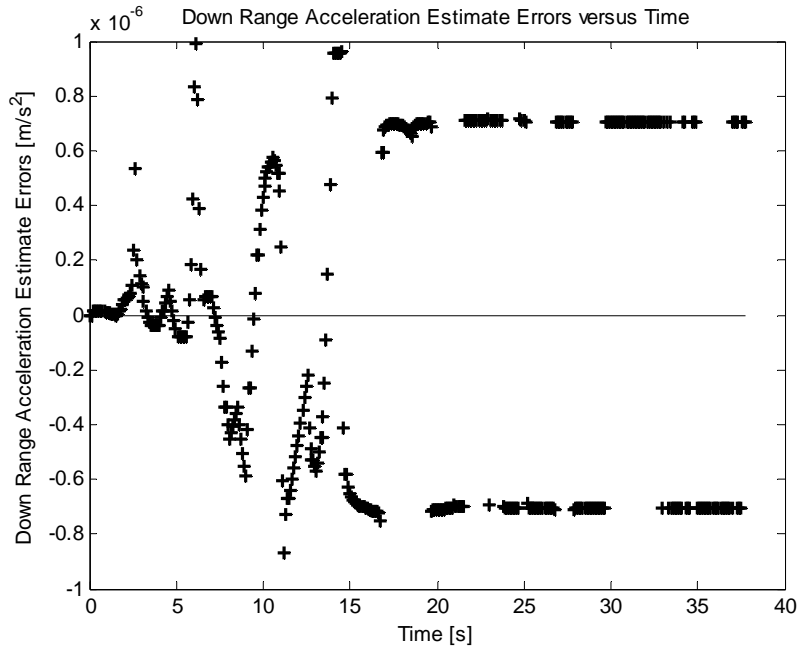


**Figure 52** Relative Down Range Velocity Estimate Error

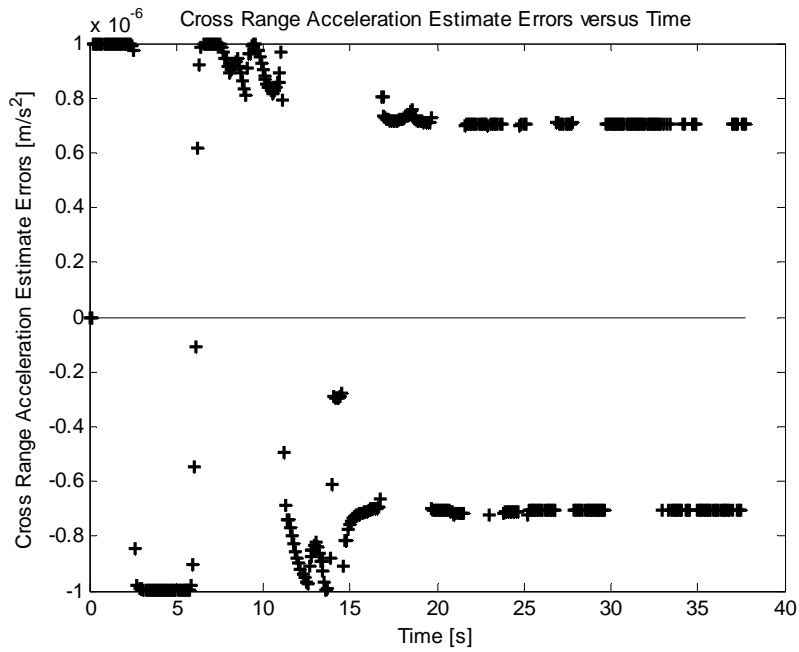


**Figure 53** Relative Cross Range Velocity Estimate Error

Figure 54 and figure 55 display the relative error of the H-infinity filter in estimating both down range acceleration and cross range acceleration of the target. Note that the target is static so the acceleration is zero.

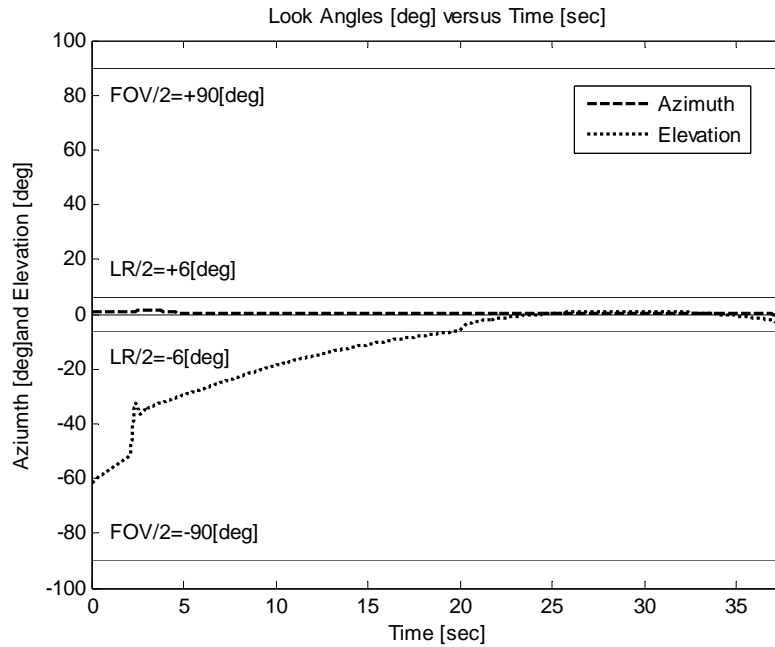


**Figure 54** Relative Down Range Acceleration Estimate Error



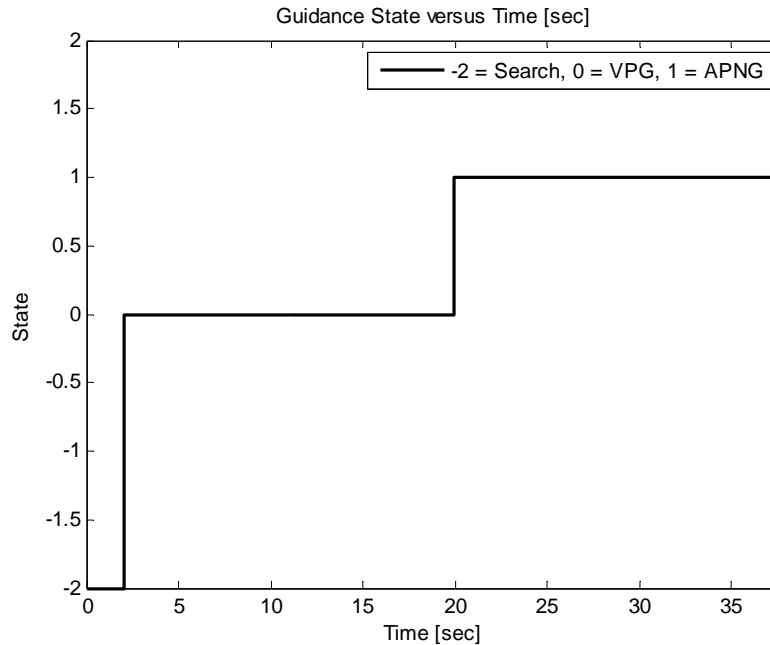
**Figure 55** Relative Cross Range Acceleration Estimate Error

Figure 56 (below) displays the seeker look angles and how they come into the field-of-view (FOV =  $\pm 90^\circ$ ) and linear region (LR =  $\pm 6^\circ$ ) of the seeker.



**Figure 56 Seeker Look Angles**

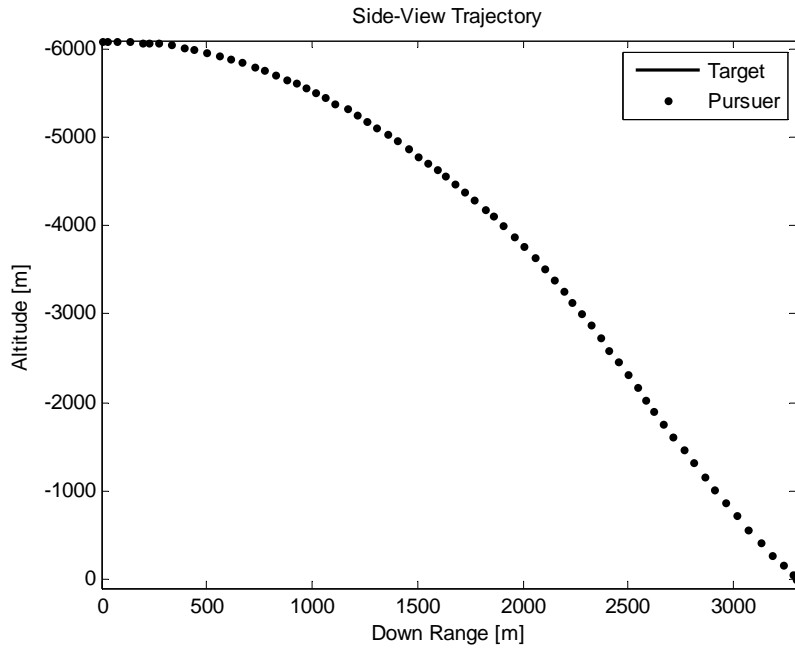
Notice at 2sec (the fin lock time for safe separation) the elevation rate changes from an intercept of  $-60^\circ$  to a rate with an intercept of  $-40^\circ$ . This is the transition from the Search state to the VPG state. VPG drives the elevation look angle toward zero. Notice the azimuth is already near zero.



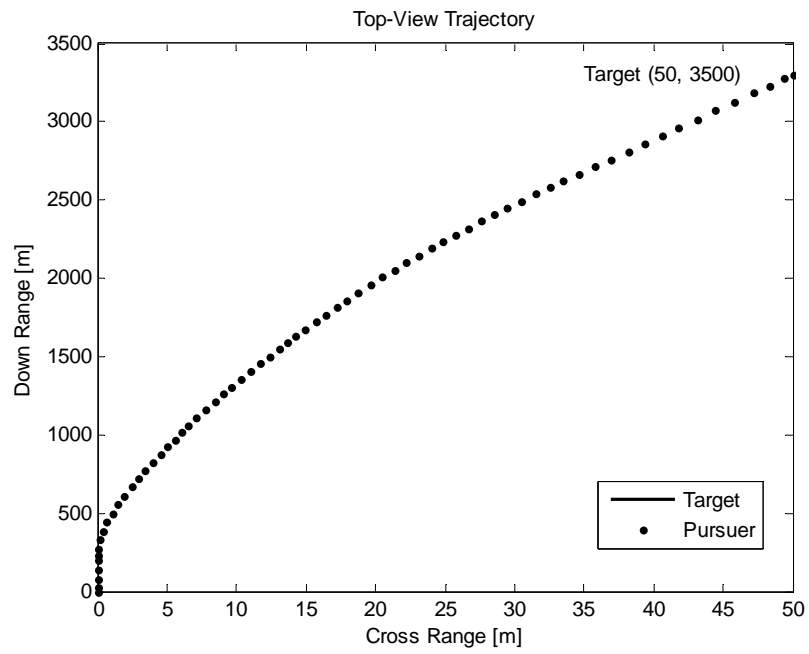
**Figure 57 Adaptive Guidance Logic**

Eventually, the elevation comes to be within the linear region of the seeker. At this time, the guidance law switches from VPG to APNG. While the APNG law is applied during this time in the engagement, the target acceleration estimate component of APNG is zero, as seen from Figure 52. Therefore, this guidance law is actually PNG, Proportional Navigation Guidance. Figure 57 (above) displays the transitions from the Search state through the VPG state to the APNG state.

Figure 58 and figure 59 (below) display the side-view and top-view trajectories of the pursuer as it intercepts the fixed (static) target.



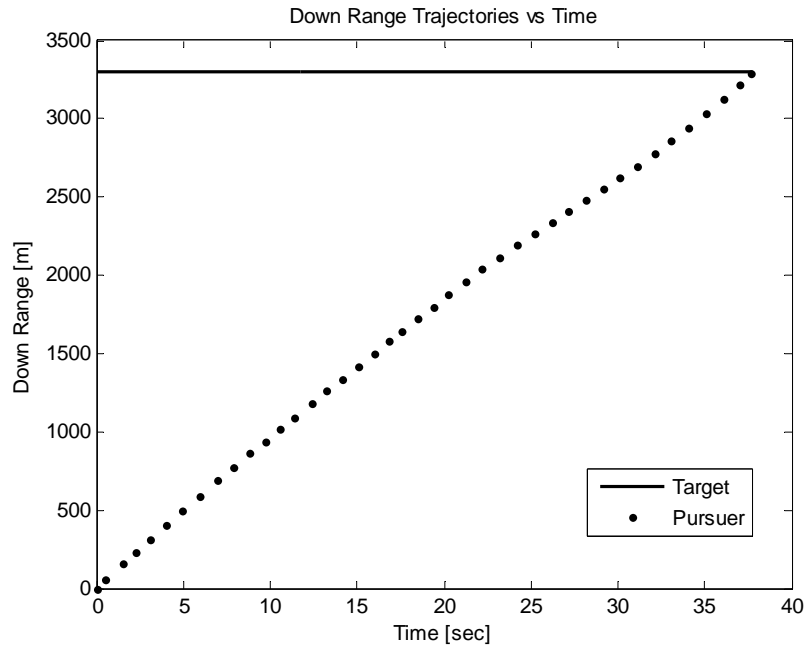
**Figure 58** Side-View Trajectories



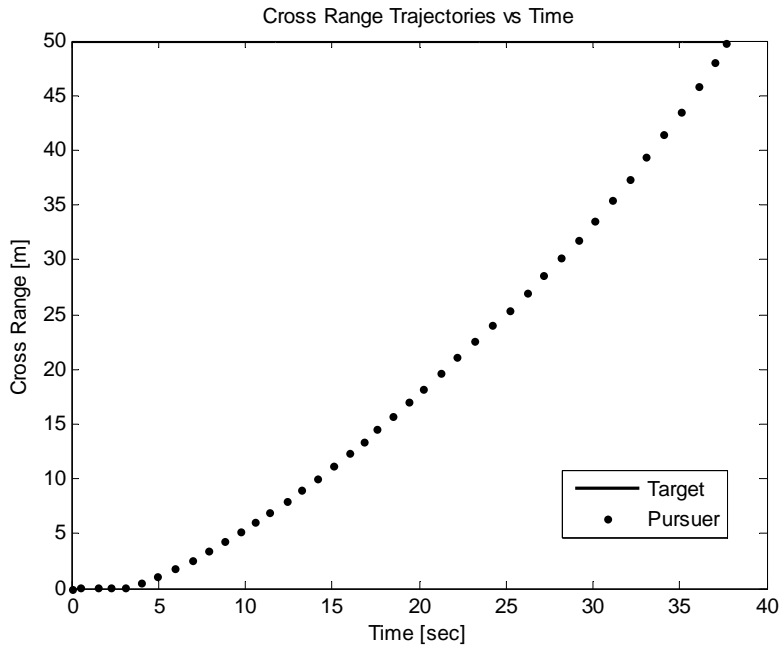
**Figure 59** Top-View Trajectories



Figure 60 and figure 61 (below) display the down range and cross range trajectories versus time.

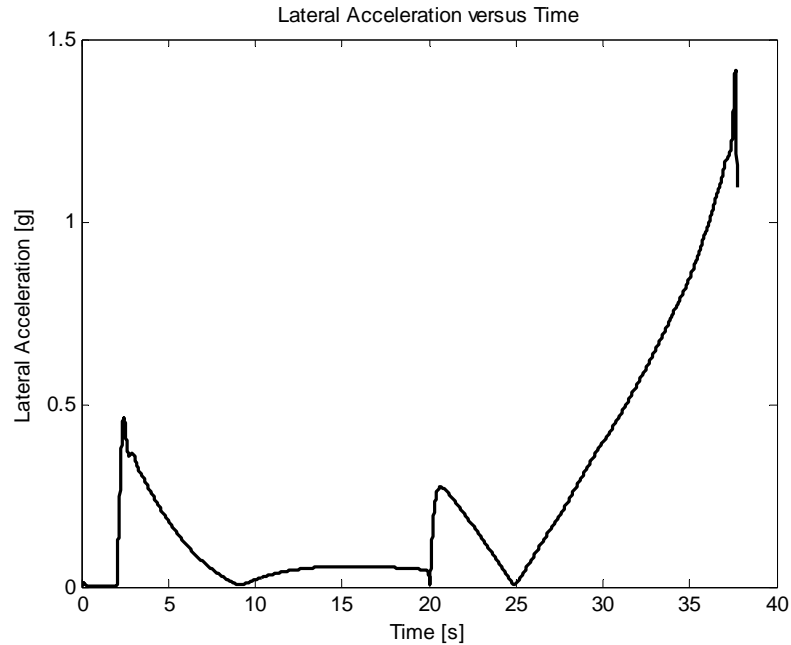


**Figure 60** Down Range Trajectories versus Time



**Figure 61** Cross Range Trajectories versus Time

Figure 62 (below) displays the total lateral acceleration of the pursuer. It is observed that the total lateral acceleration was just below 1.5g which is well within the 5g requirement.



**Figure 62** Lateral Acceleration

Finally, it is noted that for this static engagement, the miss distance was recorded to be 0.15m which is well within the 3m CEP (Circular Error Probable) required to be considered a successful intercept.

### 3.4.3 Monte Carlo Simulation of 500 Engagements

Beyond the success of the two engagements mentioned in the previous section, 500 additional pursuer/target engagements were simulated to exercise the robustness of the design.

While the target began with the same *initial position*; velocity, acceleration, and timing associated with changes in acceleration from that initial position were all changed

randomly. The following table summarizes the inputs which were varied for the Monte Carlo simulations.

**Table 5 Monte Carlo Input Data**

	Average	Standard Deviation
Down Range Velocity	-0.60m/sec	16.26m/sec
Cross Range Velocity	-0.46m/sec	16.40m/sec
Down Range Acceleration	1.09m/sec <sup>2</sup>	0.00m/sec <sup>2</sup>
Cross Range Acceleration	1.10m/sec <sup>2</sup>	0.00m/sec <sup>2</sup>
Frequency of Down Range Acceleration Change	1.28sec	0.00sec
Frequency of Cross Range Acceleration Change	0.71sec	0.00sec

The following table compares miss distance and lateral acceleration statistics (generated from 500 Monte Carlo simulations) with their requirements:

**Table 6 Monte Carlo Results**

	Average	Standard Deviation	Maximum	Requirement
Miss Distance	0.97m	1.39m	2.36m	3m
Lateral Acceleration	3.87g	0.95g	4.82g	5g

Both the maximum miss distance of 2.36m and the maximum lateral acceleration of 4.82g were within the 3m and 5g requirements, respectively. This includes 68% of the population (1 standard deviation).

### 3.5 Guidance and Control Summary

The discrete-time, constrained, H-infinity filter provided very good estimates of the target's kinematics. With these estimates, the adaptive guidance law based on velocity pursuit guidance (VPG) and augmented proportional navigation guidance

(APNG) intercepted a maneuvering target with a 0.09m miss distance which was within the 3m requirement. Furthermore, the lateral acceleration on the airframe was under 1.5g which was within the 5g requirement. This engagement is considered realistic and demanding of the pursuer in the sense that:

(1) The UAV measurements were delayed by 500msec to reflect the time it takes for the UAV to obtain the measurement through the laser range finder, perform the necessary geo-location calculations, and communicate it to the pursuer for estimation and guidance use. This 500msec delay occurs during command mode. However, when in semi-active laser (SAL) mode, the UAV designates the target while the target location measurements are being performed by the pursuer so the delay is reduced to only 100msec.

(2) The target's kinematic properties consisted of high velocity and high acceleration, including frequent maneuvering, within the limits of representative surface vehicle properties.

(3) The pursuer model includes actual aerodynamic coefficient parameters from recent flight tests and  $\frac{1}{2}^\circ$  of seeker radial bore sight error.

Next, a fixed (static) target engagement was simulated to be sure that the system was not biased for maneuvering target engagements only. It was shown that the fixed target was intercepted with a miss distance of 0.15m and applying a total lateral acceleration of 1.5g, which were both well within the limits of the requirements.

Finally, 500 Monte Carlo simulations were run to exercise the robustness of the design. It was shown that the statistics of the results proved to be within the limits of the requirements, i.e. miss distances were within 3m and total lateral accelerations were within 5g.

All of this data combines to suggest that cooperative navigation and adaptive guidance are feasible for actual implementation, which was the purpose of this research.

## CHAPTER FOUR: CONCLUSIONS AND FUTURE WORK

### 4.1 Conclusions

Chapter 2 “Cooperative Navigation” showed UAVs could estimate the kinematics of a (non-cooperative) target using a discrete-time, constrained, H-infinity filter. Clearly, the H-infinity filter outperformed the Kalman filter in the case of a maneuvering target. It was also shown that the Kalman filter took 15sec to converge while the H-infinity filter converged within 2sec. Furthermore, the H-infinity filter operates without any knowledge of the noise statistics, whereas the development of the Kalman filter equations is highly dependent on knowledge of noise statistics, i.e. zero-mean, uncorrelated, white noise. It is noted that the innovative steps associated with these results were modeling the target as noisy jerk, i.e. acceleration rate and placing limitations (constraints) on the filter’s estimate of the target’s kinematics.

Chapter 3 “Guidance and Control” showed that a relatively simple guidance algorithm could be implemented with an adaptive combination of velocity pursuit guidance (VPG) and augmented proportional navigation guidance (APNG) based on the field-of-view and linear region of the pursuer’s seeker. It was also shown that a maneuvering target was intercepted with a miss distance of 0.9m. This is within the 3m CEP which indicates target intercept. Furthermore, the lateral acceleration requirement of the pursuer was 2.8g. This is also within the requirement which is 5g.

These combined results lead the author to make a recommendation to incorporate this adaptive guidance algorithm with cooperative navigation into a low-cost, light-weight, precision-guided pursuer system with limited acceleration capability. It is noted

that the innovative step associated with these results was using the field-of-view and the linear region of the seeker as the guidance logic for switching from VPG to APNG.

## 4.2 Assumptions and Limitations

### 4.2.1 Time-Delay for UAV to Transmit Target's Position

It was assumed that the time-delay for the UAV to measure and calculate the target's range using an on-board laser range finder and to transmit that information to the pursuer was 500msec. Using the single-engagement scenario, the effects of time-delay were studied. Table 3 below was constructed from simulation where only the time-delay was changed:

**Table 7 Effects of Time-Delay on Miss Distance**

Time-Delay (msec)	Miss Distance (m)
10	0.5
100	0.5
300	0.6
500	1.0
700	1.0
900	1.4

Considering 500msec to be a nominal time-delay, it is seen from Table 4.1 that increasing the time-delay to 700msec had no immediate impact on the miss distance. However, a time-delay of 900msec increased the miss distance by 40%. Of course, reducing the time-delay has the benefit of lowering the miss distance.

### 4.3 Recommendations and Improvements

#### 4.3.1 Estimator Performance Improvements

Although no performance issues occurred, if there were issues, the filter's performance could be enhanced by either increasing the arithmetic precision, i.e. if the software developer used `float` (32 bits with 6 digits of precision) as the initial data type, this could be changed to `double` (64 bits with 10 digits of precision) or ensuring the estimation error covariance matrix  $\mathbf{P}$  is symmetric by computing  $\mathbf{P} = \frac{1}{2}(\mathbf{P} + \mathbf{P}^T)$  at each time-step.

#### 4.3.2 Algorithm Execution Time

In order to investigate the time required to execute the estimate and guidance algorithms, real-time code would be generated and benchmarked. If the algorithms cannot be executed in real-time, either increase the microprocessor's speed, or calculate off-line steady-state values for the estimation error covariance matrix and gains and hard-code them in the software.

#### 4.3.3 Including Velocity Measurement for the Estimator

The current discrete-time, constrained, H-infinity estimator relies upon the UAV providing position measurements of the target. If the UAV is capable of determining the target's velocity too, then that information could be used to improve the algorithm for estimating the target's acceleration.



#### 4.3.4 Predictor versus Estimator

The H-infinity estimator is based on *filtering* techniques where the time at which the estimate is desired coincides with the last position measurement. If the time at which the estimate is desired occurs later than the last position measurement, the H-infinity estimator is based on *predicting* techniques. This can be implemented by simply propagating the *measured* position and the *estimated* velocity and acceleration forward in time. The guidance algorithm would utilize this information as if it were the actual kinematics of the target and intercept this point on the ground – and presumably, the target would indeed be there at that point time.

If  $r_T$  is the measured position of the target, and  $\hat{v}_T$  and  $\hat{a}_T$  are the estimated velocity and acceleration of the target, respectively, then the predicted location of the target  $\hat{r}_T$  is simply

$$\hat{r}_T = r_T + \hat{v}_T t + \frac{1}{2} \hat{a}_T t^2 \quad (4.1)$$

where  $t \leq 1$  sec.

#### 4.3.5 Designation Leading

The predictor estimator in the previous section is applicable to command mode guidance. In the case of SAL mode guidance, the UAV would have to lead the target during designation.

## **APPENDIX A: AERODYNAMIC FORCES AND MOMENTS**

In this appendix, the aerodynamic forces and moments of the pursuer are presented. They are used as the equations of motion for six degree-of-freedom (6DOF) simulation.

### A.1 Parameters

$d$  is the reference diameter [in]

$S$  is cross-sectional reference area [ft<sup>2</sup>]

$Q = \frac{1}{2} \rho V^2$  is dynamic pressure [lb/ft<sup>2</sup>]

$\rho$  is atmospheric density [slugs/ft<sup>3</sup>]

$V$  is velocity [ft/sec]

$\alpha$  is angle-of-attack [rad]

$\beta$  is angle-of-sideslip [rad]

$p$ ,  $q$ , and  $r$  are angular roll, pitch, and yaw rate, respectively [rad/sec]

### A.2 Coordinate Systems and Sign Conventions

#### *Aerodynamic Coordinate System*

The x-axis is longitudinal through the nose of the body. The z-axis is up. The y-axis completes the left-hand orthogonal system and points out the right wing.

#### *Body Coordinate System*

The x-axis is longitudinal through the nose of the body. The z-axis is down. The y-axis completes the right-hand orthogonal system and points out the right wing. In the simulation, all coefficients are converted to the body coordinate system.

#### *Fin Deflection Sign Convention*

A positive fin deflection is leading edge up for the normal force analysis and leading edge to the right for the side force analysis, defined in the aerodynamic coordinate system.

### A.3 Aerodynamic Forces

All forces can be derived from the following general expression.

$$F = (C_o + C_\alpha \alpha + C_\delta \delta) QS \quad (\text{A1.1})$$

#### A.3.1 Axial Force

$C_{A_o}$  is the axial drag coefficient independent of angle-of-attack or fin deflection and is positive aft along the missile's longitudinal, x-axis. It is dimensionless and  $C_{A_o} < 0$ . However, in body coordinates, it is negative as drag opposes velocity.

$C_{A_\alpha}$  is the axial drag coefficient due to angle-of-attack with dimensions [1/rad] and is defined as  $dC_A/d\alpha$ , where  $dC_A/d\alpha < 0$ , i.e. it reduces  $C_{A_o}$  with positive angles-of-attack.

$$F_x = ma_x = (C_{A_o} + C_{A_\alpha} \alpha) QS \quad (\text{A1.2})$$

$$a_x = (C_{A_o} + C_{A_\alpha} \alpha) \frac{QS}{m} \quad (\text{A1.3})$$

The net force is positive acting along the missile's negative x-axis. This is why a negative sign is introduced in the simulation.

#### A.3.2 Normal Force

$C_{N_\alpha}$  is the normal force coefficient from the overall airframe (with the tail fins set to 0°) acting perpendicular to the longitudinal x-axis through the C.G. (center-of-gravity)

and is positive upward along the positive z-axis in aerodynamic coordinates.  $C_{N\alpha}$  has dimensions [1/rad] and is defined as  $\partial C_N / \partial \alpha$ , where  $\partial C_N / \partial \alpha < 0$  relative to the angle-of-attack.

$C_{N\delta q}$  is the normal force coefficient from the tail fins acting perpendicular to the longitudinal x-axis through the fin hinge and is positive along the positive z-axis in aerodynamic coordinates.  $C_{N\delta q}$  has dimensions [1/rad] and is defined as  $\partial C_{N\delta q} / \partial \delta_q$ , where  $\partial C_{N\delta q} / \partial \delta_q > 0$ .

$$F_z = ma_z = (C_{N\alpha}\alpha + C_{N\delta q}\delta_q)QS \quad (\text{A1.4})$$

$$a_z = N_\alpha\alpha + N_{\delta q}\delta_q = (C_{N\alpha}\alpha + C_{N\delta q}\delta_q)\frac{QS}{m} \quad (\text{A1.5})$$

The net normal force is positive acting along the missile's negative z-axis. This is why a negative sign is introduced in the simulation.

### A.3.3 Side Force

$C_{Y\beta}$  is the side force coefficient from the overall airframe (with the tail fins set to 0°) acting perpendicular to the longitudinal x-axis through the C.G. (center-of-gravity) and is positive along the positive y-axis in aerodynamic coordinates.  $C_{Y\beta}$  has dimensions [1/rad] and is defined as  $\partial C_Y / \partial \beta$ , where  $\partial C_Y / \partial \beta > 0$  relative to the angle-of-sideslip.

$C_{Y\delta r}$  is the side force coefficient from the tail fins acting perpendicular to the longitudinal x-axis through the fin hinge and is positive along the positive y-axis in aerodynamic coordinates.  $C_{Y\delta r}$  has dimensions [1/rad] and is defined as  $\partial C_{Y\delta r} / \partial \delta_r$ , where  $\partial C_{Y\delta r} / \partial \delta_r > 0$ .

$$F_y = ma_y = (C_{Y\beta}\beta + C_{Y\delta r}\delta_r)QS \quad (\text{A1.6})$$

$$a_y = L_\beta\beta + L_{\delta r}\delta_r = (C_{Y\beta}\beta + C_{Y\delta r}\delta_r) \frac{QS}{m} \quad (\text{A1.7})$$

The net side force is positive acting along the missile's positive y-axis. A negative sign is not introduced because the positive y-axis is the same in both aerodynamic and body coordinates.

#### A.4 Aerodynamic Moments

All moments can be derived from the following general expression.

$$M = (C_o + C_\alpha\alpha + C_\delta\delta)QSd + (C_a\dot{\alpha} + C_q\dot{\theta}) \frac{QSd^2}{2V} + \dots \quad (\text{A1.8})$$

##### A.4.1 Rolling Moment

$C_{l\alpha}$  is the rolling moment coefficient due to angle-of-attack and is positive for positive angle-of-attack. It has dimensions [1/rad].

$C_{l\delta p}$  is the rolling moment coefficient due to fin deflection and is positive for positive fin deflections. It has dimensions [1/rad].

$C_{lp}$  is the rolling moment coefficient due to roll rate.  $C_{lp}$  has dimensions [1/rad] and is defined as  $\partial C_l / \partial (pd / 2V)$ . In order to take the partial derivative, it is necessary to multiply/divide by  $pd / 2V$ .  $C_{lp} < 0$ .

$$M_x = I_{xx}\ddot{\phi} = C_{l\delta p}\delta pQSd + C_{lp}p \frac{QSd^2}{2V} \quad (\text{A1.9})$$

$$\ddot{\phi} = \dot{p} = L_{\delta p}\delta p + L_p p = C_{l\delta p}\delta p \frac{QSd}{I_{xx}} + C_{lp}p \frac{QSd^2}{2VI_{xx}} \quad (\text{A1.10})$$

No changes are needed for the simulation. Whatever rolling moments occur in aerodynamic coordinates also occur in body coordinates.

#### A.4.2 Pitching Moment

$C_{m_o}$  is the pitching moment coefficient due to the lug producing a nose-up.  $C_{m_o}$  is dimensionless.

$C_{m_\alpha}$  is the pitching moment coefficient due to angle-of-attack producing a nose-down for static stability.  $C_{m_\alpha}$  has dimensions [1/rad] and is defined as  $\partial C_m / \partial \alpha$ , where  $\partial C_m / \partial \alpha < 0$ .

$C_{m_{\delta q}}$  is the pitching moment coefficient due to normal fin deflection producing a nose-down.  $C_{m_{\delta q}}$  has dimensions [1/rad] and is defined as  $\partial C_m / \partial \delta_q$ , where  $\partial C_m / \partial \delta_q < 0$ .

$C_{m_q}$  is the pitching moment coefficient due to pitch rate.  $C_{m_q}$  has dimensions [1/rad] and is defined as  $\partial C_m / \partial (qd / 2V)$ . In order to take the partial derivative, it is necessary to multiply/divide by  $qd / 2V$ .  $C_{m_q} < 0$ .

$$M_y = I_{yy} \ddot{\theta} = (C_{m_o} + C_{m_\alpha} \alpha + C_{m_{\delta q}} \delta q) Q S d + C_{m_q} q \frac{Q S d^2}{2V} \quad (\text{A1.11})$$

$$\ddot{\theta} = \dot{q} = M_\alpha \alpha + M_{\delta q} \delta q + M_q q = (C_{m_o} + C_{m_\alpha} \alpha + C_{m_{\delta q}} \delta q) \frac{Q S d}{I_{yy}} + C_{m_q} q \frac{Q S d^2}{2V I_{yy}} \quad (\text{A1.12})$$

No changes are needed for the simulation. Whatever pitching moments occur in aerodynamic coordinates also occur in body coordinates.

### A.4.3 Yawing Moment

$C_{n\beta}$  is the yawing moment coefficient due to angle-of-sideslip producing a nose-left for static stability.  $C_{n\beta}$  has dimensions [1/rad] and is defined as  $\partial C_n / \partial \beta$ , where  $\partial C_n / \partial \beta > 0$ .

$C_{n\delta_r}$  is the yawing moment coefficient due to side fin deflection producing a nose-left.  $C_{n\delta_r}$  has dimensions [1/rad] and is defined as  $\partial C_n / \partial \delta_r$ , where  $\partial C_n / \partial \delta_r < 0$ .

$C_{nr}$  is the yawing moment coefficient due to yaw rate.  $C_{nr}$  has dimensions [1/rad] and is defined as  $\partial C_n / \partial (rd/2V)$ . In order to take the partial derivative, it is necessary to multiply/divide by  $rd/2V$ .  $C_{nr} < 0$ .

$$M_z = I_{zz} \ddot{\psi} = (C_{n\beta} \beta + C_{n\delta_r} \delta_r) Q S d + C_{nr} r \frac{Q S d^2}{2V} \quad (\text{A1.13})$$

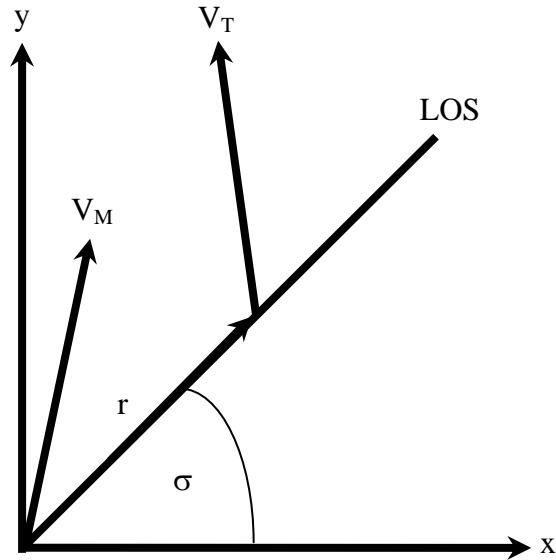
$$\ddot{\psi} = \dot{r} = M_\beta \beta + M_{\delta_r} \delta_r + M_{\dot{\psi}} \dot{\psi} = (C_{n\beta} \beta + C_{n\delta_r} \delta_r) \frac{Q S d}{I_{zz}} + C_{nr} r \frac{Q S d^2}{2V I_{zz}} \quad (\text{A1.14})$$

No changes are needed for the simulation. Whatever yawing moments occur in aerodynamic coordinates also occur in body coordinates.



**APPENDIX B: LYAPUNOV STABILITY PROOF FOR THE  
AUGMENTED PROPORTIONAL NAVIGATION GUIDANCE LAW**

The figure below represents a missile/target engagement where  $V_M$  is the missile's velocity,  $V_T$  is the target's velocity,  $r$  is the range between the missile and the target along the line-of-sight (LOS), and  $\sigma$  is the line-of-sight angle.



Since the Augmented Proportional Navigation Guidance law is based upon the rate of the line-of-sight angle ( $\dot{\sigma}$ ), a relationship needs to be defined involving  $\dot{\sigma}$ . From the figure above

$$\sin \sigma = y / r \quad (\text{A2.1})$$

Assuming the small angle approximation

$$\sigma = y / r \quad (\text{A2.2})$$

Taking the time derivative

$$r\dot{\sigma} + \dot{r}\sigma = \dot{y} \quad (\text{A2.3})$$

Taking the time derivative again

$$r\ddot{\sigma} + 2\dot{r}\dot{\sigma} + \ddot{r}\sigma = \ddot{y} \quad (\text{A2.4})$$

Rewriting

$$\ddot{\sigma} = -\frac{2\dot{r}}{r}\dot{\sigma} - \frac{\ddot{r}}{r}\sigma + \frac{1}{r}\ddot{y} \quad (\text{A2.5})$$

It is noted that  $\ddot{y} = -a_M + a_T$ , where  $a_M$  is the acceleration of the missile (a control variable of the dynamic system) and  $a_T$  is the acceleration of the target (a disturbance to the dynamic system). By letting  $x_1$  represent the line-of-sight angle ( $\sigma$ ) and  $x_2$  represent the rate of the line-of-sight angle ( $\dot{\sigma}$ ), the following state-space system is formed

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{2\dot{r}}{r}x_2 - \frac{\ddot{r}}{r}x_1 + \frac{1}{r}(-a_M + a_T) \end{aligned} \quad (\text{A2.6})$$

The next step is to select a Lyapunov function ( $V$ ) based on the line-of-sight rate ( $x_2$ )

$$V = \frac{1}{2}x_2^2 \quad (\text{A2.7})$$

which is *positive definite* for all  $x_2$ . The time derivative of  $V$  is

$$\dot{V} = x_2\dot{x}_2 \quad (\text{A2.8})$$

Substituting the dynamics from equation (A2.6) yields

$$\dot{V} = x_2 \left[ -\frac{2\dot{r}}{r}x_2 - \frac{\ddot{r}}{r}x_1 + \frac{1}{r}(-a_M + a_T) \right] \quad (\text{A2.9})$$

Provided

$$\dot{V} \leq 0 \quad (\text{A2.10})$$

the guidance law that stabilizes the system given by equation (A2.6) is

$$a_M = k_1 v_c \dot{\sigma} + k_2 a_T \quad (\text{A2.11})$$

where  $k_1$  and  $k_2$  are constants to be determined,  $v_c$  is called the “closing velocity” defined as  $v_c = -\dot{r}$ , and all other parameters have been defined earlier. Substituting equation (A2.11) into equation (A2.9) yields

$$\dot{V} = x_2 \left[ -\frac{2\dot{r}}{r} x_2 - \frac{\ddot{r}}{r} x_1 + \frac{1}{r} (-k_1 v_c \dot{\sigma} - k_2 a_T + a_T) \right] \leq 0 \quad (\text{A2.12})$$

Recall that  $x_1 = \sigma$  and  $x_2 = \dot{\sigma}$

$$-\frac{2\dot{r}}{r} \dot{\sigma} - \frac{\ddot{r}}{r} \sigma + \frac{1}{r} (-k_1 v_c \dot{\sigma} - k_2 a_T + a_T) \leq 0 \quad (\text{A2.13})$$

Grouping like terms

$$-\frac{1}{r} (2\dot{r} + k v_c) \dot{\sigma} - \left( \frac{\ddot{r}}{r} \right) \sigma - \frac{a_T}{r} (k_2 - 1) \leq 0 \quad (\text{A2.14})$$

Assuming the missile has zero acceleration along the line-of-sight ( $\ddot{r} = 0$ )

$$-\frac{1}{r} (2\dot{r} + k_1 v_c) \dot{\sigma} - \frac{a_T}{r} (k_2 - 1) \leq 0 \quad (\text{A2.15})$$

For the inequality to hold each of the remaining two terms must be less than or equal to zero

$$\begin{aligned} -\frac{1}{r} (2\dot{r} + k_1 v_c) \dot{\sigma} &\leq 0 \\ -\frac{a_T}{r} (k_2 - 1) &\leq 0 \end{aligned} \quad (\text{A2.16})$$

Recalling that  $v_c = -\dot{r}$ , the first equation in (A2.16) yields

$$k_1 \geq 2 \quad (\text{A2.17})$$

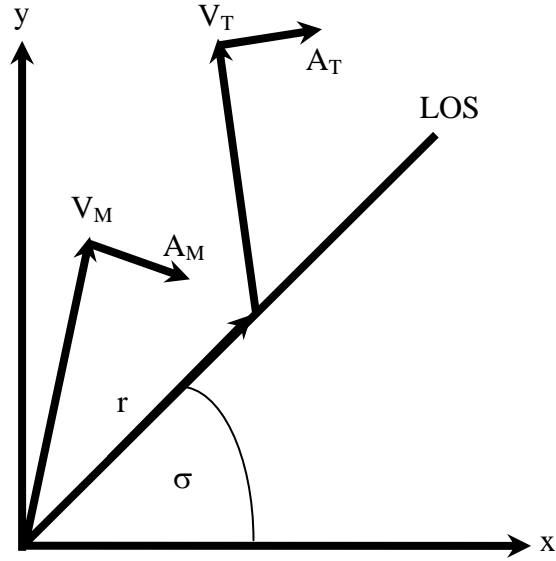
The second equation in (A2.16) yields

$$\begin{aligned} k_2 \geq 1 \quad \text{sgn}(a_T) &\geq 0 \\ k_2 \leq 1 \quad \text{sgn}(a_T) &\leq 0 \end{aligned} \quad (\text{A2.18})$$

Provided  $k_1 \geq 2$  and either  $k_2 \geq 1$  with  $\text{sgn}(a_T) \geq 0$  or  $k_2 \leq 1$  and  $\text{sgn}(a_T) \leq 0$ , the guidance law  $a_M = k_1 v_c \dot{\sigma} + k_2 a_T$  guarantees asymptotic stability with respect to  $\dot{\sigma}$ . Note this conclusion is based on two assumptions: (1) the small angle approximation and (2) the missile has zero acceleration along the line-of-sight.

**APPENDIX C: OPTIMAL GAIN FOR THE AUGMENTED  
PROPORTIONAL NAVIGATION GUIDANCE LAW**

The figure below represents a missile/target engagement where  $V_M$  is the missile's velocity,  $V_T$  is the target's velocity,  $r$  is the range between the missile and the target along the line-of-sight (LOS), and  $\sigma$  is the line-of-sight angle.



Assuming the target maneuvers with normal acceleration  $A_T$  and the missile maneuvers with normal acceleration  $A_M$ , the equation for the acceleration of the system is

$$\ddot{y} = -A_M + A_T \quad (\text{A3.1})$$

By letting  $x_1$  represent position ( $y$ ) and  $x_2$  represent velocity ( $\dot{y}$ ), the following state-space system is formed

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -A_M + A_T \end{aligned} \quad (\text{A3.2})$$

If the normal acceleration of the missile ( $A_M$ ) is considered to be the control ( $u$ ) and the normal acceleration of the target ( $A_T$ ) is considered to be a disturbance ( $w$ ), then the system may be written as

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} + \mathbf{Dw} \\ \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u + \begin{pmatrix} 0 \\ 1 \end{pmatrix} w \end{aligned} \quad (\text{A3.3})$$

Applying optimal control to this problem, the goal is to minimize the following cost function

$$\min_u J = \frac{b}{2} x_1(t_f) + \int_{t_0}^{t_f} u^2(t) dt \quad (\text{A3.4})$$

where  $b > 0$  if the miss distance is to be minimized. Adjoining the state in equation (A3.3) to the cost function in equation (A3.4) yields the Hamiltonian of the system

$$H = \frac{1}{2} u^2 + \boldsymbol{\lambda}^T (\mathbf{Ax} + \mathbf{Bu} + \mathbf{Dw}) \quad (\text{A3.5})$$

The optimal solution is found by solving the following set of equations

$$\begin{aligned} \dot{\boldsymbol{\lambda}}^T &= -\frac{\partial H}{\partial \mathbf{x}} \\ 0 &= \frac{\partial H}{\partial \mathbf{u}} \\ \lambda_i(t_f) &= \frac{\partial \phi}{\partial x_i} \end{aligned} \quad (\text{A3.6})$$

where  $\phi = \frac{b}{2} x_1(t_f)$  represents the terminal portion of the cost function. Solving the set of equations yields

$$\begin{aligned} \dot{\boldsymbol{\lambda}} &= -\mathbf{A}^T \boldsymbol{\lambda} \\ \dot{\lambda}_1 &= 0 \\ \dot{\lambda}_2 &= -\lambda_1 \end{aligned} \quad (\text{A3.7})$$

$$\begin{aligned} 0 &= \mathbf{u} + \boldsymbol{\lambda}^T \mathbf{B} \\ \mathbf{u} &= -\boldsymbol{\lambda}^T \mathbf{B} \\ \mathbf{u} &= -(\lambda_1 \quad \lambda_2) \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ u &= -\lambda_2 \end{aligned} \quad (\text{A3.8})$$



$$\begin{aligned}\lambda_1(t_f) &= bx_1(t_f) \\ \lambda_2(t_f) &= 0\end{aligned}\tag{A3.9}$$

Equation (A3.7) gives  $\dot{\lambda}_1 = 0$ , hence  $\lambda_1 = \text{const.}$  Since equation (A3.9) gives

$$\lambda_1(t_f) = bx_1(t_f)$$

$$\lambda_1(t) = bx_1(t_f)\tag{A3.10}$$

Equation (A3.7) also gives  $\dot{\lambda}_2 = -\lambda_1$ . From equation (A3.10)  $\dot{\lambda}_2 = -bx_1(t_f)$ . Integrating

from  $t$  to  $t_f$  and noting that equation (A3.9) gives  $\lambda_2(t_f) = 0$

$$\lambda_2(t) = bx_1(t_f)(t_f - t)\tag{A3.11}$$

From equation (A3.8)

$$u(t) = -bx_1(t_f)(t_f - t)\tag{A3.12}$$

Returning to the state equation (A3.3)

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -u + w\end{aligned}\tag{A3.13}$$

This with equation (A3.12) implies

$$\dot{x}_2 = bx_1(t_f)(t_f - t) + w\tag{A3.14}$$

Integrating from  $t_0$  to  $t_f$  yields

$$x_2(t) = \frac{1}{2}bx_1(t_f)\left[(t_f - t)^2 - (t_f - t_0)^2\right] + w(t - t_0) + x_2(t_0)\tag{A3.15}$$

Integrating  $\dot{x}_1 = x_2$  from  $t_0$  to  $t_f$  yields

$$\begin{aligned}x_1(t) &= \frac{1}{2}bx_1(t_f)\left\{-\frac{1}{3}\left[(t_f - t)^3 - (t_f - t_0)^3\right] - (t_f - t_0)^2(t - t_0)\right\} \\ &\quad + \frac{1}{2}w(t - t_0)^2 + x_2(t_0)(t - t_0) + x_1(t_0)\end{aligned}\tag{A3.16}$$

At  $t = t_f$

$$\begin{aligned}
x_1(t_f) &= -\frac{1}{3}bx_1(t_f)(t_f - t_0)^3 + \frac{1}{2}w(t_f - t_0)^2 + x_2(t_0)(t_f - t_0) + x_1(t_0) \\
x_1(t_f) + \frac{1}{3}bx_1(t_f)(t_f - t_0)^3 &= \frac{1}{2}w(t_f - t_0)^2 + x_2(t_0)(t_f - t_0) + x_1(t_0) \\
\left[1 + \frac{1}{3}b(t_f - t_0)^3\right]x_1(t_f) &= \frac{1}{2}w(t_f - t_0)^2 + x_2(t_0)(t_f - t_0) + x_1(t_0) \\
x_1(t_f) &= \frac{\frac{1}{2}w(t_f - t_0)^2 + x_2(t_0)(t_f - t_0) + x_1(t_0)}{1 + \frac{1}{3}b(t_f - t_0)^3}
\end{aligned} \tag{A3.17}$$

Recall from equation (A3.12)  $u(t) = -bx_1(t_f)(t_f - t)$  therefore

$$\begin{aligned}
u(t_0) &= -bx_1(t_f)(t_f - t_0) \\
u(t_0) &= -b \left[ \frac{\frac{1}{2}w(t_f - t_0)^2 + x_2(t_0)(t_f - t_0) + x_1(t_0)}{1 + \frac{1}{3}b(t_f - t_0)^3} \right] (t_f - t_0) \\
u(t_0) &= -b \left[ \frac{\frac{1}{2}w(t_f - t_0)^2 + x_2(t_0)(t_f - t_0) + x_1(t_0)}{1 + \frac{1}{3}b(t_f - t_0)^2} \right] \\
u(t_0) &= - \left[ \frac{\frac{1}{2}w(t_f - t_0)^2 + x_2(t_0)(t_f - t_0) + x_1(t_0)}{\frac{1}{b} + \frac{1}{3}(t_f - t_0)^2} \right]
\end{aligned} \tag{A3.18}$$

In general

$$\begin{aligned}
u(t) &= - \left[ \frac{\frac{1}{2}w(t_f - t)^2 + x_2(t)(t_f - t) + x_1(t)}{\frac{1}{b} + \frac{1}{3}(t_f - t)^2} \right] \\
u(t) &= -3 \left[ \frac{\frac{1}{2}w(t_f - t)^2 + x_2(t)(t_f - t) + x_1(t)}{\frac{3}{b} + (t_f - t)^2} \right] \\
u(t) &= -3 \left[ \frac{x_1(t) + x_2(t)(t_f - t) + \frac{1}{2}w(t_f - t)^2}{\frac{3}{b} + (t_f - t)^2} \right]
\end{aligned} \tag{A3.19}$$

If the miss distance is required to be zero, then the cost  $b \rightarrow \infty$ . In this case,

$$u(t) = -3 \left[ \frac{x_1(t)}{(t_f - t)^2} + \frac{x_2(t)}{(t_f - t)} + \frac{1}{2}w \right] \tag{A3.20}$$

Since the Augmented Proportional Navigation Guidance law is based upon the rate of the line-of-sight angle ( $\dot{\sigma}$ ), a relationship needs to be defined involving  $\dot{\sigma}$ . From the figure above

$$\sin \sigma = y / r \quad (\text{A3.21})$$

Assuming the small angle approximation

$$\sigma = y / r \quad (\text{A3.22})$$

Taking the time derivative

$$\begin{aligned} r\dot{\sigma} + \dot{r}\sigma &= \dot{y} \\ \dot{\sigma} &= \frac{\dot{y}}{r} - \frac{\dot{r}\sigma}{r} \end{aligned} \quad (\text{A3.23})$$

Defining the closing velocity  $V_c = -\dot{r}$ , equation (A3.23) becomes

$$\dot{\sigma} = \frac{\dot{y}}{r} + \frac{V_c\sigma}{r} \quad (\text{A3.24})$$

Substituting  $\sigma = y / r$  from equation (A3.22) yields

$$\dot{\sigma} = \frac{\dot{y}}{r} + \frac{V_c y}{r^2} \quad (\text{A3.25})$$

Also taking note that  $r = V_c t$

$$\begin{aligned} \dot{\sigma} &= \frac{\dot{y}}{V_c t} + \frac{V_c y}{V_c^2 t^2} \\ \dot{\sigma} &= \frac{\dot{y}}{V_c t} + \frac{y}{V_c t^2} \\ \dot{\sigma} V_c &= \frac{\dot{y}}{t} + \frac{y}{t^2} \end{aligned} \quad (\text{A3.26})$$

Generalizing

$$\dot{\sigma} V_c = \frac{x_2(t)}{(t_f - t)} + \frac{x_1(t)}{(t_f - t)^2} \quad (\text{A3.27})$$

Substituting this into equation (A3.20) yields

$$u(t) = -3\left(\dot{\sigma} V_c + \frac{1}{2} w\right) \quad (\text{A3.28})$$

which shows that the optimal gain for the augmented proportional navigation guidance law is 3.

## LIST OF REFERENCES

1. Ablavsky V. and Snorrason, M., “*Optimal Search for a Moving Target: A Geometric Approach*,” AIAA Guidance, Navigation, and Control Conference and Exhibit, Denver, Colorado, Aug. 2000.
2. Arslan, G., Wolfe, J., Shamma, J., and Speyer, J., “*Optimal Planning for Autonomous Air Vehicle Battle Management*,” Proceedings of the 41<sup>st</sup> IEEE Conference on Decision and Control, pp. 3782–3787, Dec. 2002.
3. Atkinson, M., “Contract Nets for Control of Distributed Agents in Unmanned Air Vehicles,” 2<sup>nd</sup> AIAA Unmanned Unlimited Systems, Technologies, and Operations, San Diego, CA, 15-18 Sep. 2003.
4. Bellingham, J., Tillerson, M., Alighanbari, M., and How, J., “*Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environment*,” Proceedings of the 41<sup>st</sup> IEEE Conference on Decision and Control, pp. 2816–2822, Dec. 2002.
5. Ben-Asher, J. and Yaesh, I., “*Advances in Missile Guidance Theory*,” AIAA Progress in Astronautics and Aeronautics, Vol. 180, 1998.
6. Boeing Company report, “*Boeing Scores Direct Hit in Laser JDAM Moving Target Test*,” The Boeing Company, 11 Jul. 2006.  
[http://www.boeing.com/defense-space/missiles/jdam/news/2006/q3/060711a\\_nr.html](http://www.boeing.com/defense-space/missiles/jdam/news/2006/q3/060711a_nr.html)
7. Brown, M. and Carter, D., “*Geolocation of Unmanned Aerial Vehicles in GPS-Degraded Environments*,” AIAA Infotech Conference and Exhibit, Arlington, Virginia, 26-29 Sep. 2005.

8. Brunke, S. and Campbell, M. E., “*Square Root Sigma Point Filtering for Aerodynamic Model Estimation,*” AIAA Journal of Guidance, Control, and Dynamics, Vol. 27, No. 2, pp. 314–317, 2004.
9. Bryson, A. and Ho, Y., “*Applied Optimal Control: Optimization, Estimation, and Control,*” Hemisphere Publishing, 1975.
10. Chadwick, W. and Rose, C., “*A Guidance Law for General Surface Targets*” AIAA Journal of Guidance and Control, Vol. 6, No. 6, pp. 526-529, Nov.-Dec. 1983.
11. Chandler, P., Pachter, M., and Rasmussen, S., “*UAV Cooperative Control*” Proceedings of the American Control Conference, Arlington, Virginia, pp. 50-55, Jun. 25-27, 2001.
12. Chatterji, G. and Pachter, M., “*Modified Velocity Pursuit Guidance Law with Crosswind Correction for Missiles against Surface Targets,*” AIAA Guidance, Navigation and Control Conference, Reston Virginia, pp. 1566-1571, 1991.
13. Childers, M., “*Near Autonomous Unmanned Systems,*” AIAA Infotech Conference and Exhibit, Rohnert Park, California, 7-10 May 2007.
14. Christophersen, H., Pickell, W., Koller, A., Kannan, S., and Johnson, E., “*Small Adaptive Flight Control Systems for UAVs using FPGA/DSP Technology,*” 2<sup>nd</sup> AIAA Unmanned Unlimited Technologies Conference, Workshop and Exhibit, Chicago, IL, 20-23 Sep. 2004.
15. Clifford, J., Towhidnejad, M., and Behi, F., “*Target Acquisition Cooperative Unmanned Vehicle System,*” AIAA Infotech Conference and Exhibit, Rohnert Park, California, 7-10 May 2007.

16. Clough, B., “*Unmanned Aerial Vehicles: Autonomous Control Challenges, A Researcher’s Perspective*,” 2<sup>nd</sup> AIAA Unmanned Unlimited Systems, Technologies, and Operations, San Diego, CA, 15-18 Sep. 2003.
17. Cochran, C., Gorman, D., and Dumoulin, J., “*Space Handbook*,” Air Command and Staff College, 1985.
18. Defense Advanced Research Projects Agency (DARPA) report, “*Darpa Demonstrates Affordable Moving Surface Target Engagement Using JDAM Weapon*,” 12 Sep. 2002.  
<http://www.darpa.mil/body/news/2002/AMSTE02.pdf>
19. Defense Advanced Research Projects Agency (DARPA) report, “*AMSTE Demonstrates Improved Tracking Capability with Live JDAM*,” 23 Oct. 2003.  
[http://www.darpa.mil/body/news/2003/amste\\_10\\_03.pdf](http://www.darpa.mil/body/news/2003/amste_10_03.pdf)
20. Deyst, J. and Price, C., “*Optimal Stochastic Guidance Laws for Tactical Missiles*” AIAA Journal of Spacecraft, Vol. 10, No. 5, pp. 301-308, May 1973.
21. Flint, M., Polycarpou, M., and Fernandez, E., “*Cooperative Control for Multiple Autonomous UAV’s Searching for Targets*,” Proceedings of the 41<sup>st</sup> IEEE Conference on Decision and Control, pp. 2823–2828, Dec. 2002.
22. Franklin, G., Powell, J., and Workman, M., “*Digital Control of Dynamic Systems*,” 2nd Edition, Addison-Wesley, 1990.
23. Franklin, G., Powell, J., and Emami-Naeini, A., “*Feedback Control of Dynamic Systems*,” 2nd Edition, Addison-Wesley, 1991.

24. Frew, E. W., and Lawrence, D., “*Cooperative Stand-Off Tracking of Moving Targets by a Team of Autonomous Aircraft*,” AIAA Guidance, Navigation, and Control Conference, Vol. 7, AIAA, Reston, Virginia, pp. 4885–4895, 2005.
25. Frew, E., Lawrence, D., and Morris, S., “*Coordinated Standoff Tracking of Moving Targets Using Lyapunov Guidance Vector Fields*,” AIAA Journal of Guidance and Control, Vol. 31, No. 2, Mar.-Apr. 2008.
26. General Atomics, “*I-GNAT Combat-Proven Long Endurance*.”  
<http://www.ga-asi.com/products/pdf/I-GNAT.pdf>
27. Johnson, E., Proctor, A., Ha, J., Tannenbaum, A. “*Development and Test of Highly Autonomous Unmanned Aerial Vehicles*,” Journal of Aerospace Computing, Information, and Communication, Vol. 1, Dec. 2004.
28. Julier, S., Uhlmann, J., and Durrant-Whyte, H., “*A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators*,” IEEE Transactions on Automatic Control, Vol. 45, No. 3, pp. 477–482, 2000.
29. Kalman, R., “A New Approach to Linear Filtering and Prediction Problems,” Transactions of the ASME, Journal of Basic Engineering, 82, Series D, pp. 35-35, 1960.
30. Leonard, N. E., and Fiorelli, E., “*Virtual Leaders, Artificial Potentials, and Coordinated Control of Groups*,” 40th IEEE Conference on Decision and Control, Vol. 3, IEEE Publications, Piscataway, New Jersey, pp. 2968–2973, 2001.
31. Lin, C., “*Modern Navigation, Guidance, and Control Processing*,” Vol. 2, Prentice Hall, 1991.
32. Lundquist, D., (Retired), Personal interview, Lockheed-Martin Inc., 2008.



33. McLain, T., Chandler, P., Rasmussen, S., and Pachter, M., “*Cooperative Control of UAV Rendezvous*,” Proceedings of the American Control Conference, pp. 2309–2314, Jun. 2001.
34. Minor, J., Warner, D., Cook, N., Adelgren, R., and Doster, J., “*Transforming Training to Support Warfighter Needs: The First USAF Test Pilot School UAV/UCAV Flight Test Course*,” USAF T&E Days, Nashville, Tennessee, 6-8 Dec. 2005
35. Murray, R., “*Recent Research in Cooperative Control of Multi-Vehicle Systems*” ASME Journal of Dynamic Systems, Measurement, and Control, pp. 1-27, 10 Sep. 2006.
36. Mutambara, A., “*Information Based Estimation for Both Linear and Nonlinear Systems*,” American Control Conference, San Diego, California, 1999.
37. Narendra, K. and Annaswamy A., “*Stable Adaptive Systems*,” Prentice Hall, 1989.
38. Pastrick, H., Seltzer, S., and Warren, M., “*Guidance Laws for Short-Range Tactical Missiles*” AIAA Journal of Guidance and Control, Vol. 4, No. 2, pp. 98-108, Mar.-Apr. 1981.
39. Pierce, J., Frederick, R., and Winkeler, J., “*Conceptual Design of an Unmanned Hybrid Vehicle for the Battlefield of 2012*,” AIAA’s 1<sup>st</sup> Conference and Workshop on Unmanned Aerospace Vehicles, Portsmouth, Virginia, 20-23 May 2002.
40. Polycarpou, M., Yang, Y., and Passino, K., “*A Cooperative Search Framework for Distributed Agents*,” Proceedings of the 2001 IEEE International Symposium on Intelligent Control, pp. 1–6, 2001.
41. Price, C., “*The Virtual UAV Leader*,” AIAA Infotech Conference and Exhibit, Rohnert Park, California, 7-10 May 2007.

42. Rao, B., Durrant-Whyte, H., and Sheen, J., “*A Fully Decentralized Multi-Sensor System for Tracking and Surveillance,*” International Journal of Robotics Research, Vol. 12, No. 1, pp. 20–44, Feb 1993.
43. Reichard, K., Crow, E., Stover, J., and Roeckel, M., “*Integration of Self-Awareness in Multi-Vehicle Autonomous Control,*” 2<sup>nd</sup> AIAA Unmanned Unlimited Systems, Technologies, and Operations, San Diego, CA, 15-18 Sep. 2003.
44. Roemerma, S., “*Moving Target Requirements Summary,*” Lone Star Aerospace, Inc. 2006.
45. Rusnak, I., “*Advanced Guidance Laws for Acceleration-Constrained Missile, Randomly Maneuvering Target and Noisy Measurements,*” IEEE Transactions on Aerospace and Electronic Systems, Vol. 32, No. 1, Jan. 1996.
46. Rysdyk, R., “*Unmanned Aerial Vehicle Path Following for Target Observation in Wind,*” Journal of Guidance, Control, and Dynamics, Vol. 29, No. 5, pp. 1092–1100, 2006.
47. Rysdyk, R., Lum, C., and Vagners, J., “*Autonomous Orbit Coordination for Two Unmanned Aerial Vehicles,*” AIAA Guidance, Navigation, and Control Conference, Vol. 6, AIAA, Reston, Virginia, pp. 4876–4884, 2005.
48. Schulze, K. and Buescher, J., “*A Scalable, Economic Autonomous Flight Control and Guidance Package for UAVs,*” 2<sup>nd</sup> AIAA Unmanned Unlimited Systems, Technologies, and Operations, San Diego, CA, 15-18 Sep. 2003.
49. Sharma, M., and Calise, A., “*Application of an Adaptive Autopilot Design to a Family of Guided Munitions,*” AIAA Guidance, Navigation, and Control Conference and Exhibit, 14-17 Aug. 2000.

50. Sharma, M., and Lavretsky, E., "*Application and Flight Testing of an Adaptive Autopilot on Precision Guided Munitions,*" AIAA Guidance, Navigation, and Control Conference and Exhibit, 21-24 Aug. 2006.
51. Simon, D., "*From Here to Infinity,*" Embedded Systems Programming, 2000.
52. Simon, D., "A Game Theory Approach to Constrained Minimax State Estimation," IEEE Transactions on Signal Processing, Vol. 54, Issue 2, Feb. 2006.
53. Simon, D., "*Optimal State Estimation: Kalman,  $H_\infty$ , and Nonlinear Approaches,*" Wiley-Interscience, 2006.
54. Song, J., "*Research on Attitude Pursuit Guidance Law for Strapdown Homing System,*" Chinese Journal of Aeronautics, Vol. 17, No. 4, pp. 235-239, Nov. 2004.
55. Stepanyan, V., and Hovakimyan, N., "*Active Disturbance Rejection Guidance Law for Visual Tracking of a Maneuvering Target,*" AIAA Guidance, Navigation, and Control Conference and Exhibit, 15-18 August 2005.
56. Stevenson, J., Li, D., Fuciarelli, D., and Zuber, J., "*Group Cooperation and Onboard Controllers for Distributed Weapon Systems & ISR Platforms,*" AIAA Infotech Conference and Exhibit, Rohnert Park, California, 7-10 May 2007.
57. Takehira, T., Vinh, N., and Kabamba, P., "*Analytical Solution of Missile Terminal Guidance,*" AIAA Journal of Guidance and Control, Vol. 21, No. 2, Mar.-Apr. 1998.
58. Thienel, J., VanEpoel, J., and Sanner, R., "*Accurate State Estimation and Tracking of a Non-Cooperative Target Vehicle,*" AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado, 21-24 Aug. 2006.

59. Vergez, P. and McClendon, S., "*Optimal Control and Estimation for Strapdown Seeker Guidance of Tactical Missiles*" AIAA Journal of Guidance and Control, Vol. 5, No. 3, pp. 50-55, May-Jun. 1982.
60. Whitacre, W. and Campbell, M., "*Cooperative Geolocation with UAVs Under Communication Constraints*," AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, South Carolina, 20-23 Aug. 2007.
61. White, A., "*Flexible Command and Control of Semi-Autonomous Combat UAVs*," 2<sup>nd</sup> AIAA Unmanned Unlimited Technologies Conference, Workshop and Exhibit, Chicago, IL, 20-23 Sep. 2004.
62. Wise, K. and Lavretsky, E., "*Adaptive Control of Flight: Theory, Applications, and Open Problems*," Proceedings of the 2006 American Control Conference, 14-16 Jun. 2006.
63. Wise, K., Lavretsky, E., Zimmerman, J., Francis-Jr., J., Dixon, D., and Whitehead, B., "*Adaptive Flight Control of a Sensor Guided Munition*," The Boeing Company, 2005.
64. Wise, R. and Rysdyk, R. T., "*UAV Coordination for Autonomous Target Tracking*," AIAA Guidance, Navigation, and Control Conference and Exhibit, Vol. 5, AIAA, Reston, Virginia, pp. 3210–3231, 2006.
65. Yang, Y., Minai, A., and Polycarpou, M., "*Decentralized Cooperative Search in UAV's Using Opportunistic Learning*," AIAA Guidance, Navigation, and Control Conference and Exhibit, Aug. 2002.
66. Yang, I. H. and Zhao, Y. J., "*Real-Time Trajectory Planning for Autonomous Aerospace Vehicles Amidst Static Obstacles*," 1st AIAA Technical Conference and

Workshop on UAV Systems, Technologies, and Operations, AIAA, Reston, Virginia, 2002.

67. Yang, Y., Minai, A., and Polycarpou, M., “*Decentralized Cooperative Search by Networked UAVs in an Uncertain Environment,*” Proceedings of the American Control Conference, Boston, Massachusetts, pp. 5558-5563, 30 Jun. – 2 Jul. 2004.
68. Zarchan, P. and Musoff, H., “*Fundamentals of Kalman Filtering: A Practical Approach,*” Second Edition, AIAA Progress in Astronautics and Aeronautics, Vol. 208, 2005.
69. Zipfel, P., “*Modeling and Simulation of Aerospace Vehicle Dynamics,*” AIAA Education Series, 2000.